In [1]:

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

In [2]:

```
df=pd.read_csv('/kaggle/input/netflix-movies/netflix.csv')
```

Overview the DataSet

In [3]:

```
df.head()
```

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, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA
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

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(11)
memory usage: 825.8+ KB
```

Checking Unique values in dataset

In [5]:

```
for i in df.columns:
  print(i,':',df[i].nunique())
```

```
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
```

In [6]:

```
df.shape
```

Out[6]:

```
(8807, 12)
```

In [7]:

```
df.describe()
```

Out[7]:

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

In [8]:

```
df.describe(include='object').T
```

Out[8]:

	count	unique	top	freq
show_id	8807	8807	s1	1
type	8807	2	Movie	6131
title	8807	8807	Dick Johnson Is Dead	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

Checking Percentage of null values

In [9]:

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

Out[9]:

```
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
```

Checking the percentage of 'Type of content'

In [10]:

```
df['type'].value_counts(normalize=True)*100
```

Out[10]:

```
Movie      69.615079
TV Show    30.384921
Name: type, dtype: float64
```

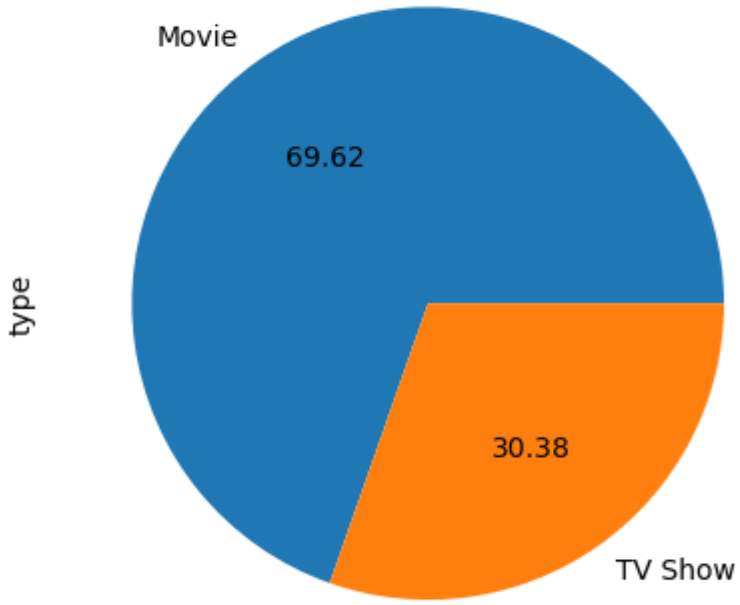
Plot for better Visualization

In [11]:

```
df['type'].value_counts().plot(kind='pie',autopct="%.2f")
```

Out[11]:

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



In [12]:

```
df.head()
```

Out[12]:

	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, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA
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

Columns Preprocessing

Preprocessing for Director column

In [13]:

```
#unnesting the directors column, i.e- creating separate columns for each director in a m
constraint1=df['director'].apply(lambda x: str(x).split(', ')).tolist()
df_new1=pd.DataFrame(constraint1,index=df['title'])
df_new1=df_new1.stack()
df_new1=pd.DataFrame(df_new1.reset_index())
df_new1.rename(columns={0:'Directors'},inplace=True)
df_new1.drop(['level_1'],axis=1,inplace=True)
df_new1.head()
```

Out[13]:

	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

Preprocessing for Cast column

In [14]:

```
#unnesting the cast column, i.e- creating separate lines for each cast member in a movie
constraint2=df['cast'].apply(lambda x: str(x).split(', ')).tolist()
df_new2=pd.DataFrame(constraint2,index=df['title'])
df_new2=df_new2.stack()
df_new2=pd.DataFrame(df_new2.reset_index())
df_new2.rename(columns={0:'Actors'},inplace=True)
df_new2.drop(['level_1'],axis=1,inplace=True)
df_new2.head()
```

Out[14]:

	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

Preprocessing for Type of Movie Genre column

In [15]:

```
#unnesting the listed_in column, i.e- creating separate lines for each genre in a movie
constraint3=df['listed_in'].apply(lambda x: str(x).split(', ')).tolist()
df_new3=pd.DataFrame(constraint3,index=df['title'])
df_new3=df_new3.stack()
df_new3=pd.DataFrame(df_new3.reset_index())
df_new3.rename(columns={0:'Genre'},inplace=True)
df_new3.drop(['level_1'],axis=1,inplace=True)
df_new3.head()
```

Out[15]:

	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

Preprocessing for Country column

In [16]:

```
#unnesting the country column, i.e- creating separate lines for each country in a movie
constraint4=df['country'].apply(lambda x: str(x).split(', ')).tolist()
df_new4=pd.DataFrame(constraint4,index=df['title'])
df_new4=df_new4.stack()
df_new4=pd.DataFrame(df_new4.reset_index())
df_new4.rename(columns={0:'country'},inplace=True)
df_new4.drop(['level_1'],axis=1,inplace=True)
df_new4.head()
```

Out[16]:

	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

Merging all the Preprocessed Dataframes

In [17]:

```
#merging the unnested director data with unnested actors data
df_new5=df_new2.merge(df_new1,on=['title'],how='inner')
#merging the above merged data with unnested genre data
df_new6=df_new5.merge(df_new3,on=['title'],how='inner')
#merging the above merged data with unnested country data
df_new=df_new6.merge(df_new4,on=['title'],how='inner')
```

Replacing Nan Values in Actors,Directors and Country

In [18]:

```
#replacing nan values of director and actor by Unknown Actor and Director
df_new['Actors'].replace(['nan'],['Unknown Actor'],inplace=True)
df_new['Directors'].replace(['nan'],['Unknown Director'],inplace=True)
df_new['country'].replace(['nan'],[np.nan],inplace=True)
df_new.head()
```

Out[18]:

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

Merging the Preprocessed dataframe with Original dataframe

In [19]:

```
#merging our unnested data with the original data
df_final=df_new.merge(df[['show_id', 'type', 'title', 'date_added',
       'release_year', 'rating', 'duration']],on=['title'],how='left')
df_final.head()
```

Out[19]:

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

Dealing with Null values

In [20]:

```
df_final.isnull().sum()
```

Out[20]:

```
title              0
Actors             0
Directors          0
Genre              0
country        11897
show_id            0
type               0
date_added       158
release_year       0
rating            67
duration           3
dtype: int64
```

Null Values in duration Column

In [21]:

```
df_final[df_final['duration'].isnull()]
```

Out[21]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_y
126537	Louis C.K. 2017	Louis C.K.	Louis C.K.	Movies	United States	s5542	Movie	April 4, 2017	20
131603	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	Movies	United States	s5795	Movie	September 16, 2016	20
131737	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	Movies	United States	s5814	Movie	August 15, 2016	20

It is observed that the Duration Columns which have null values, their data is written in rating column

In [22]:

```
df_final.loc[df_final['duration'].isnull(),'duration'] = df_final.loc[df_final['duration
```

In [23]:

```
df_final.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            67
duration           0
dtype: int64
```

Null Values in rating column

> It is observed that there is 'min' values in rating column; ratings cant be min, So replace with NR(Non-Rated)
> Also,We replace Nan values with NR

In [24]:

```
df_final.loc[df_final['rating'].str.contains('min', na=False),'rating']='NR'
df_final['rating'].fillna('NR',inplace=True)
```

In [25]:

```
df_final.isnull().sum()
```

Out[25]:

```
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
```

Null Values in Date column

In [26]:

```
df_final[df_final['date_added'].isnull()].head()
```

Out[26]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	relea
136893	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	NaN	
136894	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	TV Comedies	United Kingdom	s6067	TV Show	NaN	
136895	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	TV Dramas	United Kingdom	s6067	TV Show	NaN	
136896	A Young Doctor's Notebook and Other Stories	Jon Hamm	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	NaN	
136897	A Young Doctor's Notebook and Other Stories	Jon Hamm	Unknown Director	TV Comedies	United Kingdom	s6067	TV Show	NaN	

In [27]:

```
#date added column is imputed on the basis of release year,i.e- suppose there's a null fo
#when release year was 2013.So below piece of code just checks the mode of date added fo
# and imputes in place of nulls the corresponding mode

for i in df_final[df_final['date_added'].isnull()]['release_year'].unique():
  imp=df_final[df_final['release_year']==i]['date_added'].mode().values[0]
  df_final.loc[df_final['release_year']==i,'date_added']=df_final.loc[df_final['release_
```

In [28]:

```
df_final.isnull().sum()
```

Out[28]:

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

Null Values in Country column

In [29]:

```
#country column is imputed on the basis of director,i.e- suppose there's a null for coun
#when we have a director whose other movies have a country given.So below piece of code
#country for the director
# and imputes in place of nulls the corresponding mode

for i in df_final[df_final['country'].isnull()]['Directors'].unique():
  if i in df_final[~df_final['country'].isnull()]['Directors'].unique():
    imp=df_final[df_final['Directors']==i]['country'].mode().values[0]
    df_final.loc[df_final['Directors']==i,'country']=df_final.loc[df_final['Directors']=
```

In [30]:

```
df_final.isnull().sum()
```

Out[30]:

```
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 [31]:

```
for i in df_final[df_final['country'].isnull()]['Actors'].unique():
  if i in df_final[~df_final['country'].isnull()]['Actors'].unique():
    imp=df_final[df_final['Actors']==i]['country'].mode().values[0]
    df_final.loc[df_final['Actors']==i,'country']=df_final.loc[df_final['Actors']==i,'co
```

In [32]:

```
#If there are still nulls, I just replace it by Unknown Country
df_final['country'].fillna('Unknown Country',inplace=True)
df_final.isnull().sum()
```

Out[32]:

```
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
```

Modifying Columns

Modifying the Duration column

> Removing 'min'

In [33]:

```
#removing mins from data
df_final['duration']=df_final['duration'].str.replace(" min","")
df_final.head()
```

Out[33]:

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

In [34]:

```
df_final['duration'].nunique()
```

Out[34]:

```
220
```

In [35]:

```
df_final['duration_copy']=df_final['duration'].copy()
df_final1=df_final.copy()
```

In [36]:

```
df_final1.loc[df_final1['duration_copy'].str.contains('Season'),'duration_copy']=0
df_final1['duration_copy']=df_final1['duration_copy'].astype('int')
df_final1.head()
```

Out[36]:

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

In [37]:

```
import seaborn as sns
sns.distplot(df_final1['duration_copy'], hist=True, kde=True,
bins=int(36), color = 'darkblue',
hist_kws={'edgecolor':'black'},
kde_kws={'linewidth': 4})
plt.show()
```

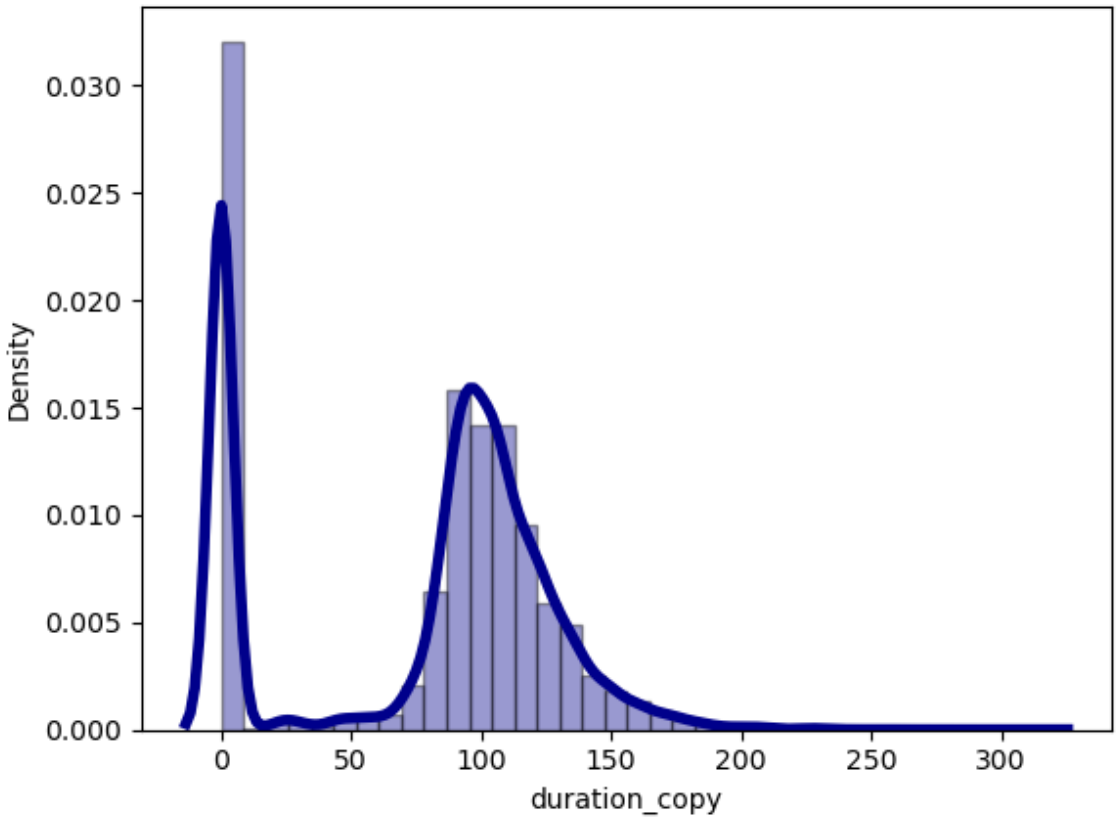
```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:5: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.
0.

Please adapt your code to use either `displot` (a figure-level function wi
th
similar flexibility) or `histplot` (an axes-level function for histogram
s).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://
gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  """
```



Since most of the movies are in 0 category; means they are seasons

In [38]:

```
bins1 = [-1,1,50,80,100,120,150,200,315]
labels1 = ['<1','1-50','50-80','80-100','100-120','120-150','150-200','200-315']
df_final1['duration_copy'] = pd.cut(df_final1['duration_copy'],bins=bins1,labels=labels1
df_final1.head()
```

Out[38]:

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

In [39]:

```
df_final1.loc[~df_final1['duration'].str.contains('Season'),'duration']=df_final1.loc[~d
df_final1.drop(['duration_copy'],axis=1,inplace=True)
df_final1.head()
```

Out[39]:

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

In [40]:

```
df_final1['duration'].value_counts()
```

Out[40]:

```
80-100        52937
100-120       48724
1 Season      35035
120-150       26691
2 Seasons      9559
50-80          7700
150-200        6737
3 Seasons      5084
1-50           2530
4 Seasons      2134
5 Seasons      1698
7 Seasons       843
6 Seasons       633
200-315         524
8 Seasons       286
9 Seasons       257
10 Seasons      220
13 Seasons      132
12 Seasons      111
15 Seasons       96
17 Seasons       30
11 Seasons       30
Name: duration, dtype: int64
```

Modifying the Date column

Seperaring Month, Week and year

In [41]:

```
from datetime import datetime
from dateutil.parser import parse
arr=[]
for i in df_final1['date_added'].values:
  dt1=parse(i)
  arr.append(dt1.strftime('%Y-%m-%d'))
df_final1['Modified_Added_date'] =arr
df_final1['Modified_Added_date']=pd.to_datetime(df_final1['Modified_Added_date'])
df_final1['month_added']=df_final1['Modified_Added_date'].dt.month
df_final1['week_Added']=df_final1['Modified_Added_date'].dt.week
df_final1['year']=df_final1['Modified_Added_date'].dt.year
df_final1.head()
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:10: FutureWar
ning: Series.dt.weekofyear and Series.dt.week have been deprecated.  Pleas
e use Series.dt.isocalendar().week instead.
  # Remove the CWD from sys.path while we load stuff.
```

Out[41]:

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

Modifying Title Column

Removing Duplicate titles on the basis of their language version

In [42]:

```
#Titles such as Bahubali(Hindi Version),Bahubali(Tamil Version) were there. Since it's o
#presence of brackets and content between brackets is removed.
df_final1['title']=df_final1['title'].str.replace(r"\(.*\)","")
df_final1.head()
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: FutureWarn
ing: The default value of regex will change from True to False in a future
version.
  This is separate from the ipykernel package so we can avoid doing import
s until
```

Out[42]:

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

Univariate Analysis

Genre

In [43]:

```
#number of distinct titles on the basis of genre
df_final1.groupby(['Genre']).agg({"title":"nunique"}).sort_values(by=['title'], ascendin
```

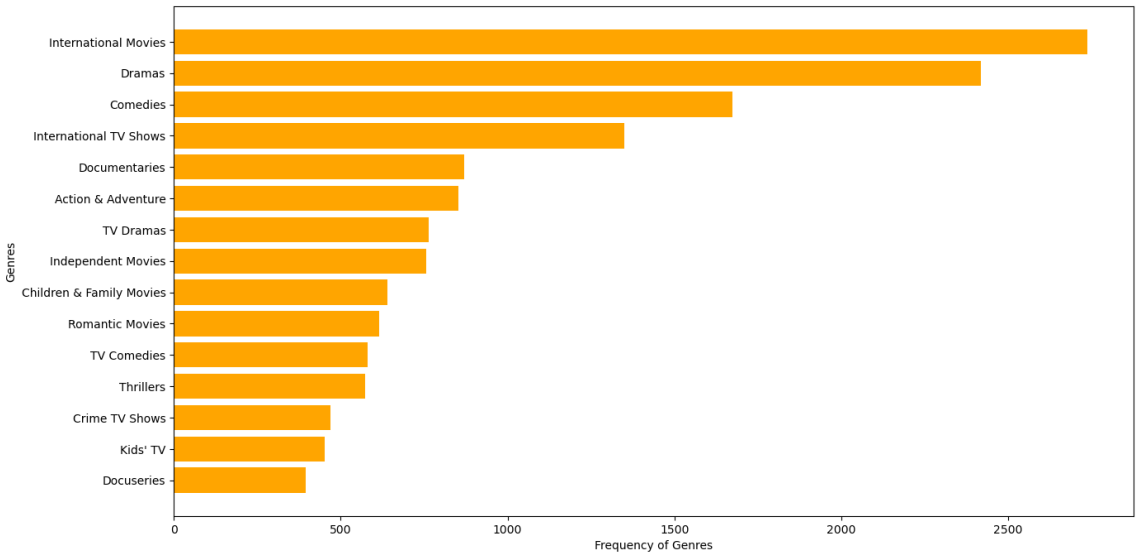
Out[43]:

Genre	title
International Movies	2738
Dramas	2418
Comedies	1673
International TV Shows	1351
Documentaries	869
Action & Adventure	854
TV Dramas	763
Independent Movies	756
Children & Family Movies	639
Romantic Movies	615
TV Comedies	581
Thrillers	573
Crime TV Shows	470
Kids' TV	451
Docuseries	395
Music & Musicals	372
Romantic TV Shows	370
Horror Movies	353
Stand-Up Comedy	343
Reality TV	255
British TV Shows	253
Sci-Fi & Fantasy	243
Sports Movies	219
Anime Series	176
Spanish-Language TV Shows	174
TV Action & Adventure	168
Korean TV Shows	151
Classic Movies	116
LGBTQ Movies	102
TV Mysteries	98
Science & Nature TV	92
TV Sci-Fi & Fantasy	84
TV Horror	75
Anime Features	71
Cult Movies	71
Teen TV Shows	69

Genre	title
Faith & Spirituality	65
TV Thrillers	57
Movies	57
Stand-Up Comedy & Talk Shows	56
Classic & Cult TV	28
TV Shows	16

In [44]:

```
df_genre=df_final1.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



International Movies, Dramas and Comedies are the most popular .

Type

In [45]:

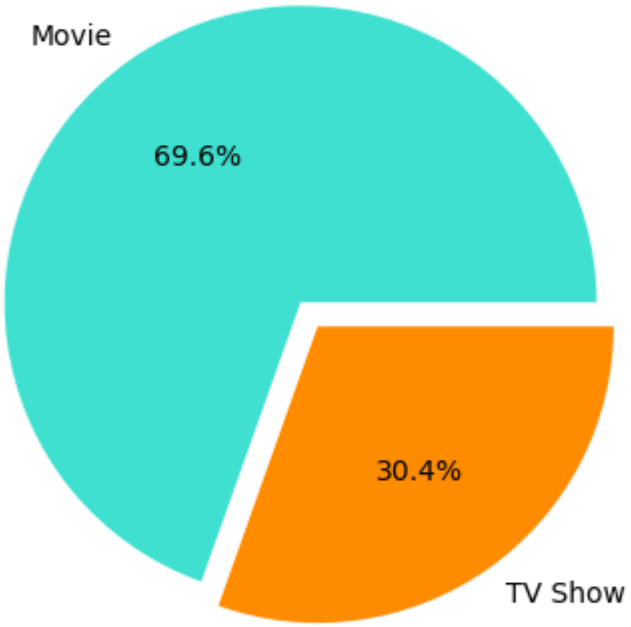
```
#number of distinct titles on the basis of type
df_final1.groupby(['type']).agg({"title":"nunique"})
```

Out[45]:

type	title
Movie	6115
TV Show	2676

In [46]:

```
df_type=df_final1.groupby(['type']).agg({"title":"nunique"}).reset_index()
plt.pie(df_type['title'],explode=(0.05,0.05), labels=df_type['type'],colors=['turquoise
plt.show()
```



- We have 70:30 ratio of Movies and TV Shows in our data

Country

In [47]:

```
#number of distinct titles on the basis of country
df_final1.groupby(['country']).agg({"title":"nunique"})
```

Out[47]:

country	title
	3
Afghanistan	1
Albania	1
Algeria	3
Angola	2
...	...
Vatican City	1
Venezuela	4
Vietnam	7
West Germany	5
Zimbabwe	3

128 rows × 1 columns

> The above dataframe shows a problem in which we are seeing countries, such as Cambodia and Cambodia, or United States and United States, are shown as different countries.They should have been same

In [48]:

```
df_final1['country'] = df_final1['country'].str.replace(',', '')
df_final1.head()
```

Out[48]:

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

In [49]:

```
#number of distinct titles on the basis of country
df_final1.groupby(['country']).agg({"title":"nunique"})
```

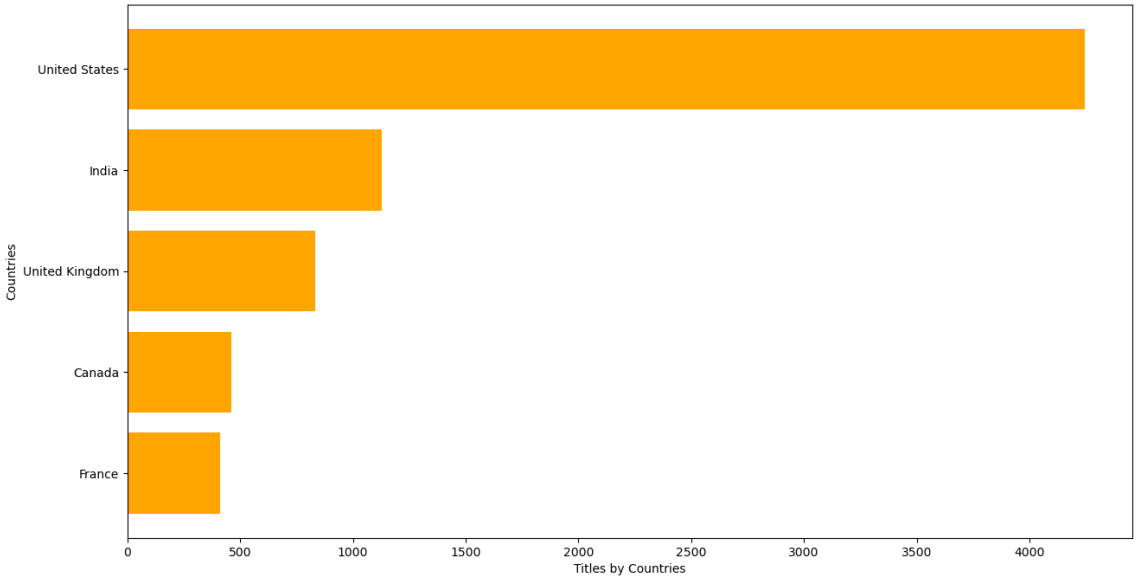
Out[49]:

country	title
	3
Afghanistan	1
Albania	1
Algeria	3
Angola	2
...	...
Vatican City	1
Venezuela	4
Vietnam	7
West Germany	5
Zimbabwe	3

124 rows × 1 columns

In [50]:

```
df_country=df_final1.groupby(['country']).agg({"title":"nunique"}).reset_index().sort_va
plt.figure(figsize=(15,8))
plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=['orange'])
plt.xlabel('Titles by Countries')
plt.ylabel('Countries')
plt.show()
```



- US,India,UK,Canada and France are leading countries in Content Creation on Netflix

Rating

In [51]:

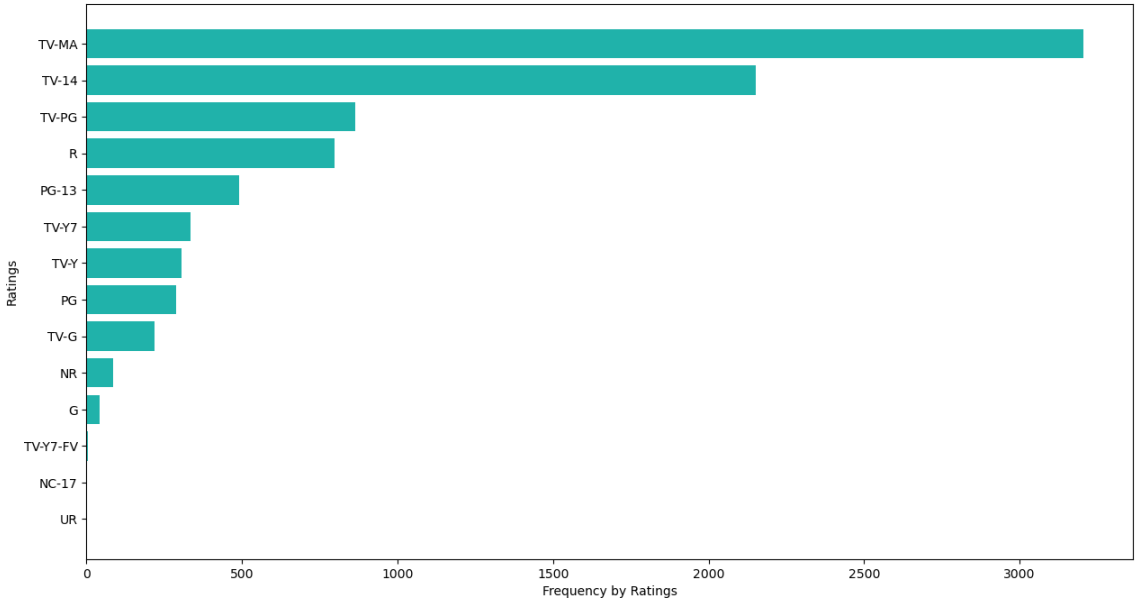
```
#number of distinct titles on the basis of rating
df_final1.groupby(['rating']).agg({"title":"nunique"})
```

Out[51]:

rating	title
G	41
NC-17	3
NR	87
PG	287
PG-13	490
R	799
TV-14	2151
TV-G	220
TV-MA	3204
TV-PG	863
TV-Y	305
TV-Y7	334
TV-Y7-FV	6
UR	3

In [52]:

```
df_rating=df_final1.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_valu
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['lightseagreen'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



Most of the highly rated content on Netflix is intended for Mature Audiences, R Rated, content not intended for audience under 14 and those which require Parental Guidance

Duration

In [53]:

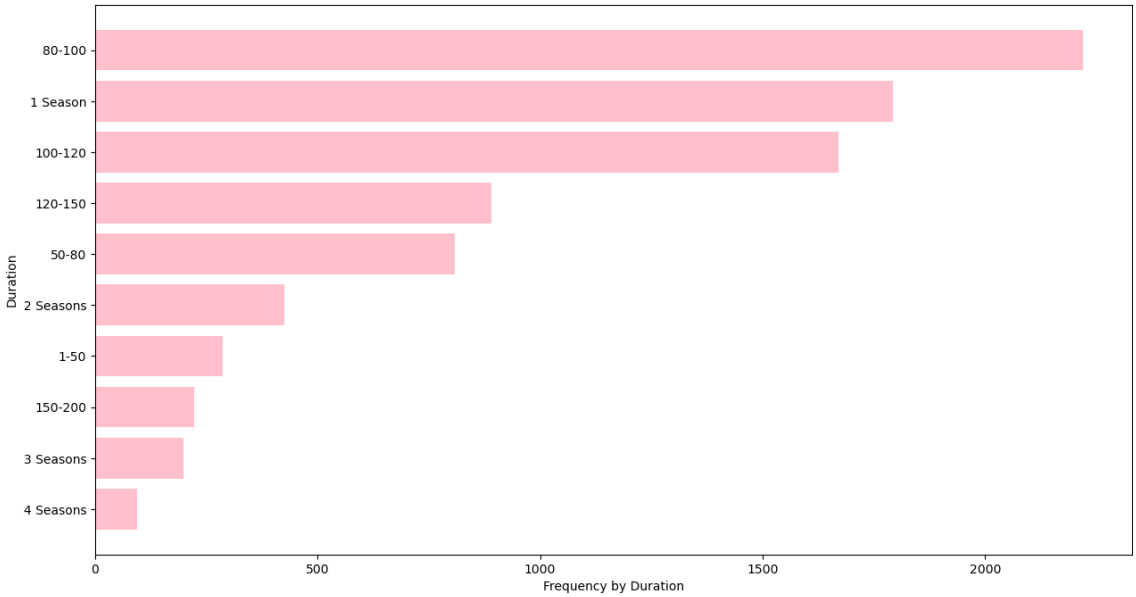
```
#number of distinct titles on the basis of duration
df_final1.groupby(['duration']).agg({"title":"nunique"})
```

Out[53]:

duration	title
1 Season	1793
1-50	287
10 Seasons	7
100-120	1671
11 Seasons	2
12 Seasons	2
120-150	891
13 Seasons	3
15 Seasons	2
150-200	222
17 Seasons	1
2 Seasons	425
200-315	19
3 Seasons	199
4 Seasons	95
5 Seasons	65
50-80	808
6 Seasons	33
7 Seasons	23
8 Seasons	17
80-100	2220
9 Seasons	9

In [54]:

```
df_duration=df_final1.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort_
plt.figure(figsize=(15,8))
plt.barh(df_duration[::-1]['duration'], df_duration[::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



The duration of Most Watched content in our whole data is 80-100 mins.These must be movies and Shows having only 1 Season.

Actors

In [55]:

```
#number of distinct titles on the basis of Actors
df_final1.groupby(['Actors']).agg({"title":"nunique"})
```

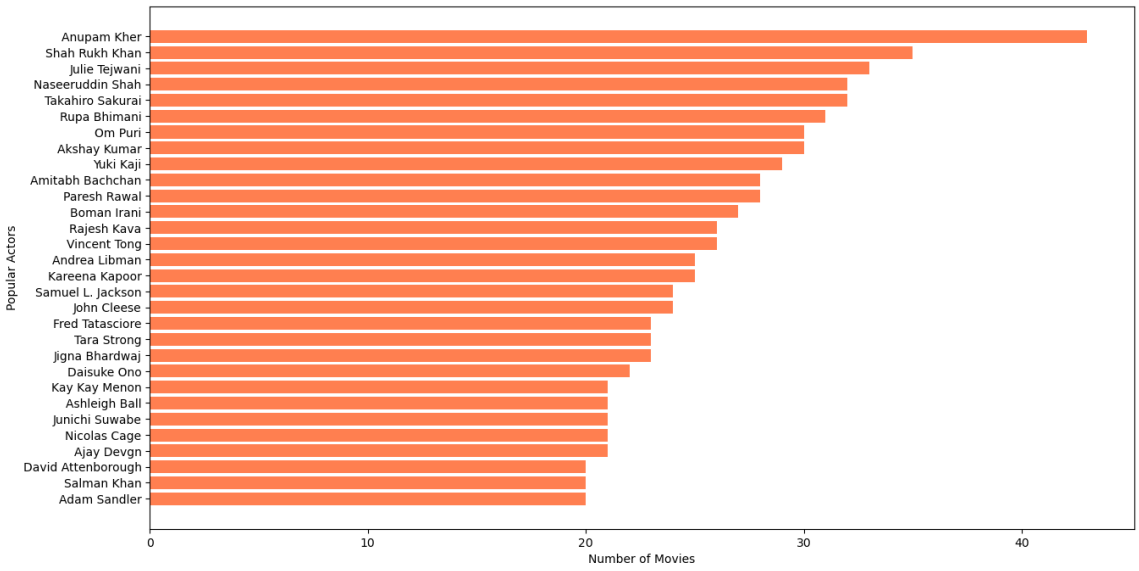
Out[55]:

Actors	title
Jr.	2
"Riley" Lakdhar Dridi	1
'Najite Dede	2
2 Chainz	1
2Mex	1
...	...
Şevket Çoruh	1
Şinasi Yurtsever	3
Şükran Ovalı	1
Şükrü Özyıldız	2
Ṣọpẹ́ Dìrísù	1

36440 rows × 1 columns

In [56]:

```
df_actors=df_final1.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_valu
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['Actors'], df_actors[::-1]['title'],color=['coral'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actors')
plt.show()
```



- Anupam Kher,SRK,Julie Tejwani, Naseeruddin Shah and Takahiro Sakurai occupy the top stop in Most Watched content.

Directors

In [57]:

```
#number of distinct titles on the basis of Directors
df_final1.groupby(['Directors']).agg({"title":"nunique"})
```

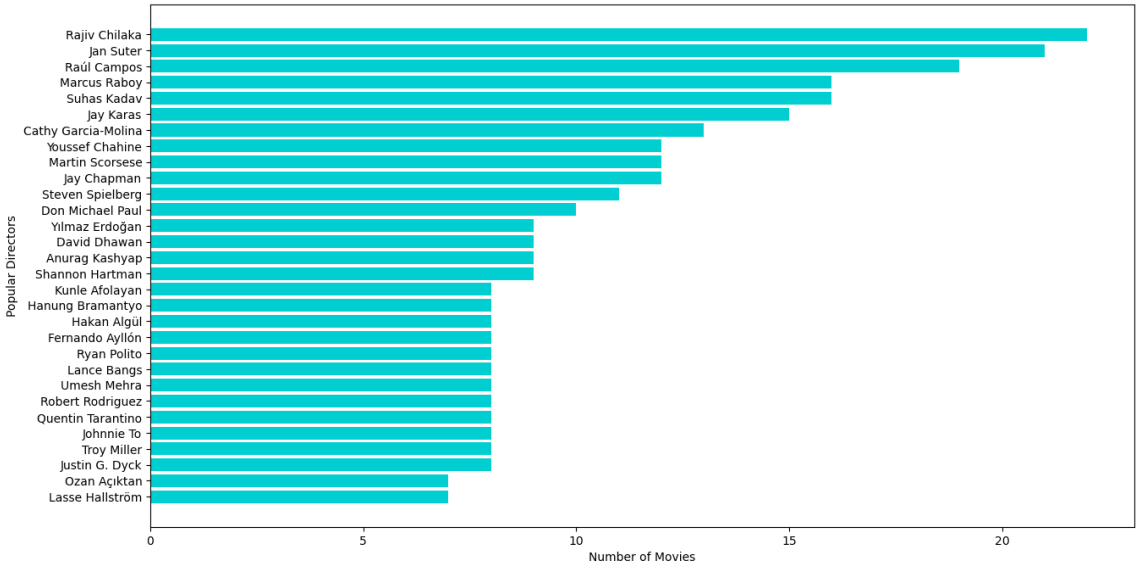
Out[57]:

Directors	title
A. L. Vijay	2
A. Raajdheep	1
A. Salaam	1
A.R. Murugadoss	2
Aadish Keluskar	1
...	...
Éric Warin	1
Ísold Uggadóttir	1
Óskar Thór Axelsson	1
Ömer Faruk Sorak	3
Şenol Sönmez	2

4994 rows × 1 columns

In [58]:

```
df_directors=df_final1.groupby(['Directors']).agg({"title":"nunique"}).reset_index().sor
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['darkturquo
plt.xlabel('Number of Movies')
plt.ylabel('Popular Directors')
plt.show()
```



- Rajiv Chilaka, Jan Suter and Raul Campos are the most popular directors across Netflix

Year

In [59]:

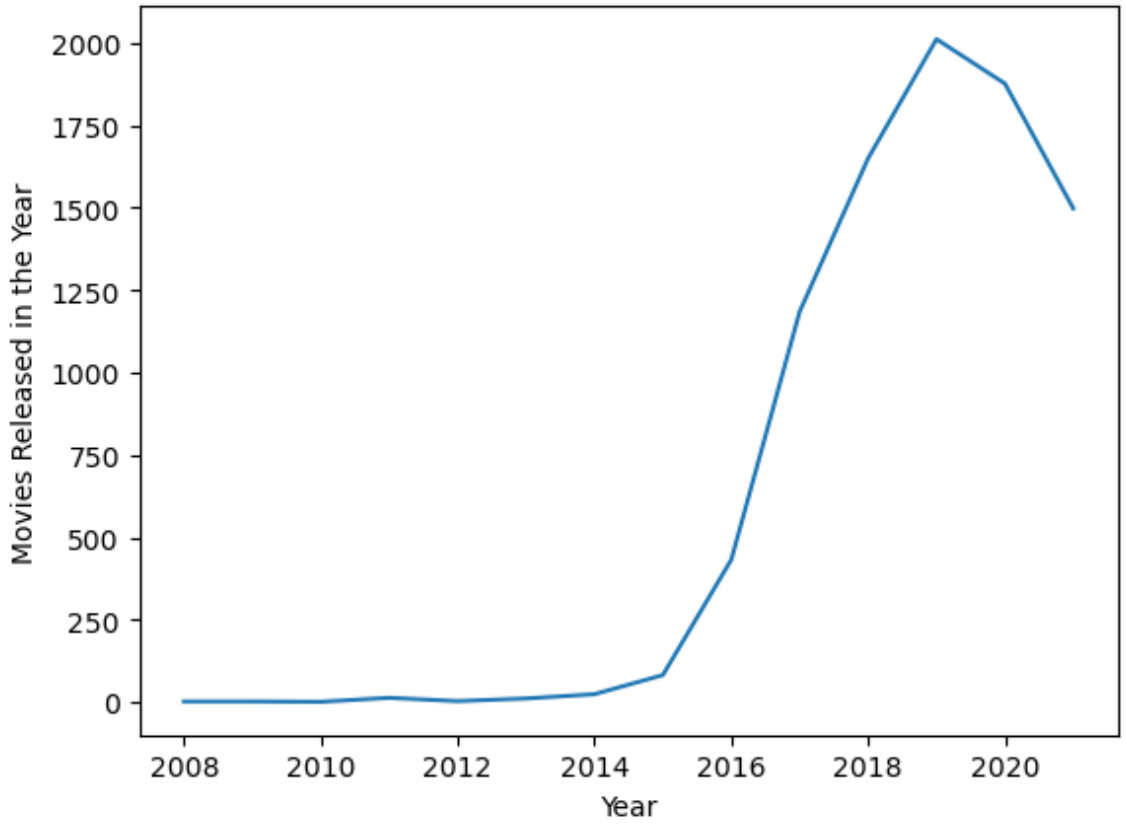
```
#number of distinct titles on the basis of year
df_final1.groupby(['year']).agg({"title":"nunique"})
```

Out[59]:

year	title
2008	2
2009	2
2010	1
2011	13
2012	3
2013	11
2014	24
2015	82
2016	432
2017	1185
2018	1650
2019	2012
2020	1877
2021	1498

In [60]:

```
df_year=df_final1.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



- The Amount of Content across Netflix has increased from 2008 continuously till 2019. Then started decreasing from here(probably due to Covid)

Week

In [61]:

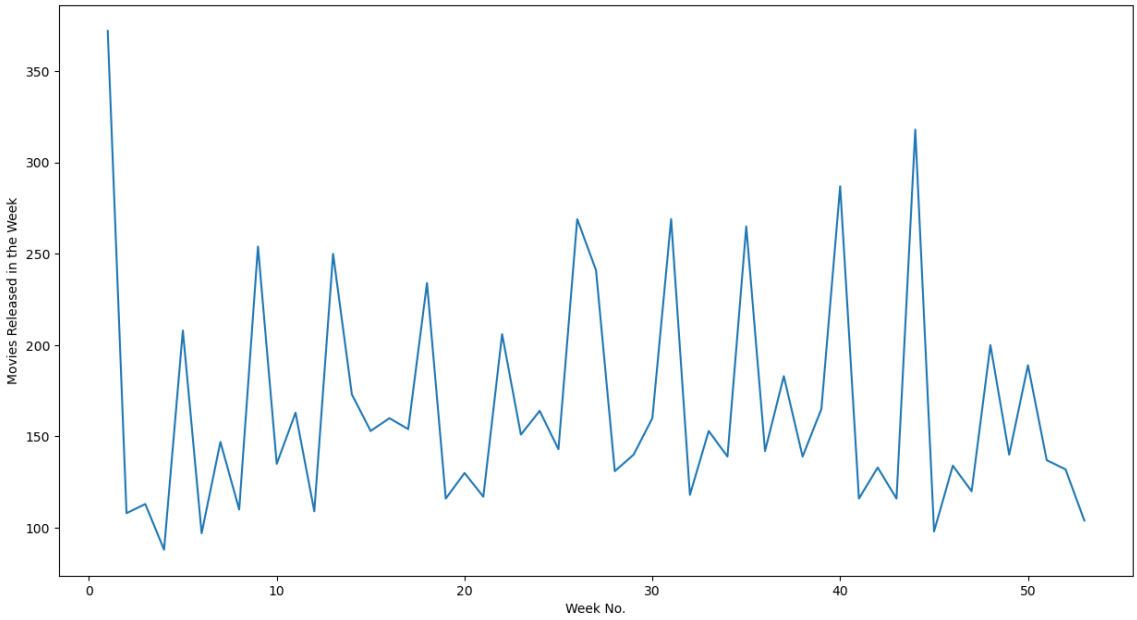
```
#number of distinct titles on the basis of week
df_final1.groupby(['week_Added']).agg({"title":"nunique"})
```

Out[61]:

week_Added	title
1	372
2	108
3	113
4	88
5	208
6	97
7	147
8	110
9	254
10	135
11	163
12	109
13	250
14	173
15	153
16	160
17	154
18	234
19	116
20	130
21	117
22	206
23	151
24	164
25	143
26	269
27	241
28	131
29	140
30	160
31	269
32	118
33	153
34	139
35	265
36	142

week_Added	title
37	183
38	139
39	165
40	287
41	116
42	133
43	116
44	318
45	98
46	134
47	120
48	200
49	140
50	189
51	137
52	132
53	104

In [62]:
```
df_week=df_final1.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



- Most of the Content across Netflix is added in the first week of the year and it follows a bit of a cyclical pattern

Month

In [63]:

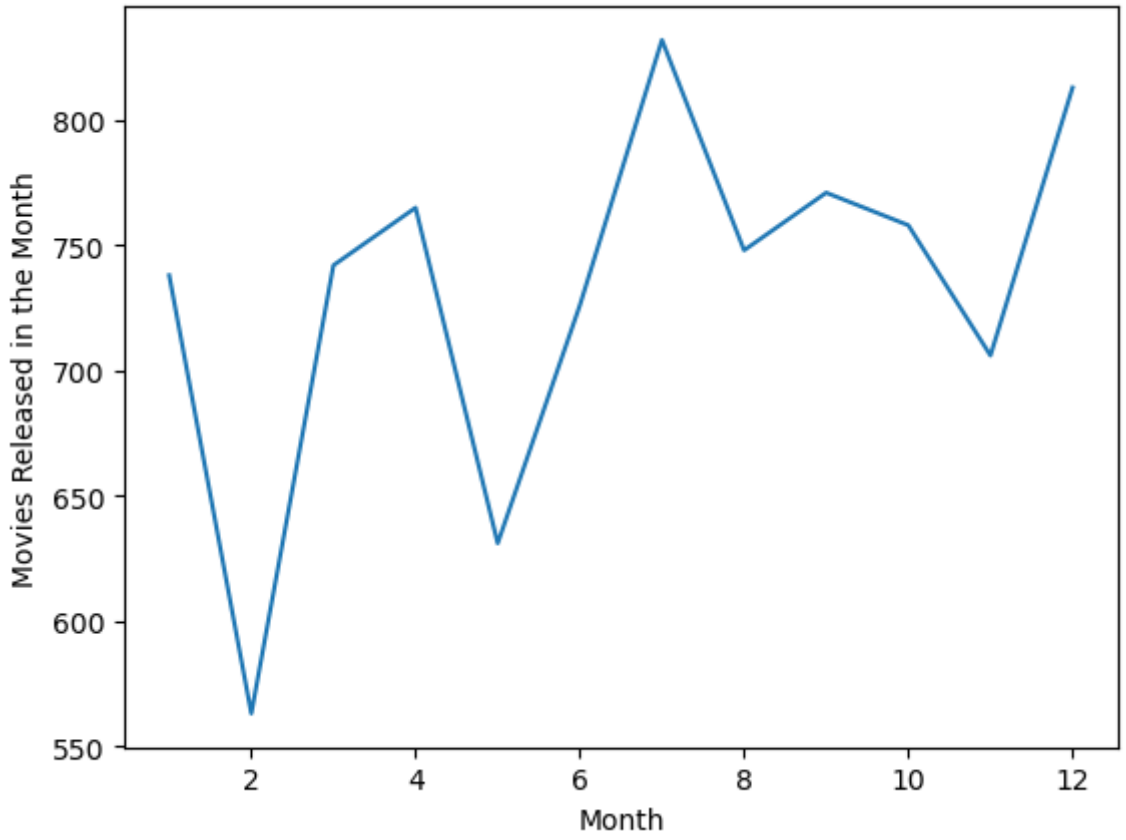
```
#number of distinct titles on the basis of month
df_final1.groupby(['month_added']).agg({"title":"nunique"})
```

Out[63]:

month_added	title
1	738
2	563
3	742
4	765
5	631
6	726
7	832
8	748
9	771
10	758
11	706
12	813

In [64]:

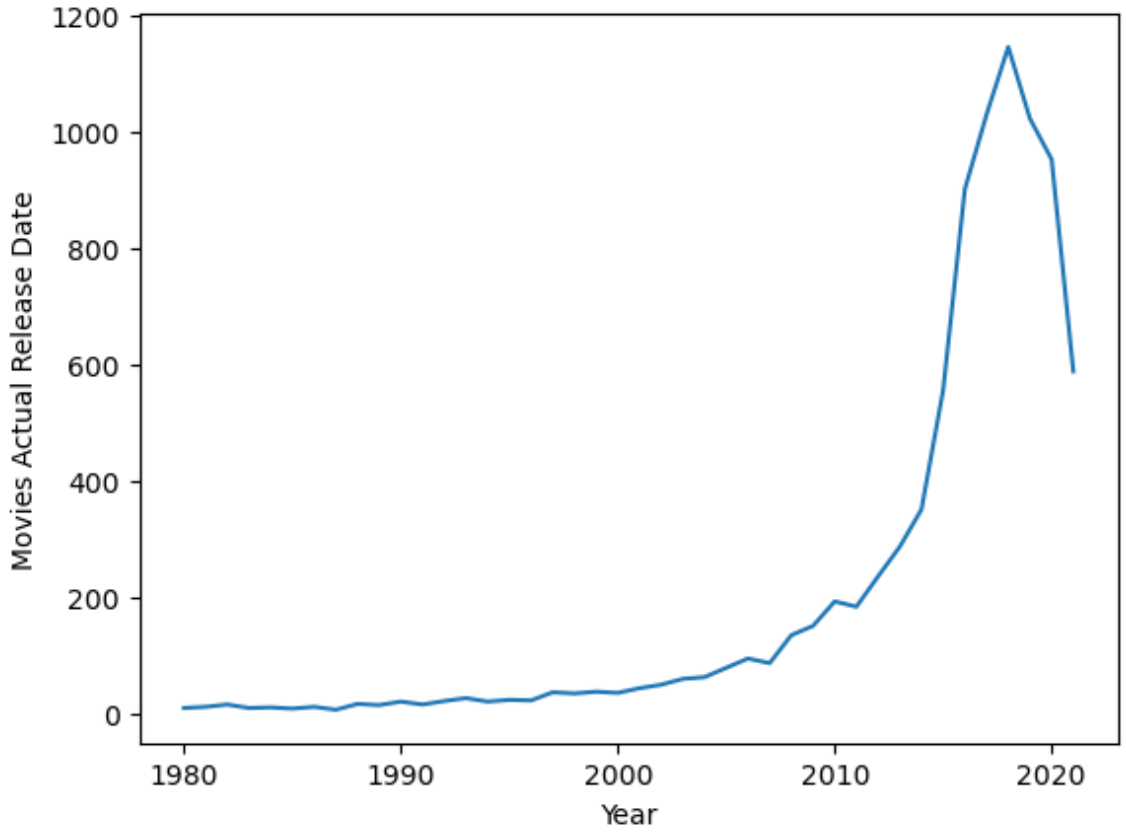
```
df_month=df_final1.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("Movies Released in the Month")
plt.xlabel("Month")
plt.show()
```



- Most of the content is added in the first and last months across Netflix

In [65]:

```
df_release_year=df_final1[df_final1['release_year']>=1980].groupby(['release_year']).agg
sns.lineplot(data=df_release_year, x='release_year', y='title')
plt.ylabel("Movies Actual Release Date")
plt.xlabel("Year")
plt.show()
```



- Net content release which are later uploaded to Netflix has increased since 1980 till 2020 though later reduced certainly due to COVID-19

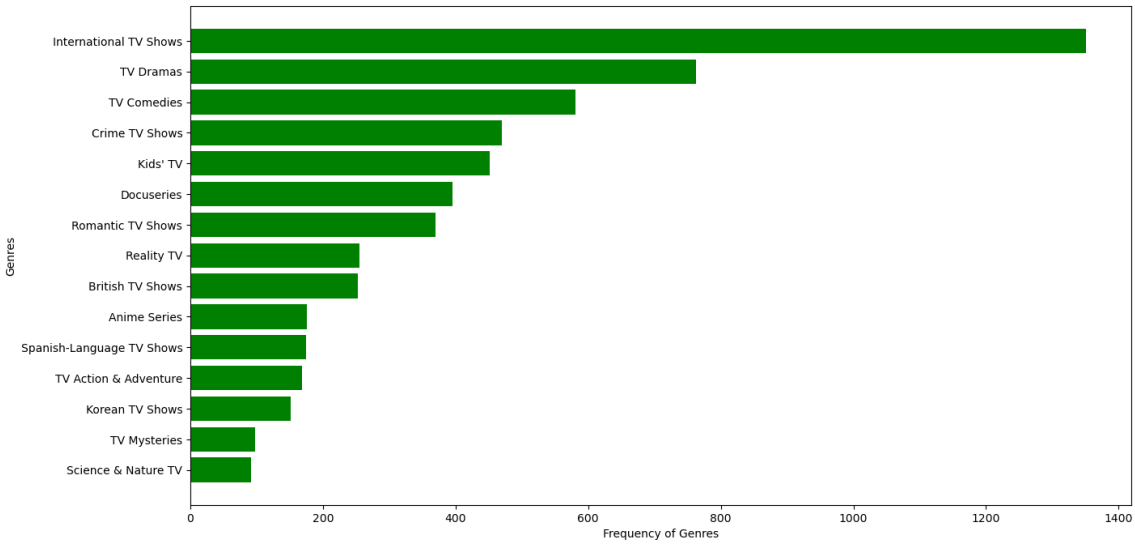
Univariate Analysis separately for shows and movies

In [66]:

```
df_shows=df_final1[df_final1['type']=='TV Show']
df_movies=df_final1[df_final1['type']=='Movie']
```

In [67]:

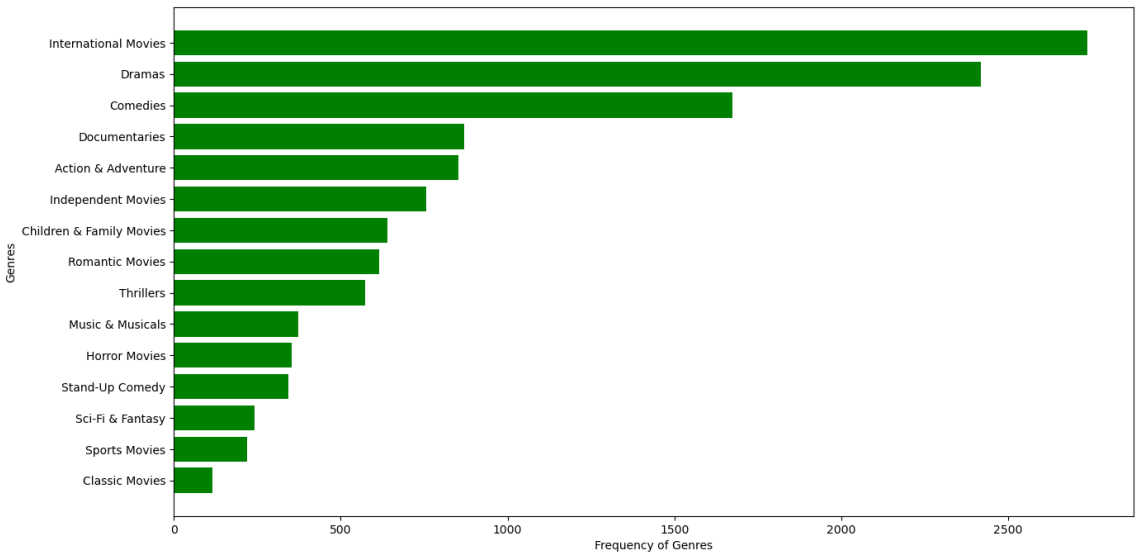
```
df_genre=df_shows.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values(
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['green'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



- Most of the TV Shows are International, Dramas and Comemdies

In [68]:

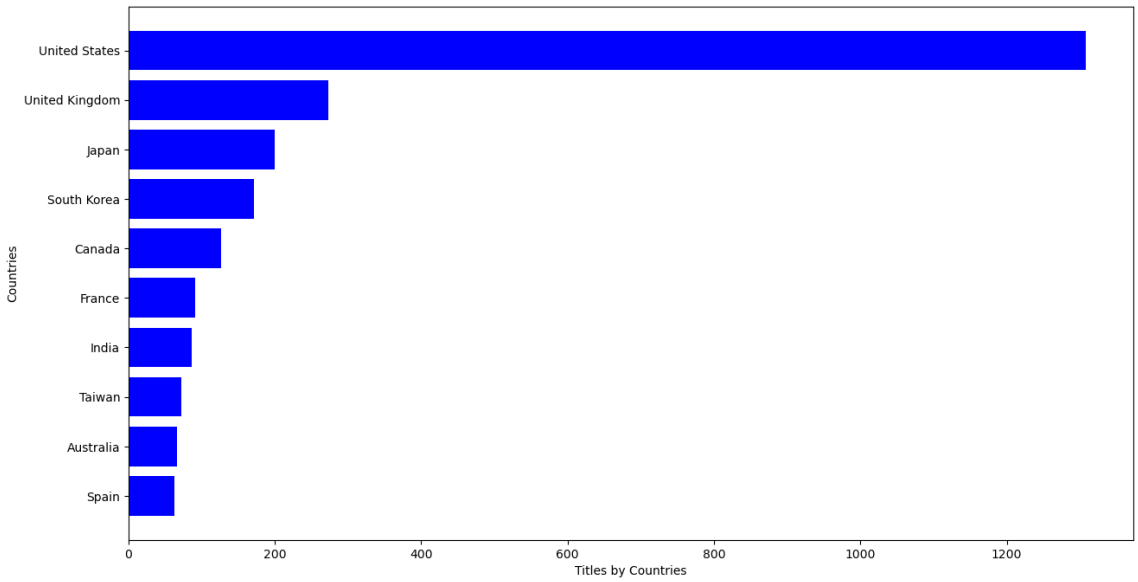
```
df_genre=df_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['green'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



- International Movies, Dramas and Comedy Genres are popular followed by Documentaries across Movies on Netflix

In [69]:

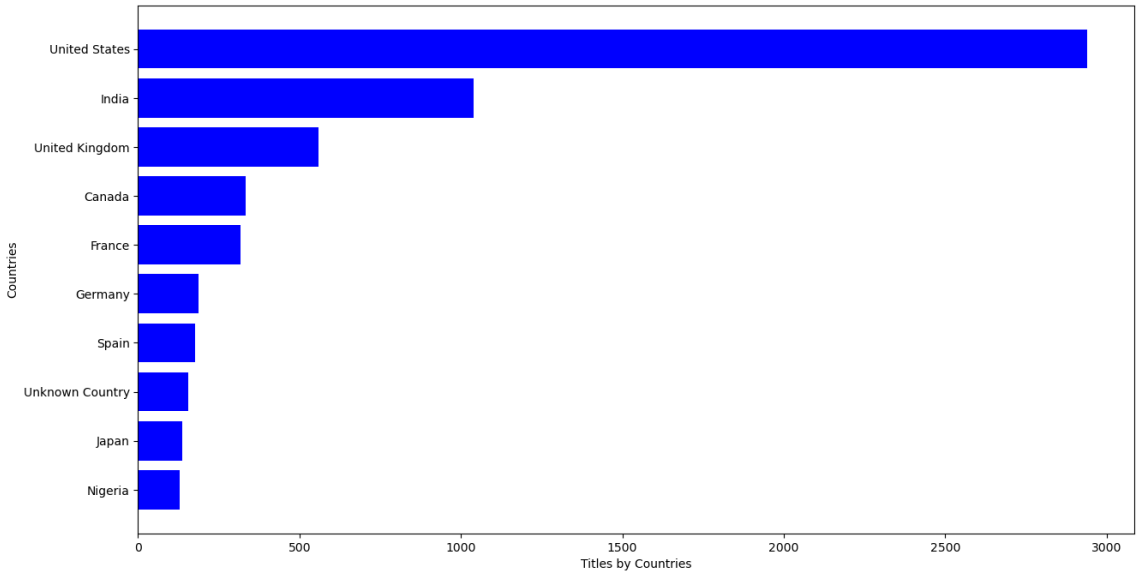
```
df_country=df_shows.groupby(['country']).agg({"title":"nunique"}).reset_index().sort_val
plt.figure(figsize=(15,8))
plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=['blue'])
plt.xlabel('Titles by Countries')
plt.ylabel('Countries')
plt.show()
```



US, UK and Japan produces most TV Shows on Netflix

In [70]:

```
df_country=df_movies.groupby(['country']).agg({"title":"nunique"}).reset_index().sort_va
plt.figure(figsize=(15,8))
plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=['blue'])
plt.xlabel('Titles by Countries')
plt.ylabel('Countries')
plt.show()
```

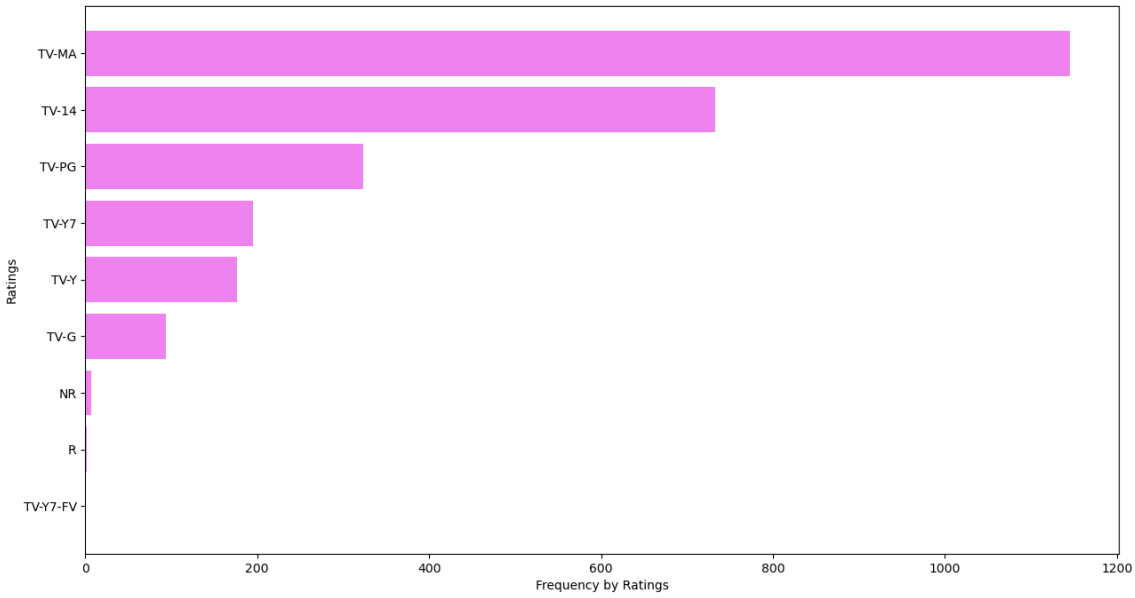


- United States is leading across both TV Shows and Movies, UK also provides great content across TV Shows and Movies.
- Surprisingly India is much more prevalent in Movies as compared TV Shows.

Moreover the number of Movies created in India outweigh the sum of TV Shows and Movies across UK since India was rated as second in net sum of whole content across Netflix.

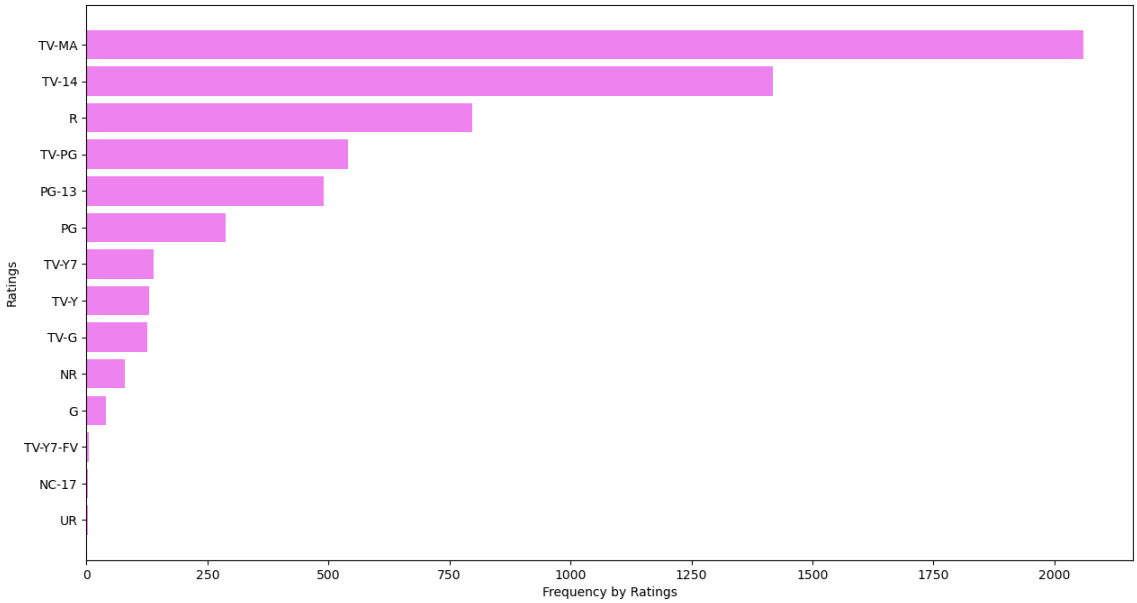
In [71]:

```
df_rating=df_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_value
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



In [72]:

```
df_rating=df_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_valu
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```

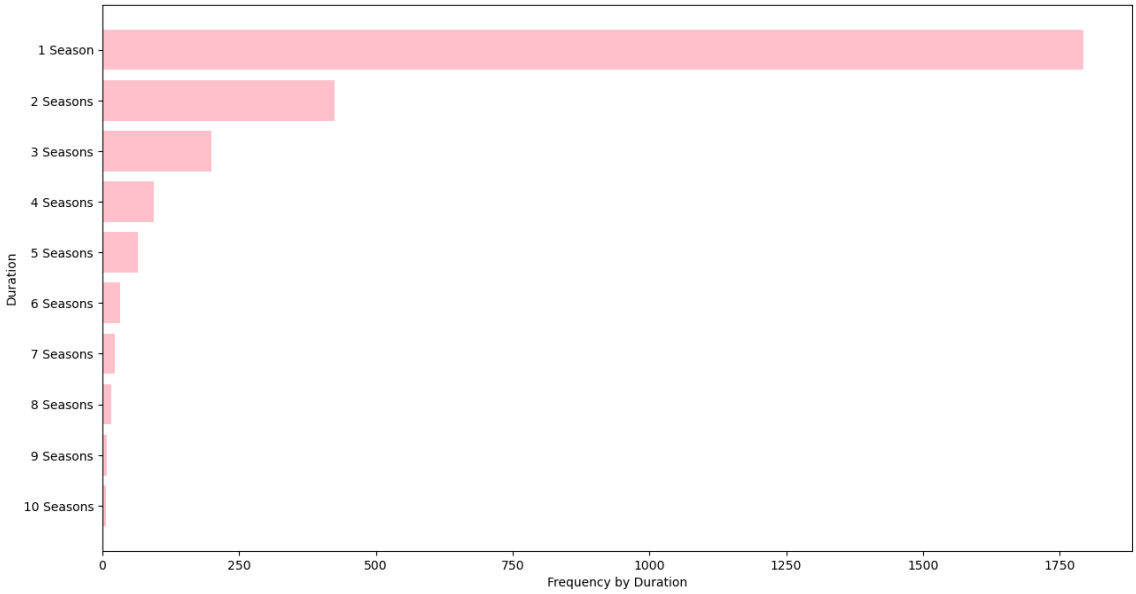


- So it seems plaussible to conclude that the popular ratings across Netflix includes Mature Audiences and those appropriate for over 14/over 17 ages.

Moreover there are no TV Shows having a rating of R

In [73]:

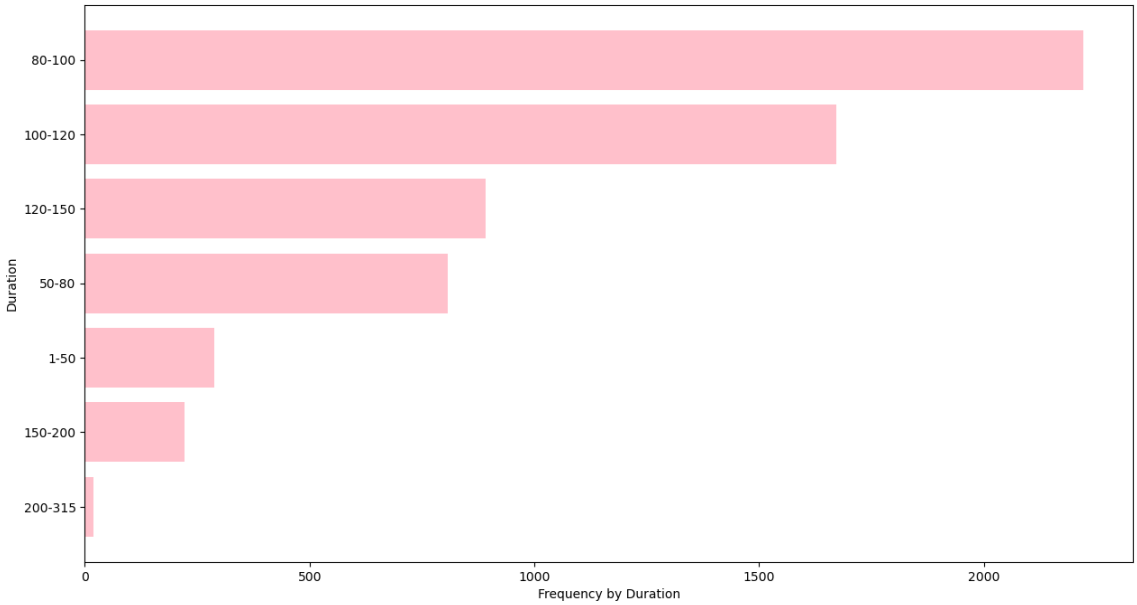
```
df_duration=df_shows.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort_v
plt.figure(figsize=(15,8))
plt.barh(df_duration[::-1]['duration'], df_duration[::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



- Across TV Shows, shows having only 1 Season are common as soon as the season length increases, the number of shows decrease and this definitely sounds as expected

In [74]:

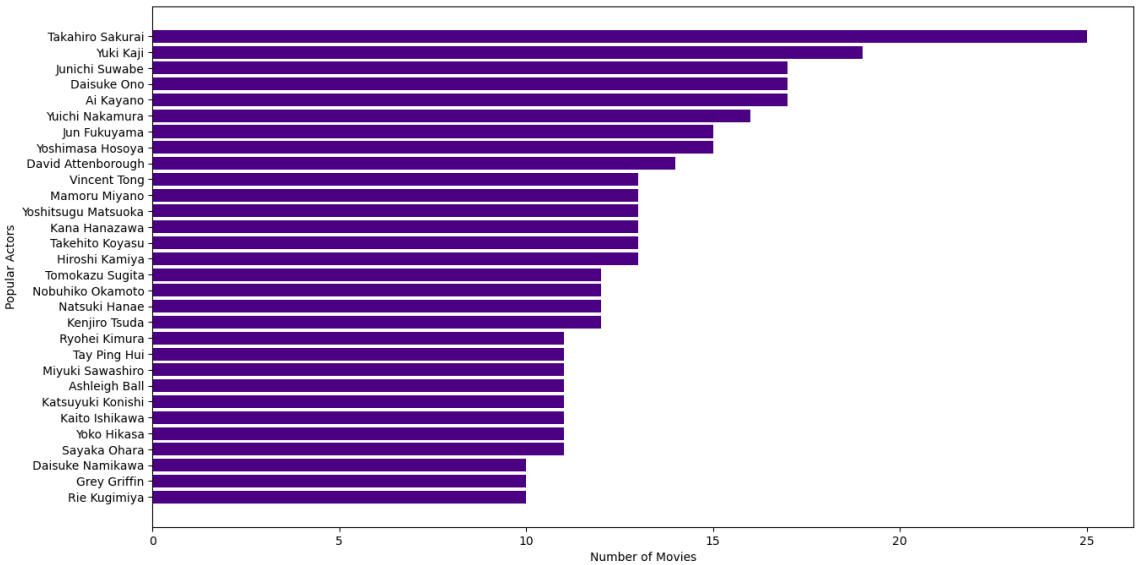
```
df_duration=df_movies.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort_
plt.figure(figsize=(15,8))
plt.barh(df_duration[::-1]['duration'], df_duration[::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



- Across movies 80-100,100-120 and 120-150 is the ranges of minutes for which most movies lie. So quite possibly 80-150 mins is the sweet spot we would be wanting for movies.

In [75]:

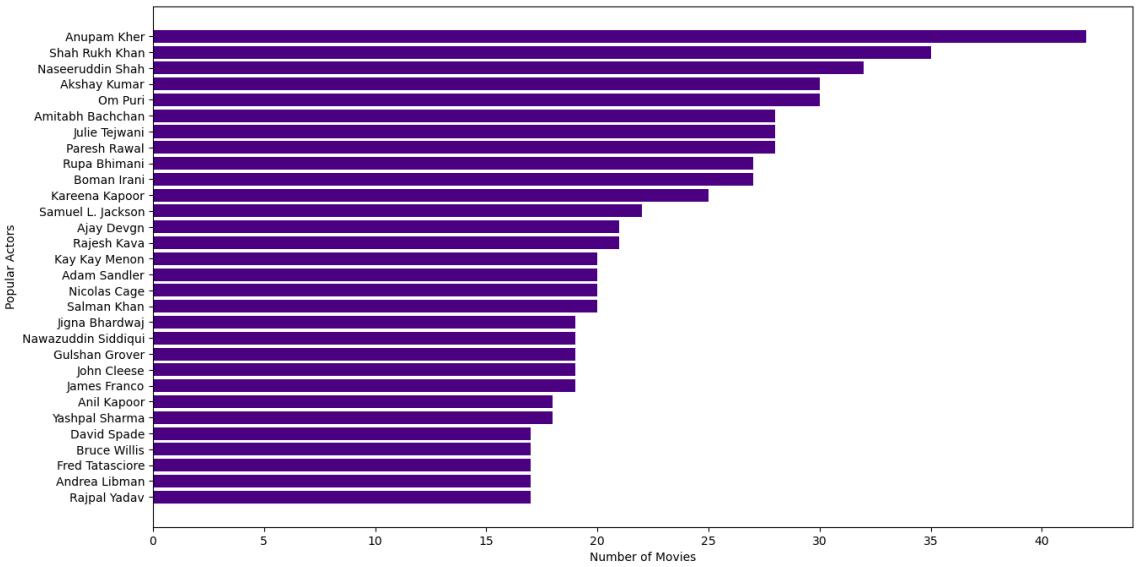
```
df_actors=df_shows.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_value
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['Actors'], df_actors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actors')
plt.show()
```



- Takahiro Sakurai,Yuki Kaji and other South Korean/Japanese actors are the most popular actors across TV Shows

In [76]:

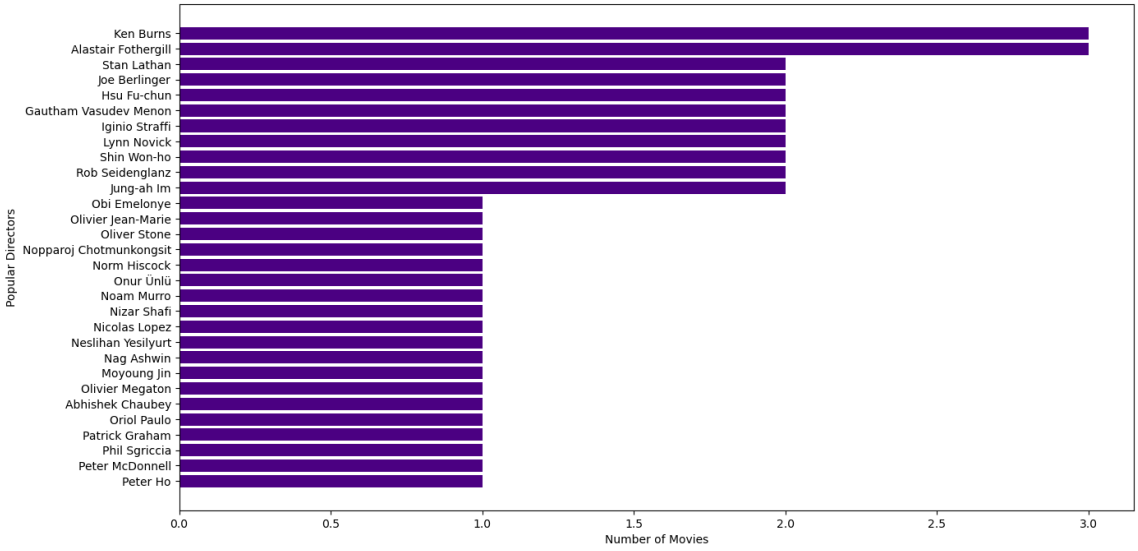
```
df_actors=df_movies.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_valu
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['Actors'], df_actors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actors')
plt.show()
```



- Our bollywood actors such as Anupam Kher, SRK, Naseeruddin Shah are very much popular acrossmovies on Netflix

In [77]:

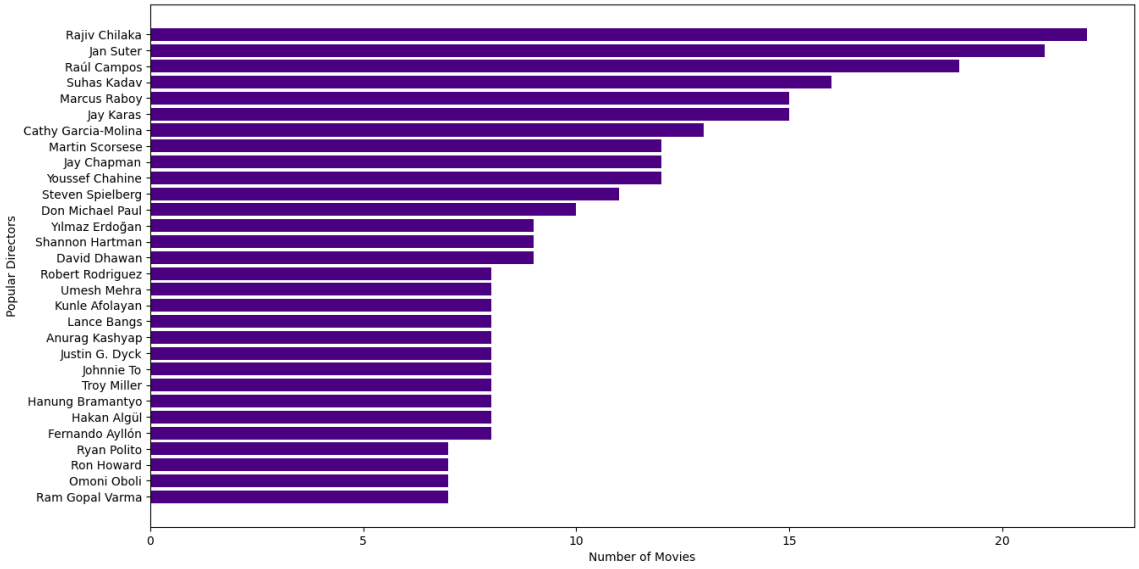
```
df_directors=df_shows.groupby(['Directors']).agg({"title":"nunique"}).reset_index().sort
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Directors')
plt.show()
```



- Ken Burns, Alastair Fothergill, Stan Lathan, Joe Barlinger are popular directors across TV Shows on Netflix

In [78]:

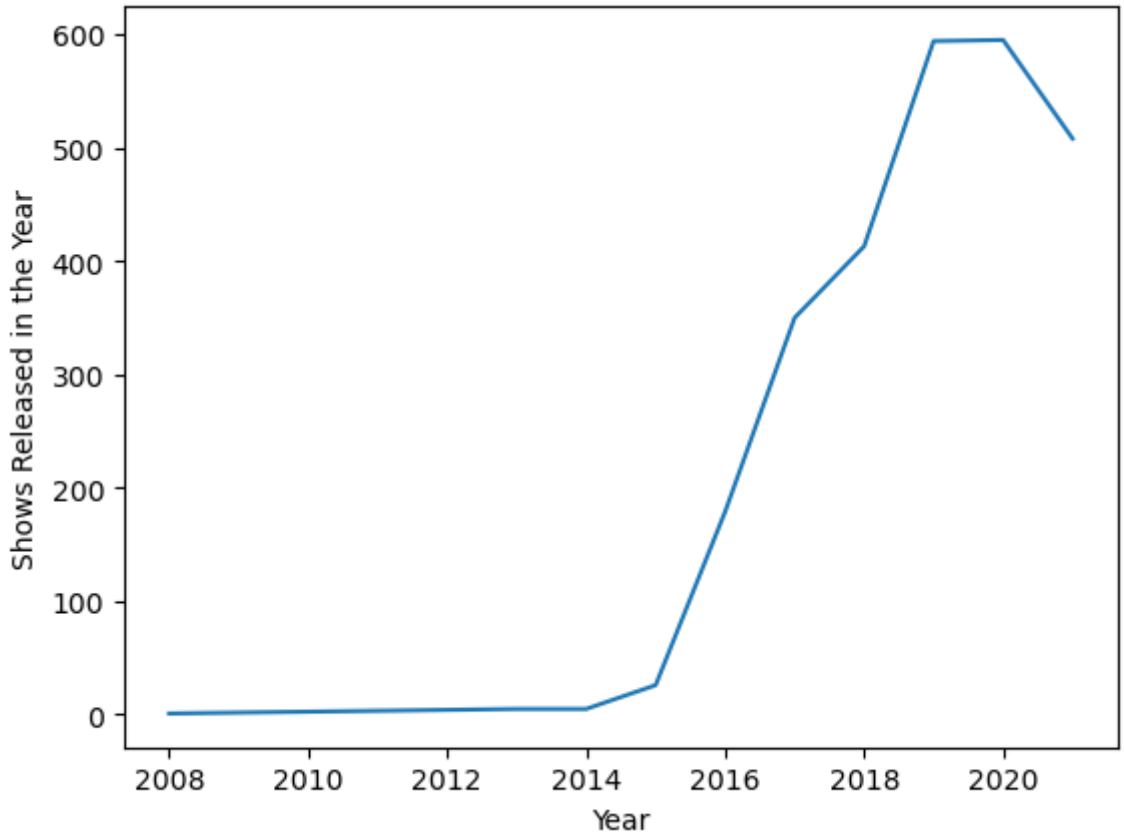
```
df_directors=df_movies.groupby(['Directors']).agg({"title":"nunique"}).reset_index().sor
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Directors')
plt.show()
```



- Rajiv Chilka, Jan Suter, Raul Campos, Suhas Kadav are popular directors across movies

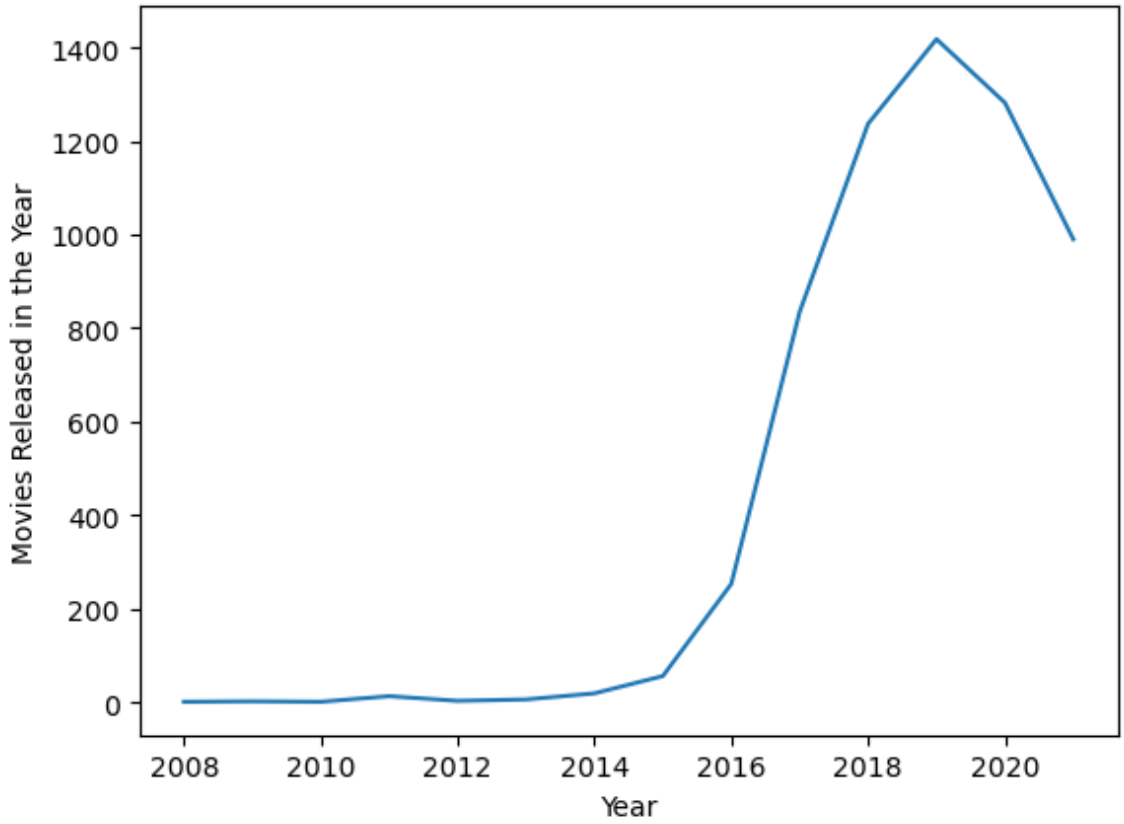
In [79]:

```
df_year=df_shows.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Shows Released in the Year")
plt.xlabel("Year")
plt.show()
```



In [80]:

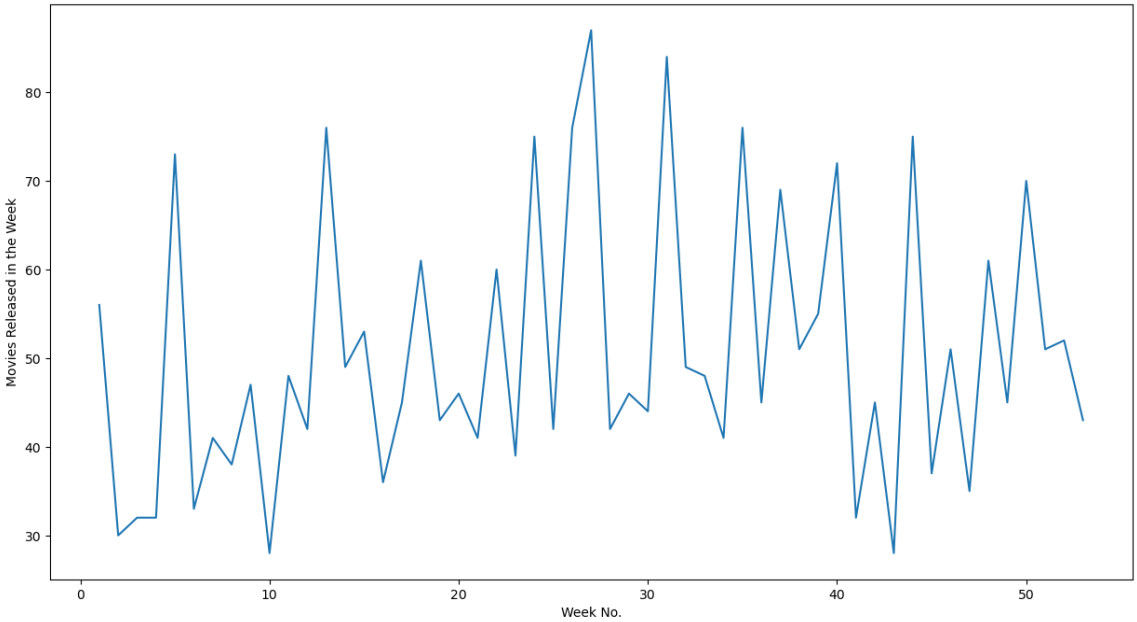
```
df_year=df_movies.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



- Till 2019, overall content across Netflix was increasing but due to Covid in 2020, though TV Shows didn't take a hit then Movies did take a hit. Well later in 2021, content across both was reduced significantly

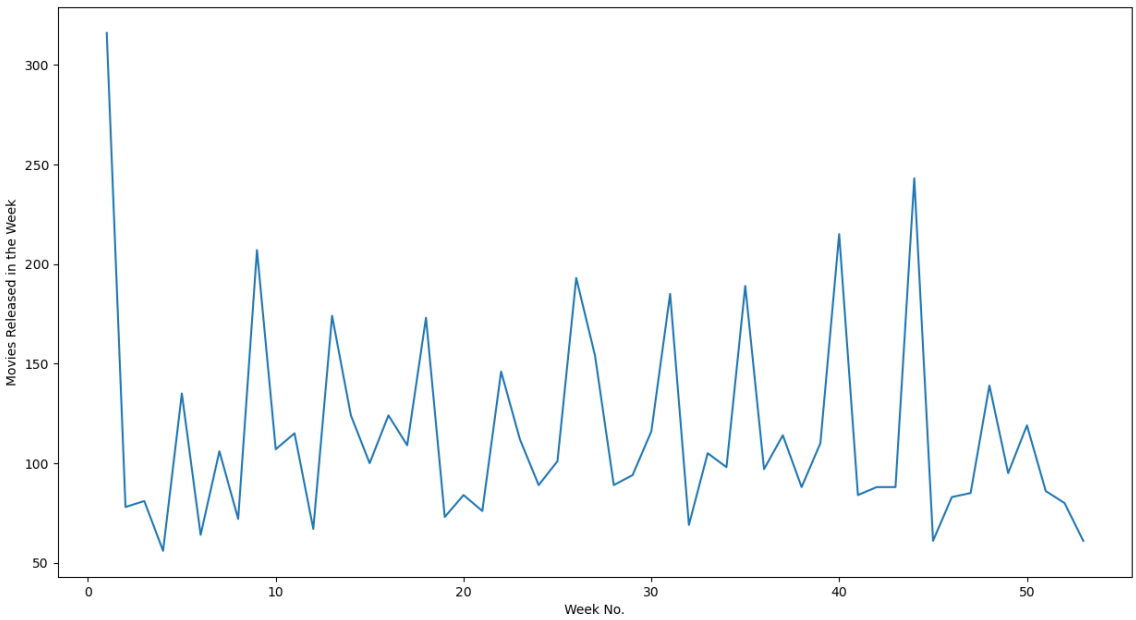
In [81]:

```
df_week=df_shows.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



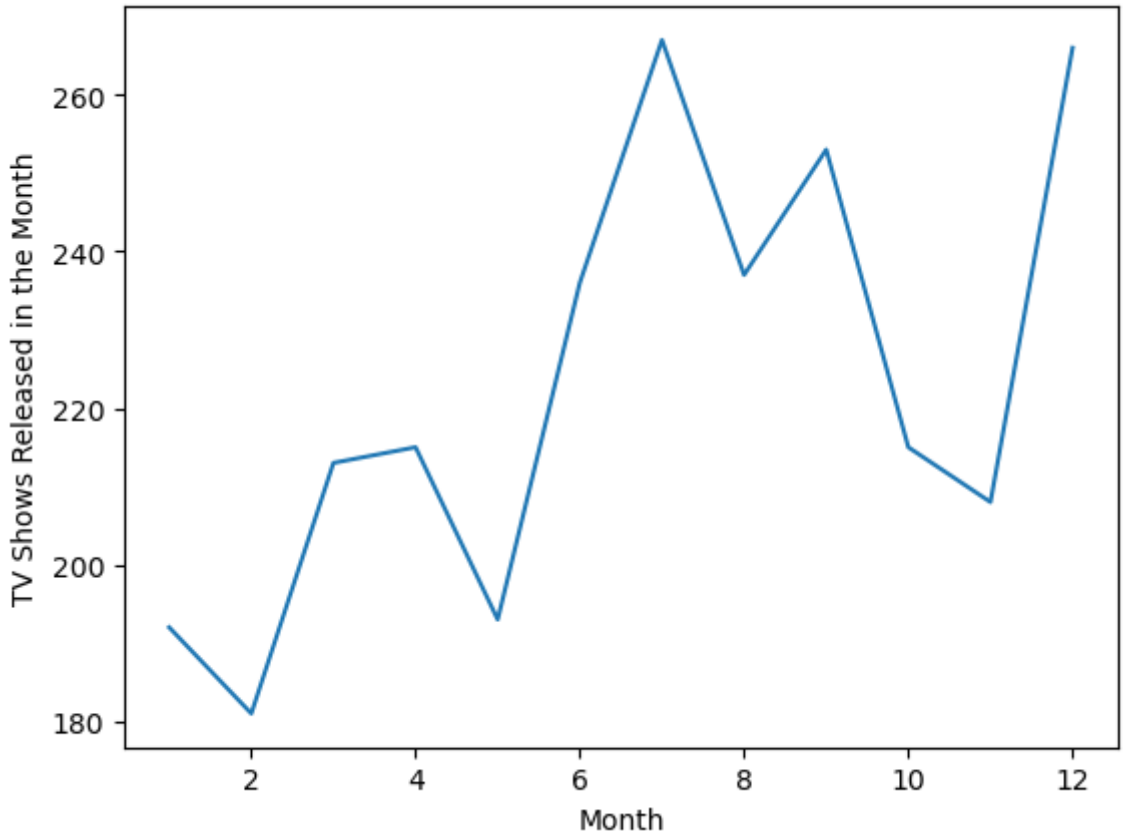
In [82]:

```
df_week=df_movies.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



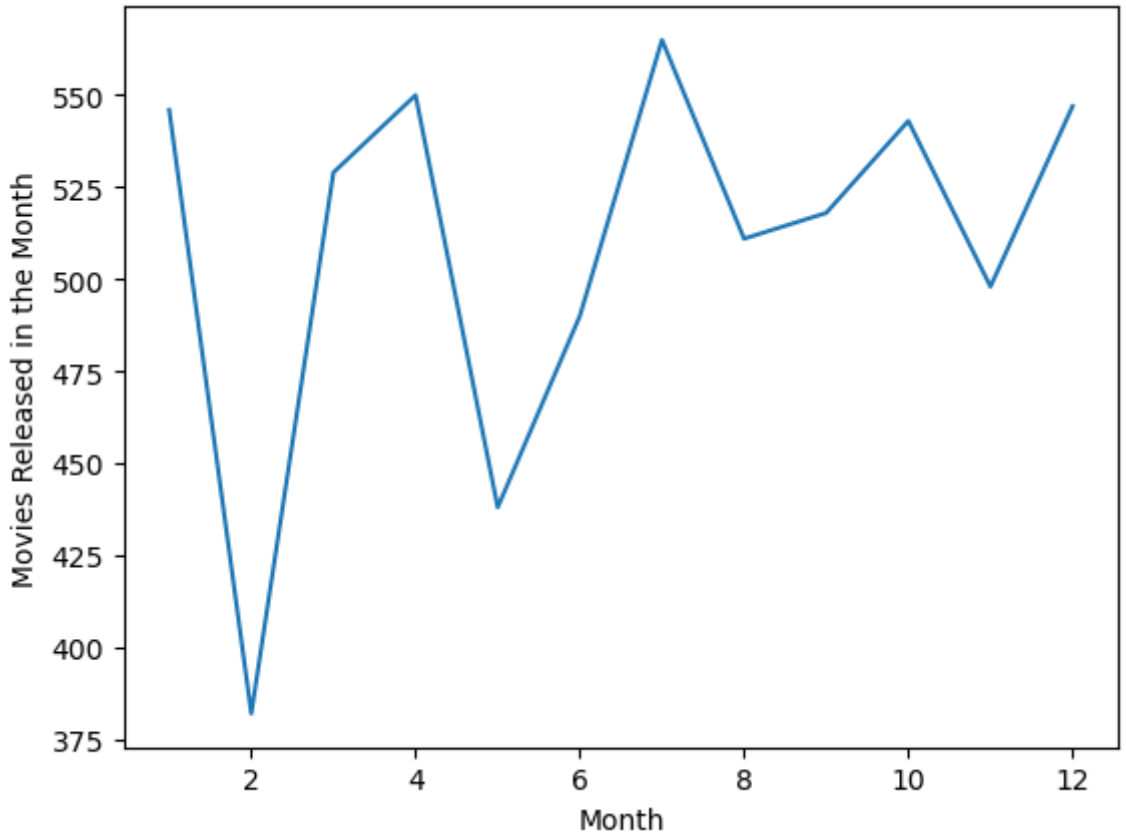
In [83]:

```
df_month=df_shows.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("TV Shows Released in the Month")
plt.xlabel("Month")
plt.show()
```



In [84]:

```
df_month=df_movies.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("Movies Released in the Month")
plt.xlabel("Month")
plt.show()
```

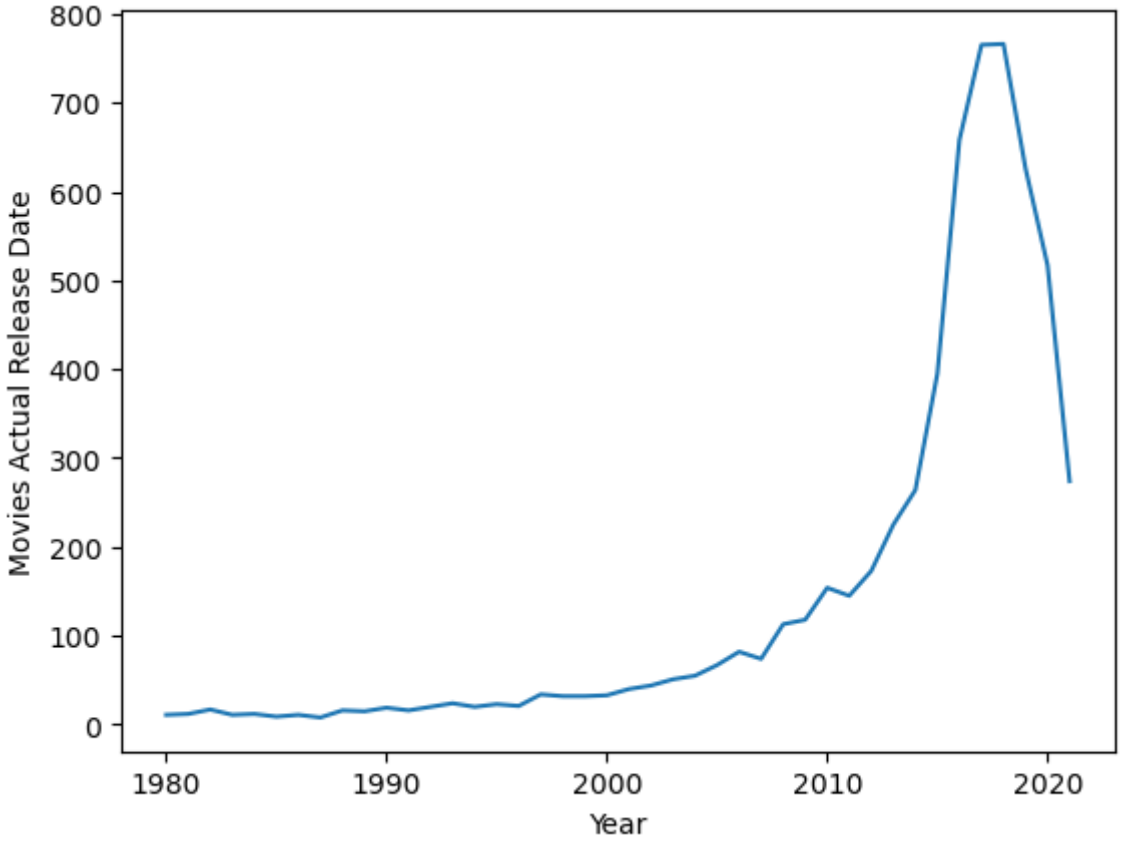


- TV Shows are added in Netflix by a tremendous amount in mid weeks/months of the year, i.e- July

Movies are added in Netflix by a tremendous amount in first week/last month of current year and first month of next year

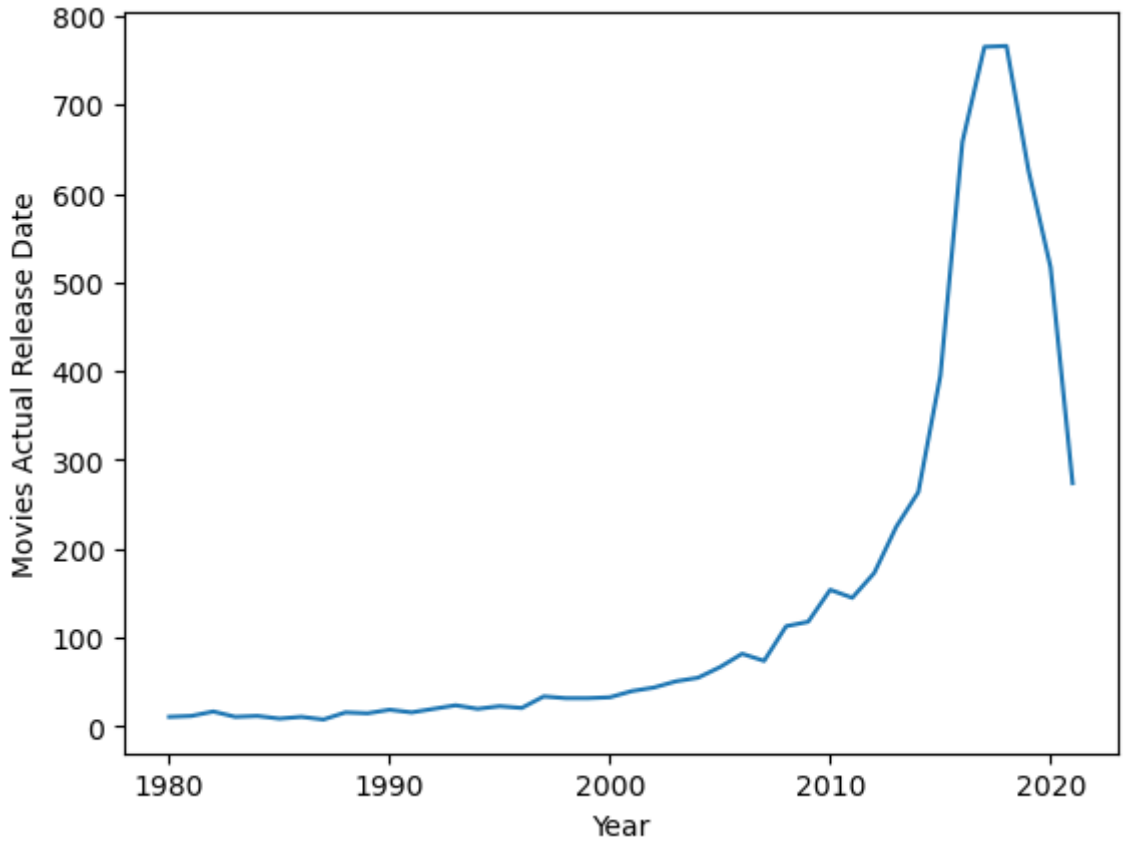
In [85]:

```
df_release_year=df_movies[df_movies['release_year']>=1980].groupby(['release_year']).agg
sns.lineplot(data=df_release_year, x='release_year', y='title')
plt.ylabel("Movies Actual Release Date")
plt.xlabel("Year")
plt.show()
```



In [86]:

```
df_release_year=df_movies[df_movies['release_year']>=1980].groupby(['release_year']).agg
sns.lineplot(data=df_release_year, x='release_year', y='title')
plt.ylabel("Movies Actual Release Date")
plt.xlabel("Year")
plt.show()
```



- Actual Releases of both TV Shows and Movies have taken a hit after 2020

Analysis for Recommendations

In [87]:

```
#below countries will be analyzed for both shows and movies
shows_and_movies=['United States','India','United Kingdom']
#below countries will be only analyzed on basis of shows
only_shows=['Japan','South Korea']
```

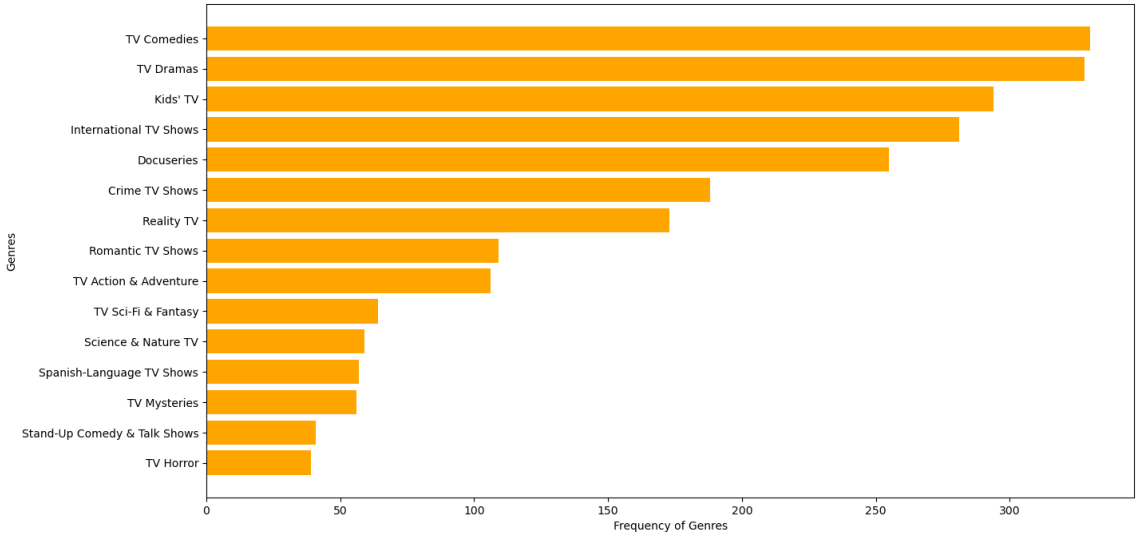
Univariate Analysis separately for shows and movies in USA

In [88]:

```
#Analyzing USA for both shows and movies
df_usa_shows=df_final1[df_final1['country']=='United States'][df_final1[df_final1['count
df_usa_movies=df_final1[df_final1['country']=='United States'][df_final1[df_final1['coun
```

In [89]:

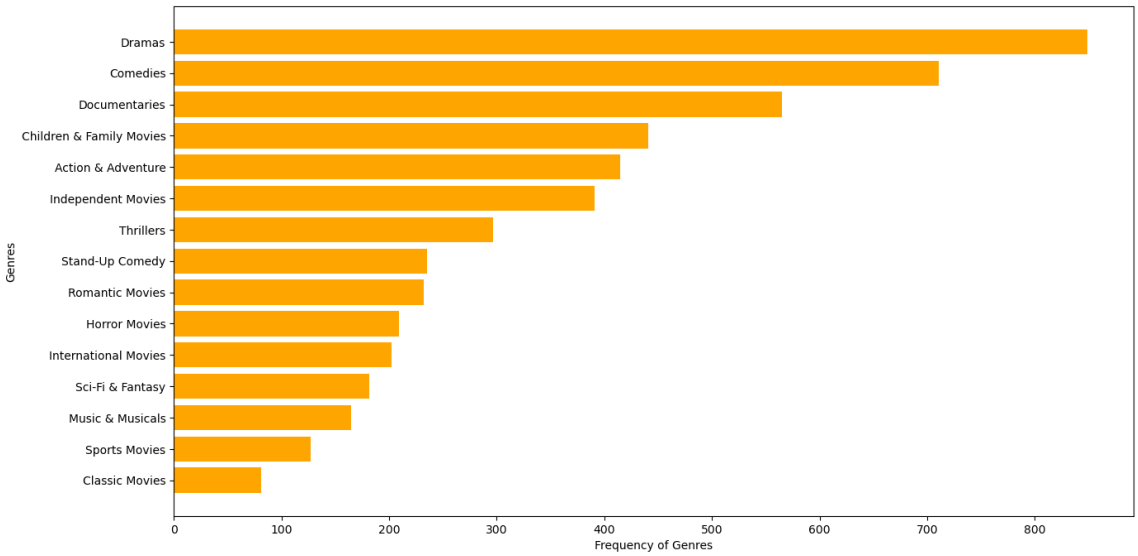
```
df_genre=df_usa_shows.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_val
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



- Dramas,Comedy, Kids 'TV Shows, International TV Shows and Docuseries, Genres are popular in TV Series in USA

In [91]:

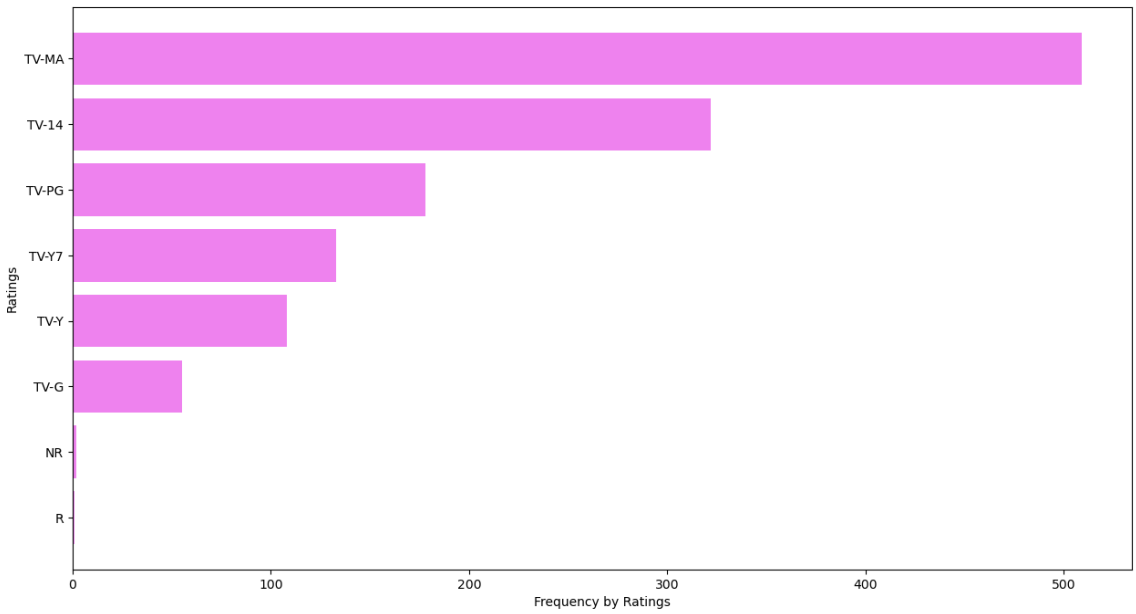
```
df_genre=df_usa_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_va
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



- Dramas,Comedy, Documentaries, Family Movies and Action Genres in Movies are popular in USA

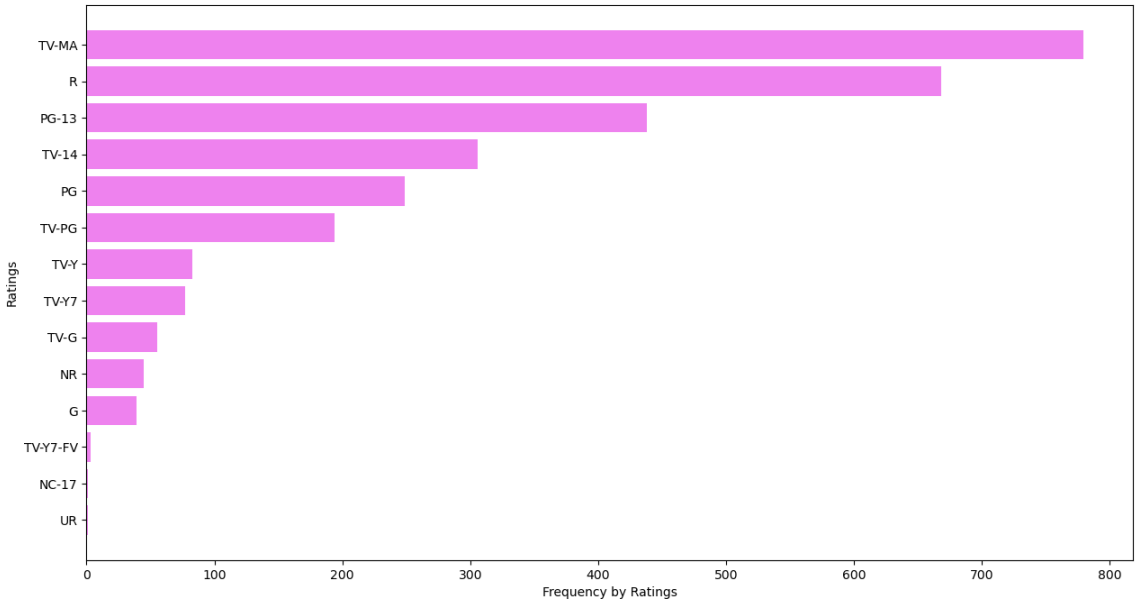
In [92]:

```
df_rating=df_usa_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_v
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



In [93]:

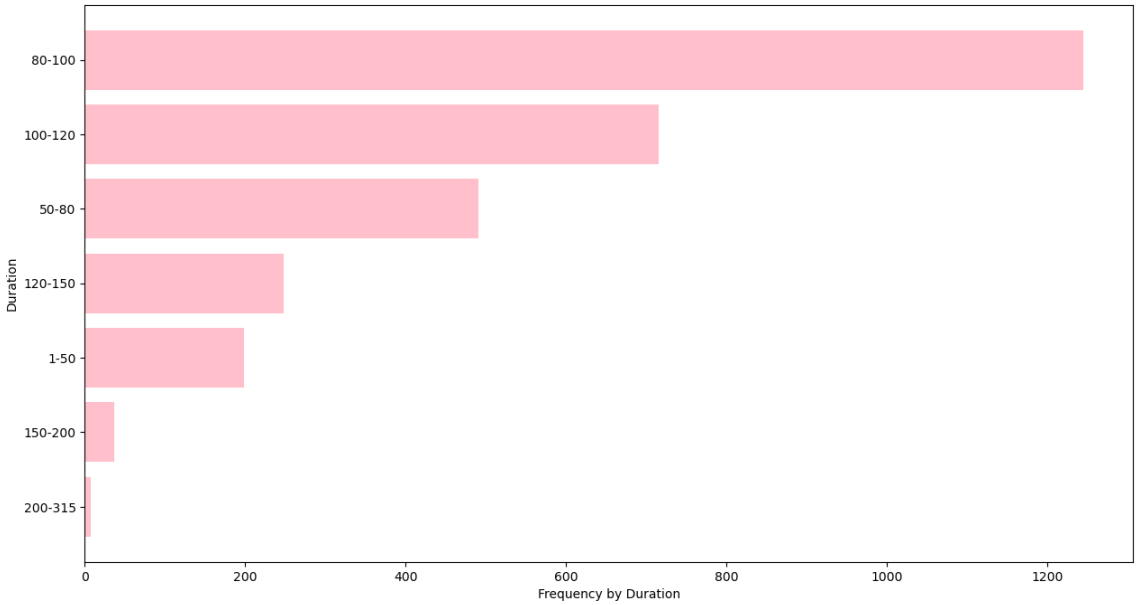
```
df_rating=df_usa_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



- So it seems plaussible to conclude that the popular ratings across Netflix includes Mature Audiences and those appropriate for over 14/over 17 ages in both Movies and TV Shows in USA

In [94]:

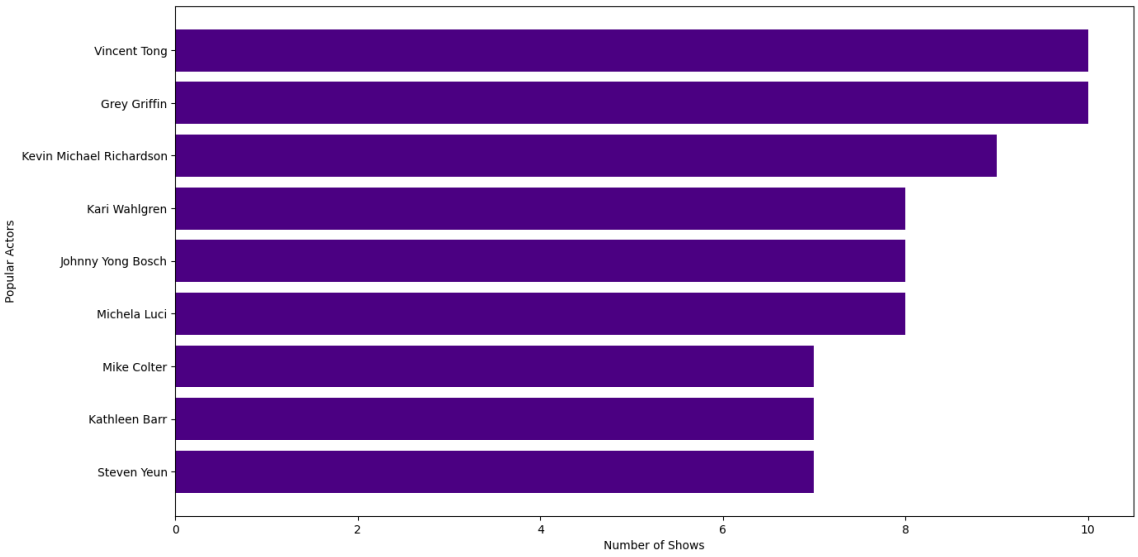
```
df_duration=df_usa_movies.groupby(['duration']).agg({"title":"nunique"}).reset_index().s
plt.figure(figsize=(15,8))
plt.barh(df_duration[::-1]['duration'], df_duration[::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



- Across movies 80-100,100-120 is the ranges of minutes for which most movies lie. So quite possibly 80-120 mins is the sweet spot we would be wanting for movies in USA

In [95]:

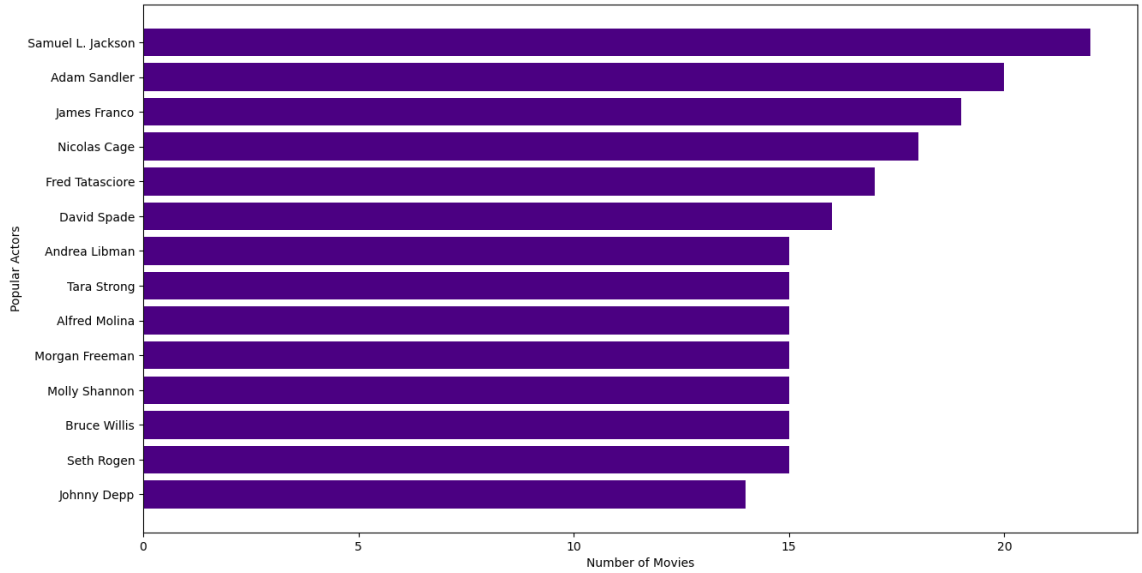
```
df_actors=df_usa_shows.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_v
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['Actors'], df_actors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Shows')
plt.ylabel('Popular Actors')
plt.show()
```



- Vincent Tong,Grey Griffin and Kevin Richardson are the most popular actors across TV Shows in USA

In [96]:

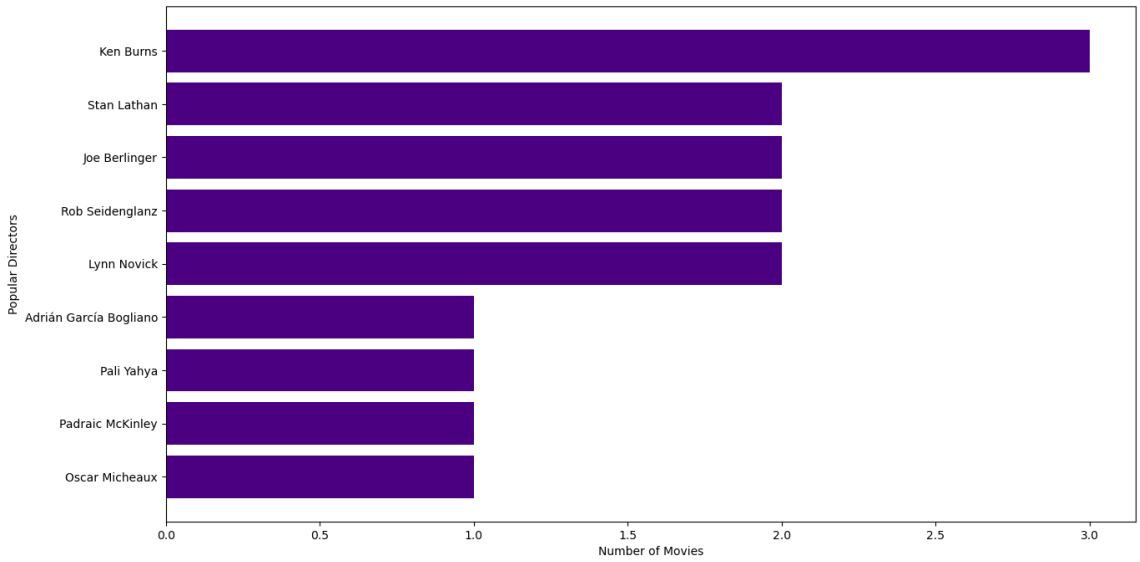
```
df_actors=df_usa_movies.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['Actors'], df_actors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actors')
plt.show()
```



- Samuel Jackson,Adam Sandler,James Franco and Nicolas Cage are very much popular across movies on Netflix in USA

In [97]:

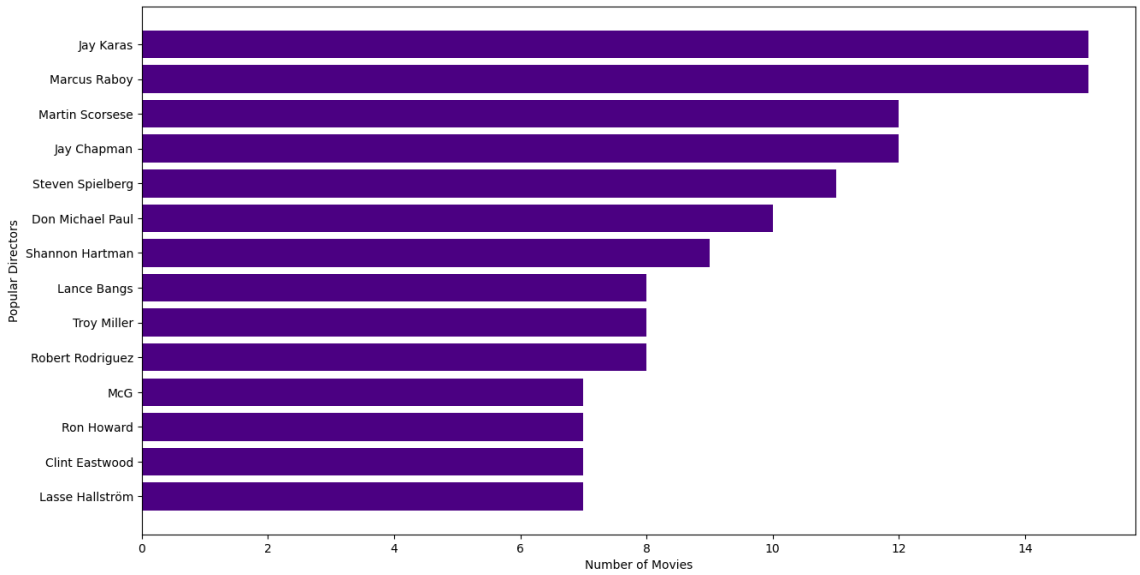
```
df_directors=df_usa_shows.groupby(['Directors']).agg({"title":"nunique"}).reset_index().
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Directors')
plt.show()
```



- Ken Burns,Stan Lathan, Joe Barlinger are popular directors across TV Shows on Netflix in USA

In [98]:

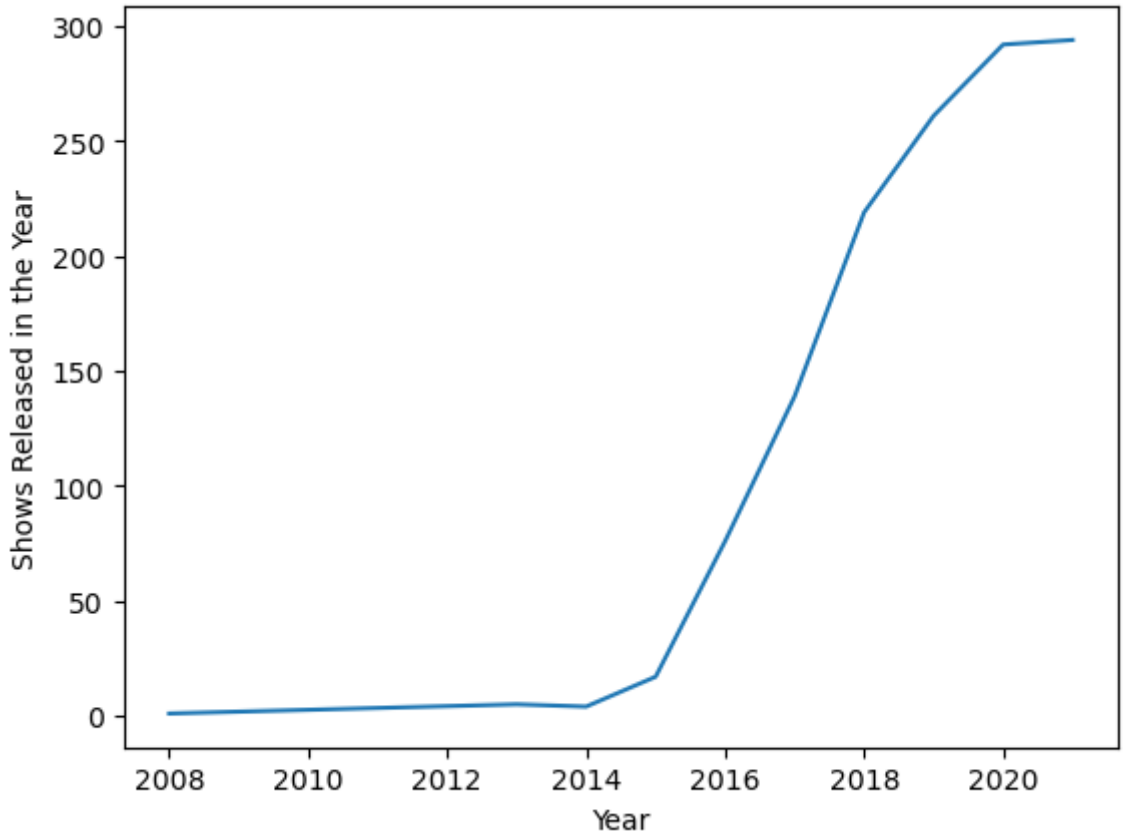
```
df_directors=df_usa_movies.groupby(['Directors']).agg({"title":"nunique"}).reset_index()
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Directors')
plt.show()
```



- Jay Karas,Marcus Raboy,Martin Scorcese and Jay Chapman are popular directors across movies in USA

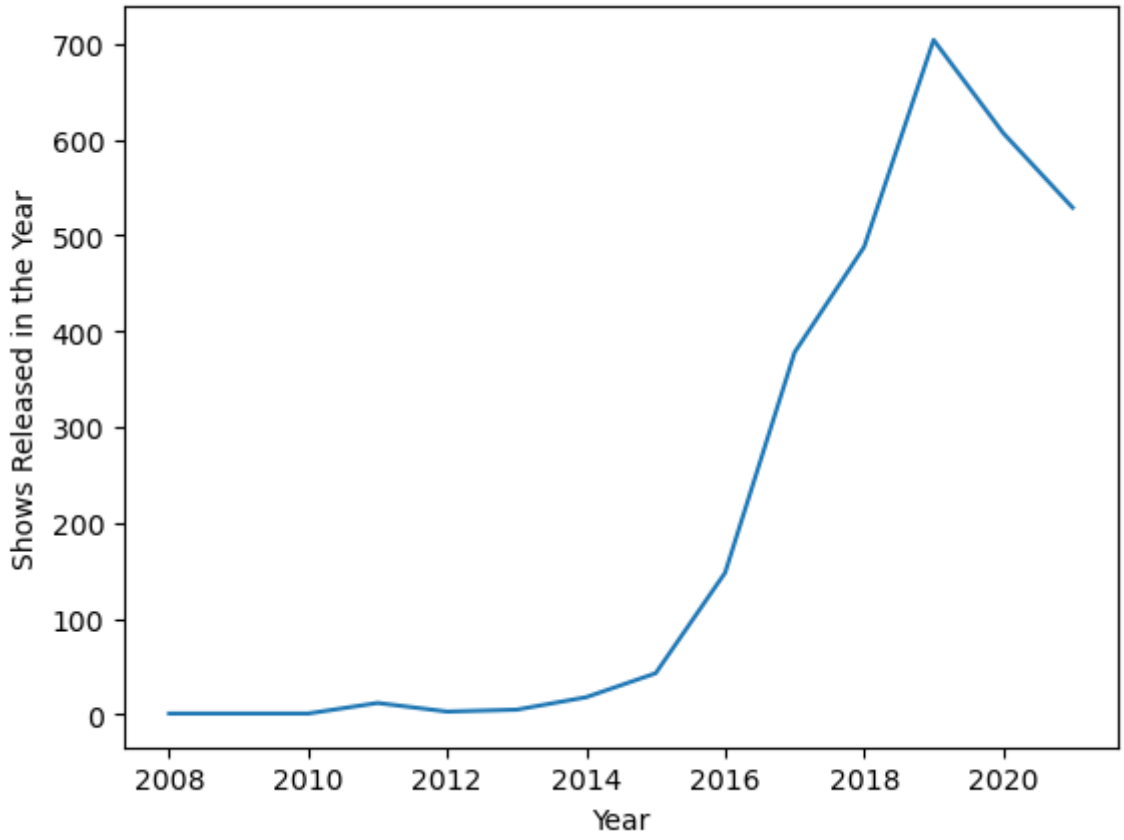
In [99]:

```
df_year=df_usa_shows.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Shows Released in the Year")
plt.xlabel("Year")
plt.show()
```



In [100]:

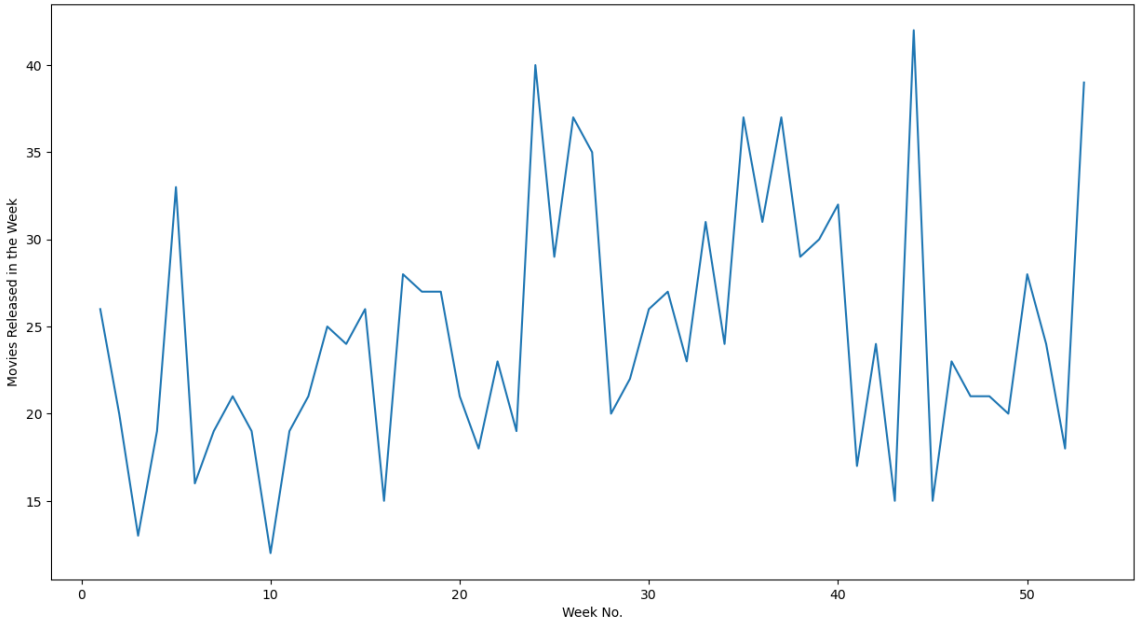
```
df_year=df_usa_movies.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Shows Released in the Year")
plt.xlabel("Year")
plt.show()
```



- In USA, number of shows remained the same in 2021 as they were in 2020 while number of movies declined:

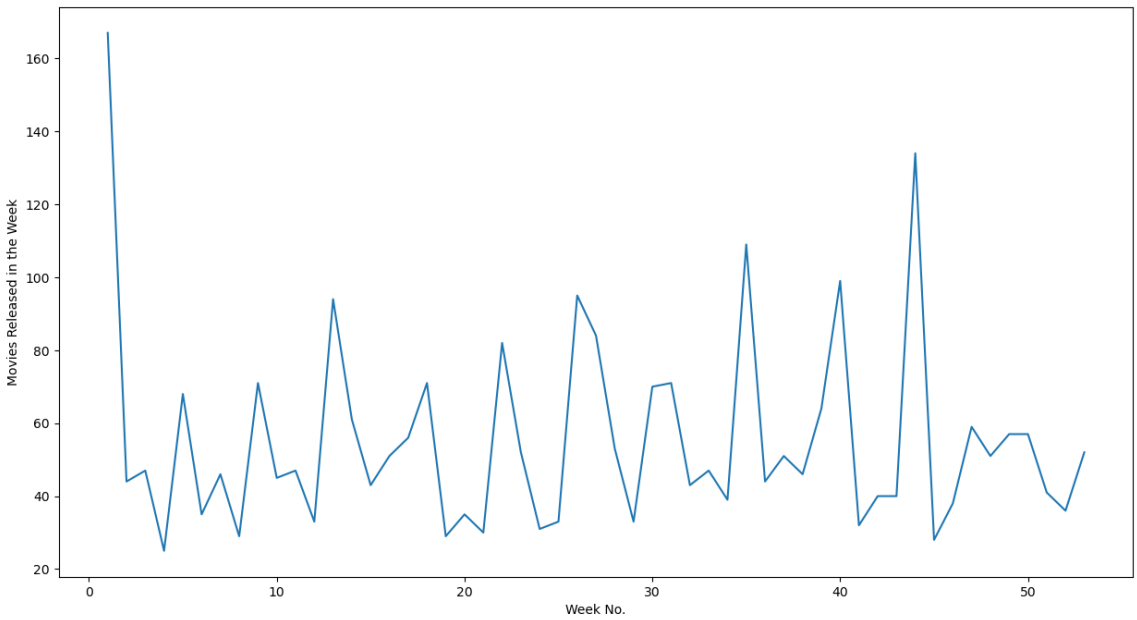
In [101]:

```
df_week=df_usa_shows.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



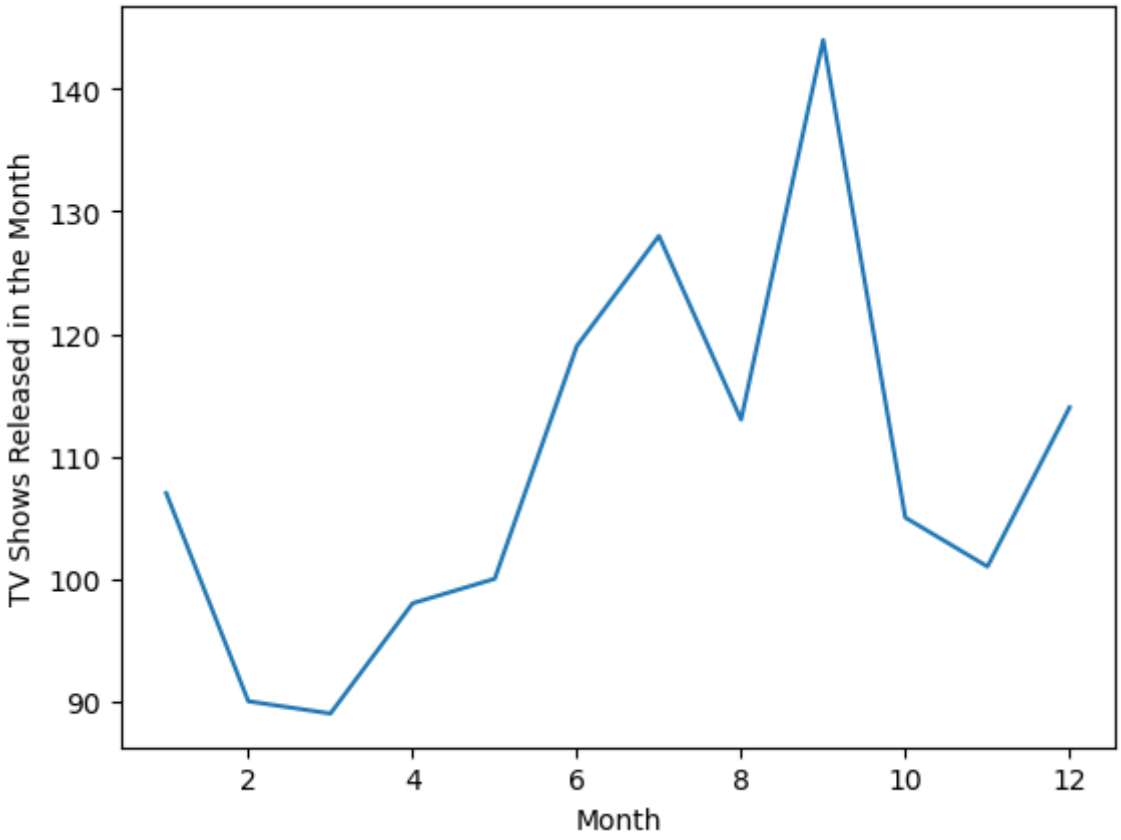
In [102]:

```
df_week=df_usa_movies.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



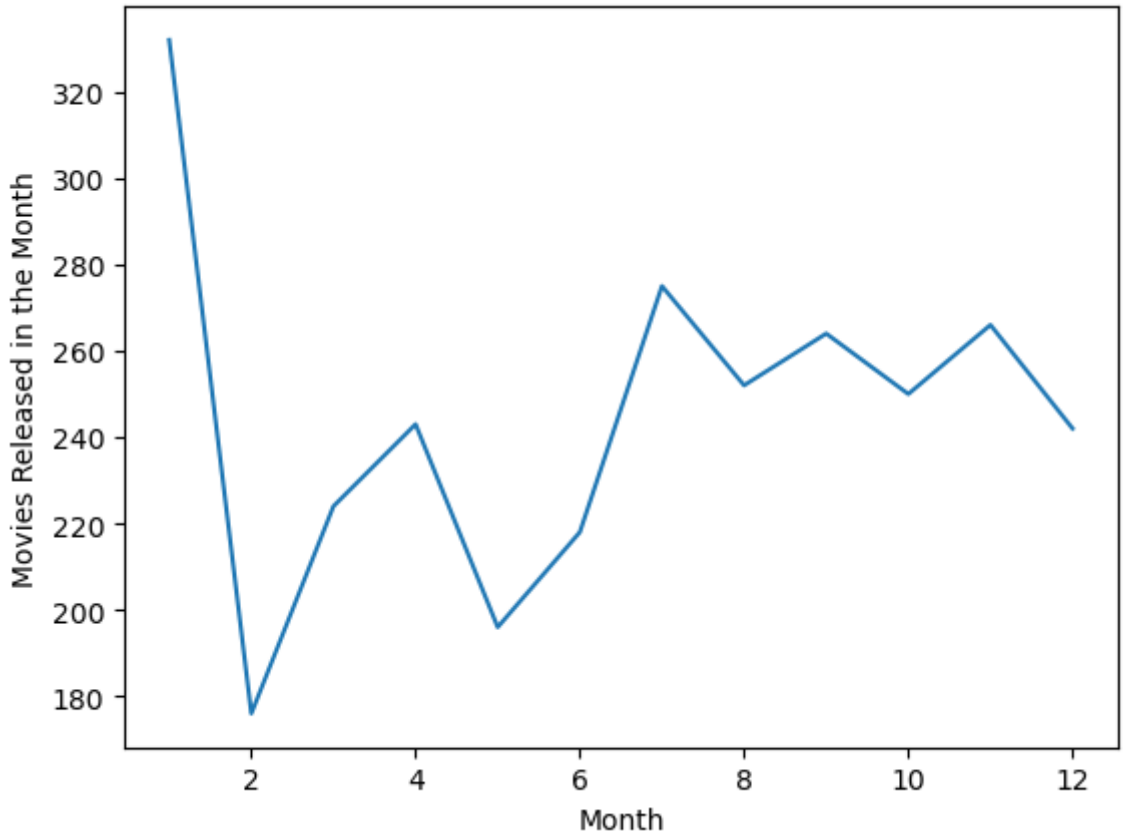
In [103]:

```
df_month=df_usa_shows.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("TV Shows Released in the Month")
plt.xlabel("Month")
plt.show()
```



In [104]:

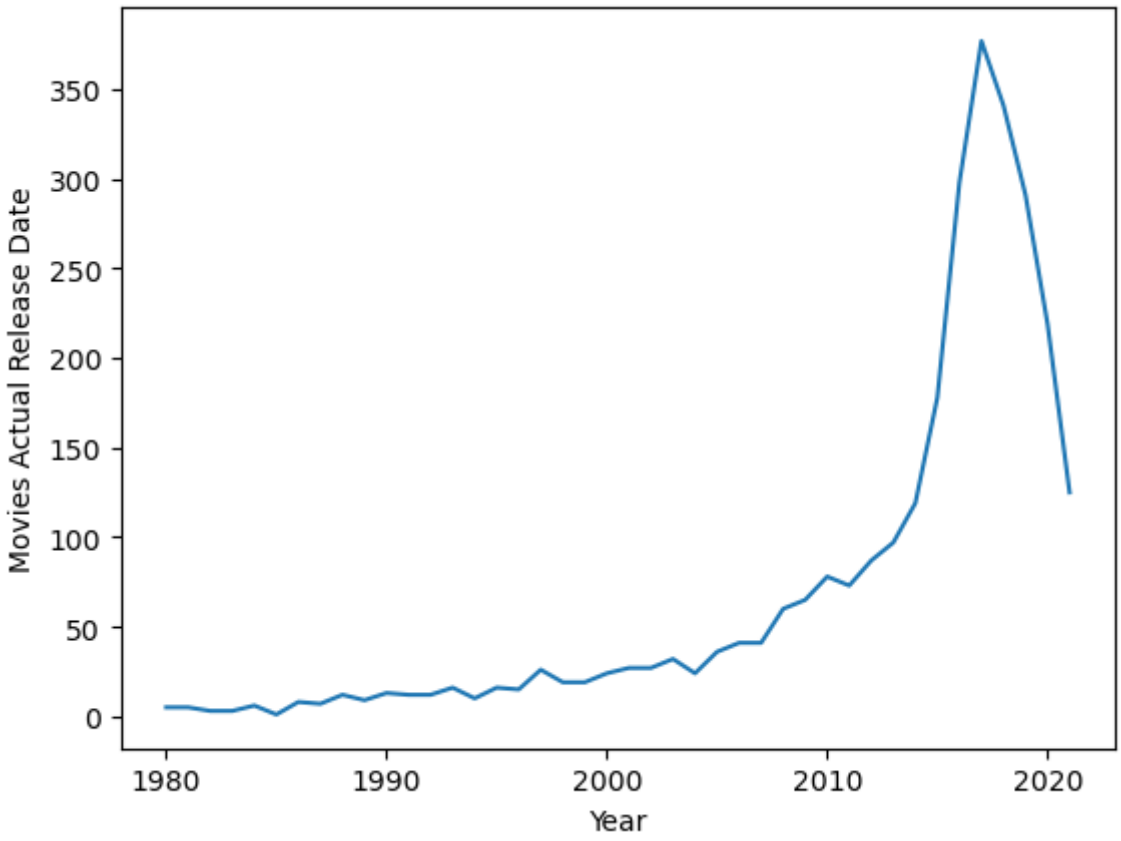
```
df_month=df_usa_movies.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("Movies Released in the Month")
plt.xlabel("Month")
plt.show()
```



- TV Shows are added in Netflix by a tremendous amount in July and September in USA
- Movies are added in Netflix in USA by a tremendous amount in first week/last month of current year and first month of next year

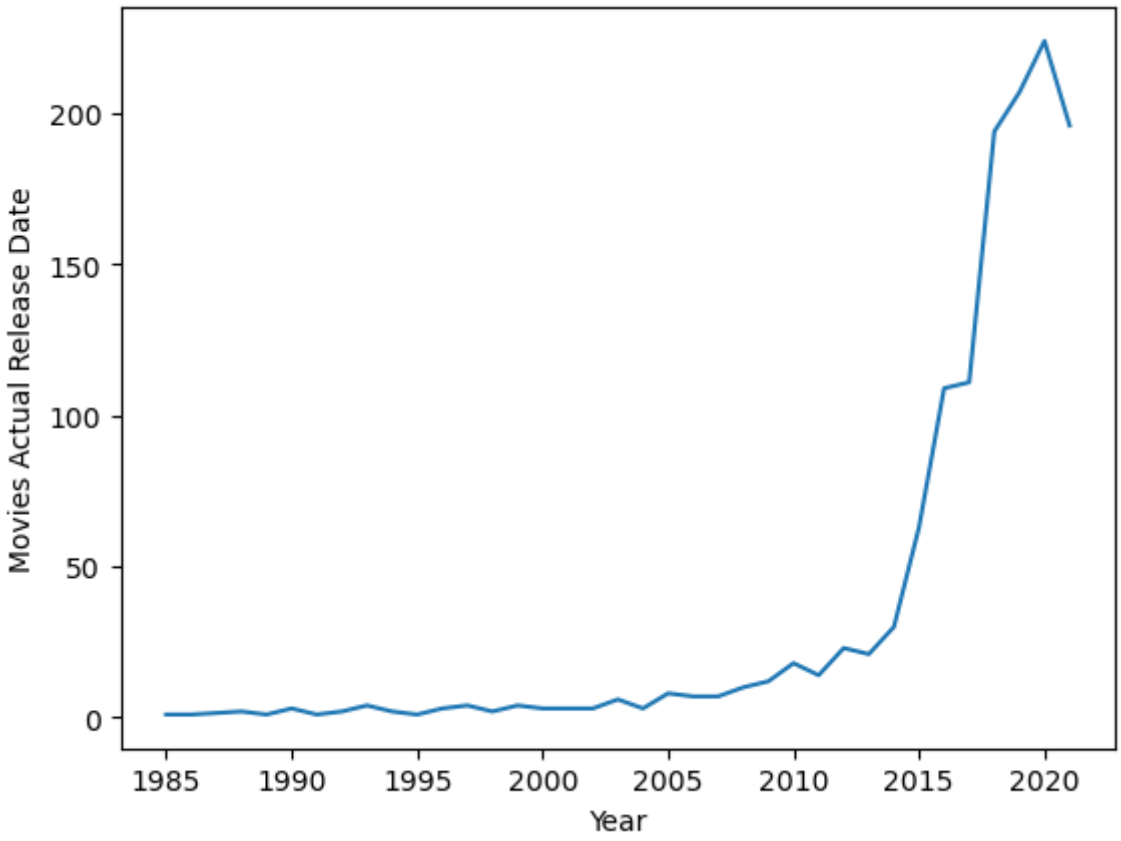
In [105]:

```
df_release_year=df_usa_movies[df_usa_movies['release_year']>=1980].groupby(['release_yea
sns.lineplot(data=df_release_year, x='release_year', y='title')
plt.ylabel("Movies Actual Release Date")
plt.xlabel("Year")
plt.show()
```



In [106]:

```
df_release_year=df_usa_shows[df_usa_shows['release_year']>=1980].groupby(['release_year'
sns.lineplot(data=df_release_year, x='release_year', y='title')
plt.ylabel("Movies Actual Release Date")
plt.xlabel("Year")
plt.show()
```



- In USA, though both Movies and Shows have reduced in 2021, the amount of decrease in number of TV Shows is small as compared to Movies

In [107]:

```
df_usa_movies.head()
```

Out[107]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	re
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	
160	My Little Pony: A New Generation	Vanessa Hudgens	José Luis Ucha	Children & Family Movies	United States	s7	Movie	September 24, 2021	
161	My Little Pony: A New Generation	Kimiko Glenn	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	
162	My Little Pony: A New Generation	Kimiko Glenn	José Luis Ucha	Children & Family Movies	United States	s7	Movie	September 24, 2021	

In [108]:

```
#Analysing a combination of actors and directors
df_usa_movies['Actor_Director_Combination'] = df_usa_movies.Actors.str.cat(df_usa_movies
df_usa_movies_subset=df_usa_movies[df_usa_movies['Actors']!='Unknown Actor']
df_usa_movies_subset=df_usa_movies_subset[df_usa_movies_subset['Directors']!='Unknown Di
df_usa_movies_subset.head()
```

Out[108]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	
160	My Little Pony: A New Generation	Vanessa Hudgens	José Luis Ucha	Children & Family Movies	United States	s7	Movie	September 24, 2021	
161	My Little Pony: A New Generation	Kimiko Glenn	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	
162	My Little Pony: A New Generation	Kimiko Glenn	José Luis Ucha	Children & Family Movies	United States	s7	Movie	September 24, 2021	
163	My Little Pony: A New Generation	James Marsden	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	

In [109]:

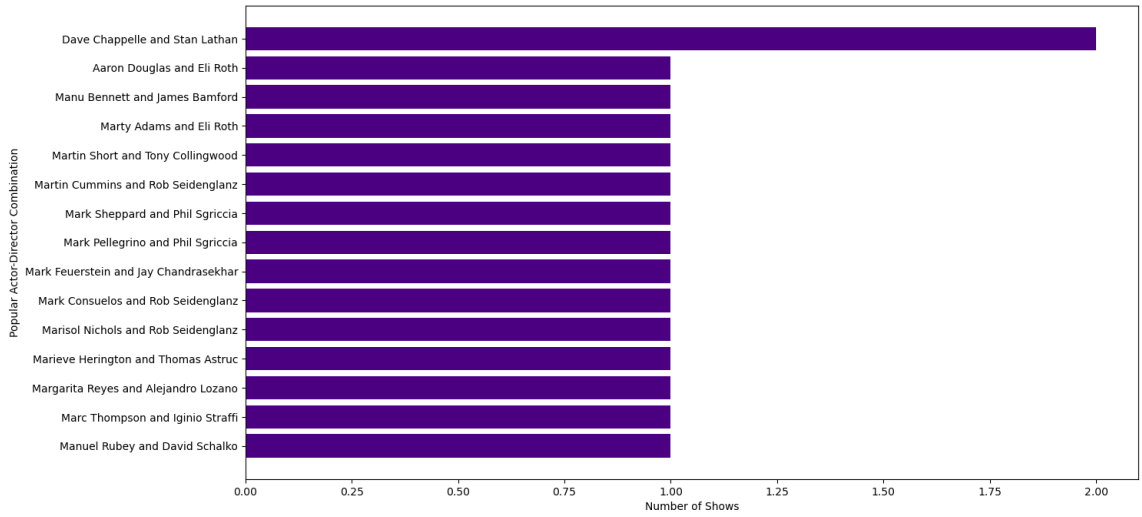
```
df_usa_shows['Actor_Director_Combination'] = df_usa_shows.Actors.str.cat(df_usa_shows.Di
df_usa_shows_subset=df_usa_shows[df_usa_shows['Actors']!='Unknown Actor']
df_usa_shows_subset=df_usa_shows_subset[df_usa_shows_subset['Directors']!='Unknown Direc
df_usa_shows_subset.head()
```

Out[109]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_yea
111	Midnight Mass	Kate Siegel	Mike Flanagan	TV Dramas	United States	s6	TV Show	September 24, 2021	202
112	Midnight Mass	Kate Siegel	Mike Flanagan	TV Horror	United States	s6	TV Show	September 24, 2021	202
113	Midnight Mass	Kate Siegel	Mike Flanagan	TV Mysteries	United States	s6	TV Show	September 24, 2021	202
114	Midnight Mass	Zach Gilford	Mike Flanagan	TV Dramas	United States	s6	TV Show	September 24, 2021	202
115	Midnight Mass	Zach Gilford	Mike Flanagan	TV Horror	United States	s6	TV Show	September 24, 2021	202

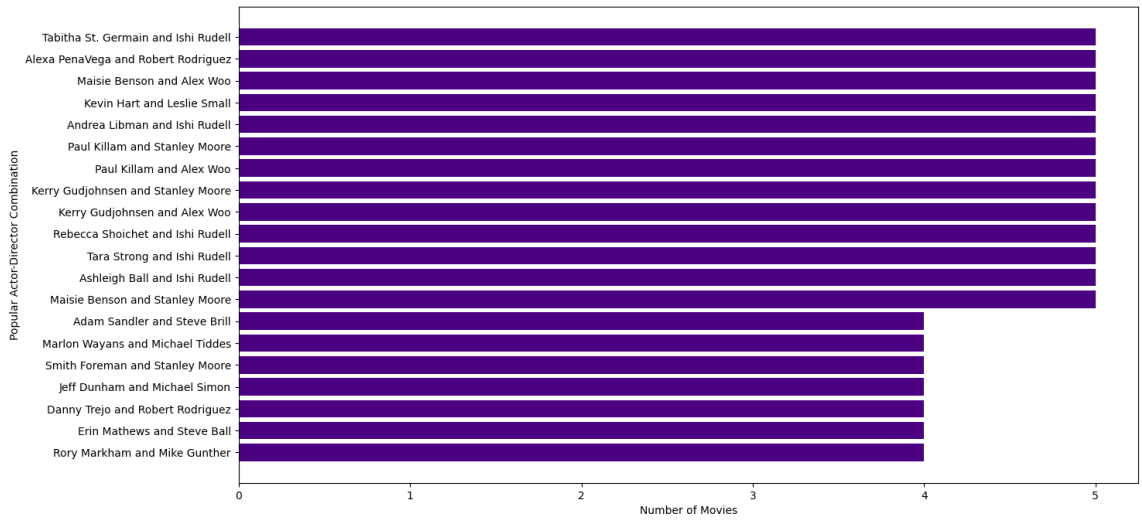
In [110]:

```
df_actors_directors=df_usa_shows_subset.groupby(['Actor_Director_Combination']).agg({"ti
plt.figure(figsize=(15,8))
plt.barh(df_actors_directors[::-1]['Actor_Director_Combination'], df_actors_directors[::
plt.xlabel('Number of Shows')
plt.ylabel('Popular Actor-Director Combination')
plt.show()
```



In [111]:

```
df_actors_directors=df_usa_movies_subset.groupby(['Actor_Director_Combination']).agg({"t
plt.figure(figsize=(15,8))
plt.barh(df_actors_directors[::-1]['Actor_Director_Combination'], df_actors_directors[::
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actor-Director Combination')
plt.show()
```



In [112]:

```
df_actors_directors[::-1]['Actor_Director_Combination'].values
```

Out[112]:

```
array(['Rory Markham and Mike Gunther', 'Erin Mathews and Steve Ball',
       'Danny Trejo and Robert Rodriguez',
       'Jeff Dunham and Michael Simon', 'Smith Foreman and Stanley Moore',
       'Marlon Wayans and Michael Tiddes', 'Adam Sandler and Steve Brill',
       'Maisie Benson and Stanley Moore', 'Ashleigh Ball and Ishi Rudell',
       'Tara Strong and Ishi Rudell', 'Rebecca Shoichet and Ishi Rudell',
       'Kerry Gudjohnsen and Alex Woo',
       'Kerry Gudjohnsen and Stanley Moore', 'Paul Killam and Alex Woo',
       'Paul Killam and Stanley Moore', 'Andrea Libman and Ishi Rudell',
       'Kevin Hart and Leslie Small', 'Maisie Benson and Alex Woo',
       'Alexa PenaVega and Robert Rodriguez',
       'Tabitha St. Germain and Ishi Rudell'], dtype=object)
```

The Most Popular Actor Director Combination in Movies Across USA are:-

'Smith Foreman and Stanley Moore',
'Marlon Wayans and Michael Tiddes',
'Adam Sandler and Steve Brill',
'Maisie Benson and Stanley Moore',
'Ashleigh Ball and Ishi Rudell',
'Tara Strong and Ishi Rudell',
'Rebecca Shoichet and Ishi Rudell',
'Kerry Gudjohnsen and Alex Woo',
'Kerry Gudjohnsen and Stanley Moore',
'Paul Killam and Alex Woo',
'Paul Killam and Stanley Moore',
'Andrea Libman and Ishi Rudell',
'Kevin Hart and Leslie Small',
'Maisie Benson and Alex Woo',
'Alexa PenaVega and Robert Rodriguez',
'Tabitha St. Germain and Ishi Rudell'

The Second Most Popular Actor Director Combination in Movies Across USA are:-
'Rory Markham and Mike Gunther',
'Erin Mathews and Steve Ball',
'Danny Trejo and Robert Rodriguez',
'Jeff Dunham and Michael Simon'

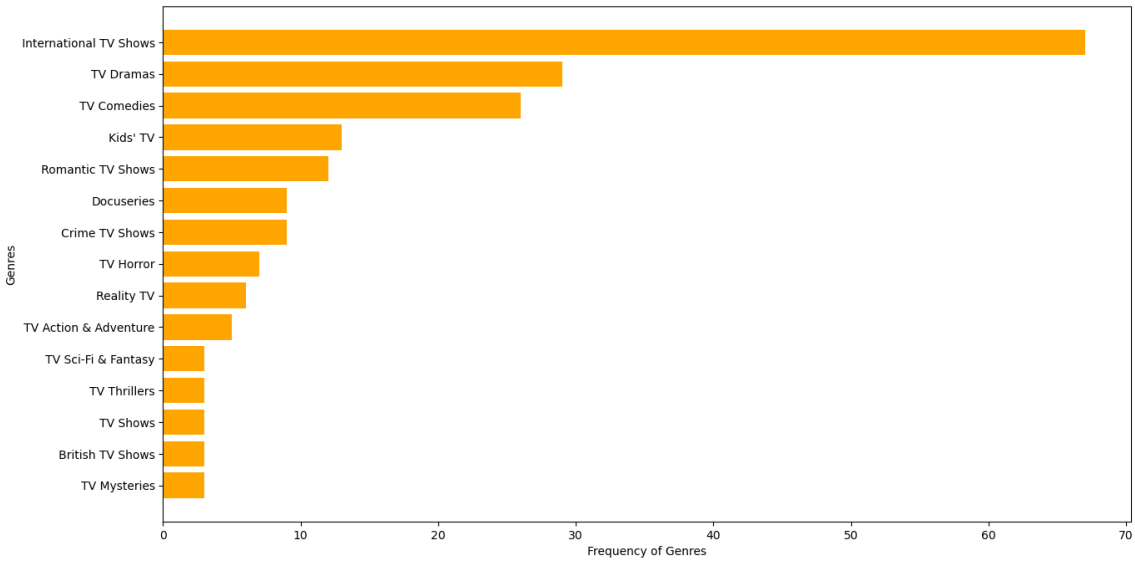
Univariate Analysis separately for shows and movies in India

In [113]:

```
#Analyzing India for both shows and movies
df_india_shows=df_final1[df_final1['country']=='India'][df_final1[df_final1['country']==
df_india_movies=df_final1[df_final1['country']=='India'][df_final1[df_final1['country']=
```

In [114]:

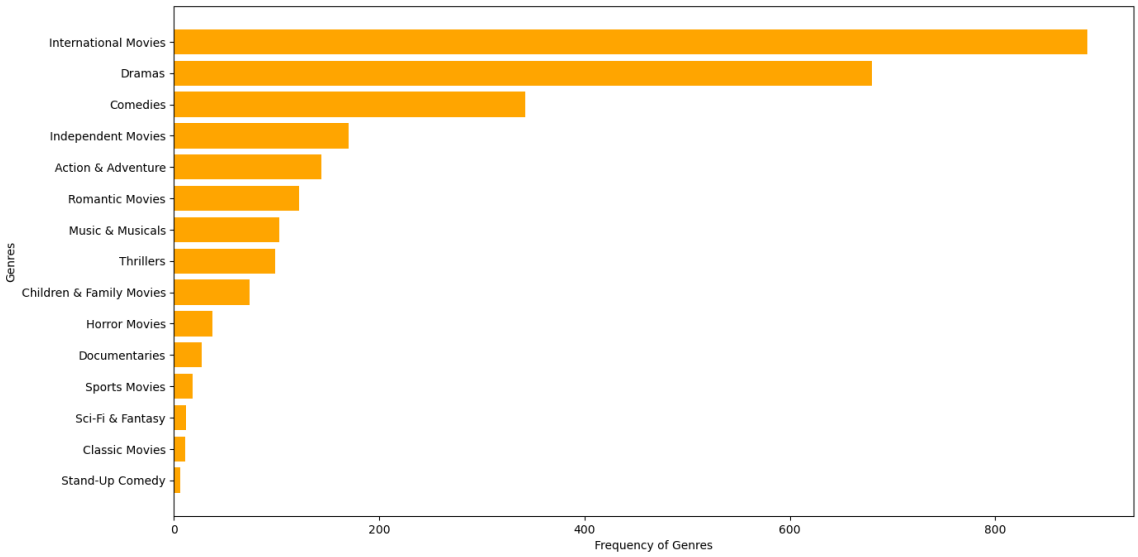
```
df_genre=df_india_shows.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_v
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



- Dramas,Comedy, Kids 'TV Shows and International TV Shows Genres are popular in TV Series in India

In [115]:

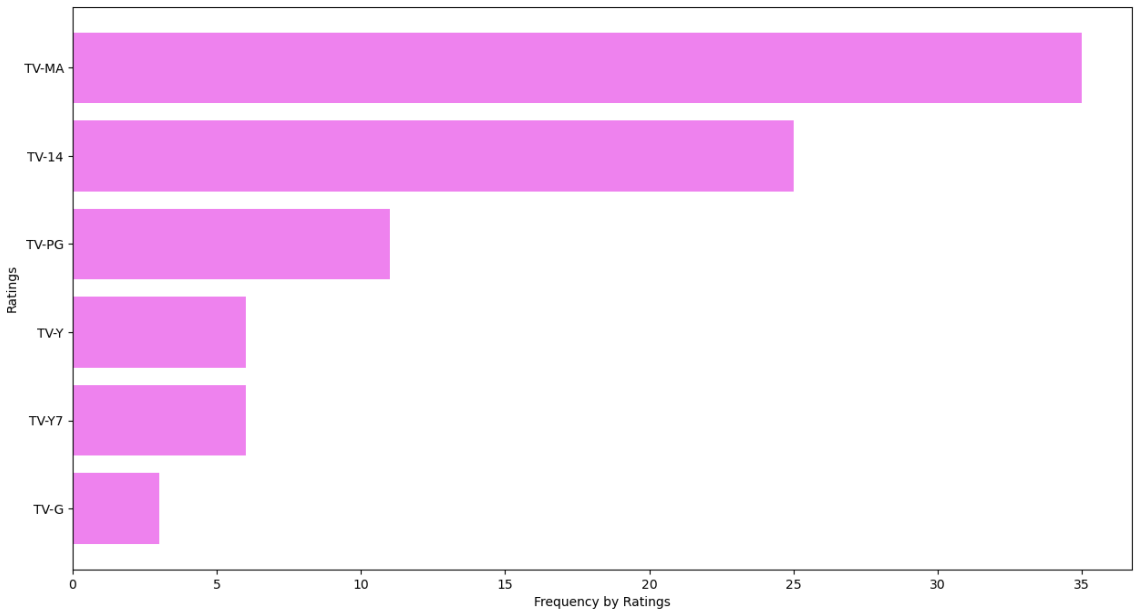
```
df_genre=df_india_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



- International Movies,Drama,Comedy,Indpeendent Movies and Action, Romance Genres are prevalent in India

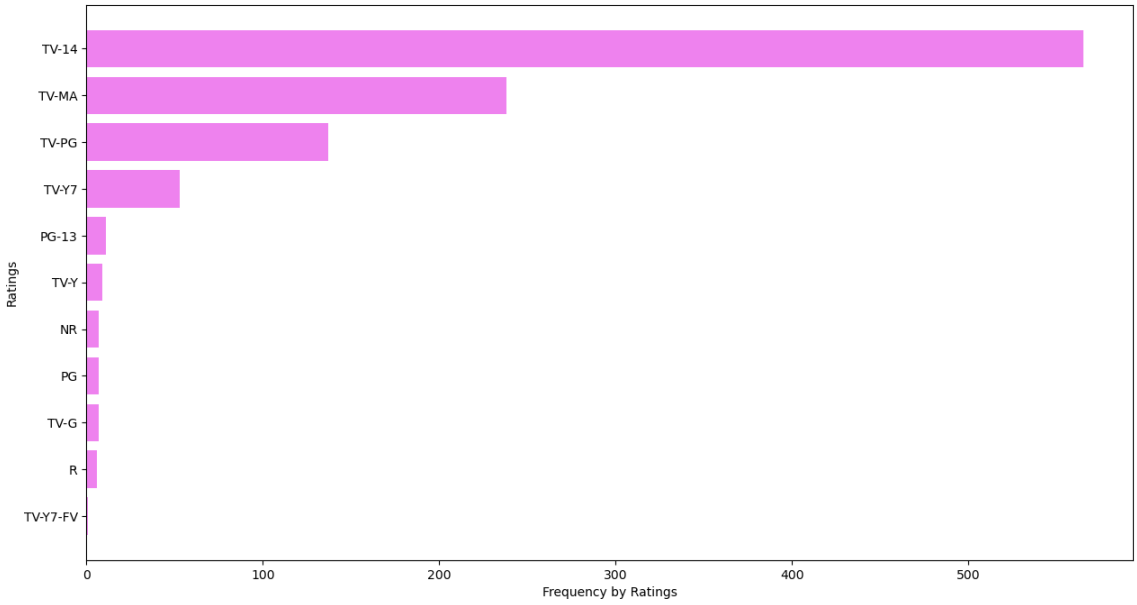
In [116]:

```
df_rating=df_india_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



In [117]:

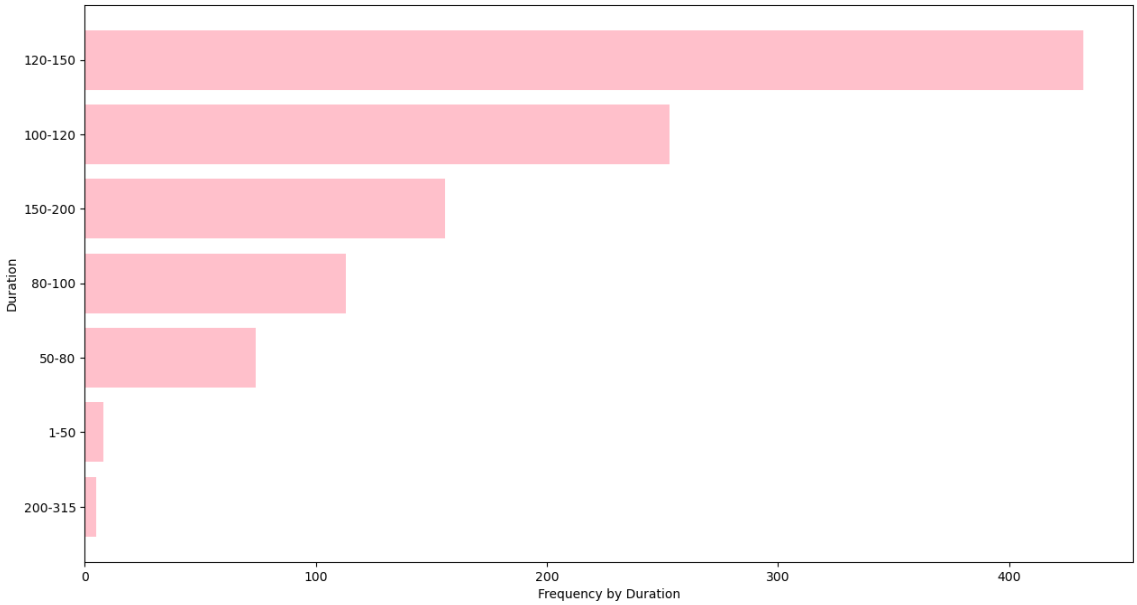
```
df_rating=df_india_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sor
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



- So it seems plaussible to conclude that the popular ratings across Netflix includes Mature Audiences in TV Shows and those appropriate for people over 14 in Movies in India.
- Now this indeed seems to be the case. Indian TV Shows in Netflix are without a shadow of doubt intended for Mature Audiences while Movies for over 14 years of age.

In [118]:

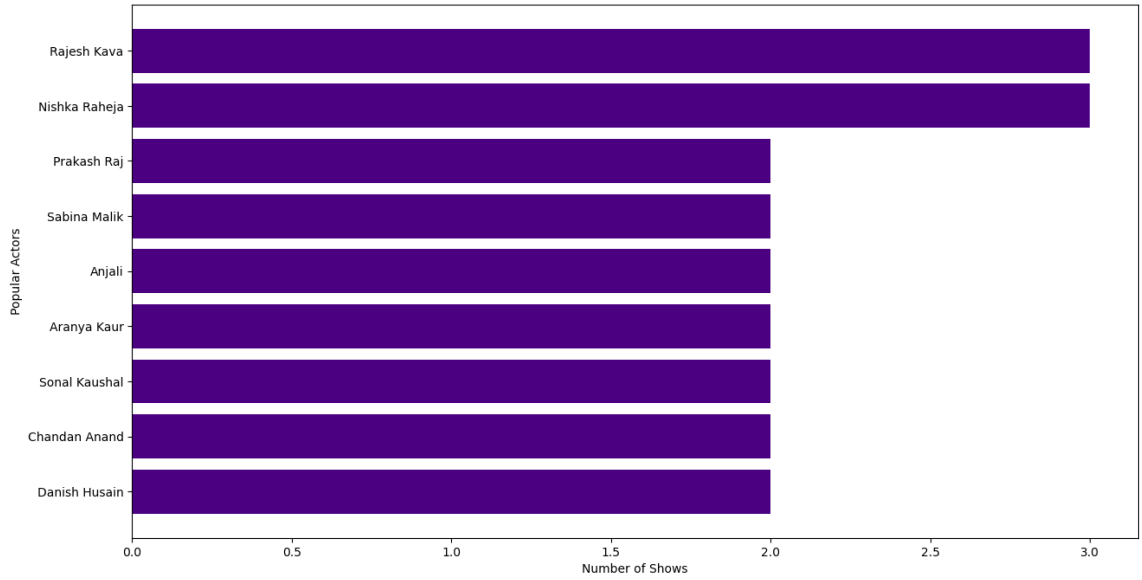
```
df_duration=df_india_movies.groupby(['duration']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
plt.barh(df_duration[::-1]['duration'], df_duration[::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



- Across movies ranges of minutes in India are comparatively greater than USA with a sweet spot at 120-150 mins.

In [119]:

```
df_actors=df_india_shows.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['Actors'], df_actors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Shows')
plt.ylabel('Popular Actors')
plt.show()
```



In [120]:

```
df_actors['Actors'].values
```

Out[120]:

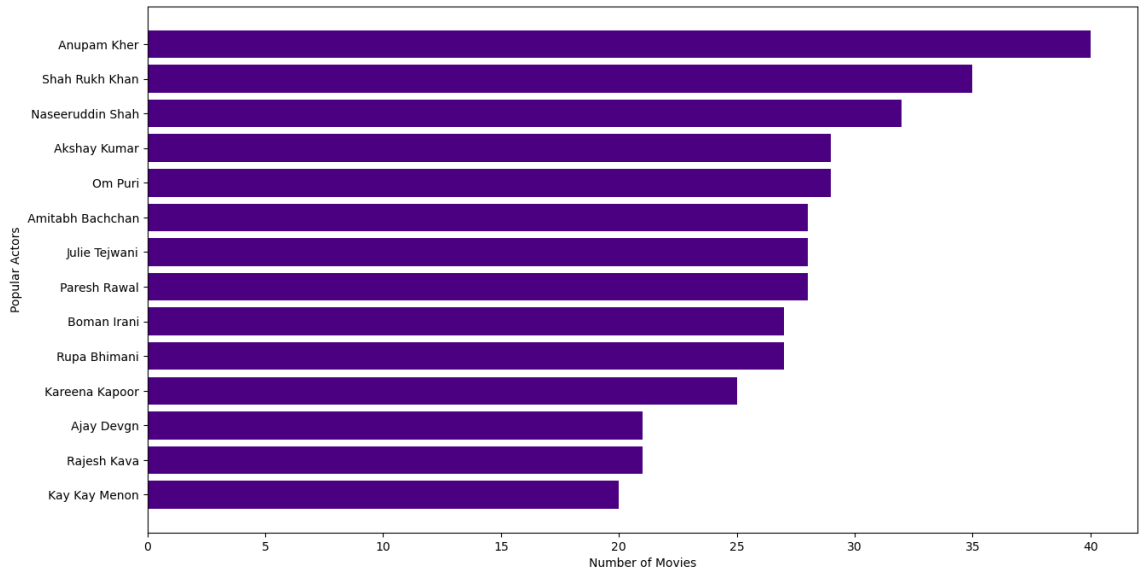
```
array(['Rajesh Kava', 'Nishka Raheja', 'Prakash Raj', 'Sabina Malik',
       'Anjali', 'Aranya Kaur', 'Sonal Kaushal', 'Chandan Anand',
       'Danish Husain'], dtype=object)
```

Popular Actors in TV Shows in India are:-

'Rajesh Kava',
'Nishka Raheja',
'Prakash Raj',
'Sabina Malik',
'Anjali',
'Aranya Kaur',
'Sonal Kaushal',
'Chandan Anand',
'Danish Husain'

In [121]:

```
df_actors=df_india_movies.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sor
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['Actors'], df_actors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actors')
plt.show()
```



In [122]:

```
df_actors['Actors'].values
```

Out[122]:

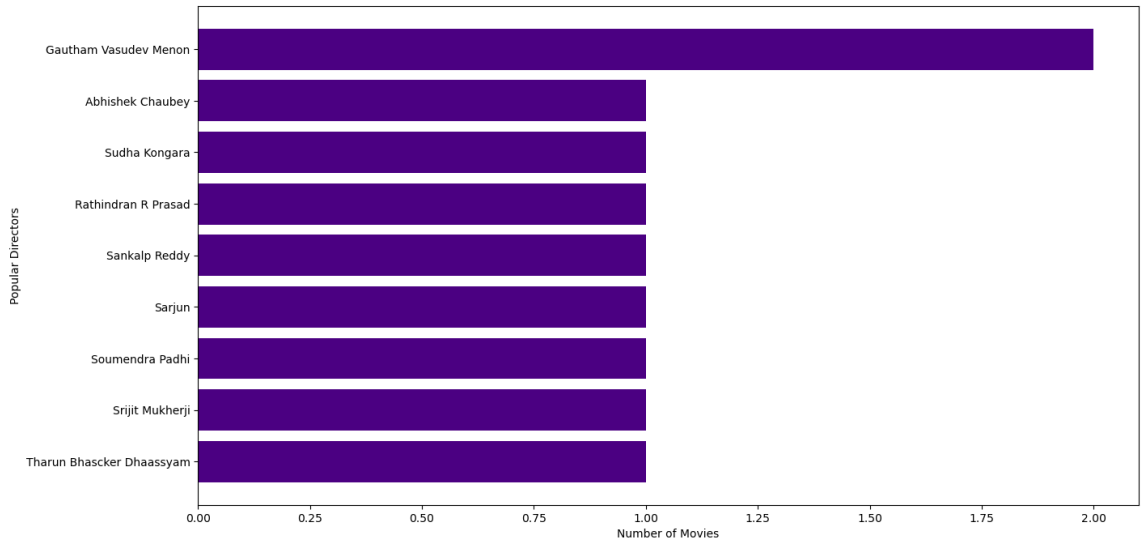
```
array(['Anupam Kher', 'Shah Rukh Khan', 'Naseeruddin Shah',
       'Akshay Kumar', 'Om Puri', 'Amitabh Bachchan', 'Julie Tejwani',
       'Paresh Rawal', 'Boman Irani', 'Rupa Bhimani', 'Kareena Kapoor',
       'Ajay Devgn', 'Rajesh Kava', 'Kay Kay Menon'], dtype=object)
```

Popular actors across Movies in India:-

'Anupam Kher',
'Shah Rukh Khan',
'Naseeruddin Shah',
'Akshay Kumar',
'Om Puri',
'Paresh Rawal',
'Julie Tejwani',
'Amitabh Bachchan',
'Boman Irani',
'Rupa Bhimani',
'Kareena Kapoor',
'Ajay Devgn',
'Rajesh Kava',
'Kay Kay Menon'

In [123]:

```
df_directors=df_india_shows.groupby(['Directors']).agg({"title":"nunique"}).reset_index(
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Directors')
plt.show()
```



In [124]:

```
df_directors['Directors'].values
```

Out[124]:

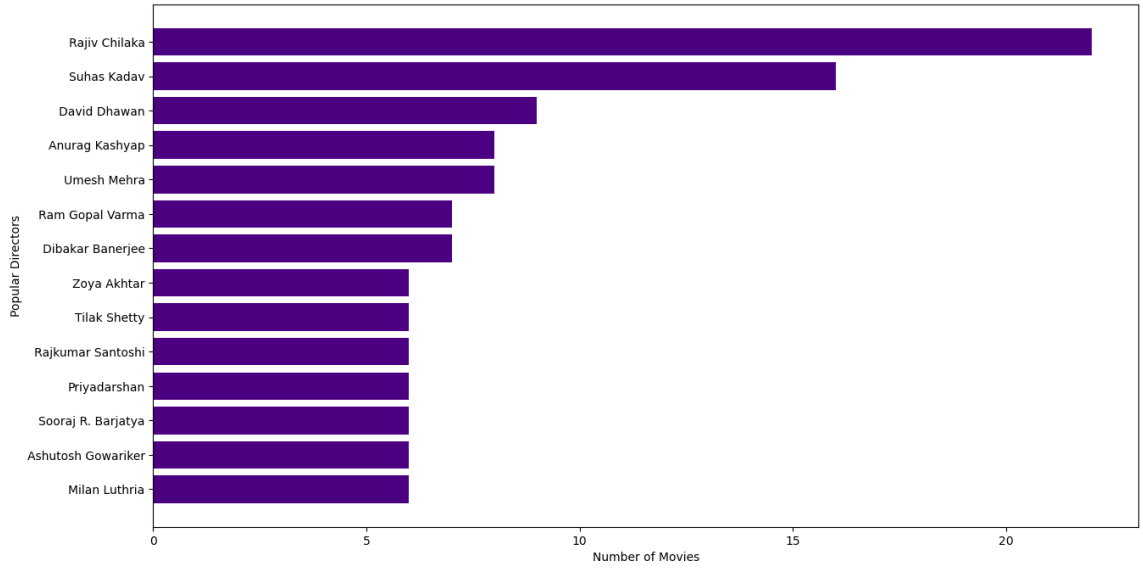
```
array(['Gautham Vasudev Menon', 'Abhishek Chaubey', 'Sudha Kongara',
       'Rathindran R Prasad', 'Sankalp Reddy', 'Sarjun',
       'Soumendra Padhi', 'Srijit Mukherji', 'Tharun Bhascker Dhaassyam'],
      dtype=object)
```

Popular Directors Across Movies in India:-

'Gautham Vasudev Menon',
'Abhishek Chaubey',
'Sudha Kongara',
'Rathindran R Prasad',
'Sankalp Reddy',
'Sarjun',
'Soumendra Padhi',
'Srijit Mukherji',
'Tharun Bhascker Dhaassyam'

In [125]:

```
df_directors=df_india_movies.groupby(['Directors']).agg({"title":"nunique"}).reset_index
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Directors')
plt.show()
```



In [126]:

```
df_directors['Directors'].values
```

Out[126]:

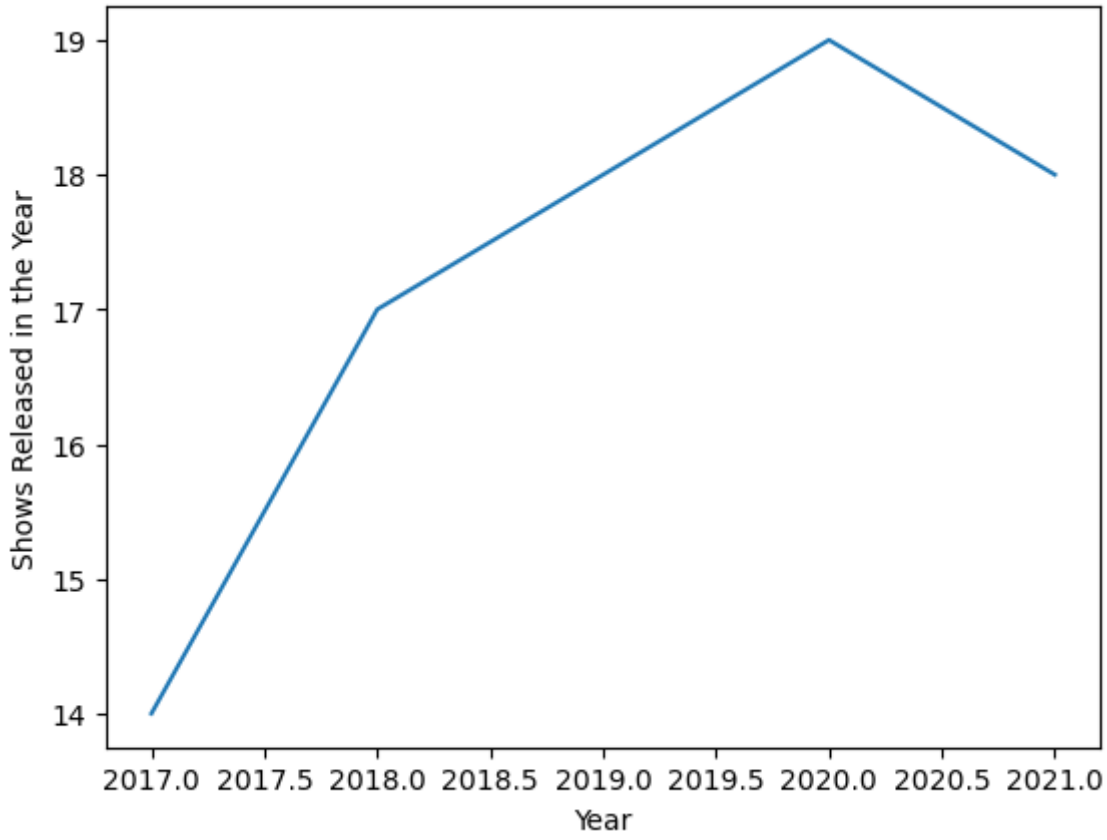
```
array(['Rajiv Chilaka', 'Suhas Kadav', 'David Dhawan', 'Anurag Kashyap',
       'Umesh Mehra', 'Ram Gopal Varma', 'Dibakar Banerjee',
       'Zoya Akhtar', 'Tilak Shetty', 'Rajkumar Santoshi', 'Priyadarshan',
       'Sooraj R. Barjatya', 'Ashutosh Gowariker', 'Milan Luthria'],
      dtype=object)
```

Popular directors across movies in India:-

'Rajiv Chilaka',
'Suhas Kadav',
'David Dhawan',
'Umesh Mehra',
'Anurag Kashyap',
'Ram Gopal Varma',
'Dibakar Banerjee',
'Zoya Akhtar',
'Tilak Shetty',
'Rajkumar Santoshi',
'Priyadarshan',
'Sooraj R. Barjatya',
'Ashutosh Gowariker',
'Milan Luthria'

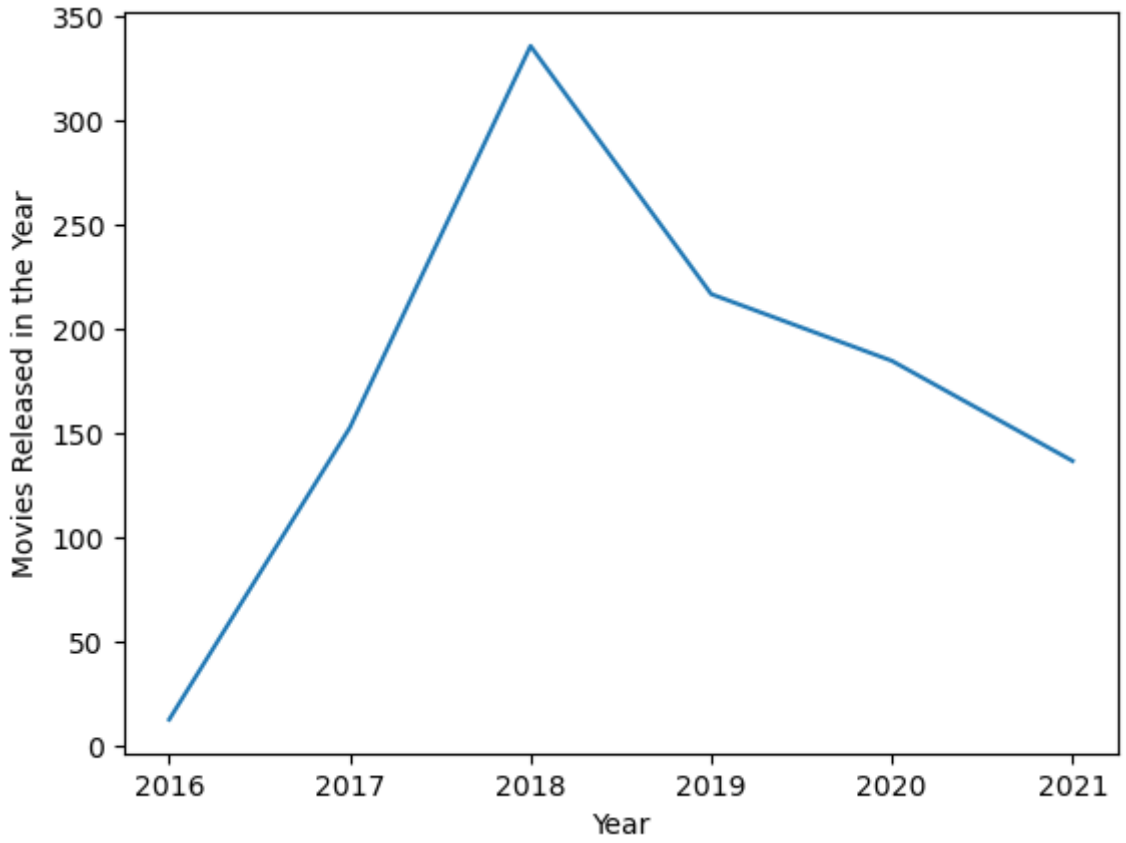
In [127]:

```
df_year=df_india_shows.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Shows Released in the Year")
plt.xlabel("Year")
plt.show()
```



In [128]:

```
df_year=df_india_movies.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```

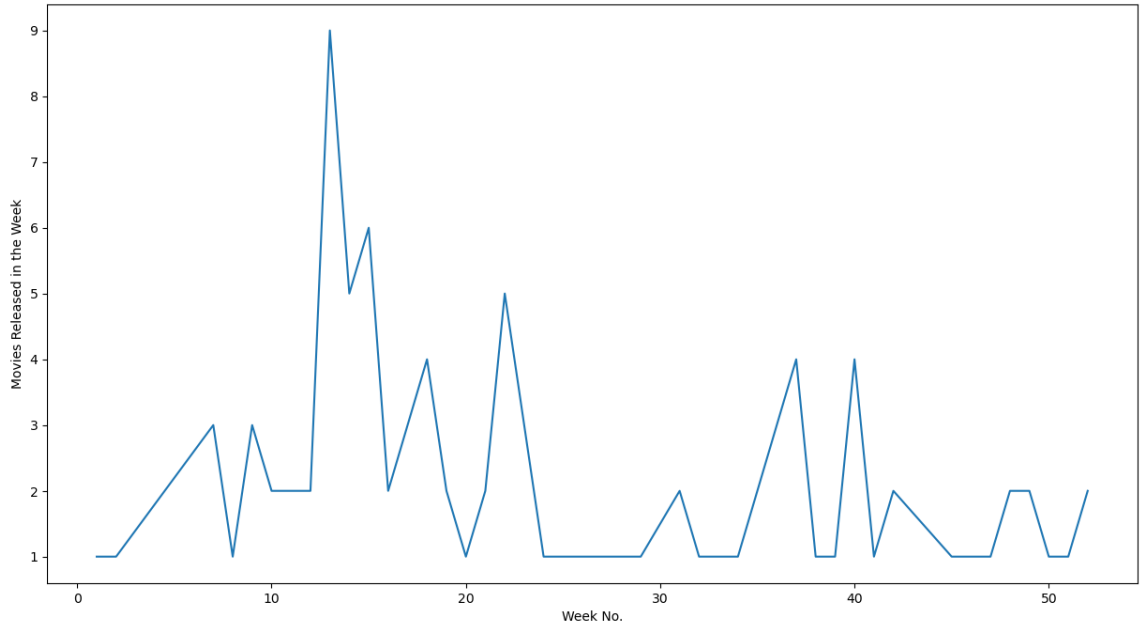


In India,TV Shows were increasingly being added till 2020, though the addition of shows reduced in 2021.

In India, Movies were increasingly added till 2018 but it has been a huge downhill since then. Now that's preposterous, since and soemthing has to be recommended to the Netflix Team with regards to that.

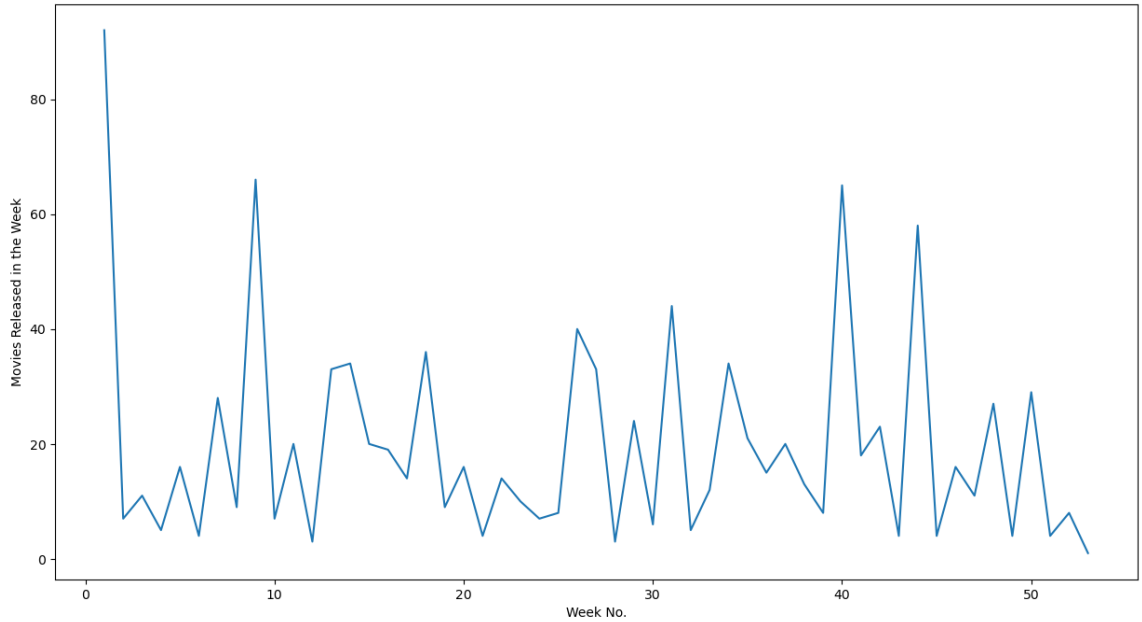
In [129]:

```
df_week=df_india_shows.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



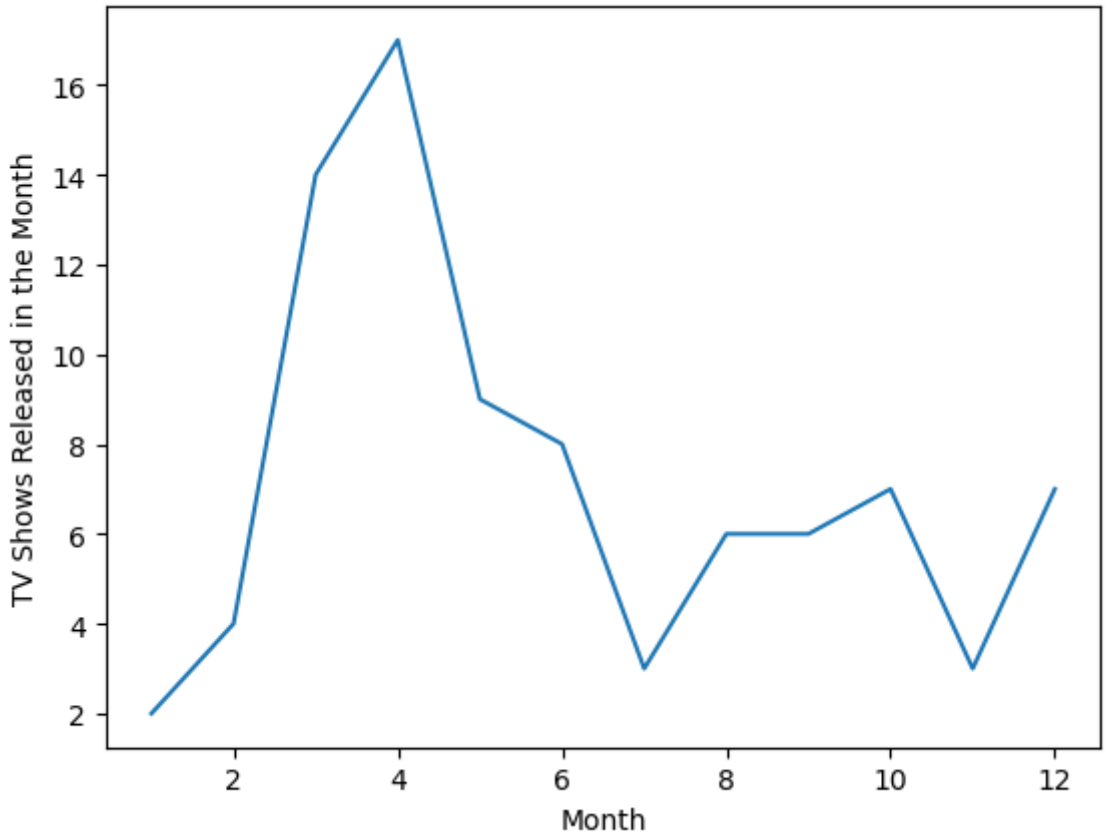
In [130]:

```
df_week=df_india_movies.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



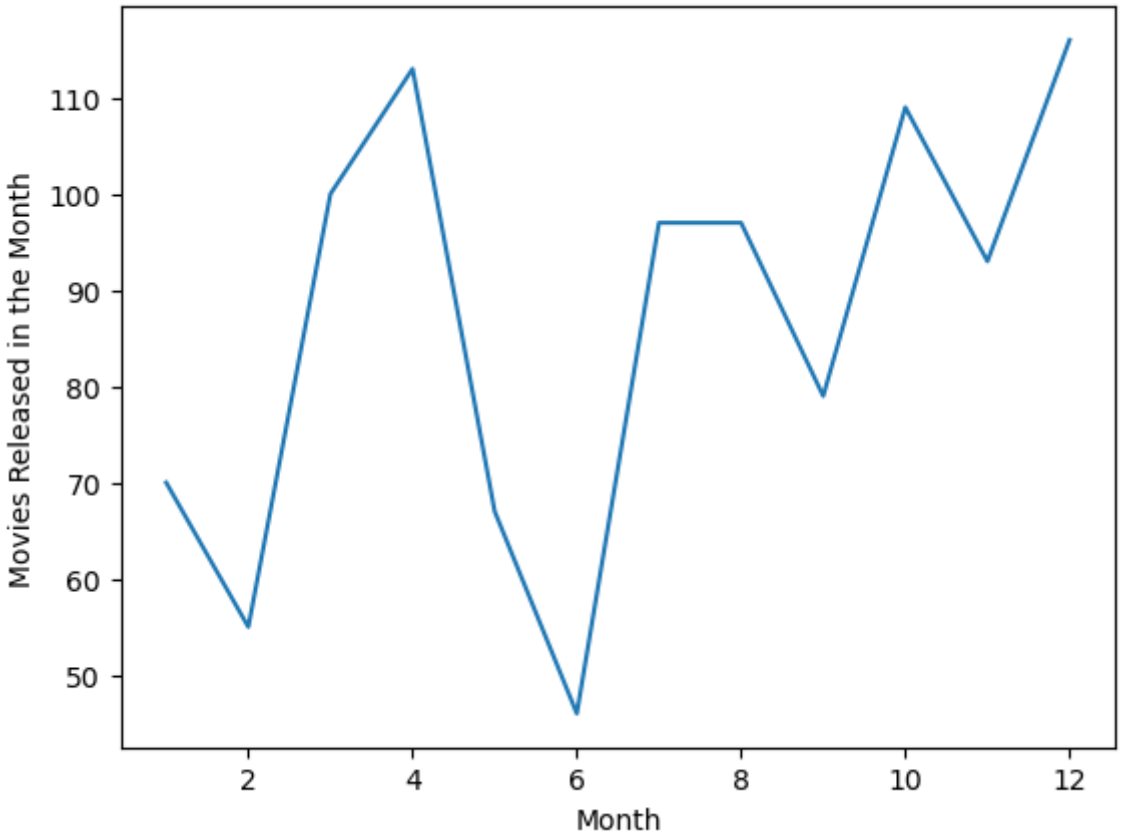
In [131]:

```
df_month=df_india_shows.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("TV Shows Released in the Month")
plt.xlabel("Month")
plt.show()
```



In [132]:

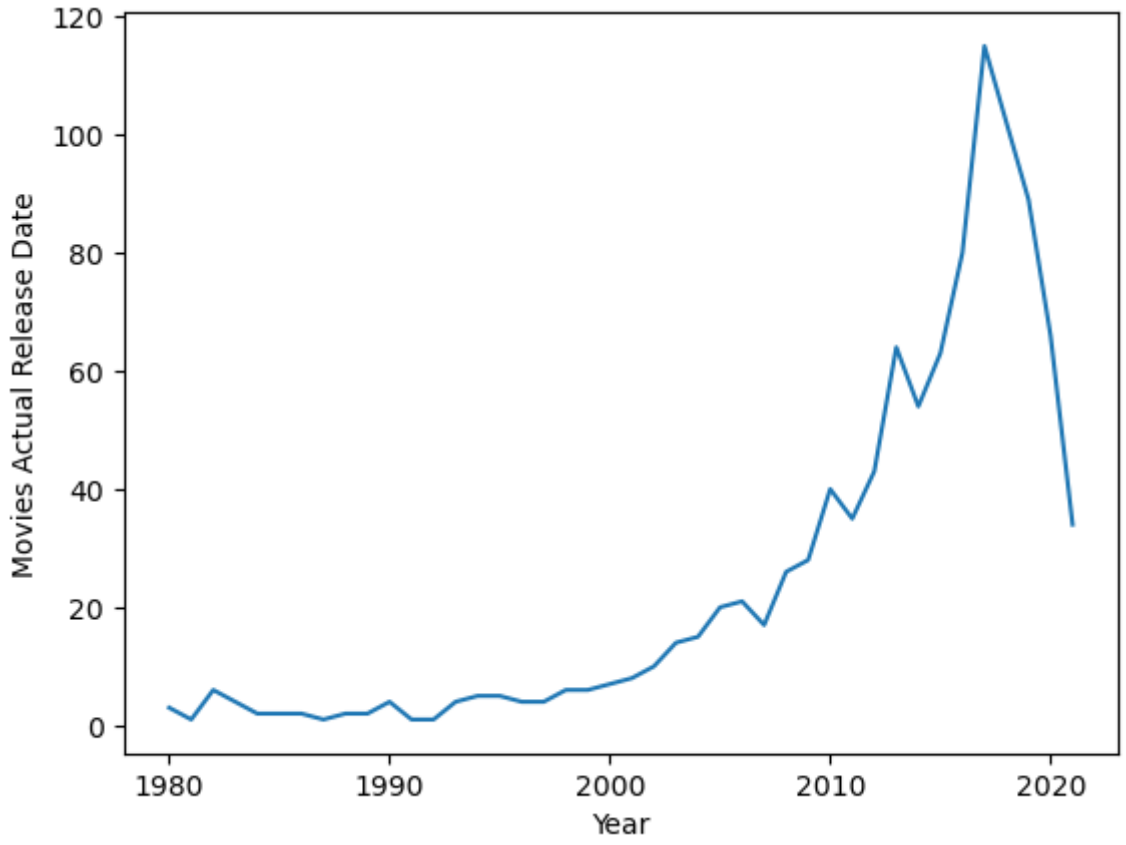
```
df_month=df_india_movies.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("Movies Released in the Month")
plt.xlabel("Month")
plt.show()
```



- TV Shows are added in Netflix by a tremendous amount in April in India
- Movies are added in Netflix in India by a tremendous amount in first week/last month of current year and first month of next year

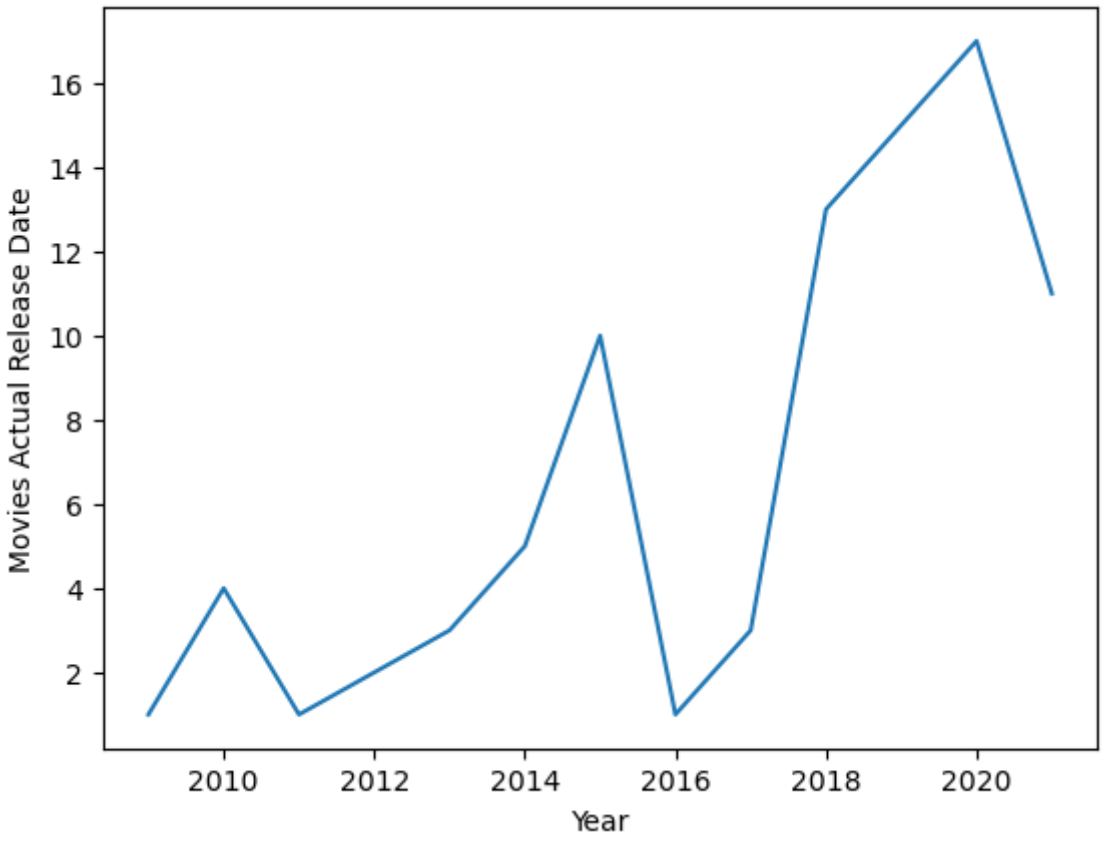
In [133]:

```
df_release_year=df_india_movies[df_india_movies['release_year']>=1980].groupby(['release_
sns.lineplot(data=df_release_year, x='release_year', y='title')
plt.ylabel("Movies Actual Release Date")
plt.xlabel("Year")
plt.show()
```



In [134]:

```
df_release_year=df_india_shows[df_india_shows['release_year']>=1980].groupby(['release_y
sns.lineplot(data=df_release_year, x='release_year', y='title')
plt.ylabel("Movies Actual Release Date")
plt.xlabel("Year")
plt.show()
```



- The understandable trend amongs movies and TV Shows across India in Netflix is the reduction of movies after 2020

In [135]:

```
#Analysing a combination of actors and directors
df_india_movies['Actor_Director_Combination'] = df_india_movies.Actors.str.cat(df_india_
df_india_movies_subset=df_india_movies[df_india_movies['Actors']!='Unknown Actor']
df_india_movies_subset=df_india_movies_subset[df_india_movies_subset['Directors']!='Unkn
df_india_movies_subset.head()
```

Out[135]:

	title	Actors	Directors	Genre	country	show_id	type	date_a
621	Avvai Shanmughi	Kamal Hassan	K.S. Ravikumar	Comedies	India	s23	Movie	Septe 21,
622	Avvai Shanmughi	Kamal Hassan	K.S. Ravikumar	International Movies	India	s23	Movie	Septe 21,
629	Avvai Shanmughi	Nassar	K.S. Ravikumar	Comedies	India	s23	Movie	Septe 21,
630	Avvai Shanmughi	Nassar	K.S. Ravikumar	International Movies	India	s23	Movie	Septe 21,
631	Avvai Shanmughi	S.P. Balasubrahmanyam	K.S. Ravikumar	Comedies	India	s23	Movie	Septe 21,

In [136]:

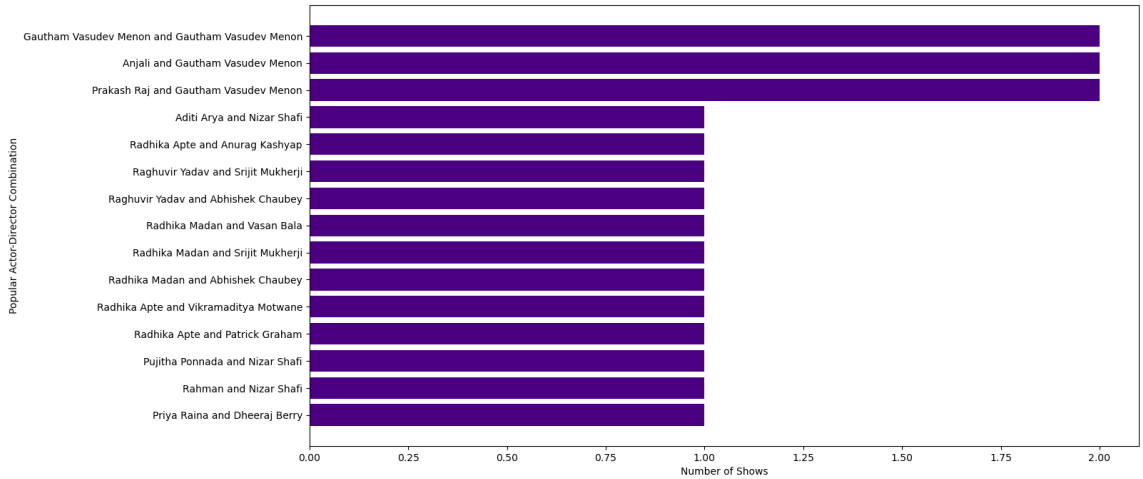
```
df_india_shows['Actor_Director_Combination'] = df_india_shows.Actors.str.cat(df_india_sh
df_india_shows_subset=df_india_shows[df_india_shows['Actors']!='Unknown Actor']
df_india_shows_subset=df_india_shows_subset[df_india_shows_subset['Directors']!='Unknown
df_india_shows_subset.head()
```

Out[136]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_
7005	Navarasa	Suriya	Bejoy Nambiar	TV Shows	India	s298	TV Show	August 6, 2021	2
7006	Navarasa	Suriya	Priyadarshan	TV Shows	India	s298	TV Show	August 6, 2021	2
7007	Navarasa	Suriya	Karthik Narain	TV Shows	India	s298	TV Show	August 6, 2021	2
7008	Navarasa	Suriya	Vasanth Sai	TV Shows	India	s298	TV Show	August 6, 2021	2
7009	Navarasa	Suriya	Karthik Subbaraj	TV Shows	India	s298	TV Show	August 6, 2021	2

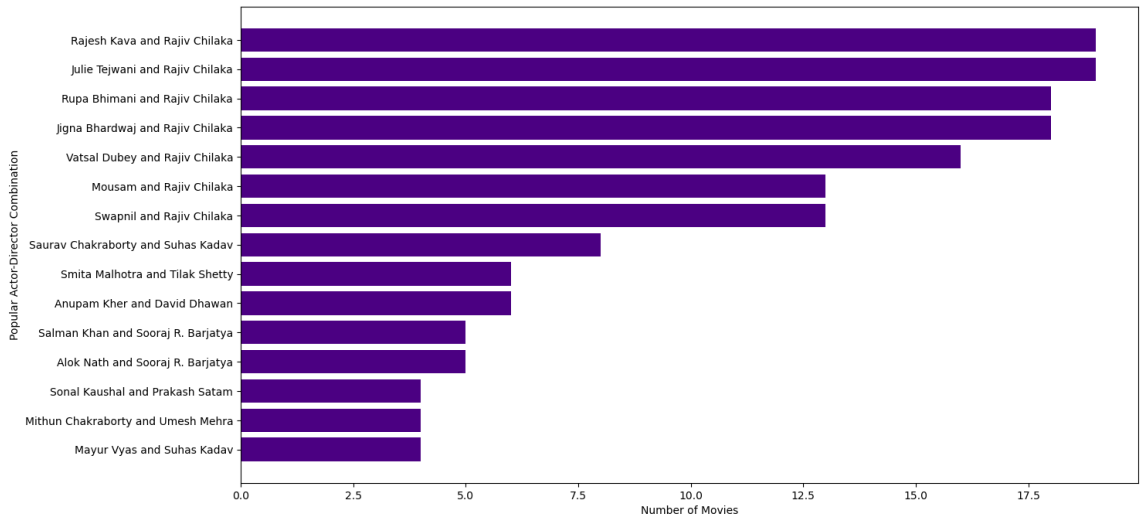
In [137]:

```
df_actors_directors=df_india_shows_subset.groupby(['Actor_Director_Combination']).agg({"
plt.figure(figsize=(15,8))
plt.barh(df_actors_directors[::-1]['Actor_Director_Combination'], df_actors_directors[::
plt.xlabel('Number of Shows')
plt.ylabel('Popular Actor-Director Combination')
plt.show()
```



In [138]:

```
df_actors_directors=df_india_movies_subset.groupby(['Actor_Director_Combination']).agg({
plt.figure(figsize=(15,8))
plt.barh(df_actors_directors[::-1]['Actor_Director_Combination'], df_actors_directors[::
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actor-Director Combination')
plt.show()
```



In [139]:

```
df_india_movies[df_india_movies['Directors']=='Rajiv Chilaka']
```

Out[139]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	rele
10058	Chhota Bheem - Neeli Pahaadi	Vatsal Dubey	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	
10059	Chhota Bheem - Neeli Pahaadi	Julie Tejwani	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	
10060	Chhota Bheem - Neeli Pahaadi	Rupa Bhimani	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	
10061	Chhota Bheem - Neeli Pahaadi	Jigna Bhardwaj	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	
10062	Chhota Bheem - Neeli Pahaadi	Rajesh Kava	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	
...	...	...	...	...	...	...	...	...	
145810	Chhota Bheem Kungfu Dhamaka	Vaibhav Thakkar	Rajiv Chilaka	Children & Family Movies	India	s6465	Movie	August 15, 2019	
145812	Chhota Bheem Kungfu Dhamaka	Samriddhi Shuklaa	Rajiv Chilaka	Children & Family Movies	India	s6465	Movie	August 15, 2019	
145814	Chhota Bheem Kungfu Dhamaka	Aditya Raj Sharma	Rajiv Chilaka	Children & Family Movies	India	s6465	Movie	August 15, 2019	
145816	Chhota Bheem Kungfu Dhamaka	Vinod Kulkarni	Rajiv Chilaka	Children & Family Movies	India	s6465	Movie	August 15, 2019	
150769	Dragonkala Ka Rahasya	Unknown Actor	Rajiv Chilaka	Children & Family Movies	India	s6646	Movie	June 18, 2019	

147 rows × 16 columns

- It seems that Rajiv Chilaka has worked on Chota Bheem and has been able to create some good content in its movies. He can be relied on for more Chota Bheem stories

In [140]:

```
df_actors_directors['Actor_Director_Combination'].values
```

Out[140]:

```
array(['Rajesh Kava and Rajiv Chilaka', 'Julie Tejwani and Rajiv Chilaka',
       'Rupa Bhimani and Rajiv Chilaka',
       'Jigna Bhardwaj and Rajiv Chilaka',
       'Vatsal Dubey and Rajiv Chilaka', 'Mousam and Rajiv Chilaka',
       'Swapnil and Rajiv Chilaka', 'Saurav Chakraborty and Suhas Kadav',
       'Smita Malhotra and Tilak Shetty', 'Anupam Kher and David Dhawan',
       'Salman Khan and Sooraj R. Barjatya',
       'Alok Nath and Sooraj R. Barjatya',
       'Sonal Kaushal and Prakash Satam',
       'Mithun Chakraborty and Umesh Mehra', 'Mayur Vyas and Suhas Kada
v'],
      dtype=object)
```

The Most Popular Actor Director Combination in Movies Across India are:-
'Rajesh Kava and Rajiv Chilaka',
'Julie Tejwani and Rajiv Chilaka',
'Rupa Bhimani and Rajiv Chilaka',
'Jigna Bhardwaj and Rajiv Chilaka',
'Vatsal Dubey and Rajiv Chilaka',
'Mousam and Rajiv Chilaka',
'Swapnil and Rajiv Chilaka',
'Saurav Chakraborty and Suhas Kadav',
'Smita Malhotra and Tilak Shetty',
'Anupam Kher and David Dhawan',
'Salman Khan and Sooraj R. Barjatya',

Recommendations

1) The most popular Genres across the countries and in both TV Shows and Movies are Drama, Comedy and International TV Shows/Movies, so content aligning to that is recommended.

2)Add TV Shows in July/August and Movies in last week of the year/first month of the next year.

3)For USA audience 80-120 mins is the recommended length for movies and Kids TV Shows are also popular along with the genres in first point, hence recommended.

4)The target audience in USA and India is recommended to be 14+ and above ratings

5)Add movies for Indian Audience, it has been declining since 2018.

6) 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.

In [ ]: