

In [1]:

```
#importing libraries for our purpose
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv('netflix_titles.csv')
df.head()
```

Out[1]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	d
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 [2]:

```
#length of data
len(df)
```

Out[2]:

8807

In [3]:

```
#checking datatypes
df.dtypes
```

Out[3]:

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

In [4]:

```
#number of unique values in our data
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 [5]:

```
#checking null values in every column of our data
df.isnull().sum()/len(df)*100
```

Out[5]:

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

In [6]:

```
#checking the occurences of each of the ratings
df[ 'rating' ].value_counts()
```

Out[6]:

TV-MA	3207
TV-14	2160
TV-PG	863
R	799
PG-13	490
TV-Y7	334
TV-Y	307
PG	287
TV-G	220
NR	80
G	41
TV-Y7-FV	6
NC-17	3
UR	3
74 min	1
84 min	1
66 min	1

Name: rating, dtype: int64

In [7]:

```
#unnesting the directors column, i.e- creating separate lines for each director in a
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[7]:

	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

In [8]:

```
#unnesting the cast column, i.e- creating separate lines for each cast member in a n
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[8]:

	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

In [9]:

```
#unnesting the listed_in column, i.e- creating separate lines for each genre in a m
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[9]:

	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

In [10]:

```
#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[10]:

	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

In [11]:

```
#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 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[11]:

	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

In [12]:

```
#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[12]:

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

In [13]:

```
#now checking nulls
df_final.isnull().sum()
```

Out[13]:

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	

In duration column, it was observed that the nulls had values which were written in corresponding ratings column, i.e- you can't expect ratings to be in min. So the duration column nulls are replaced by corresponding values in ratings column

In [15]:

```
df_final.loc[df_final['duration'].isnull(), 'duration']=df_final.loc[df_final['duration'].isnull(), 'duration']=df_final.loc[df_final['rating'].str.contains('min', na=False), 'rating']='NR'

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

Out[15]:

```
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
```

In [16]:

```
#Ratings can't be in min, so it has been made NR(i.e- Non Rated)
df_final.loc[df_final['rating'].str.contains('min', na=False), 'rating']='NR'
df_final['rating'].fillna('NR', inplace=True)
pd.set_option('display.max_rows', None)
```

In [17]:

```
#just an attempt to observe nulls in date_added column
df_final[df_final['date_added'].isnull()].head()
```

Out[17]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	release
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 [18]:

```
#date added column is imputed on the basis of release year,i.e- suppose there's a null
#when release year was 2013.So below piece of code just checks the mode of date added
# 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['rele
```

In [19]:

```
#country column is imputed on the basis of director,i.e- suppose there's a null for
#when we have a director whose other movies have a country given.So below piece of c
#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['Director
```

So we imputed the country column on the basis of directors whose other movie titles had countries given. But there might be directors who have only one occurrence in our data. In that scenario, I have used Actors as a basis. i.e- for this Actor majorly acts in movies of which country? Imputation has been done on this basis. For remaining rows, country has been filled as Unknown Country

In [20]:

```
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
#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[20]:

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	

In [21]:

```
df_final.head()
```

Out[21]:

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

In [22]:

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

101 min	3048
102 min	3017
103 min	2985
98 min	2984
99 min	2956
91 min	2915
92 min	2863
104 min	2822
88 min	2781
110 min	2711
100 min	2697
108 min	2614
112 min	2594
85 min	2486
89 min	2420
86 min	2213
4 Seasons	2134
116 min	2122
118 min	2119
---	----

In [23]:

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

Out[23]:

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

In [24]:

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

Out[24]:

```
array(['90', '2 Seasons', '1 Season', '91', '125', '9 Seasons', '104',
       '127', '4 Seasons', '67', '94', '5 Seasons', '161', '61', '16
6',
       '147', '103', '97', '106', '111', '3 Seasons', '110', '105', '9
6',
       '124', '116', '98', '23', '115', '122', '99', '88', '100',
       '6 Seasons', '102', '93', '95', '85', '83', '113', '13', '182',
       '48', '145', '87', '92', '80', '117', '128', '119', '143', '11
4',
       '118', '108', '63', '121', '142', '154', '120', '82', '109', '1
01',
       '86', '229', '76', '89', '156', '112', '107', '129', '135', '13
6',
       '165', '150', '133', '70', '84', '140', '78', '7 Seasons', '6
4',
       '59', '139', '69', '148', '189', '141', '130', '138', '81', '13
2',
       '10 Seasons', '123', '65', '68', '66', '62', '74', '131', '39',
       '46', '38', '8 Seasons', '17 Seasons', '126', '155', '159', '13
7',
       '12', '273', '36', '34', '77', '60', '49', '58', '72', '204',
       '212', '25', '73', '29', '47', '32', '35', '71', '149', '33',
       '15',
       '54', '224', '162', '37', '75', '79', '55', '158', '164', '17
3',
       '181', '185', '21', '24', '51', '151', '42', '22', '134', '17
7',
       '13 Seasons', '52', '14', '53', '8', '57', '28', '50', '9', '2
6',
       '45', '171', '27', '44', '146', '20', '157', '17', '203', '41',
       '30', '194', '15 Seasons', '233', '237', '230', '195', '253',
       '152', '190', '160', '208', '180', '144', '5', '174', '170', '1
92',
       '209', '187', '172', '16', '186', '11', '193', '176', '56', '16
9',
       '40', '10', '3', '168', '312', '153', '214', '31', '163', '19',
       '12 Seasons', '179', '11 Seasons', '43', '200', '196', '167',
       '178', '228', '18', '205', '201', '191'], dtype=object)
```

In [25]:

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

In [26]:

```
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[26]:

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

In [27]:

```
df_final1['duration_copy'].describe()
```

Out[27]:

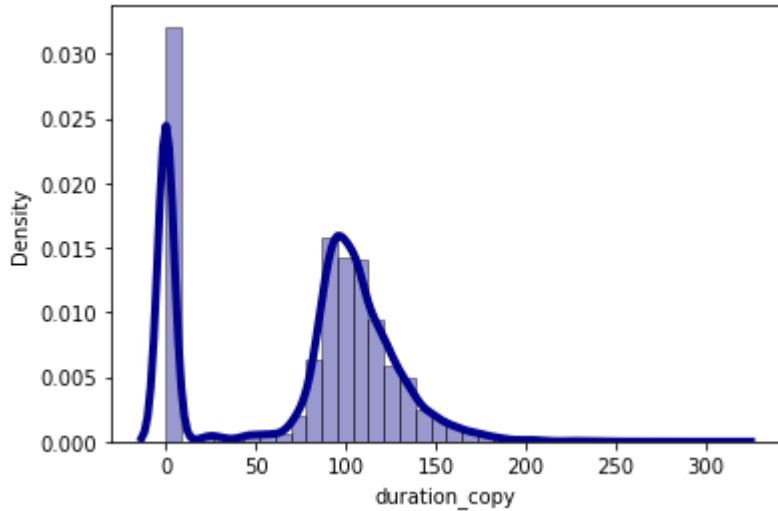
count	201991.000000
mean	77.152789
std	52.269154
min	0.000000
25%	0.000000
50%	95.000000
75%	112.000000
max	312.000000
Name:	duration_copy, dtype: float64

In [28]:

```
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()
```

/Users/suraaj/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please 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)
```



In [29]:

```
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[29]:

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

In [30]:

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

Out[30]:

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

In [31]:

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

Out[31]:

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	

In [32]:

```

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()

```

```

/var/folders/qc/486wrwtd4mj2pdzxt66k36c0000gn/T/ipykernel_38030/91795
4583.py:10: FutureWarning: Series.dt.weekofyear and Series.dt.week have been deprecated. Please use Series.dt.isocalendar().week instead.
df_final1['week_Added']=df_final1['Modified_Added_date'].dt.week

```

Out[32]:

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

In [33]:

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

```
/var/folders/qc/486wrwtd4mj2pdzxt66k36c0000gn/T/ipykernel_38030/37539
97727.py:3: FutureWarning: The default value of regex will change from
True to False in a future version.
df_final1['title']=df_final1['title'].str.replace(r"\(.*\)", "")
```

Out[33]:

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

## Univariate Analysis in terms of counts of each column

In [41]:

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

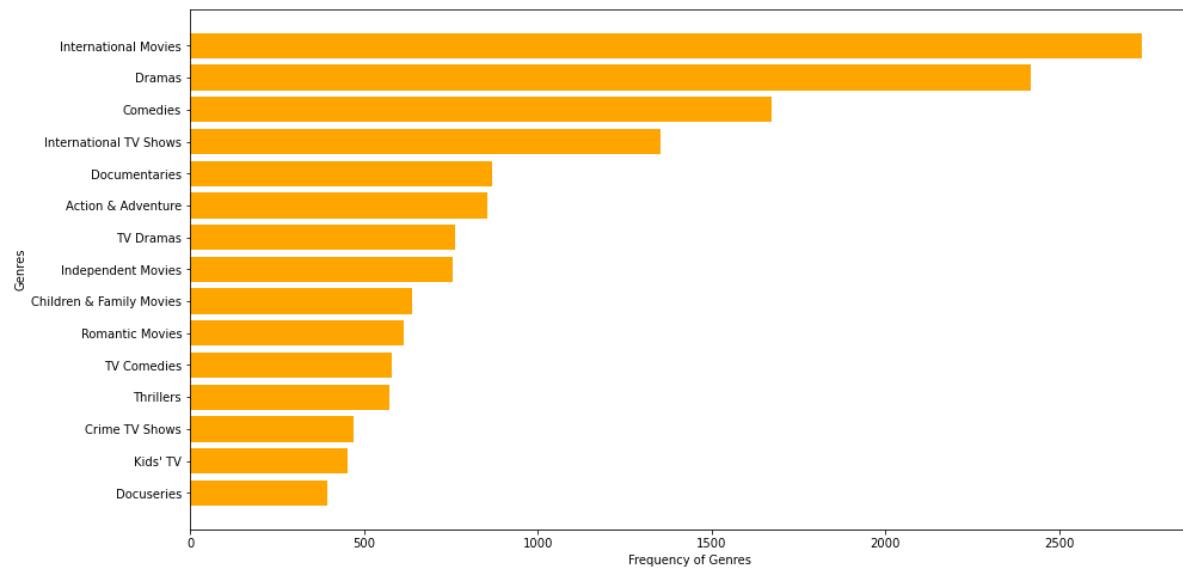
Out[41]:

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

Genre	title
<b>TV Sci-Fi &amp; Fantasy</b>	84
<b>TV Horror</b>	75
<b>Anime Features</b>	71
<b>Cult Movies</b>	71
<b>Teen TV Shows</b>	69
<b>Faith &amp; Spirituality</b>	65
<b>TV Thrillers</b>	57
<b>Movies</b>	57
<b>Stand-Up Comedy &amp; Talk Shows</b>	56
<b>Classic &amp; Cult TV</b>	28
<b>TV Shows</b>	16

In [42]:

```
df_genre=df_final1.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values('title', ascending=False)
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 .

In [43]:

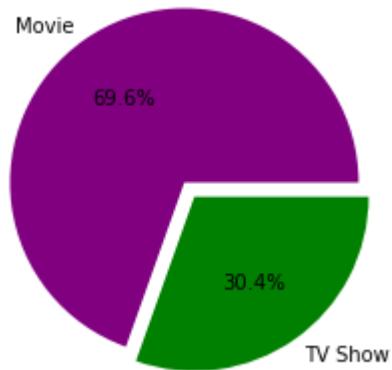
```
#number of distinct titles on the basis of type  
df_final.groupby(['type']).agg({"title":"nunique"})
```

Out[43]:

type	title
Movie	6115
TV Show	2676

In [44]:

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



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

In [45]:

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

Ecuador	1
Egypt	134
Ethiopia	1
Finland	12
France	409
Georgia	2
Germany	231
Ghana	8
Greece	11
Guatemala	2
Hong Kong	110
Hungary	11
Iceland	11

The above dataframe shows a flaw 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 [46]:

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

Out[46]:

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

In [47]:

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

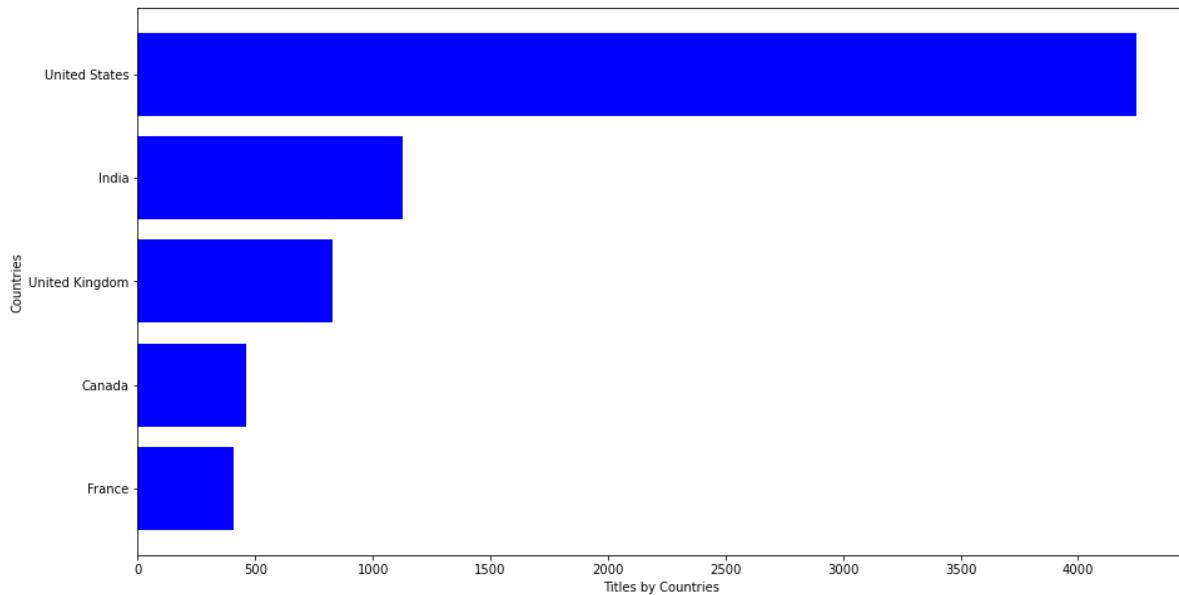
Out[47]:

country	title
	3
Afghanistan	1
Albania	1
Algeria	3
Angola	2
Argentina	94
Armenia	1
Australia	162
Austria	12

Now it looks great.

In [48]:

```
df_country=df_final1.groupby(['country']).agg({'title':"nunique"}).reset_index().sort_values('title', ascending=False)
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, India, UK, Canada and France are leading countries in Content Creation on Netflix

In [49]:

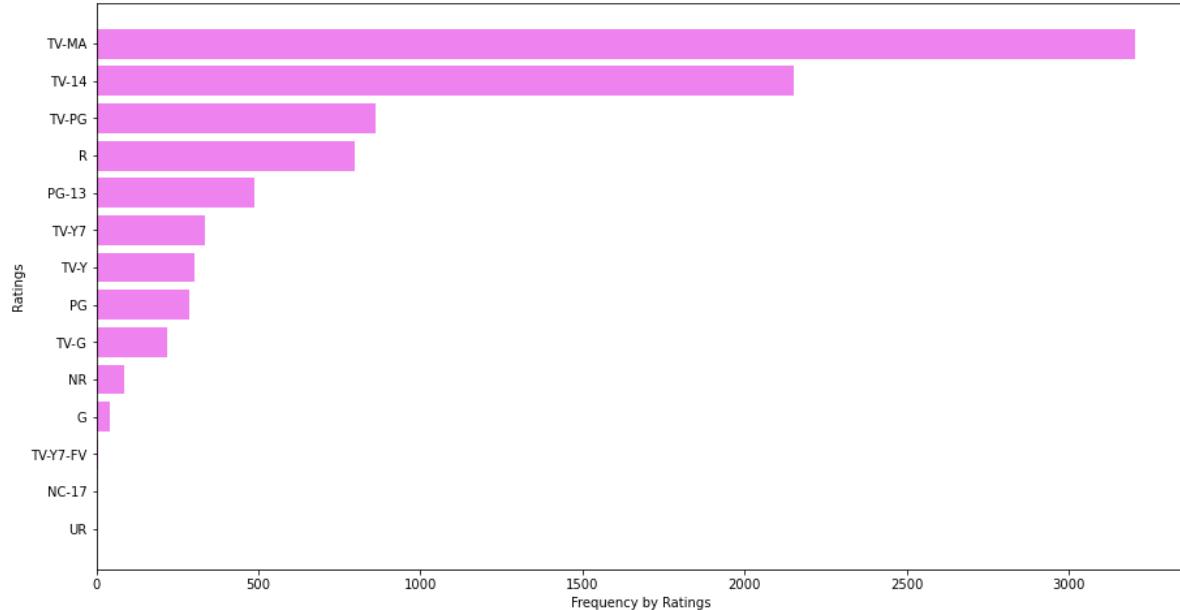
```
#number of distinct titles on the basis of rating  
df_final1.groupby(['rating']).agg({'title':"nunique"})
```

Out[49]:

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 [50]:

```
df_rating=df_final1.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()
```



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

In [51]:

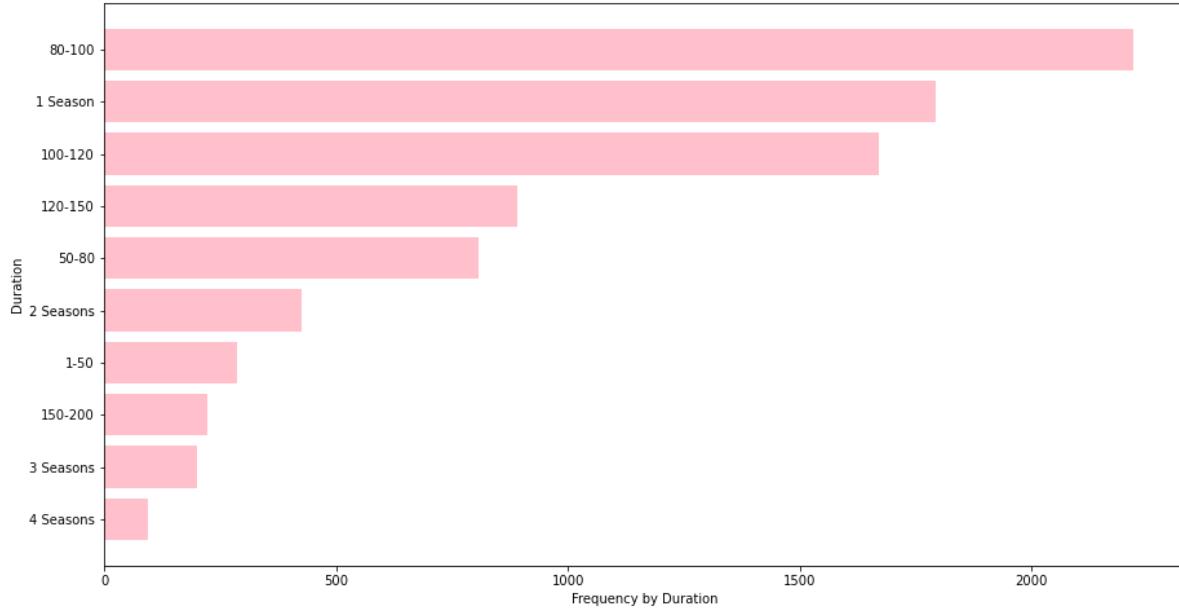
```
#number of distinct titles on the basis of duration
df_final1.groupby(['duration']).agg({'title':"nunique"})
```

Out[51]:

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

In [52]:

```
df_duration=df_final1.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
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.

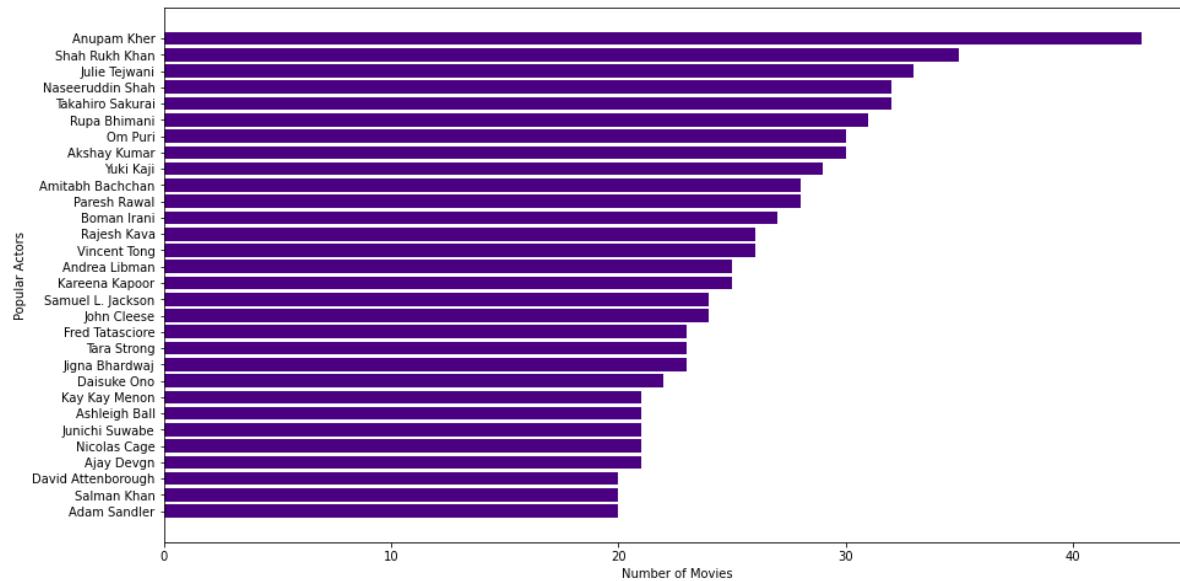
In [53]:

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

<b>A.J. Cook</b>	2
<b>A.J. Johnson</b>	1
<b>A.J. LoCascio</b>	3
<b>A.K. Hangal</b>	4
<b>A.R. Rahman</b>	1
<b>A.S. Sasi Kumar</b>	1
<b>AC Lim</b>	1
<b>AFRA</b>	1
<b>AJ Bowen</b>	1
<b>AJ Michalka</b>	1
<b>AJ Rivera</b>	1
<b>ARAH</b>	2
...	...

In [54]:

```
df_actors=df_final1.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()
```



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

In [55]:

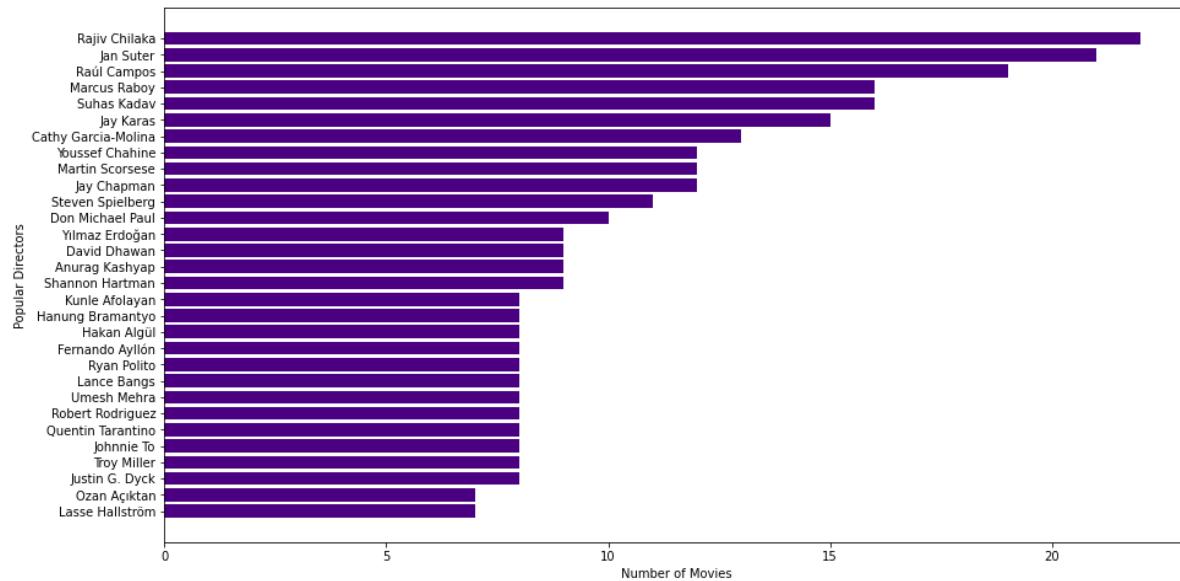
```
#number of distinct titles on the basis of Actors
df_final1.groupby(['Directors']).agg({"title":"nunique"})
```

Out[55]:

Directors	title
<b>A. L. Vijay</b>	2
<b>A. Raajdheep</b>	1
<b>A. Salaam</b>	1
<b>A.R. Murugadoss</b>	2
<b>Aadish Keluskar</b>	1
<b>Aamir Bashir</b>	1
<b>Aamir Khan</b>	1
<b>Aanand Rai</b>	1
<b>Aaron Burns</b>	1

In [56]:

```
df_directors=df_final1.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()
```



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

In [57]:

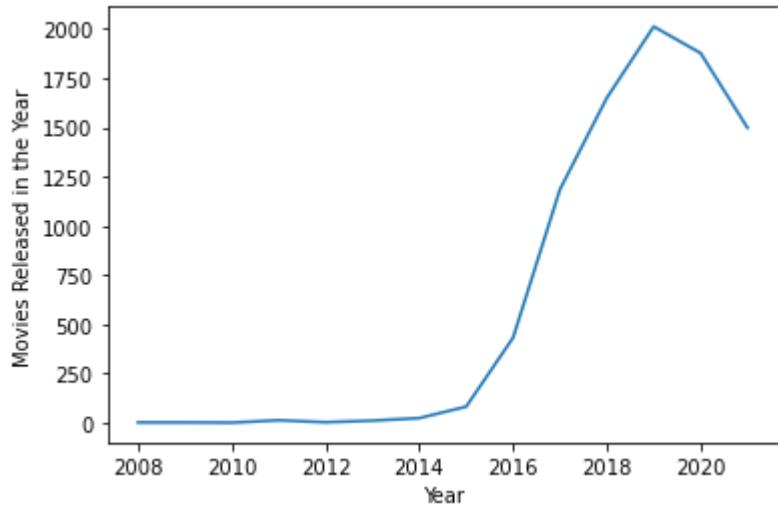
```
#number of distinct titles on the basis of year  
df_final1.groupby(['year']).agg({"title":"nunique"})
```

Out[57]:

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 [58]:

```
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)

In [69]:

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

Out[69]:

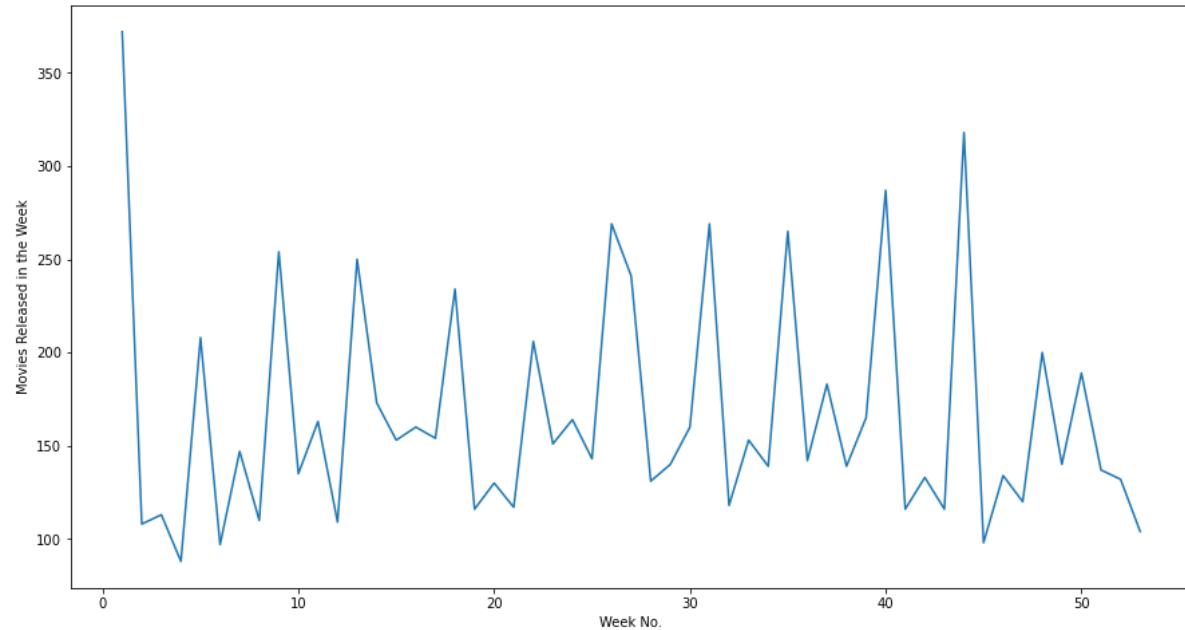
week_Added	title
1	372
2	108
3	113
4	88
5	208
6	97
7	147
8	110
9	254
10	135
11	163
12	109
13	250
14	173
15	153
16	160
17	154
18	234
19	116
20	130
21	117
22	206
23	151
24	164
25	143
26	269
27	241
28	131
29	140
30	160
31	269
32	118

**title****week\_Added**

<b>33</b>	153
<b>34</b>	139
<b>35</b>	265
<b>36</b>	142
<b>37</b>	183
<b>38</b>	139
<b>39</b>	165
<b>40</b>	287
<b>41</b>	116
<b>42</b>	133
<b>43</b>	116
<b>44</b>	318
<b>45</b>	98
<b>46</b>	134
<b>47</b>	120
<b>48</b>	200
<b>49</b>	140
<b>50</b>	189
<b>51</b>	137
<b>52</b>	132
<b>53</b>	104

In [70]:

```
df_week=df_final.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

In [71]:

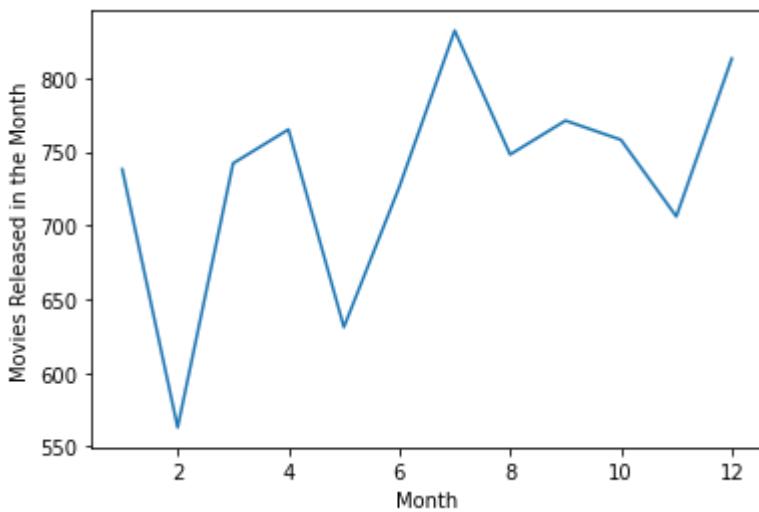
```
#number of distinct titles on the basis of week
df_final1.groupby(['month_added']).agg({'title':'nunique'})
```

Out[71]:

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 [72]:

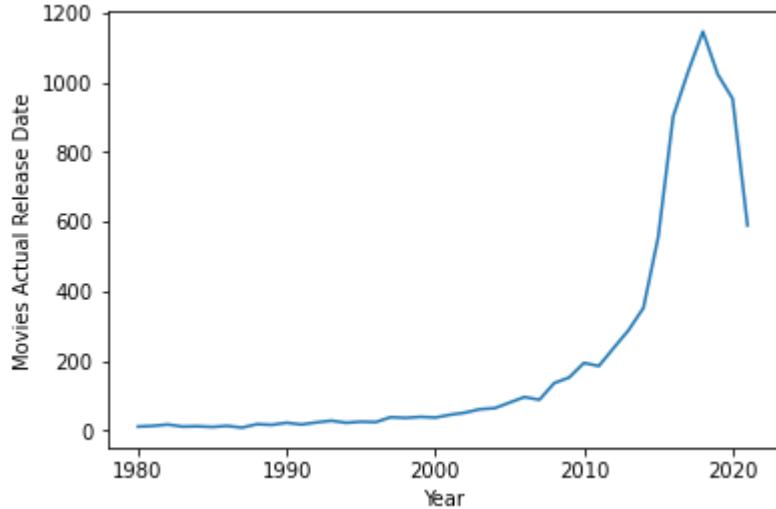
```
df_month=df_final1.groupby(['month_added']).agg({'title':'nunique'}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("Movies Released in the Month")
plt.xlabel("Month")
plt.show()
```



Most of the content is added in the first and last months across Netflix(reinstating what we observed for first week in baoove plot )

In [ 73 ]:

```
df_release_year=df_final1[df_final1['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()
```



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

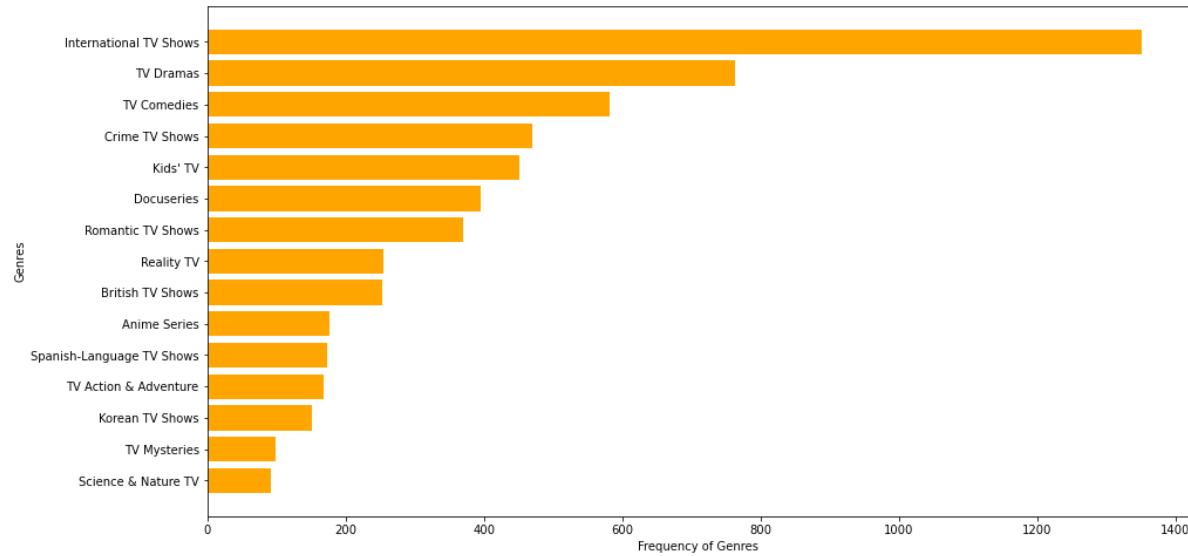
### Univariate Analysis separately for shows and movies

In [ 74 ]:

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

In [ 75 ]:

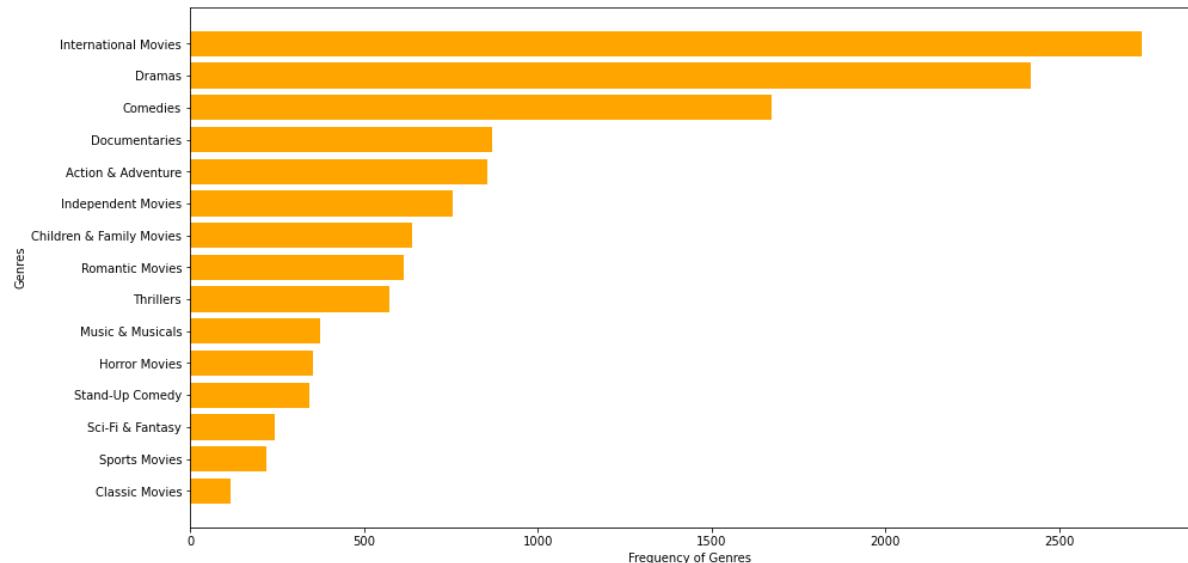
```
e=df_shows.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values(by="title",figsize=(15,8))
h(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
xlabel('Frequency of Genres')
ylabel('Genres')
w()
```



International TV Shows, Dramas and Comedy Genres are popular across TV Shows in Netflix

In [ 76 ]:

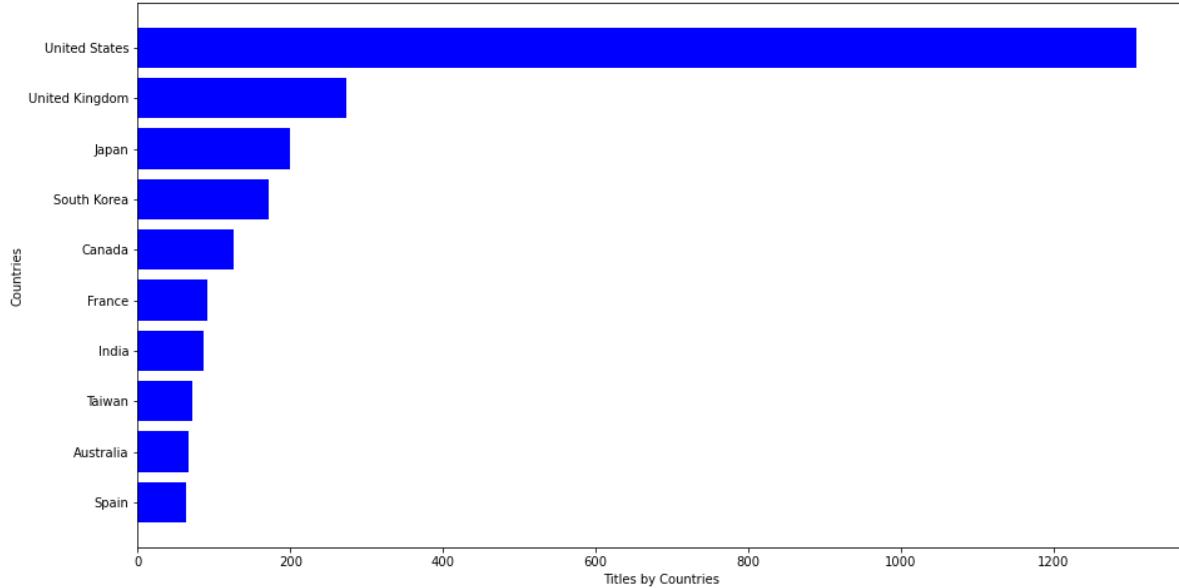
```
df_genre=df_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values(by="title",figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



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

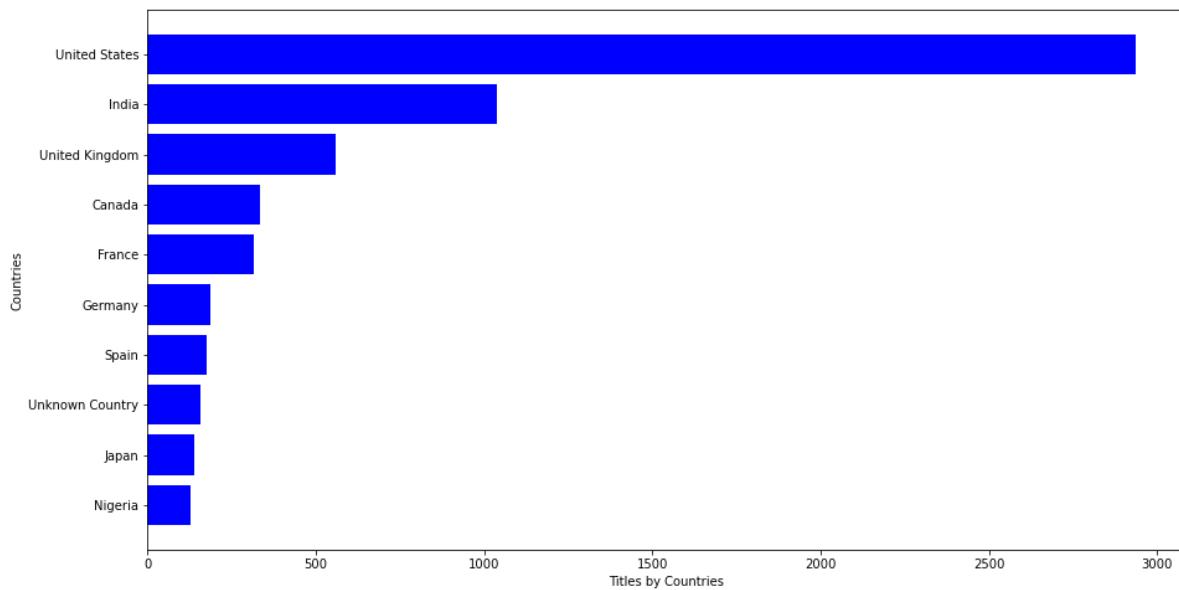
In [ 77 ]:

```
df_country=df_shows.groupby(['country']).agg({"title":"nunique"}).reset_index().sort
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()
```



In [ 78 ]:

```
df_movies.groupby(['country']).agg({"title":"nunique"}).reset_index().sort_values(by=
figsize=(15,8))
_country[::-1]['country'], df_country[::-1]['title'],color=['blue'])
'Titles by Countries')
'Countries')
```

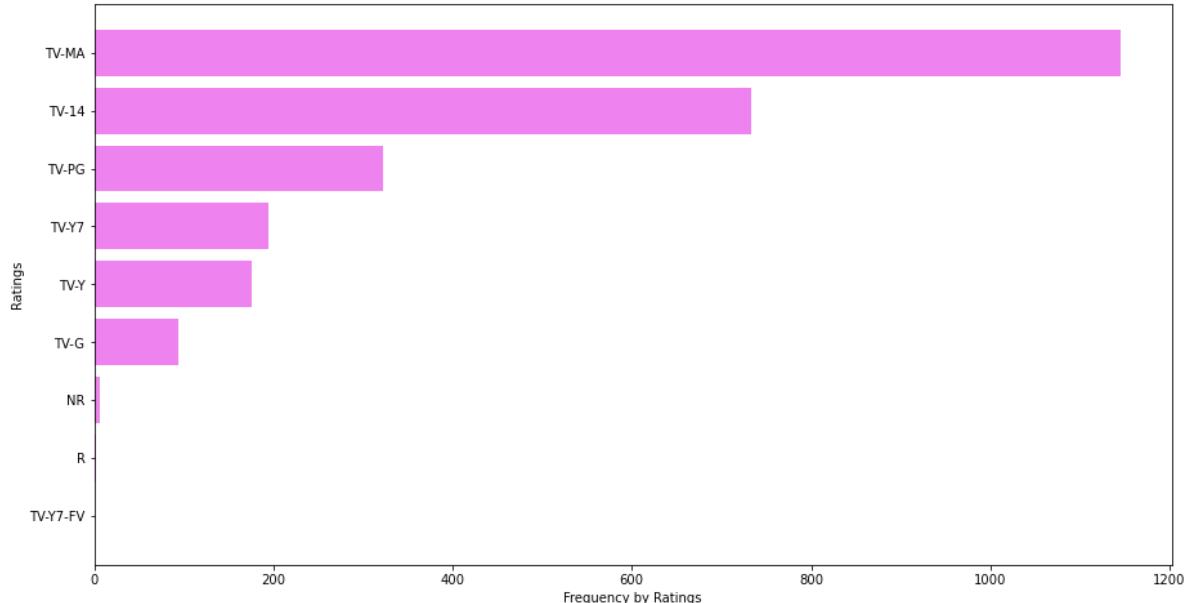


United States is leading across both TV Shows and Movies, UK also provides great content across TV Shows and Movies. Surprisingly India is much more prevalent in Movies as compared TV Shows.

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.

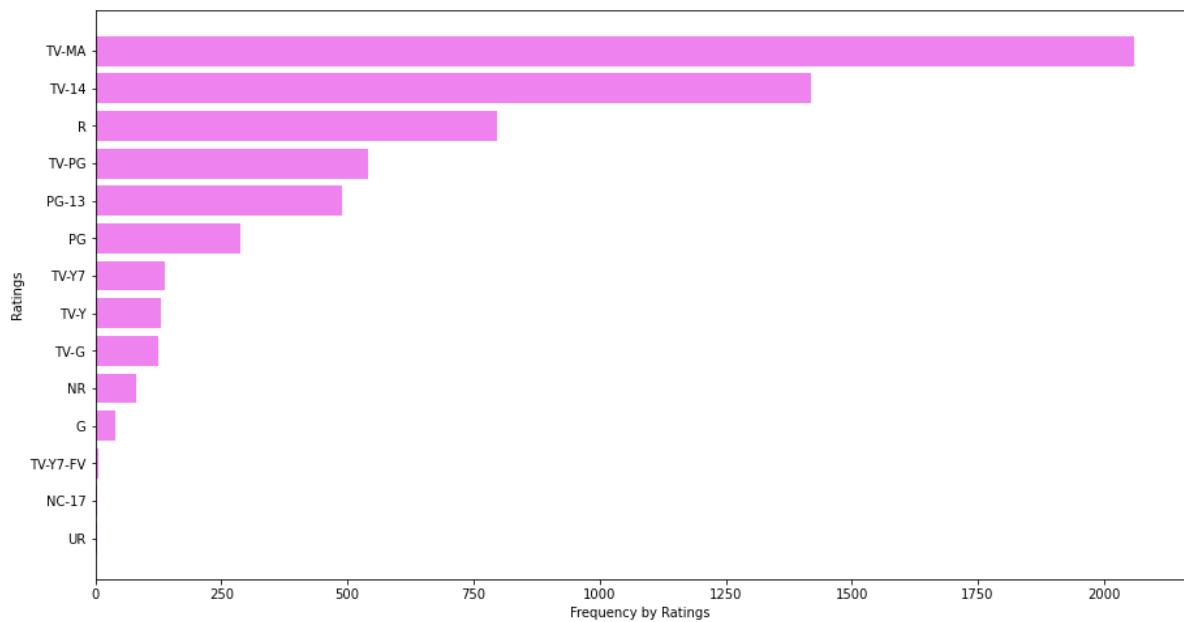
In [ 79 ]:

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



In [ 80 ]:

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



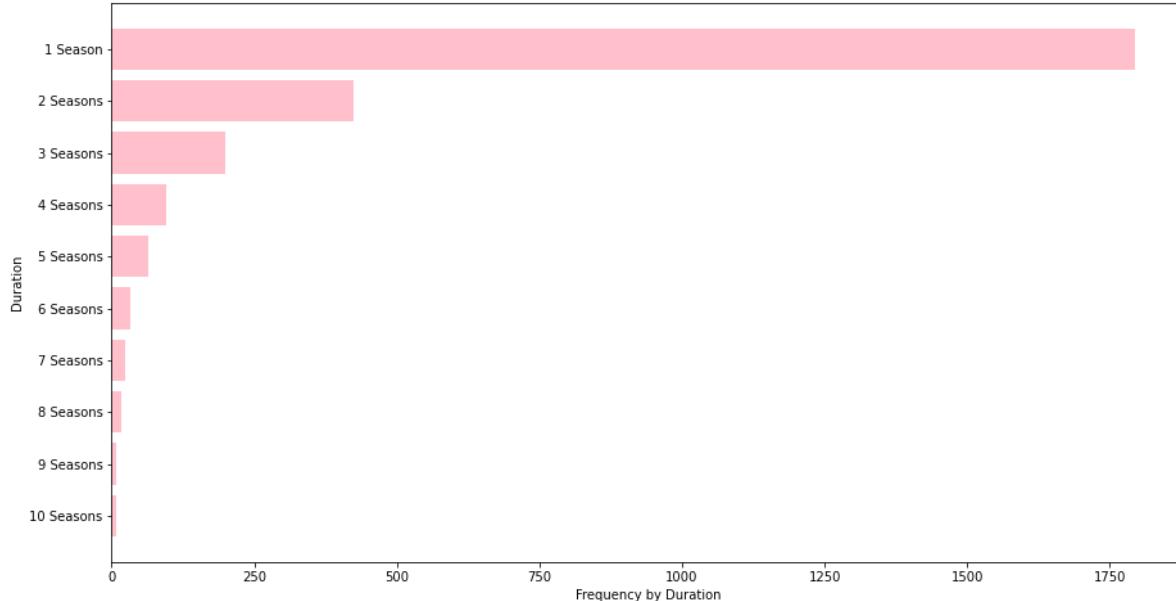
So it seems plausible to conclude that the popular ratings across Netflix includes Mature Audiences and

those appropriate for over 14/over 17 ages.

Moreover there are no TV Shows having a rating of R

In [81]:

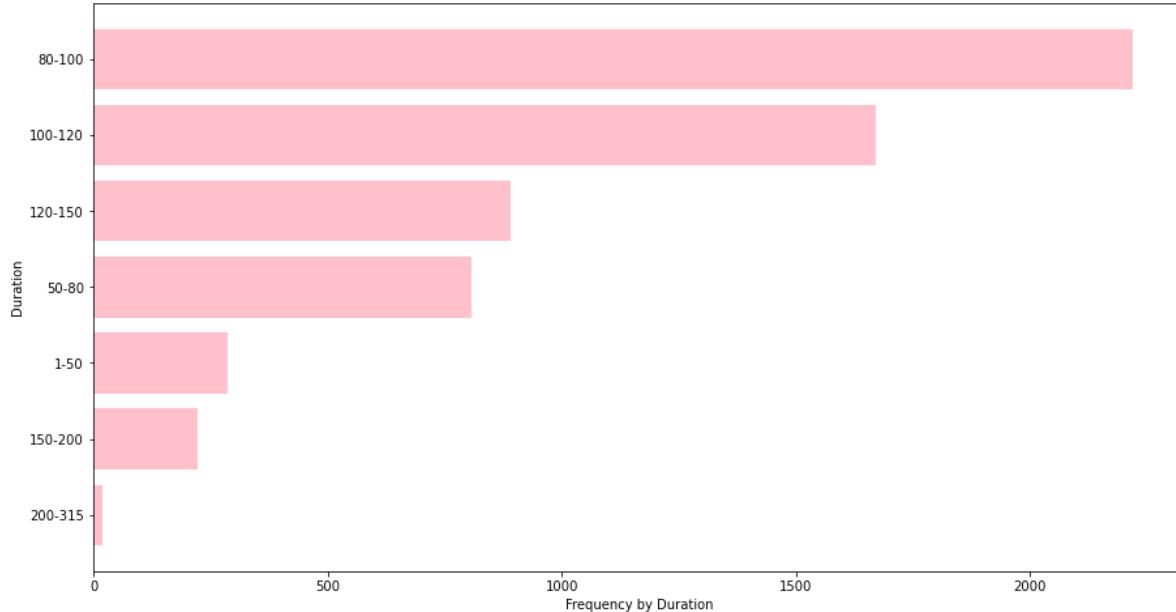
```
tion=df_shows.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort_values(by='nunique', ascending=False)
fig=Figure(figsize=(15,8))
ax=fig.add_subplot(111)
ax.bar(df_duration[::-1]['duration'], df_duration[::-1]['title'], color=['pink'])
ax.set_xlabel('Frequency by Duration')
ax.set_ylabel('Duration')
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 [82]:

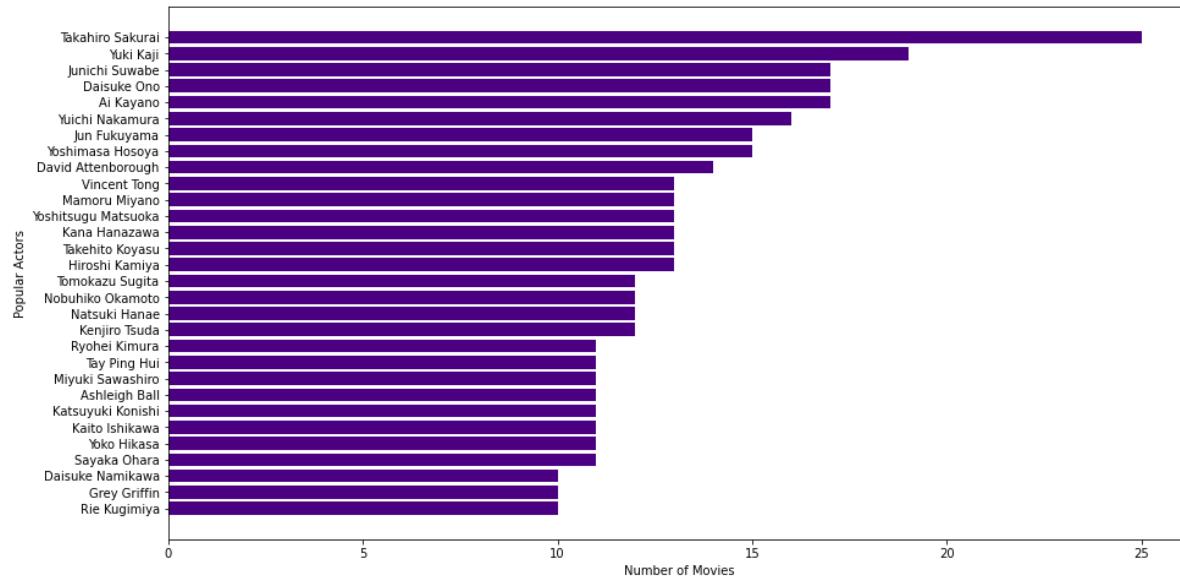
```
df_duration=df_movies.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_duration[:::-1]['duration'], df_duration[:::-1]['title'], color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()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



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 [83]:

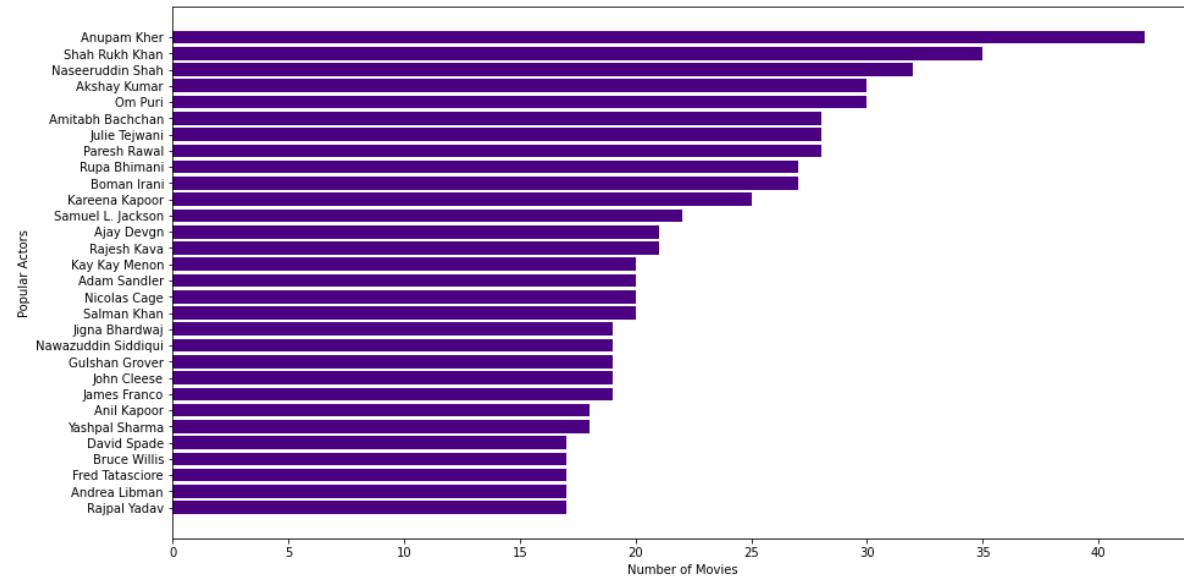
```
df_actors=df_shows.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[:::-1]['Actors'], df_actors[:::-1]['title'], color=['indigo'])
plt.xlabel('Number 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 [84]:

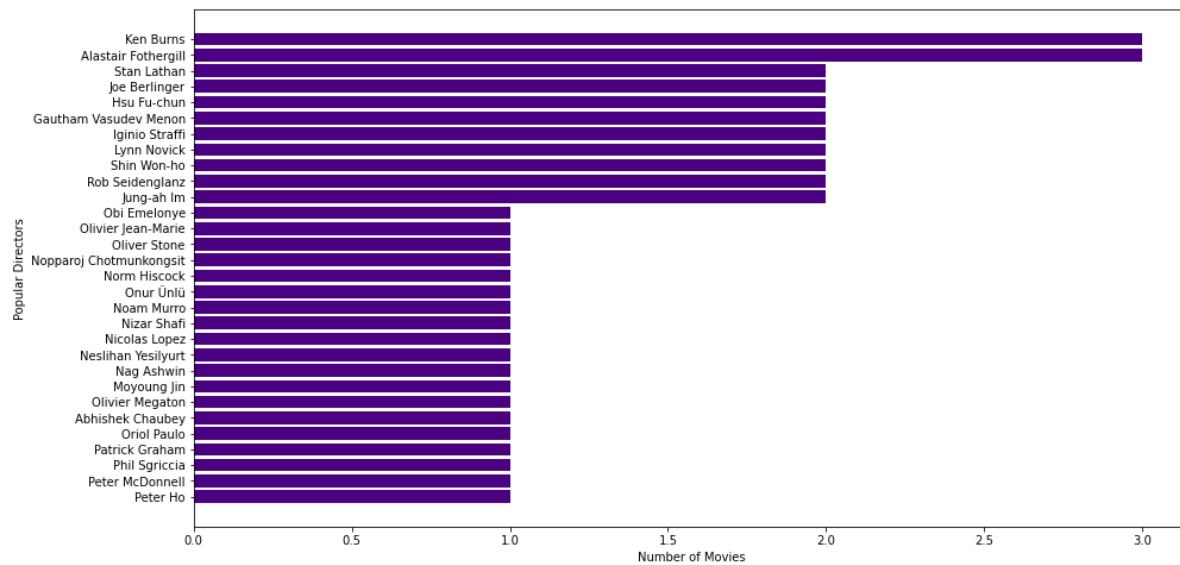
```
df_actors=df_movies.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['Actors'], df_actors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular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



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

In [85]:

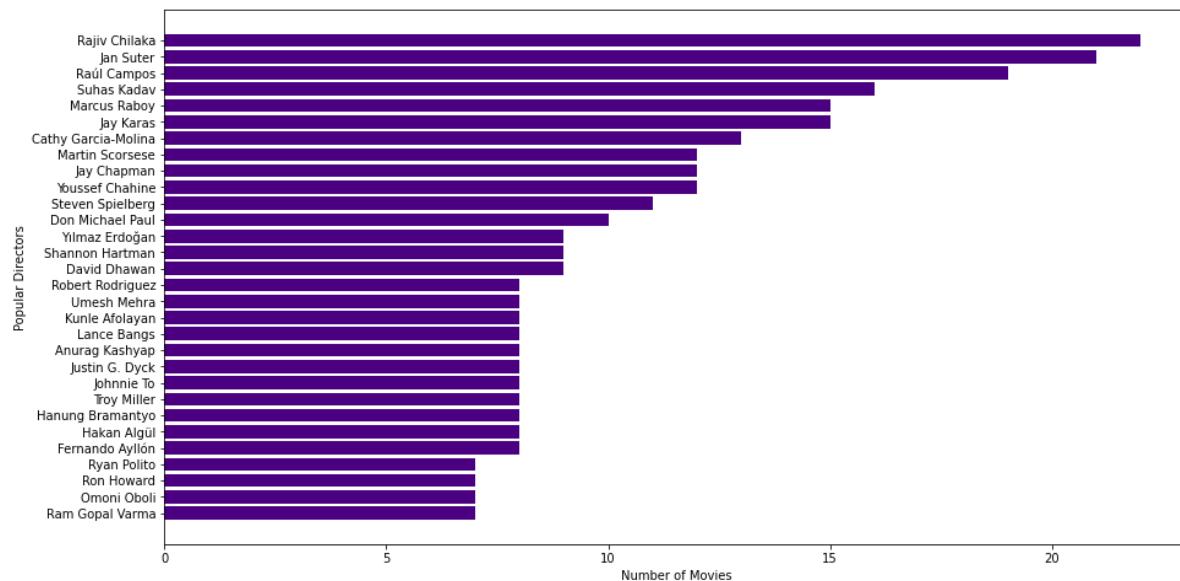
```
df_directors=df_shows.groupby(['Directors']).agg({"title":"nunique"}).reset_index()
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')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



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

In [86]:

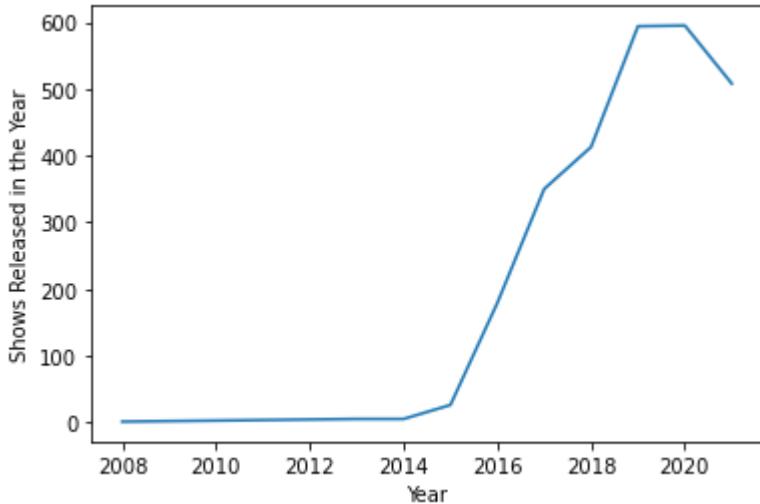
```
df_directors=df_movies.groupby(['Directors']).agg({"title":"nunique"}).reset_index()
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[:::-1]['Directors'], df_directors[:::-1]['title'],color=['indigo'])
plt.xlabel('Number of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



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

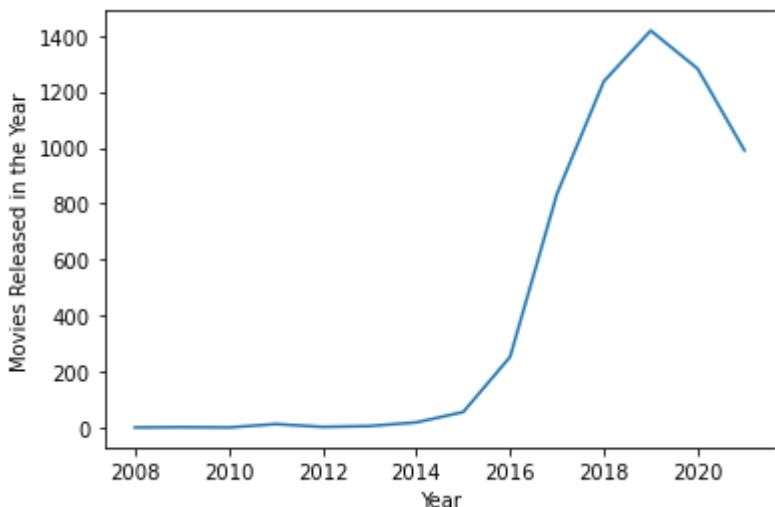
In [87]:

```
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 [88]:

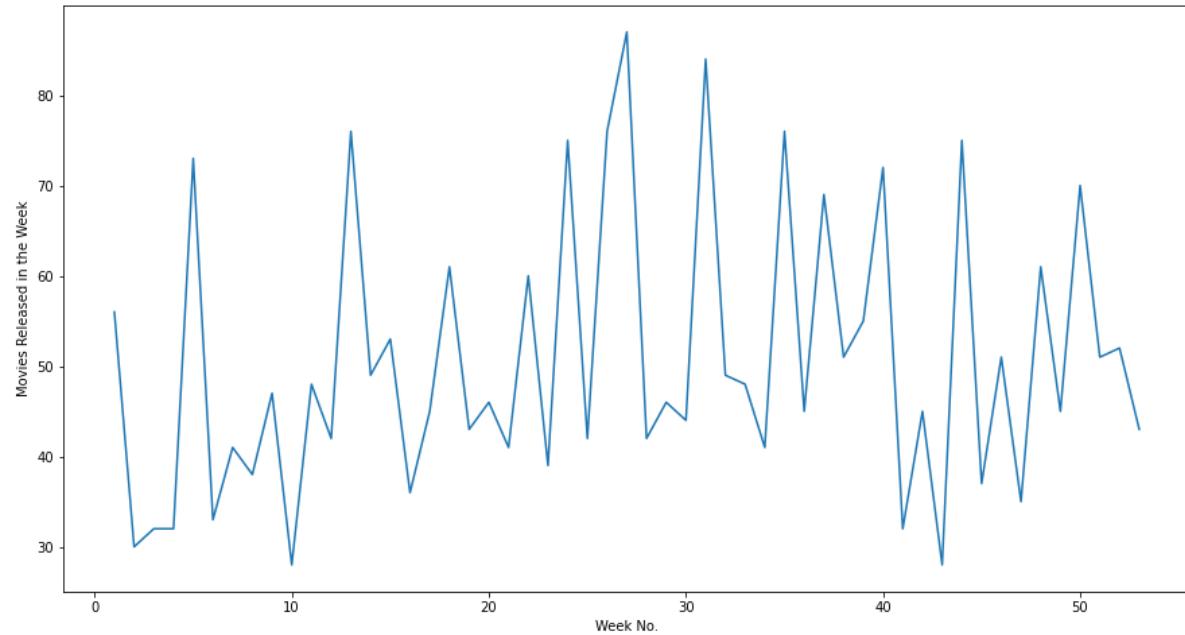
```
df_year=df_movies.groupby(['year']).agg({'title':'nunique'}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



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

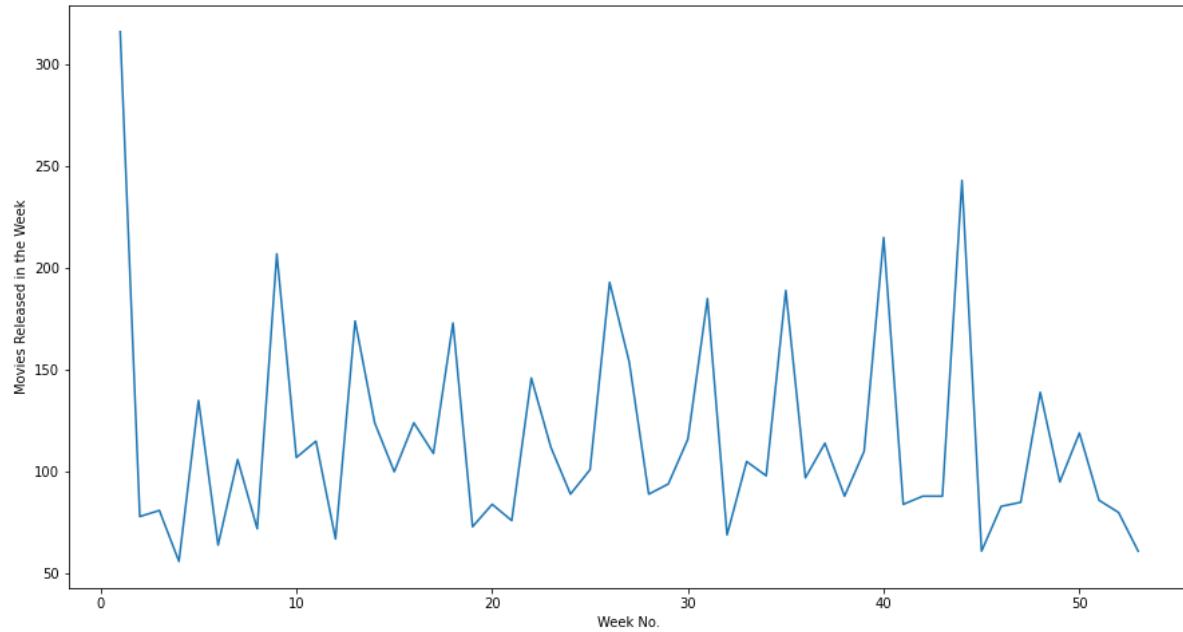
In [89]:

```
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()
```



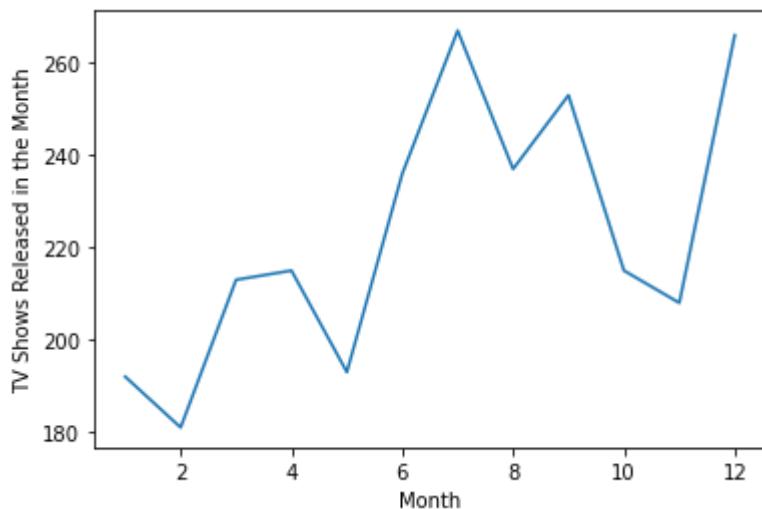
In [90]:

```
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()
```



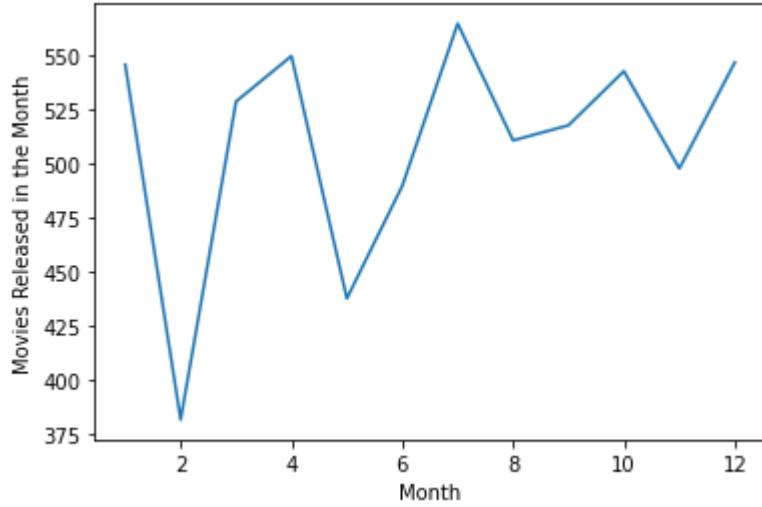
In [91]:

```
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 [92]:

```
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()
```

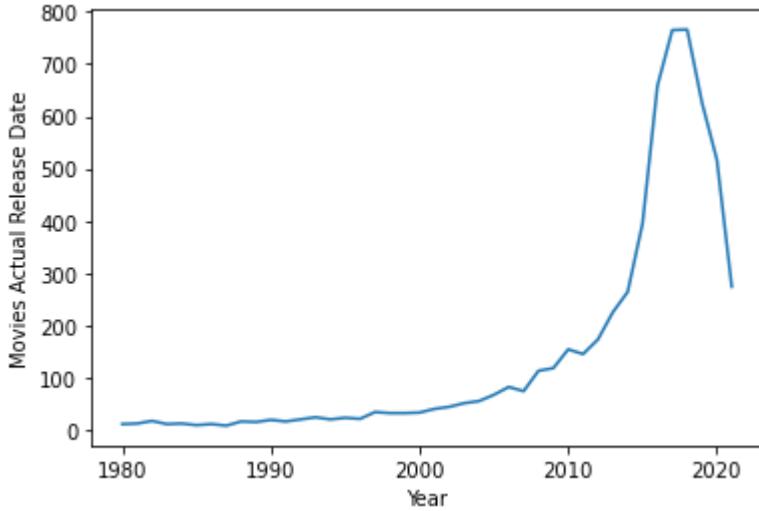


TV Shows are added in Netflix by a tremendous amount in mid weeks/months of the year, i.e- July

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

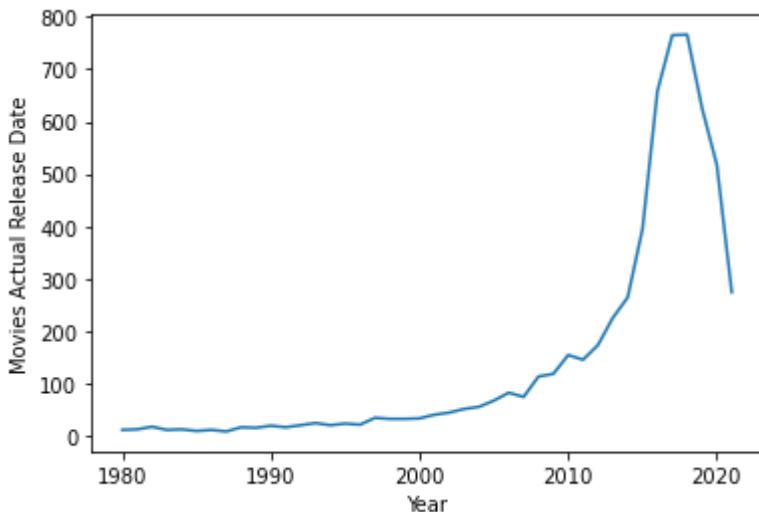
In [93]:

```
df_release_year=df_movies[df_movies['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 [94]:

```
df_release_year=df_movies[df_movies['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()
```



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

## Questions to be Explored Now for Recommendations

- 1) So this time, the granularity level is country and analysis of TV Shows/Movies the country brings. I am going to consider only the top countries individually for TV Shows and Movies. There are definitely some common countries too which bring out quality content in both TV Shows and Movies.

2) Which Genres do these countries offer and what are the intended audiences(Ratings) which are popular in Netflix?

3)In case of Movies, what is the duration/length of movies which makes them special and depicts attention span?

4)Who are the popular actors/directors across TV Shows and Movies in these countries?

5)In what time of the year, people tend to watch movies and shows in these countries?

6)Popular Actor and Director Combinations in these countries

In [95]:

```
#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']
```

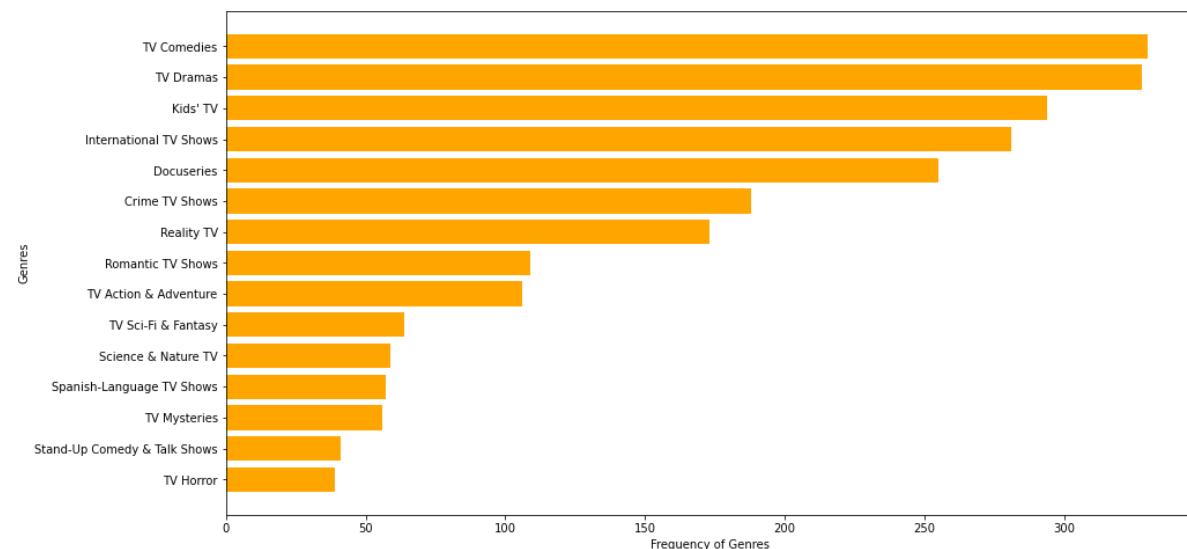
## Univariate Analysis separately for shows and movies in USA

In [96]:

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

In [97]:

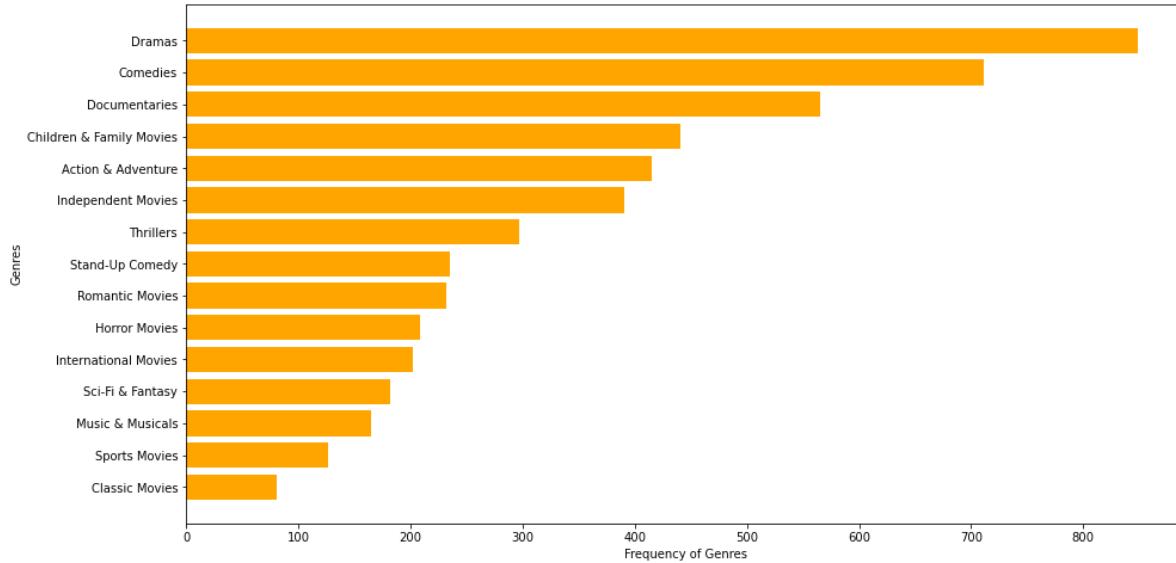
```
_shows.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values(by=['ti
size=(15,8))
genre[:::-1]['Genre'], df_genre[:::-1]['title'],color=['orange'])
frequency of Genres')
ures')
```



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

In [98]:

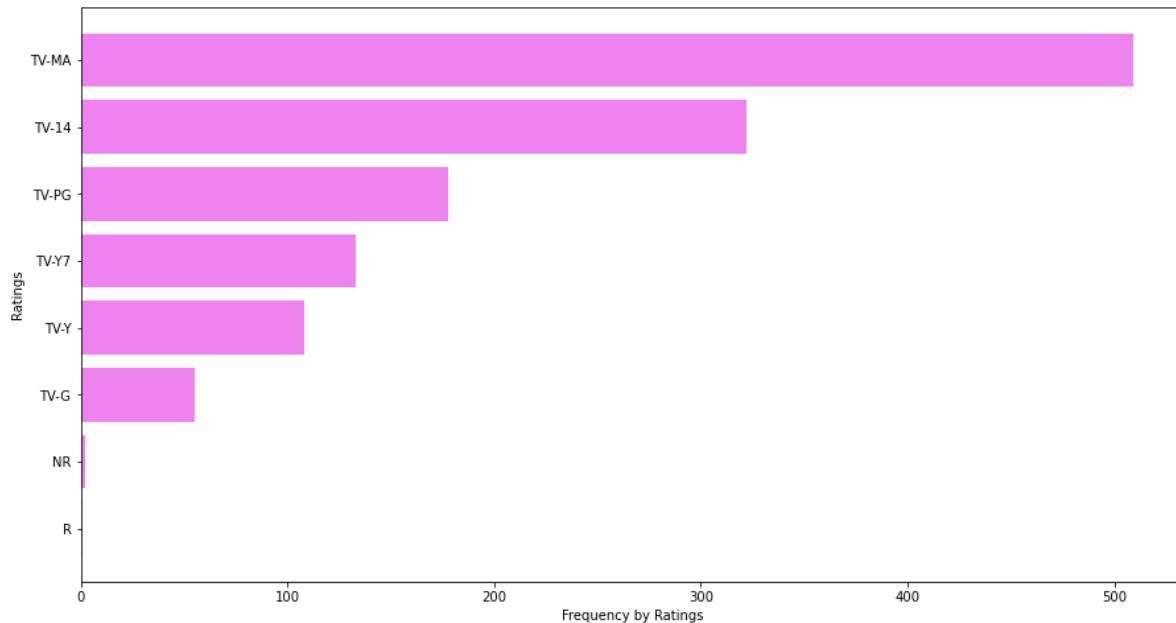
```
df_genre=df_usa_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_genre[:::-1]['Genre'], df_genre[:::-1]['title'], color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



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

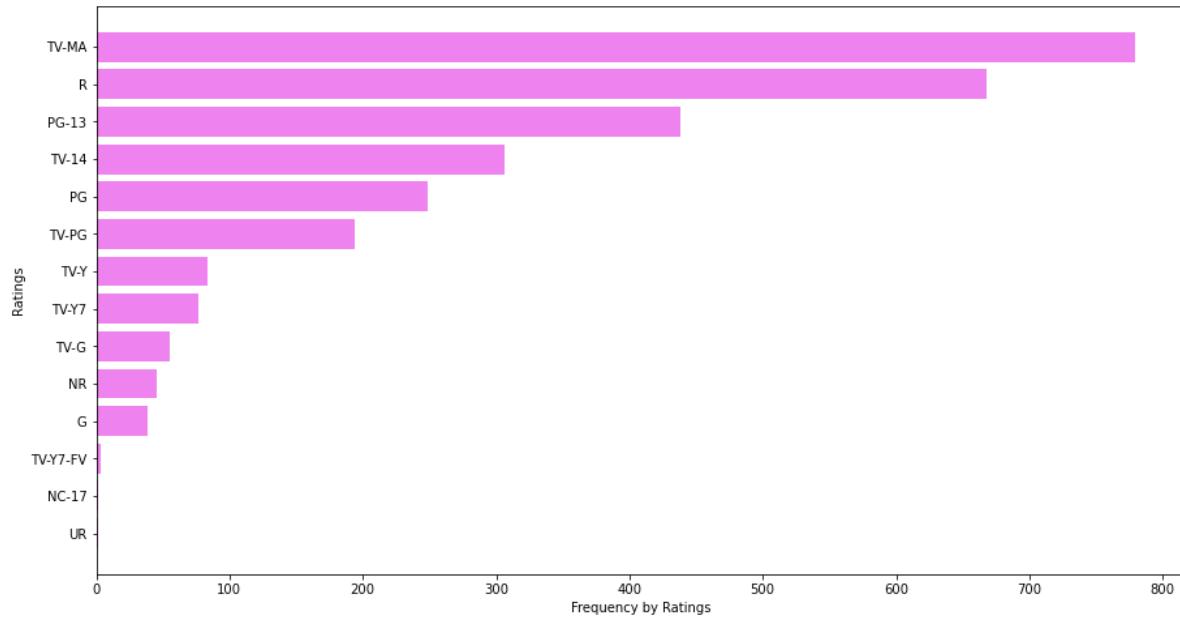
In [99]:

```
df_rating=df_usa_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
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 [100]:

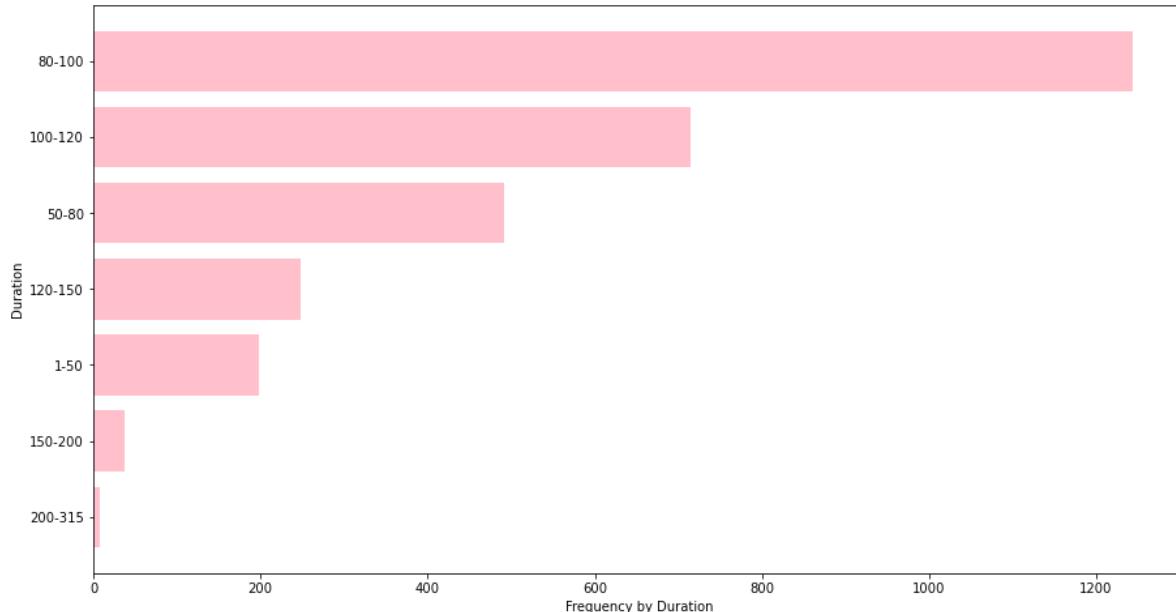
```
df_rating=df_usa_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'], color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()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



So it seems plausible 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 [101]:

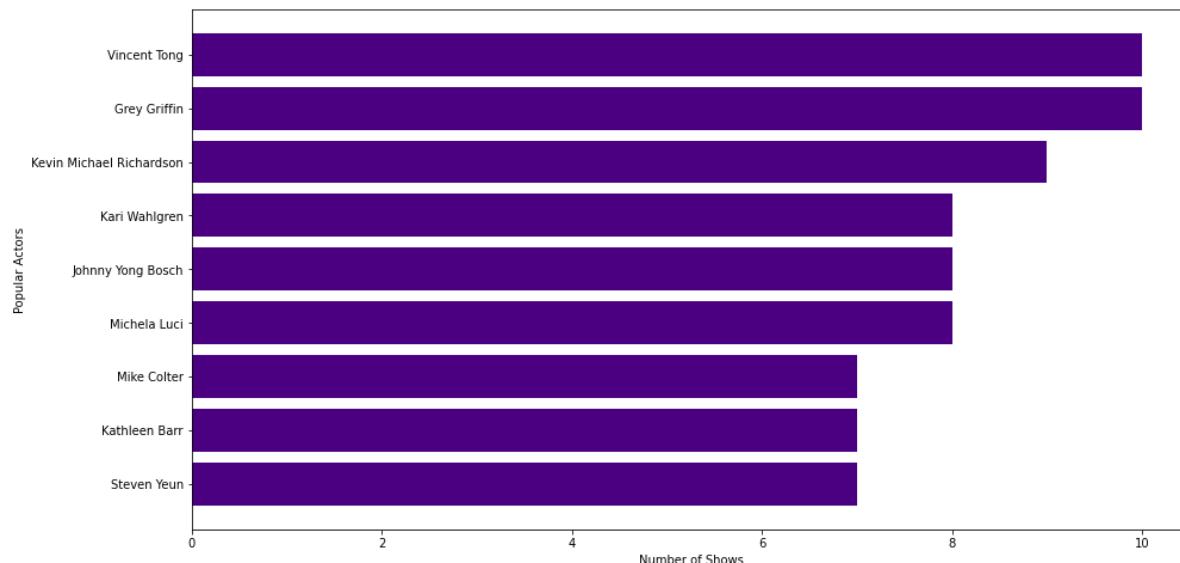
```
df_duration=df_usa_movies.groupby(['duration']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
plt.barh(df_duration[:::-1]['duration'], df_duration[:::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



Across movies 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 [102]:

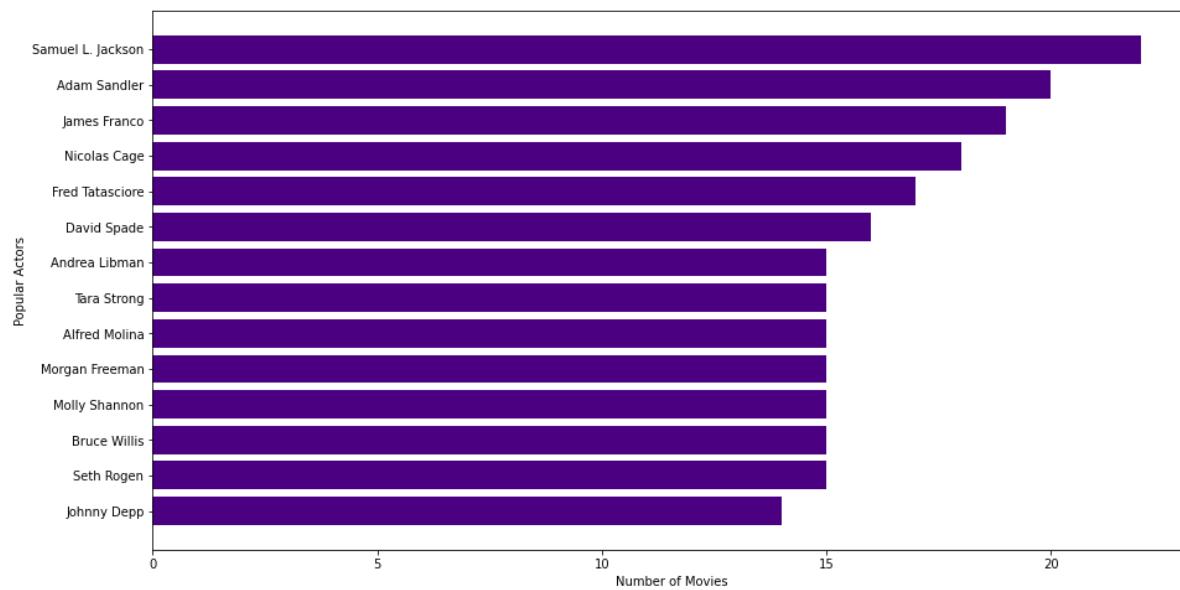
```
df_actors=df_usa_shows.groupby(['Actors']).agg({"title":"nunique"}).reset_index()
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[:::-1]['Actors'], df_actors[:::-1]['title'],color=['indigo'])
plt.xlabel('Number of Shows')
plt.ylabel('Popular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



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

In [103]:

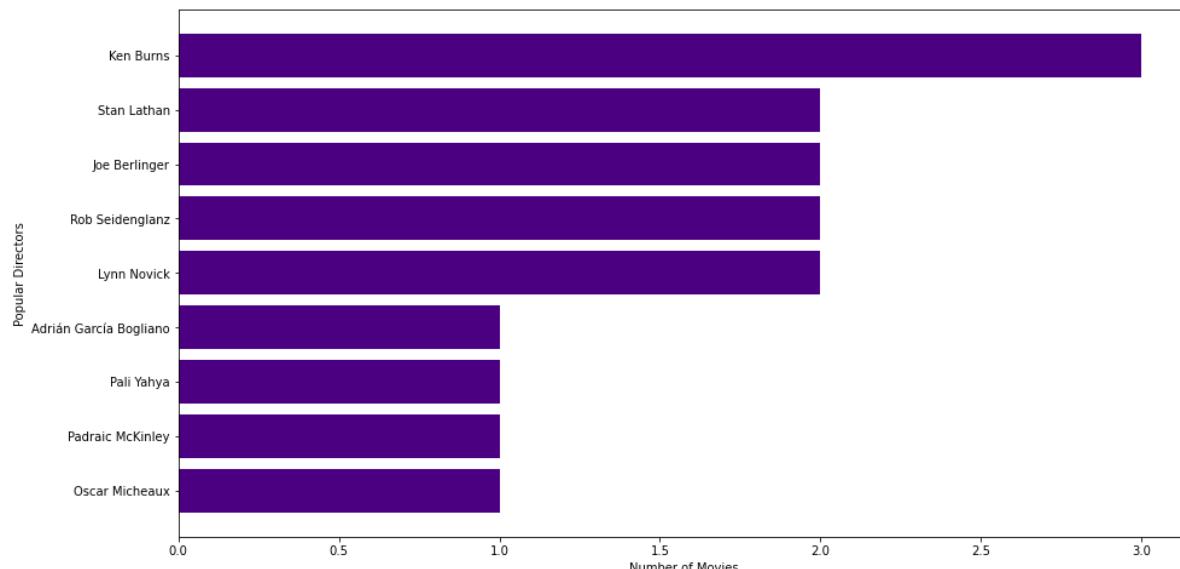
```
df_actors=df_usa_movies.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_values('title', ascending=False)
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[:::-1]['Actors'], df_actors[:::-1]['title'],color=['indigo'])
plt.xlabel('Number 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 [104]:

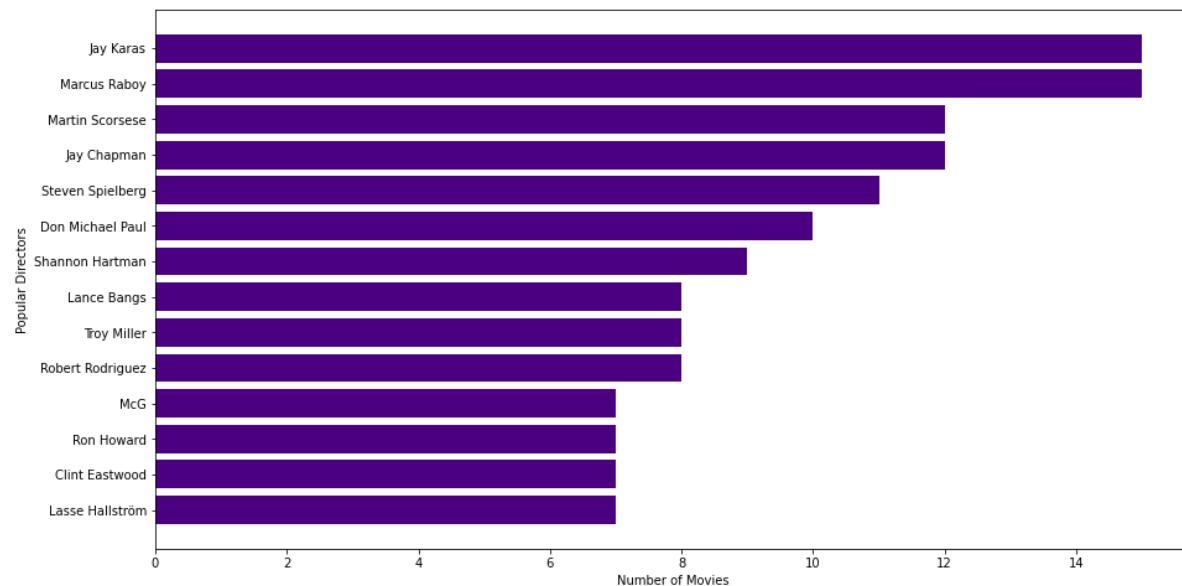
```
df_directors=df_usa_shows.groupby(['Directors']).agg({"title":"nunique"}).reset_index().sort_values('title', ascending=False)
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[:::-1]['Directors'], df_directors[:::-1]['title'],color=['indigo'])
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 [105]:

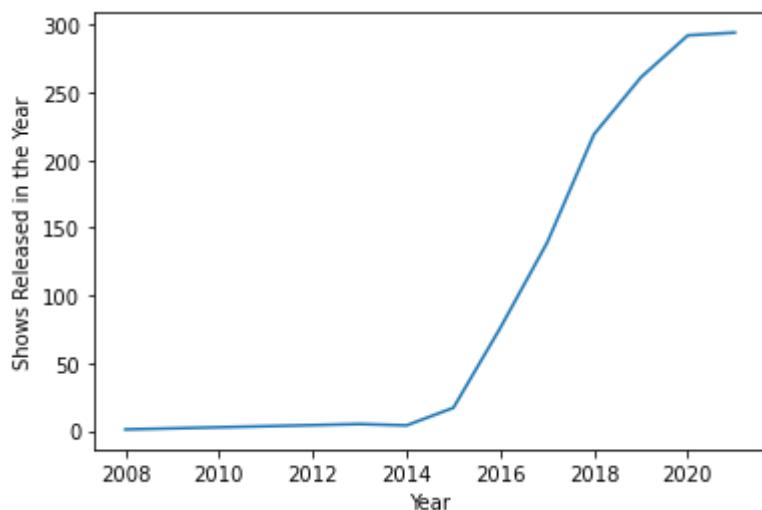
```
df_directors=df_usa_movies.groupby(['Directors']).agg({"title":"nunique"}).reset_index()
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of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



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

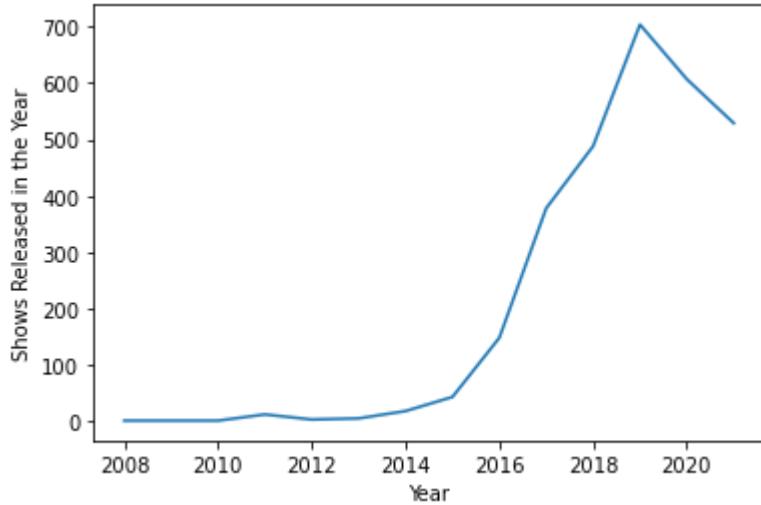
In [106]:

```
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 [107]:

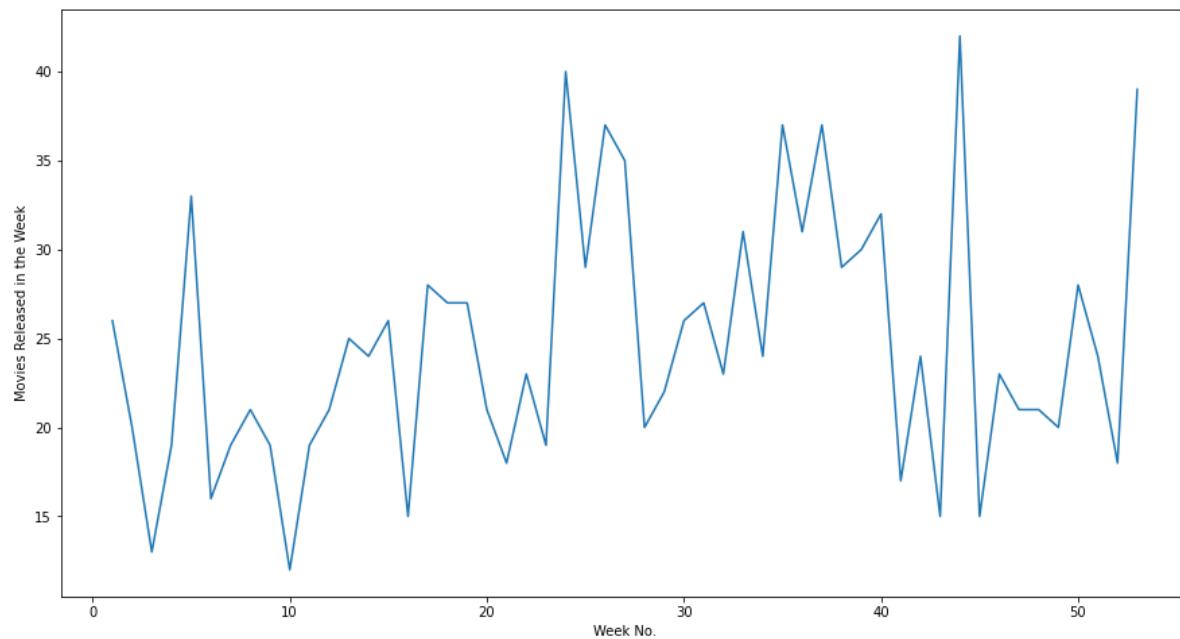
```
df_year=df_usa_movies.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Shows Released in the Year")
plt.xlabel("Year")
plt.show()
```



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

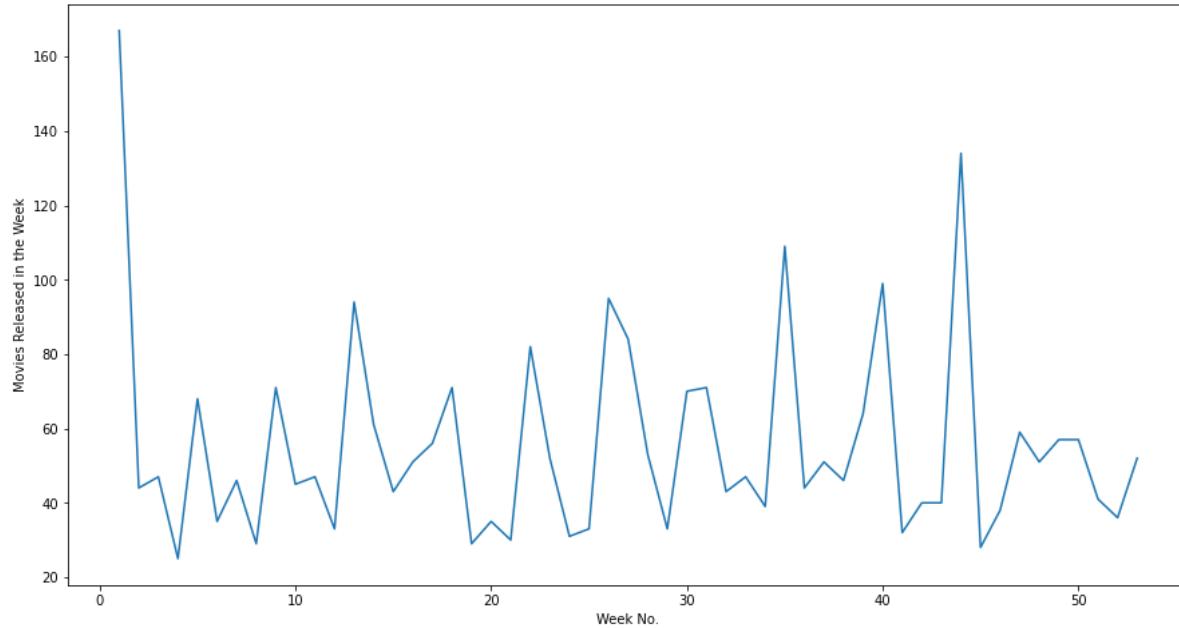
In [108]:

```
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()
```



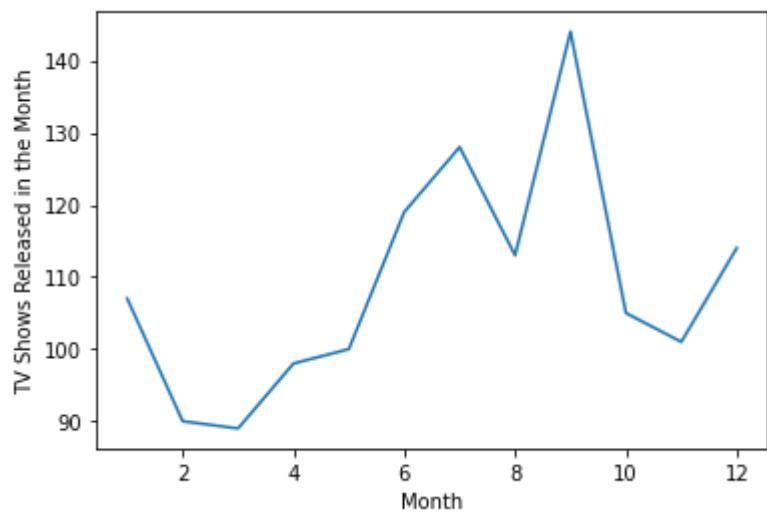
In [109]:

```
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()
```



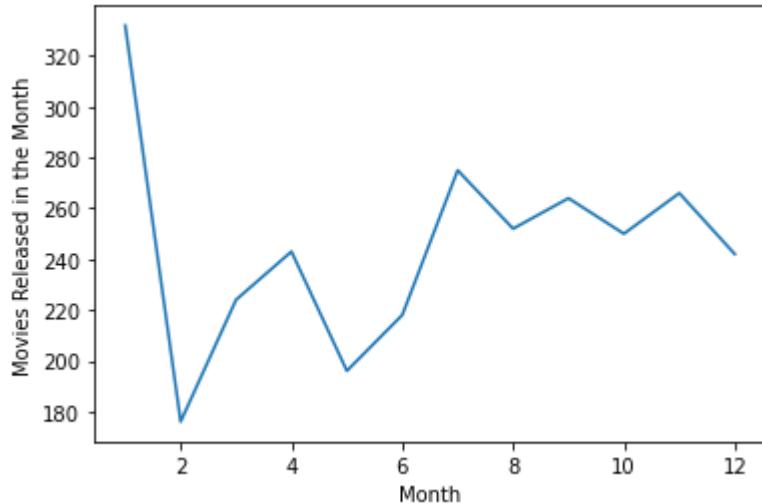
In [110]:

```
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 [111]:

```
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()
```

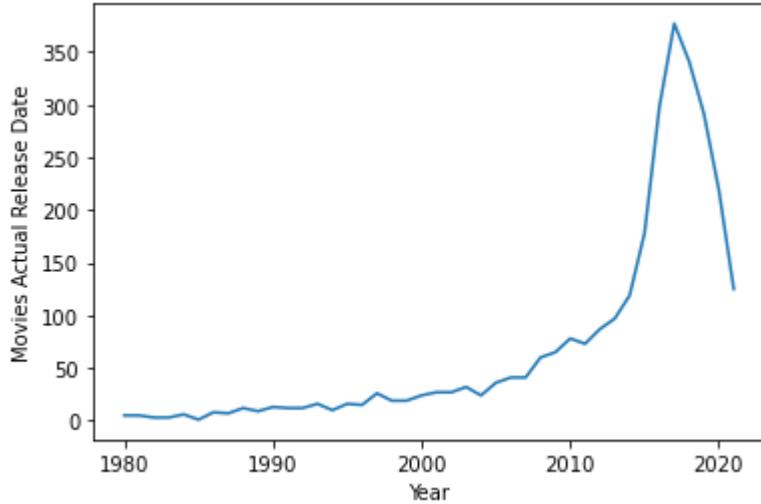


TV Shows are added in Netflix by a tremendous amount in July and September in USA

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

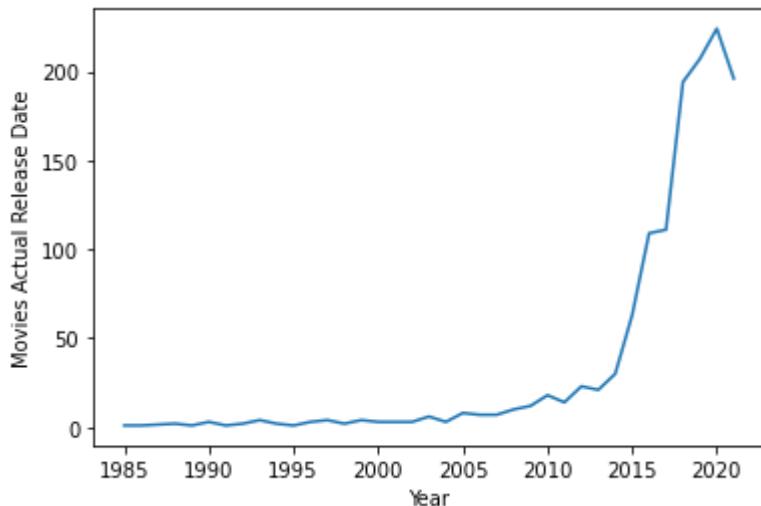
In [112]:

```
df_release_year=df_usa_movies[df_usa_movies['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 [113]:

```
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 [114]:

df\_usa\_movies.head()

Out[114]:

	<b>title</b>	<b>Actors</b>	<b>Directors</b>	<b>Genre</b>	<b>country</b>	<b>show_id</b>	<b>type</b>	<b>date_added</b>	<b>releas</b>
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 [115]:

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

Out[115]:

	<b>title</b>	<b>Actors</b>	<b>Directors</b>	<b>Genre</b>	<b>country</b>	<b>show_id</b>	<b>type</b>	<b>date_added</b>	<b>release_yea</b>
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	202
160	My Little Pony: A New Generation	Vanessa Hudgens	José Luis Ucha	Children & Family Movies	United States	s7	Movie	September 24, 2021	202
161	My Little Pony: A New Generation	Kimiko Glenn	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	202
162	My Little Pony: A New Generation	Kimiko Glenn	José Luis Ucha	Children & Family Movies	United States	s7	Movie	September 24, 2021	202
163	My Little Pony: A New Generation	James Marsden	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	202

In [116]:

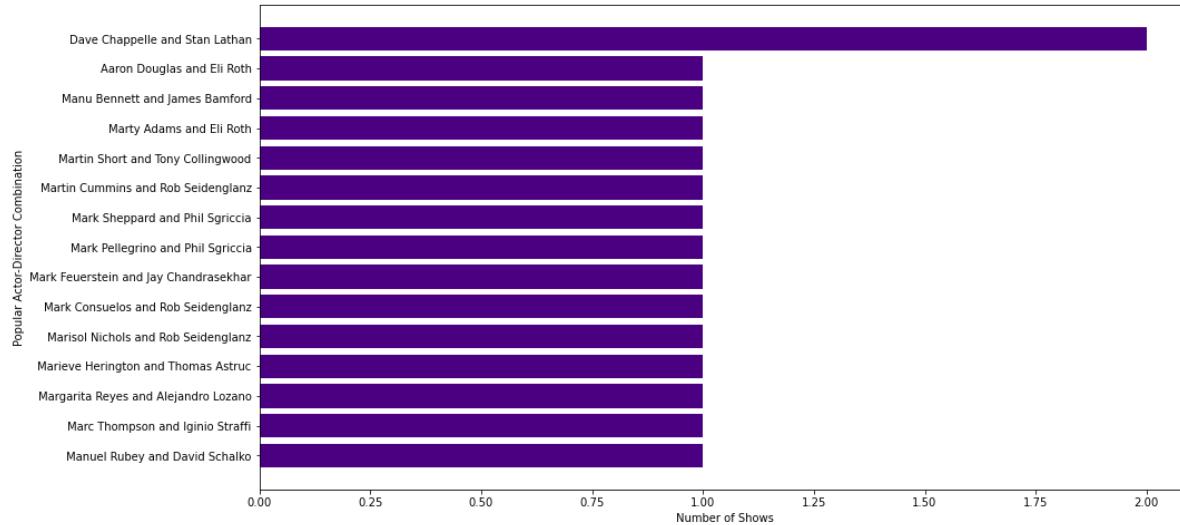
```
df_usa_shows['Actor_Director_Combination'] = df_usa_shows.Actors.str.cat(df_usa_shows.Directors)
df_usa_shows_subset=df_usa_shows[df_usa_shows['Actors']!='Unknown Actor']
df_usa_shows_subset=df_usa_shows_subset[df_usa_shows_subset['Directors']!='Unknown Director']
df_usa_shows_subset.head()
```

Out[116]:

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

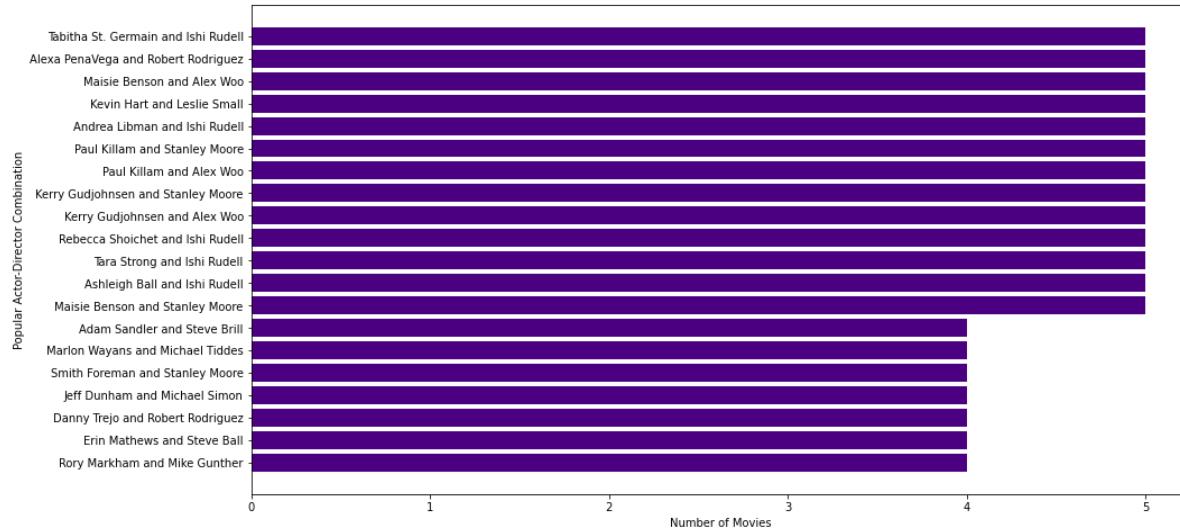
In [117]:

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



In [118]:

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



**In [119]:**

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

**Out[119]:**

```
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'

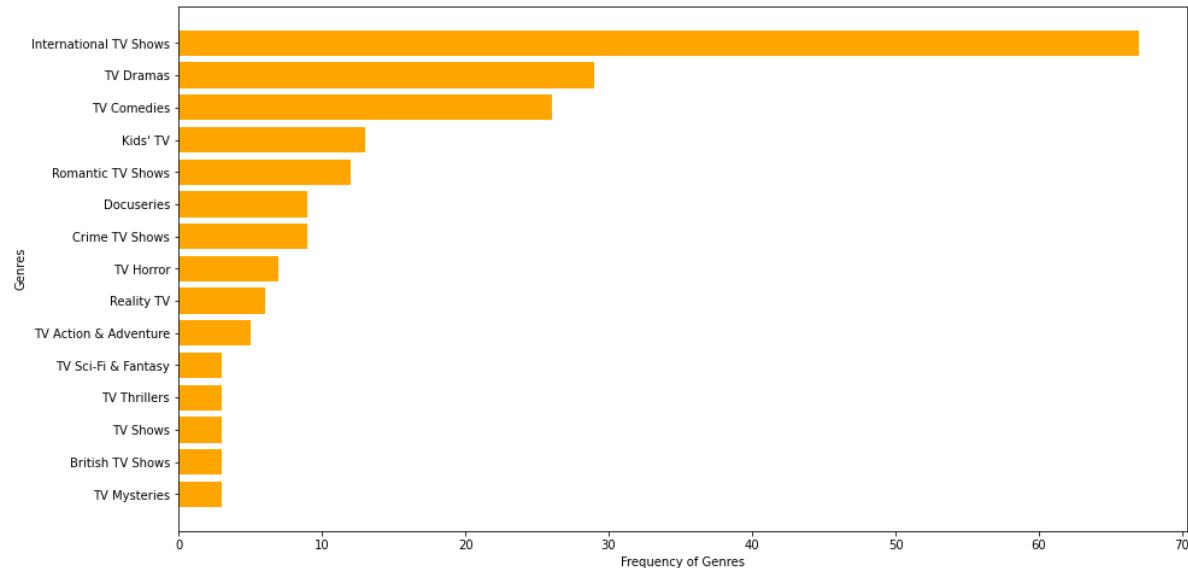
**Univariate Analysis separately for shows and movies in India**

In [120]:

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

In [121]:

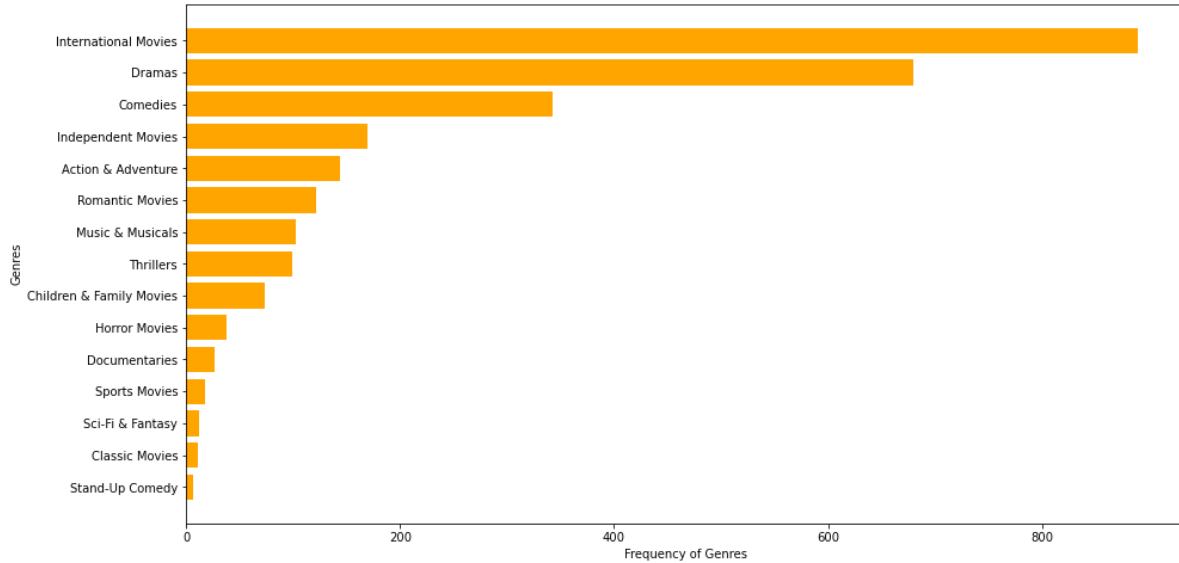
```
df_genre=df_india_shows.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_genre[:::-1]['Genre'], df_genre[:::-1]['title'], color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



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

In [122]:

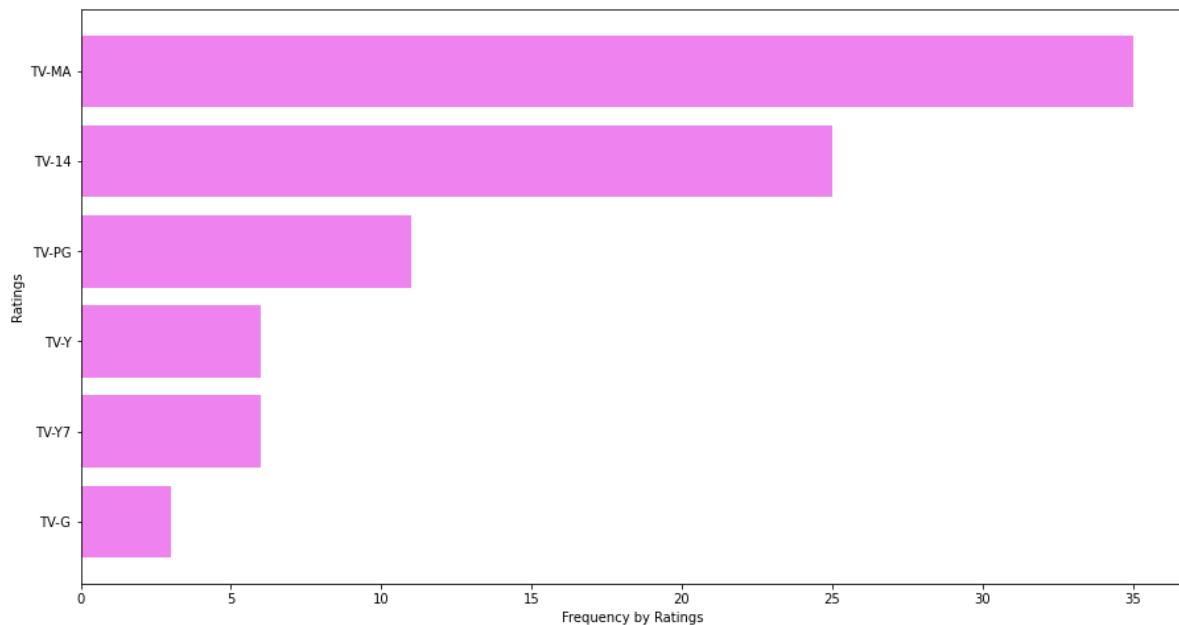
```
df_genre=df_india_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values('title', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_genre[:::-1]['Genre'], df_genre[:::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



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

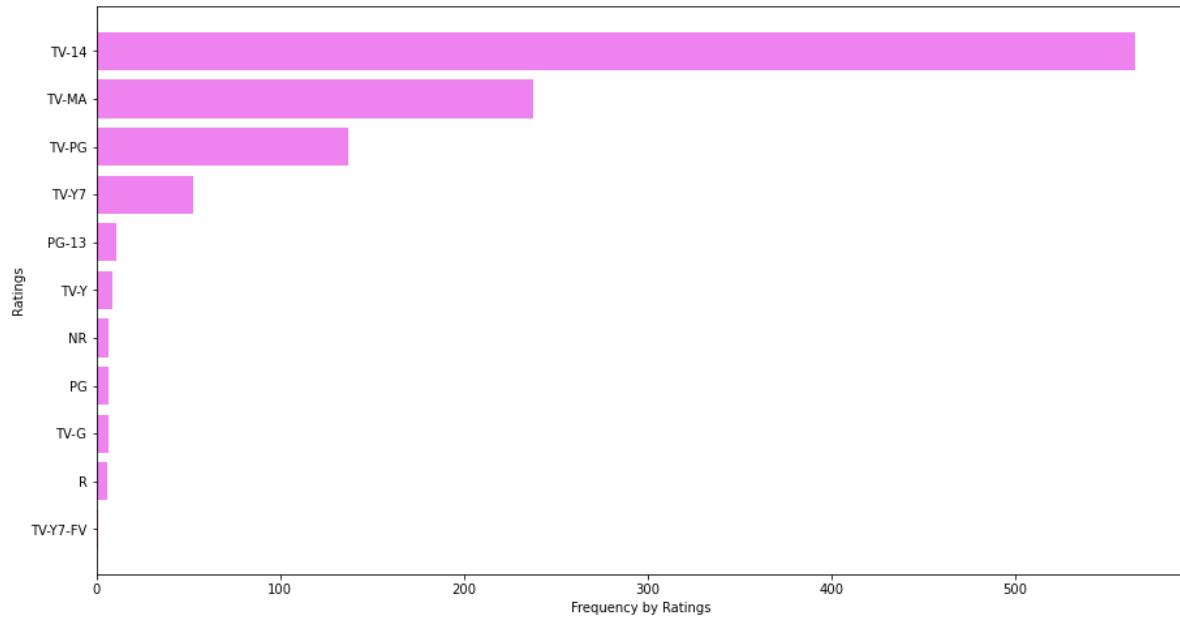
In [123]:

```
df_rating=df_india_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index()
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 [124]:

```
df_rating=df_india_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index()
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()
```

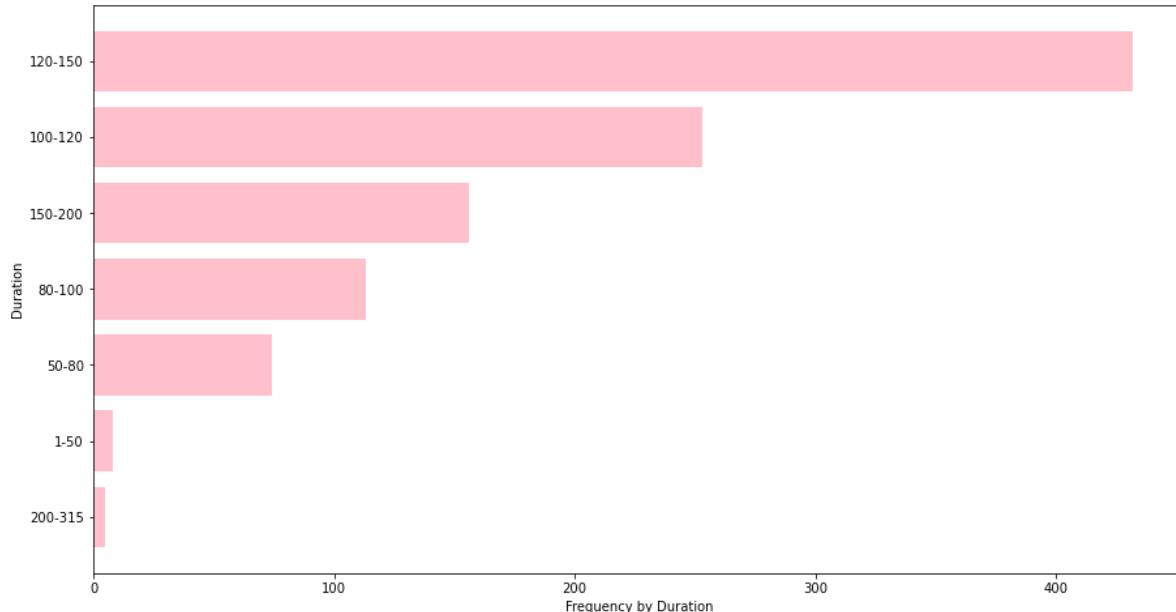


So it seems plausible to conclude that the popular ratings across Netflix includes Mature Audiences in TV Shows and those appropriate for people over 14 in Movies in India.

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 [125]:

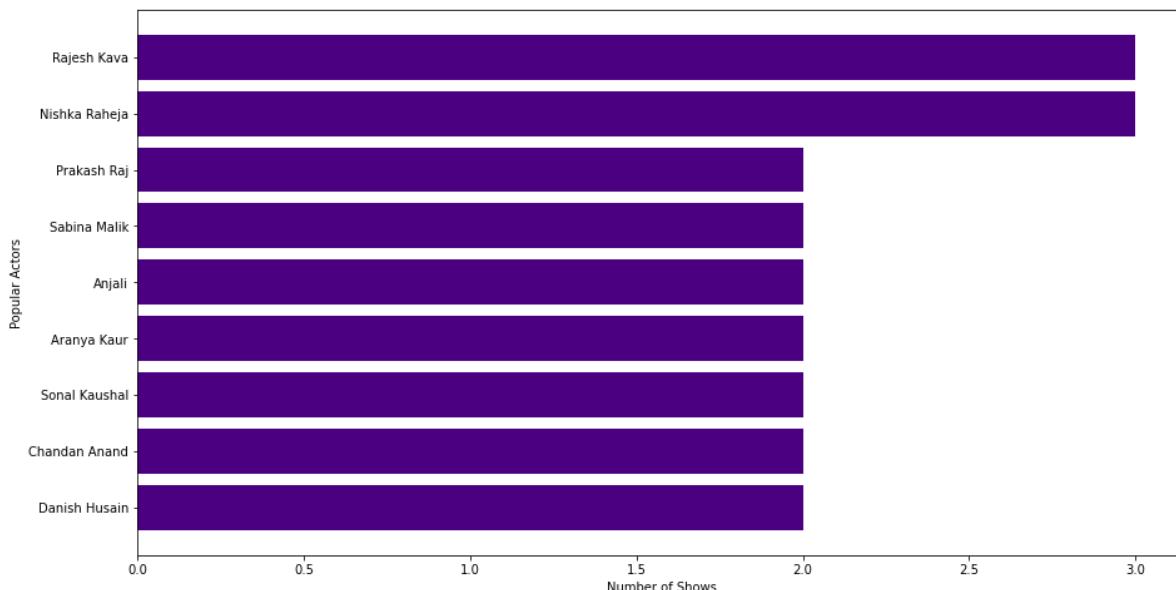
```
df_duration=df_india_movies.groupby(['duration']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
plt.barh(df_duration[:::-1]['duration'], df_duration[:::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



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

In [126]:

```
df_actors=df_india_shows.groupby(['Actors']).agg({"title":"nunique"}).reset_index()
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 [127]:

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

Out[127]:

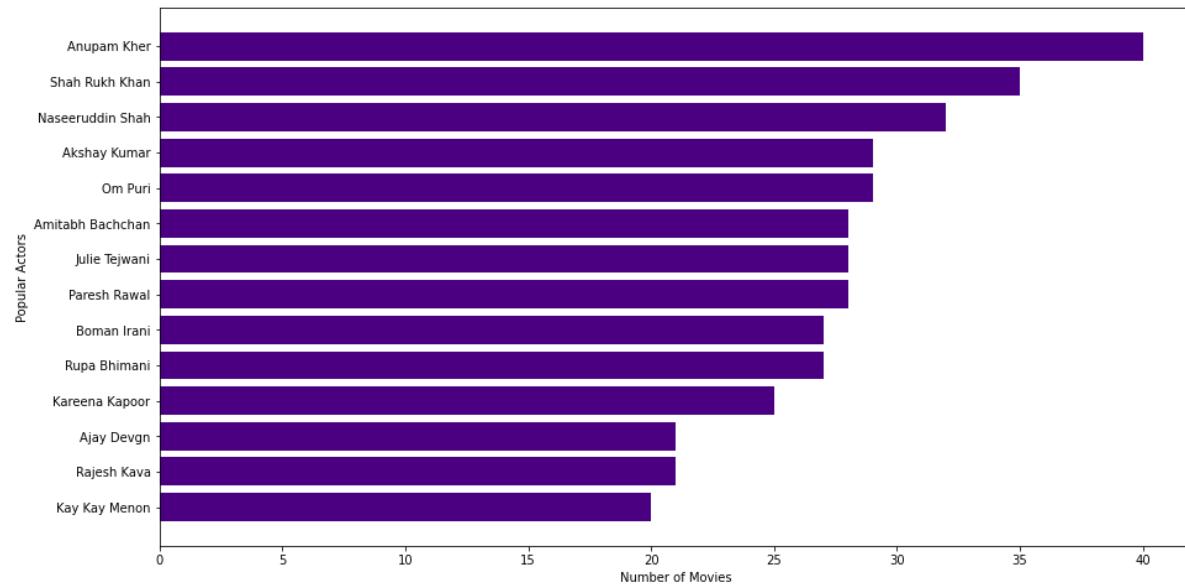
```
array(['Rajesh Kava', 'Nishka Raheja', 'Prakash Raj', 'Sabina Malik',
       'Anjali', 'Aranya Kaur', 'Sonal Kaushal', 'Chandan Anand',
       'Danish Husain'], dtype=object)
```

**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 [128]:

```
df_actors=df_india_movies.groupby(['Actors']).agg({"title":"nunique"}).reset_index()
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 [129]:

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

Out[129]:

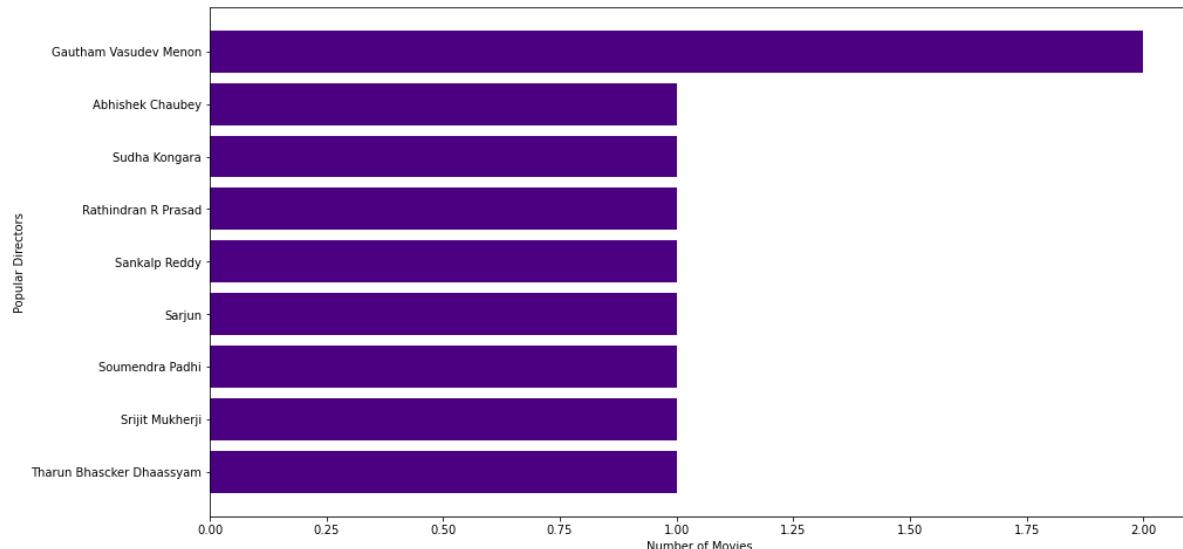
```
array(['Anupam Kher', 'Shah Rukh Khan', 'Naseeruddin Shah',
       'Akshay Kumar', 'Om Puri', 'Amitabh Bachchan', 'Julie Tejwani',
       'Paresh Rawal', 'Boman Irani', 'Rupa Bhimani', 'Kareena Kapoor',
       'Ajay Devgn', 'Rajesh Kava', 'Kay Kay Menon'], dtype=object)
```

**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 [130]:

```
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 [131]:

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

Out[131]:

```
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:-**

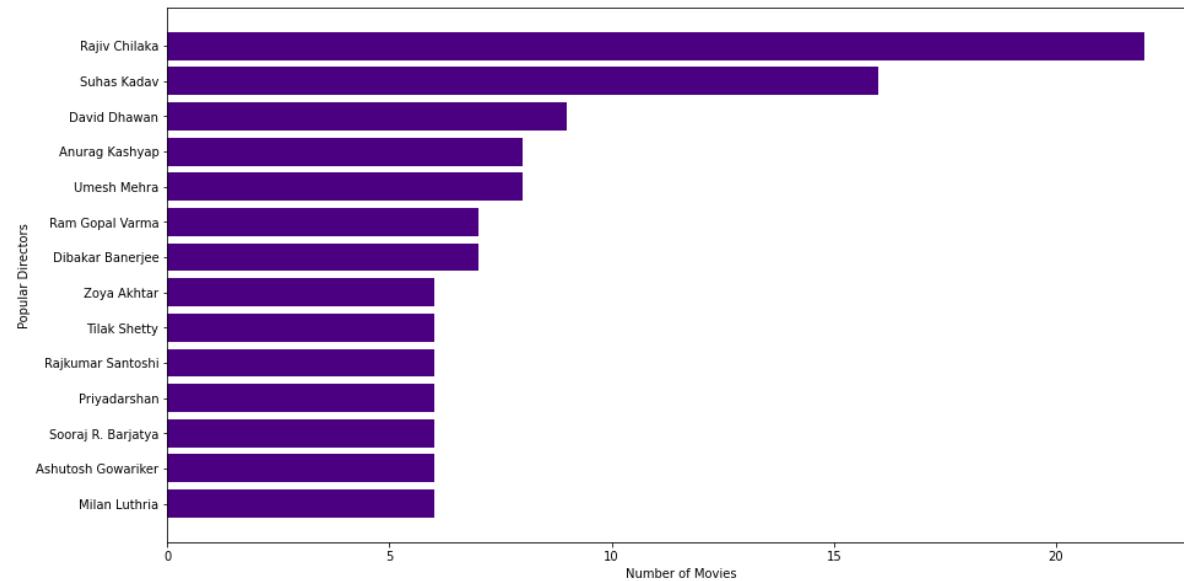
'Gautham Vasudev Menon',  
 'Abhishek Chaubey',  
 'Sudha Kongara',  
 'Rathindran R Prasad',  
 'Sankalp Reddy',  
 'Sarjun',  
 'Soumendra Padhi',  
 'Srijit Mukherji',  
 'Tharun Bhascker Dhaassyam'

In [132]:

```
es.groupby(['Directors']).agg({"title": "nunique"}).reset_index().sort_values(by=['tit
if_directors['Directors']!='Unknown Director']

['Directors'], df_directors[::-1]['title'],color=['indigo'])

es')
ors')
```



In [133]:

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

Out[133]:

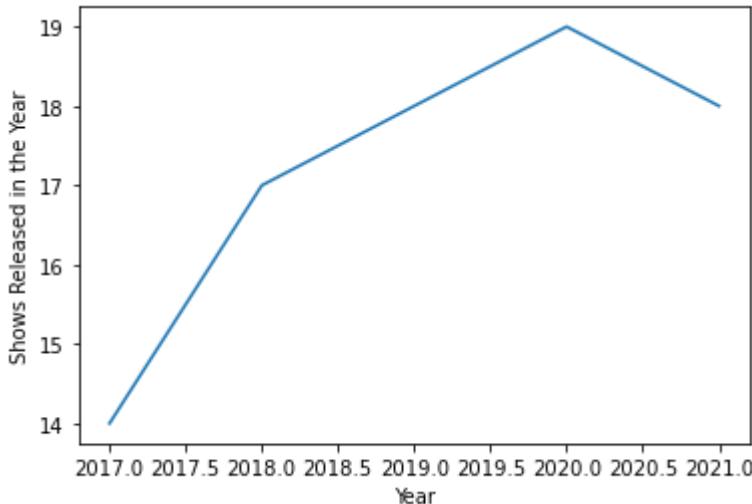
```
array(['Rajiv Chilaka', 'Suhas Kadav', 'David Dhawan', 'Anurag Kashyap',
       'Umesh Mehra', 'Ram Gopal Varma', 'Dibakar Banerjee',
       'Zoya Akhtar', 'Tilak Shetty', 'Rajkumar Santoshi', 'Priyadarshan',
       'Sooraj R. Barjatya', 'Ashutosh Gowariker', 'Milan Luthria'],
      dtype=object)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

**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'

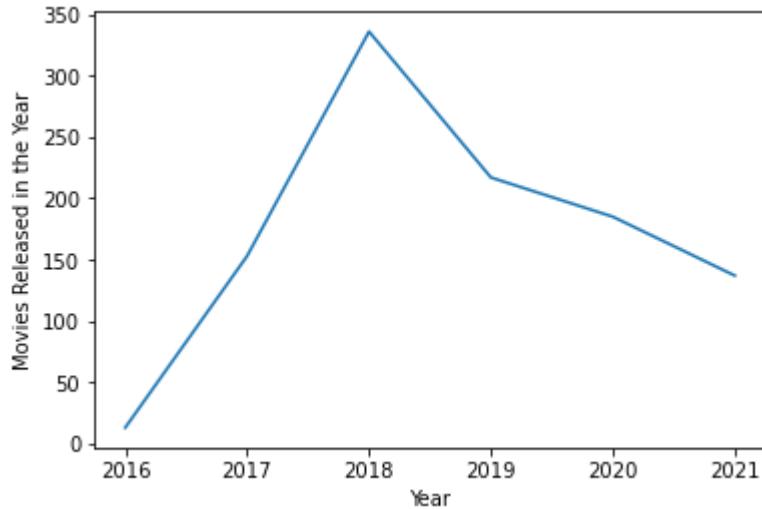
In [134]:

```
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 [135]:

```
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()
```

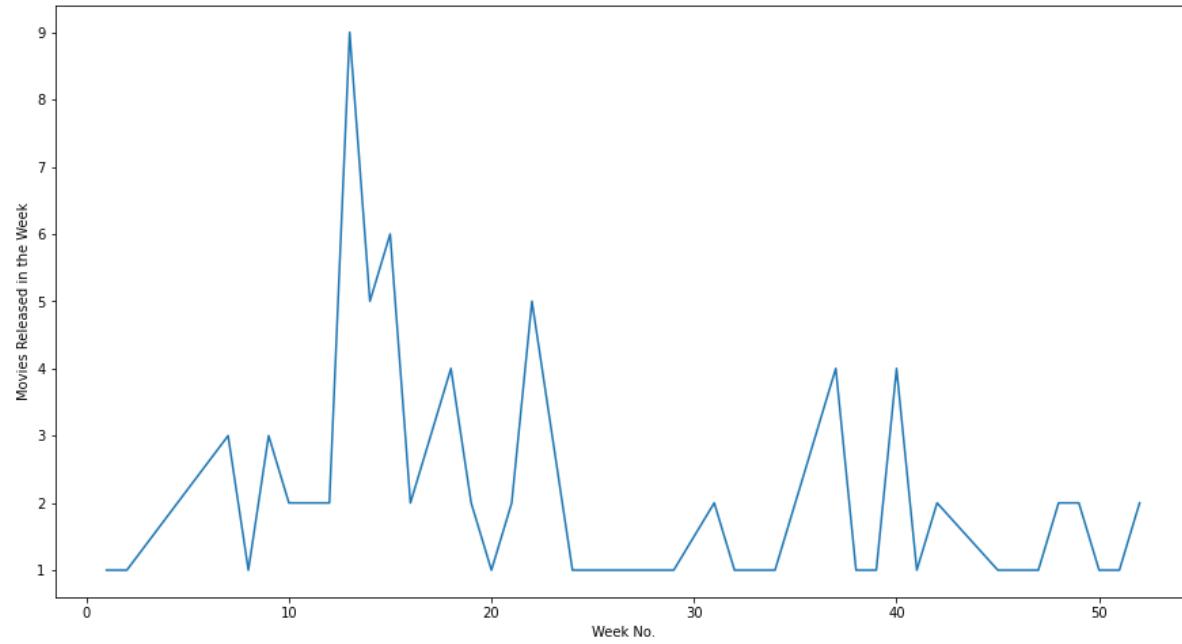


In India, TV Shows were increasingly being added till 2020, though the addition of shows reduced in 2021.

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

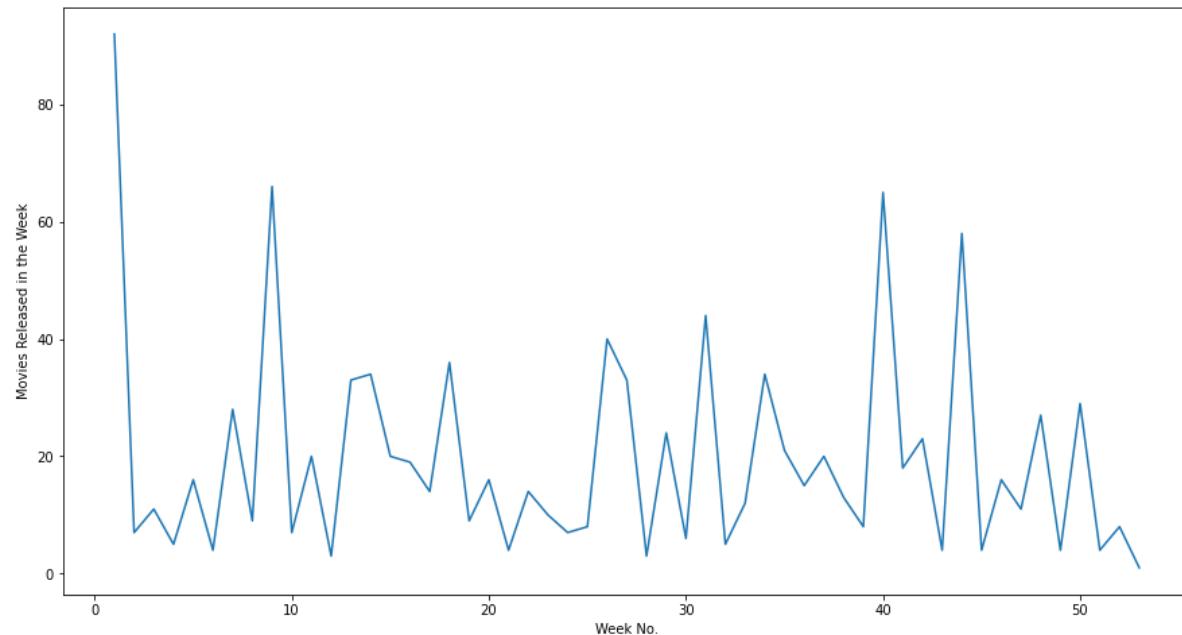
In [136]:

```
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()
```



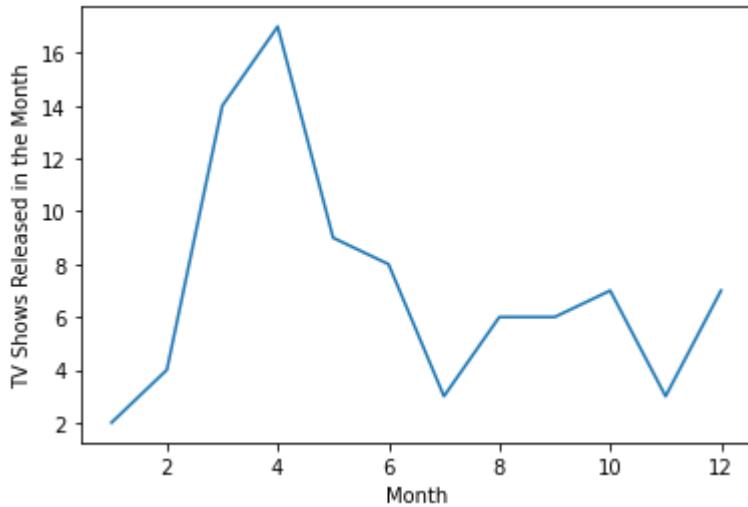
In [137]:

```
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()
```



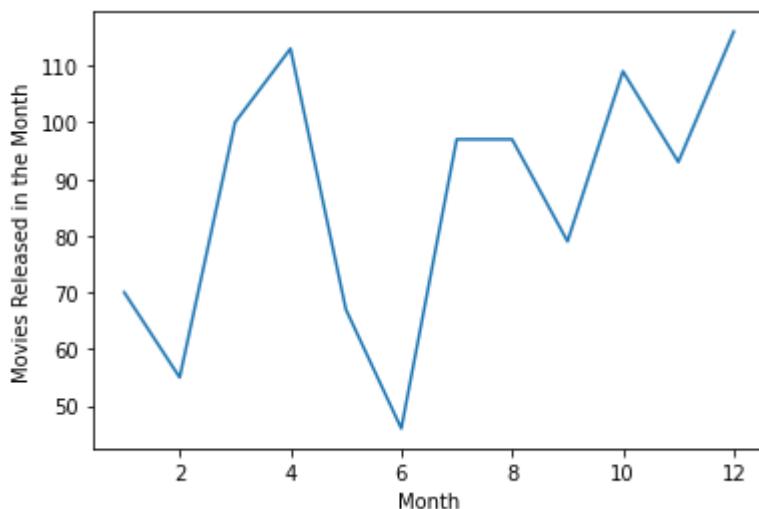
In [138]:

```
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 [139]:

```
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()
```

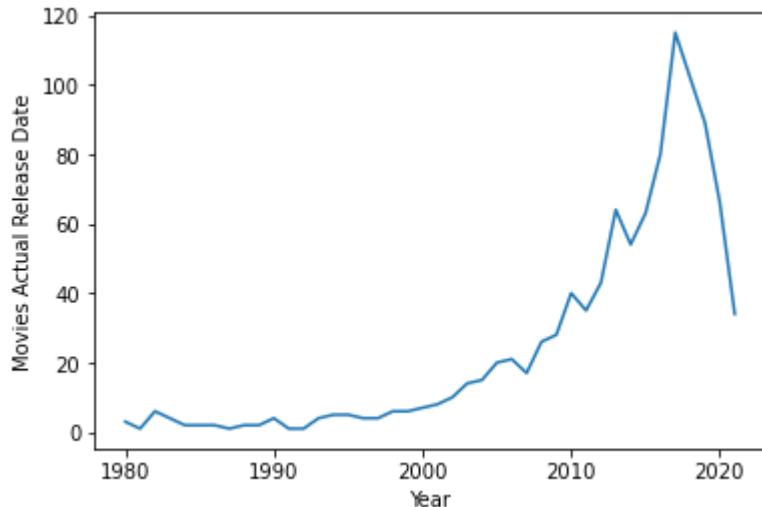


TV Shows are added in Netflix by a tremendous amount in April in India

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

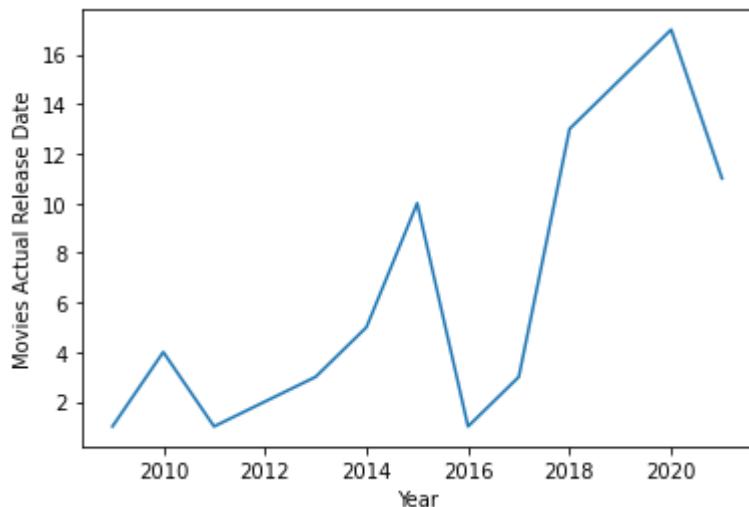
In [140]:

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



In [141]:

```
df_release_year=df_india_shows[df_india_shows['release_year']>=1980].groupby(['rele
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 [142]:

```
#Analysing a combination of actors and directors
```

```
df_india_movies['Actor_Director_Combination'] = df_india_movies.Actors.str.cat(df_ir
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']!=''
df_india_movies_subset.head()
```

Out[142]:

	title	Actors	Directors	Genre	country	show_id	type	date_added
621	Avvai Shanmughi	Kamal Hassan	K.S. Ravikumar	Comedies	India	s23	Movie	September 21, 2022
622	Avvai Shanmughi	Kamal Hassan	K.S. Ravikumar	International Movies	India	s23	Movie	September 21, 2022
629	Avvai Shanmughi	Nassar	K.S. Ravikumar	Comedies	India	s23	Movie	September 21, 2022
630	Avvai Shanmughi	Nassar	K.S. Ravikumar	International Movies	India	s23	Movie	September 21, 2022
631	Avvai Shanmughi	S.P. Balasubrahmanyam	K.S. Ravikumar	Comedies	India	s23	Movie	September 21, 2022

In [143]:

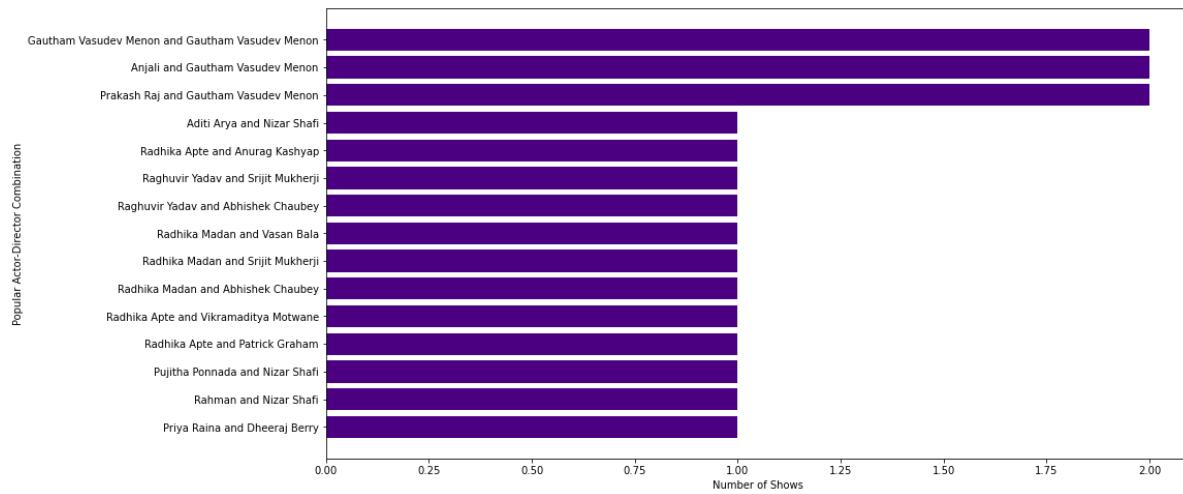
```
df_india_shows['Actor_Director_Combination'] = df_india_shows.Actors.str.cat(df_ir
df_india_shows_subset=df_india_shows[df_india_shows['Actors']!='Unknown Actor']
df_india_shows_subset=df_india_shows_subset[df_india_shows_subset['Directors']!=''
df_india_shows_subset.head()
```

Out[143]:

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

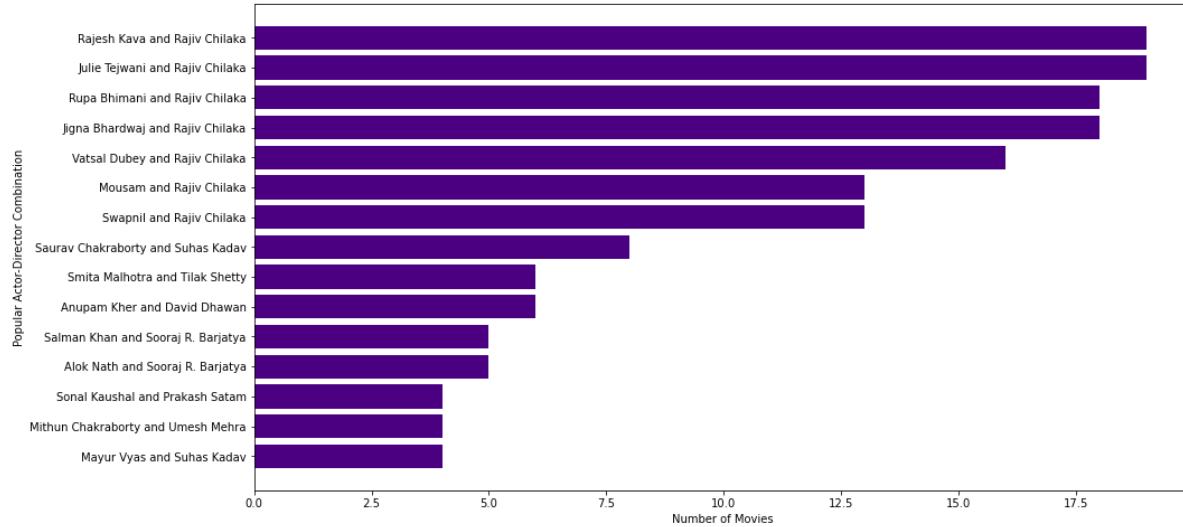
In [144]:

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



In [145]:

```
df_actors_directors=df_india_movies_subset.groupby(['Actor_Director_Combination']).size()
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 [146]:

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

Out[146]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating
10058	Chhota Bheem - Neeli Pahaadi	Vatsal Dubey	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	2013	TV-Y7
10059	Chhota Bheem - Neeli Pahaadi	Julie Tejwani	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	2013	TV-Y7
10060	Chhota Bheem - Neeli Pahaadi	Rupa Bhimani	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	2013	TV-Y7
10061	Chhota Bheem - Neeli Pahaadi	Jigna Bhardwaj	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	2013	TV-Y7

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 [147]:

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

Out[147]:

```
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 Kadav'],
      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',

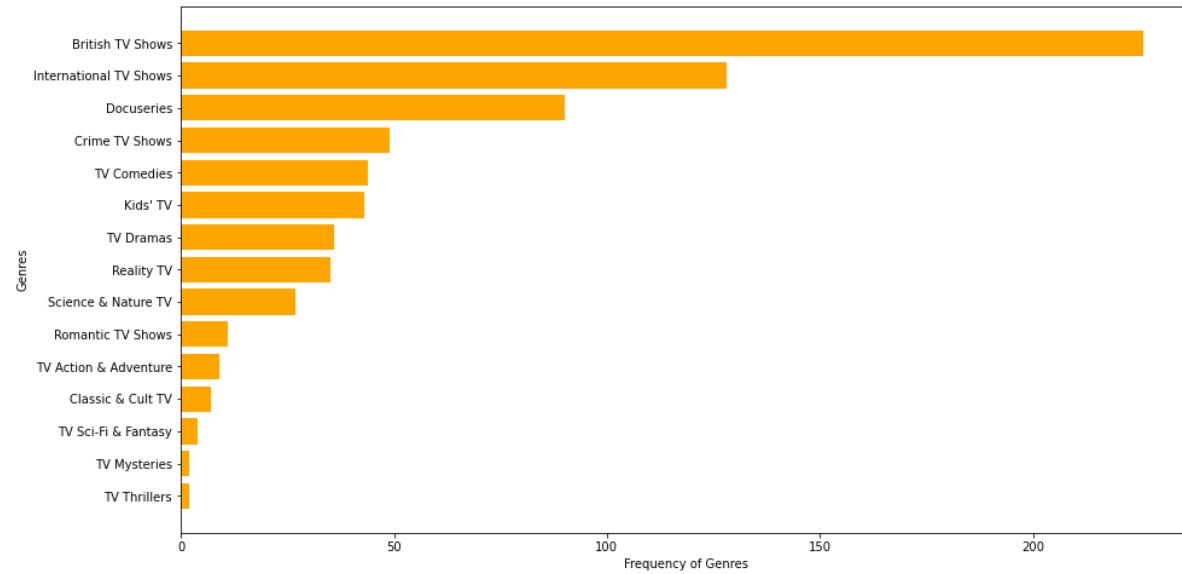
**Univariate Analysis separately for shows and movies in United Kingdom**

In [148]:

```
#Analyzing India for both shows and movies
df_uk_shows=df_final1[df_final1['country']=='United Kingdom'][df_final1[df_final1['country']=='United Kingdom'][df_final1['type']=='TV Show']]
df_uk_movies=df_final1[df_final1['country']=='United Kingdom'][df_final1[df_final1['type']=='Movie']]
```

In [149]:

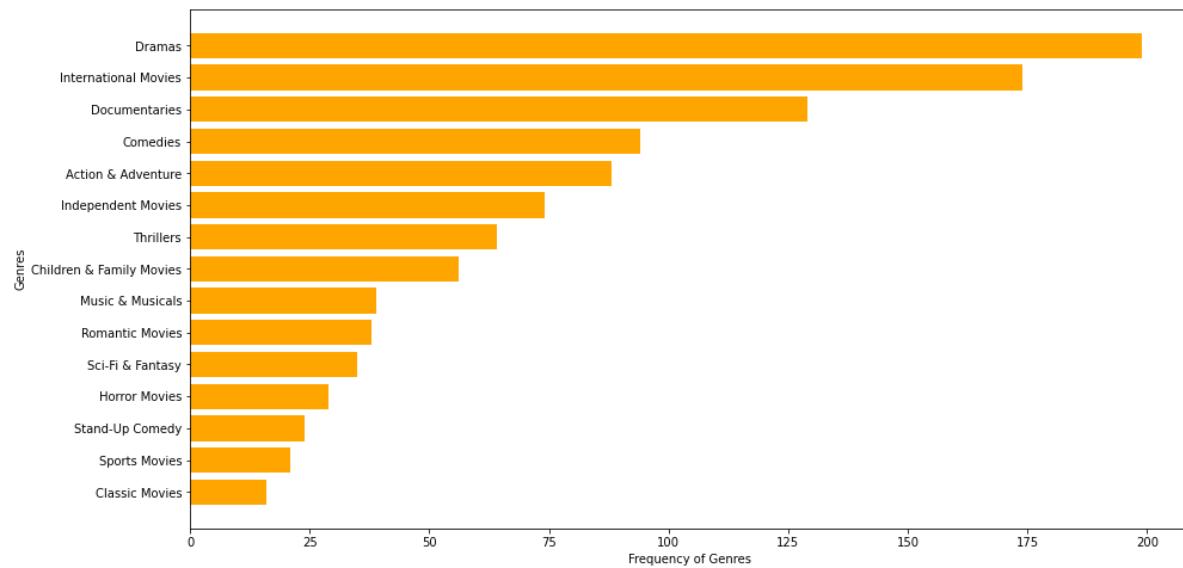
```
df_genre=df_uk_shows.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_
plt.figure(figsize=(15,8))
plt.barh(df_genre[:::-1]['Genre'], df_genre[:::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



British TV Shows, International TV Shows, Docuseries, Crime, Comedy are widely watched Genres in TV Shows in UK

In [150]:

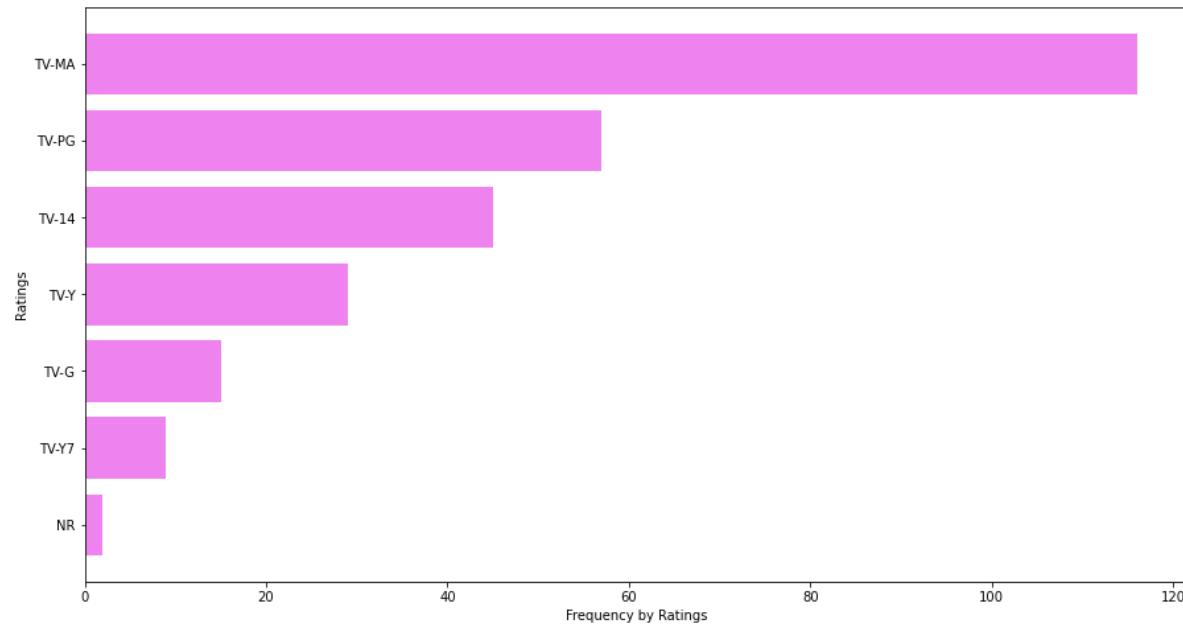
```
_uk_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values(by='nunique', ascending=False)
figsize=(15,8))
genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
'Frequency of Genres')
'Genres')
```



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

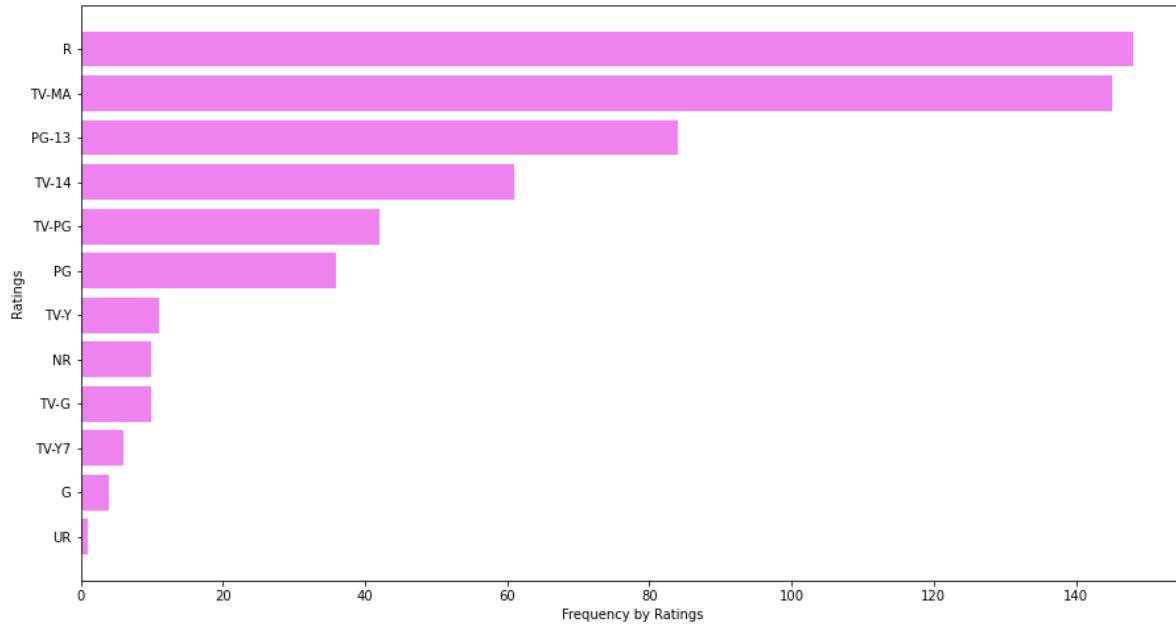
In [151]:

```
df_rating=df_uk_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values(by='nunique', ascending=False)
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 [152]:

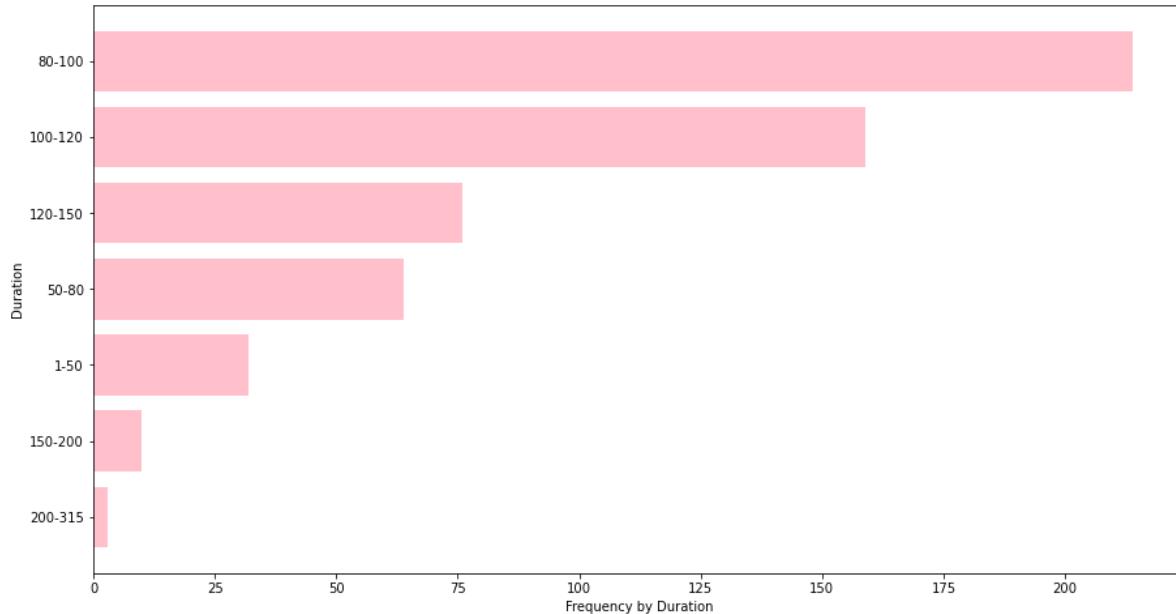
```
df_rating=df_uk_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_rating['rating'], df_rating['nunique'], color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



So it seems plausible to conclude that the popular ratings across Netflix includes Parental Guidance and Mature Audiences in TV Shows and R Rated+MA Rated in Movies in UK

In [153]:

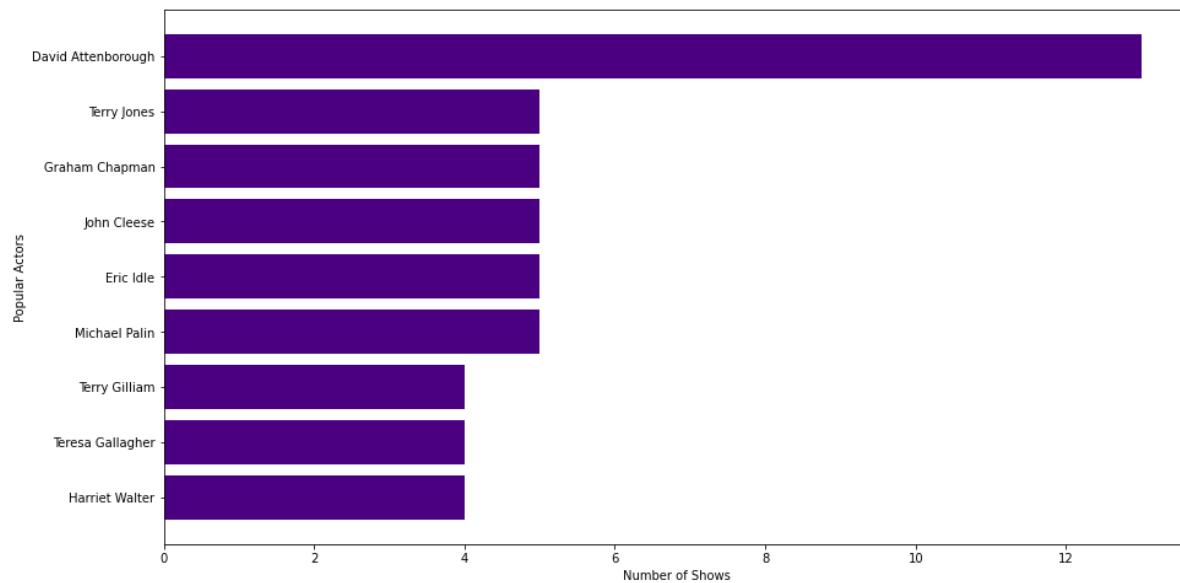
```
df_duration=df_uk_movies.groupby(['duration']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
plt.barh(df_duration[:::-1]['duration'], df_duration[:::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



Across movies ranges of minutes in UK have a sweet spot at 80-120 mins.

In [154]:

```
df_actors=df_uk_shows.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_values('title', ascending=False)
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 [155]:

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

Out[155]:

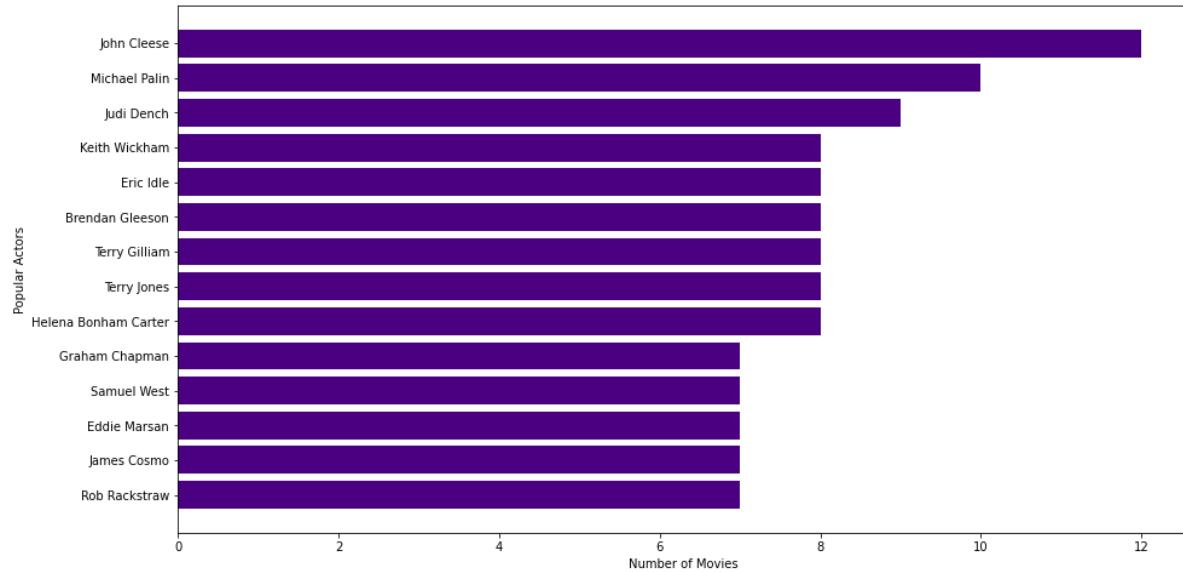
```
array(['David Attenborough', 'Terry Jones', 'Graham Chapman',
       'John Cleese', 'Eric Idle', 'Michael Palin', 'Terry Gilliam',
       'Teresa Gallagher', 'Harriet Walter'], dtype=object)
```

**Popular Actors in TV Shows in UK are:-**

'David Attenborough',  
 'Terry Jones',  
 'Graham Chapman',  
 'John Cleese',  
 'Eric Idle',  
 'Michael Palin',  
 'Terry Gilliam',  
 'Teresa Gallagher',  
 'Harriet Walter'

In [156]:

```
df_actors=df_uk_movies.groupby(['Actors']).agg({"title":"nunique"}).reset_index().sort_values("title", ascending=False)
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 [157]:

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

Out[157]:

```
array(['John Cleese', 'Michael Palin', 'Judi Dench', 'Keith Wickham',
       'Eric Idle', 'Brendan Gleeson', 'Terry Gilliam', 'Terry Jones',
       'Helena Bonham Carter', 'Graham Chapman', 'Samuel West',
       'Eddie Marsan', 'James Cosmo', 'Rob Rackstraw'], dtype=object)
```

**Popular actors across Movies in UK:-**

'John Cleese',  
 'Michael Palin',  
 'Judi Dench',  
 'Keith Wickham',  
 'Eric Idle',  
 'Brendan Gleeson',  
 'Terry Gilliam',  
 'Terry Jones',  
 'Helena Bonham Carter',  
 'Graham Chapman',  
 'Samuel West',  
 'Eddie Marsan',  
 'James Cosmo',  
 'Rob Rackstraw'

In [158]:

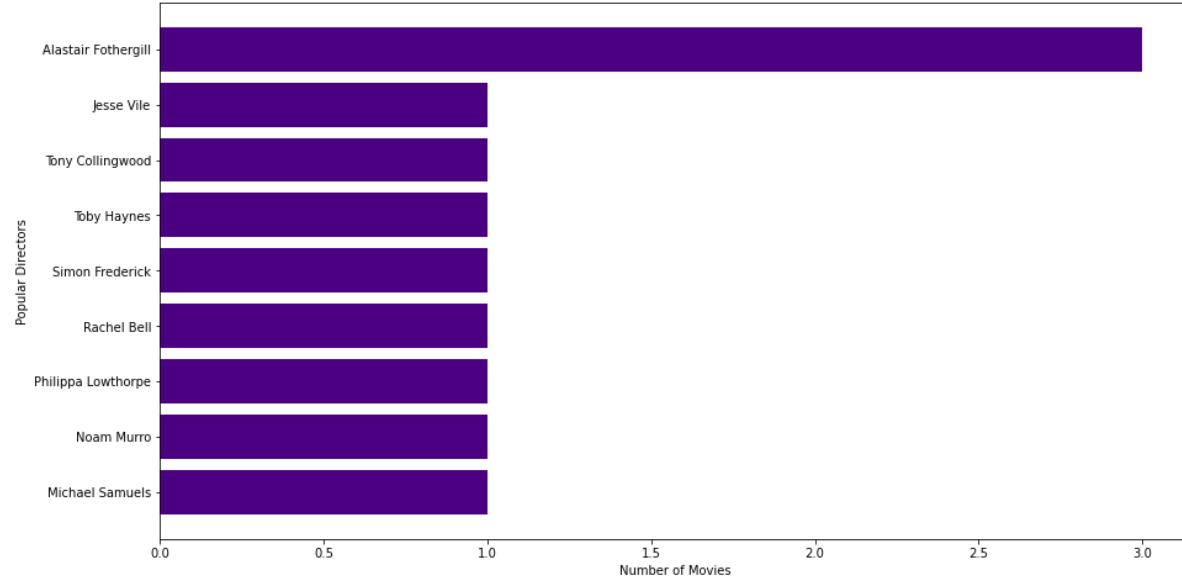
```
shows.groupby(['Directors']).agg({"title":"nunique"}).reset_index().sort_values(by=[Directors[df_directors['Directors']!='Unknown Director']]  

(15,8))  

rs[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])  

of Movies')  

Directors')
```



In [159]:

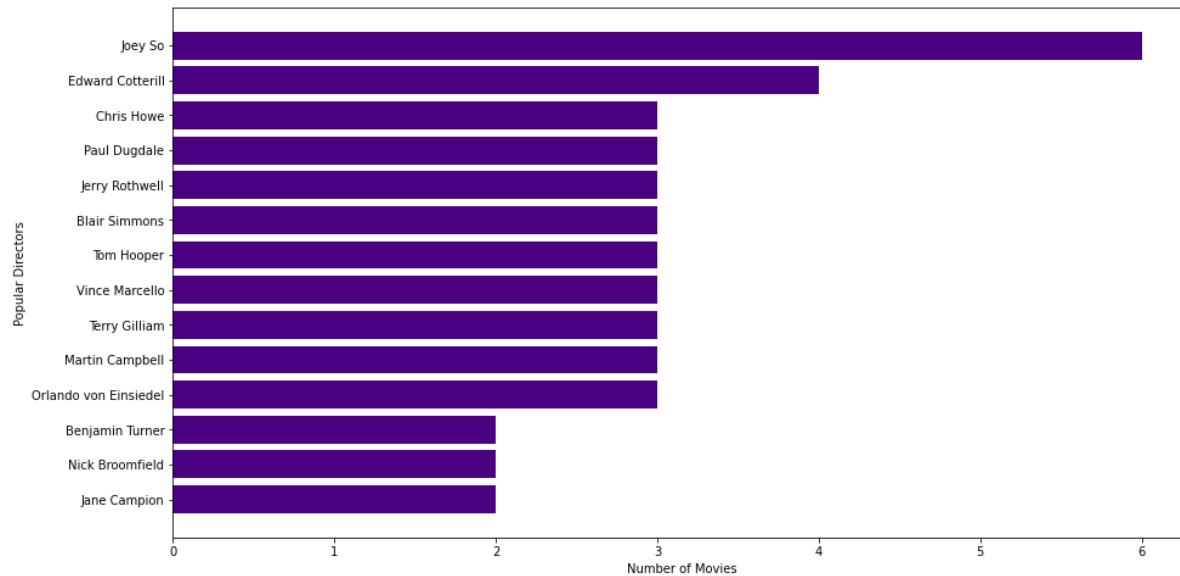
```
df_directors['Directors'].values
```

Out[159]:

```
array(['Alastair Fothergill', 'Jesse Vile', 'Tony Collingwood',
       'Toby Haynes', 'Simon Frederick', 'Rachel Bell',
       'Philippa Lowthorpe', 'Noam Murro', 'Michael Samuels'],
      dtype=object)
```

In [160]:

```
df_directors=df_uk_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 [161]:

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

Out[161]:

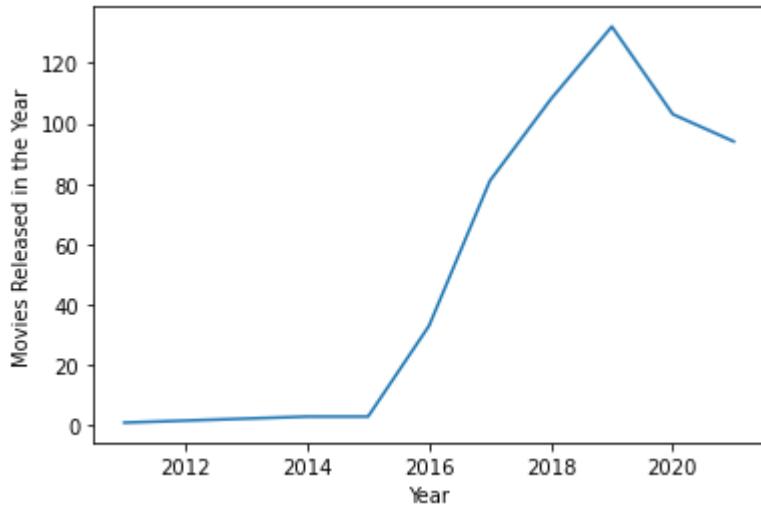
```
array(['Joey So', 'Edward Cotterill', 'Chris Howe', 'Paul Dugdale',
       'Jerry Rothwell', 'Blair Simmons', 'Tom Hooper', 'Vince Marcell
o',
       'Terry Gilliam', 'Martin Campbell', 'Orlando von Einsiedel',
       'Benjamin Turner', 'Nick Broomfield', 'Jane Campion'], dtype=ob
ject)
```

**Popular directors across movies in UK:-**

'Joey So',  
'Edward Cotterill'

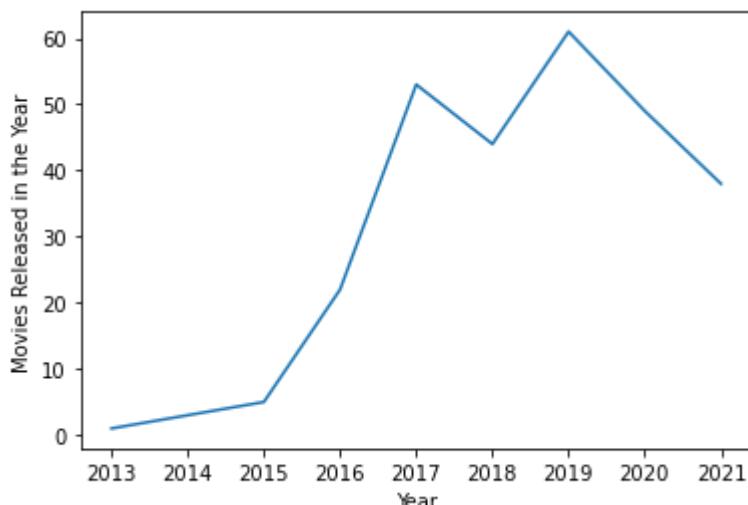
In [162]:

```
df_year=df_uk_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 [163]:

```
df_year=df_uk_shows.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()
```

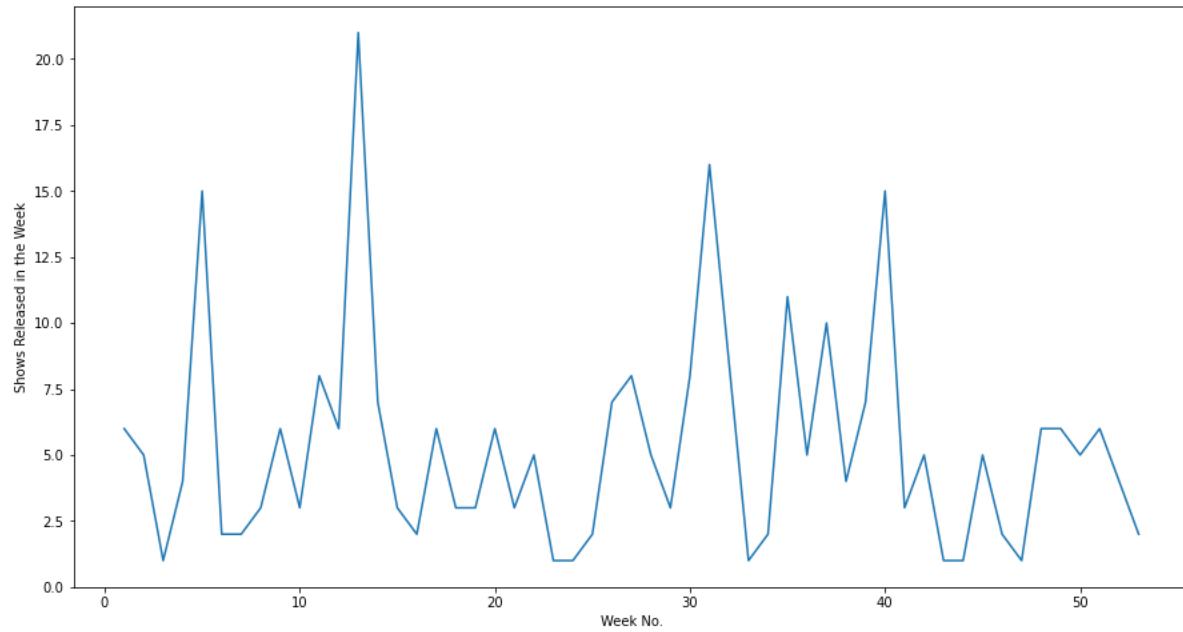


In terms of TV Shows, UK saw a downfall in 2018 from 2017, then a great increase in 2019 but has been reducing since then.

In terms of Movies, the number of popular movies in UK increased till 2019, since then it's decreasing.

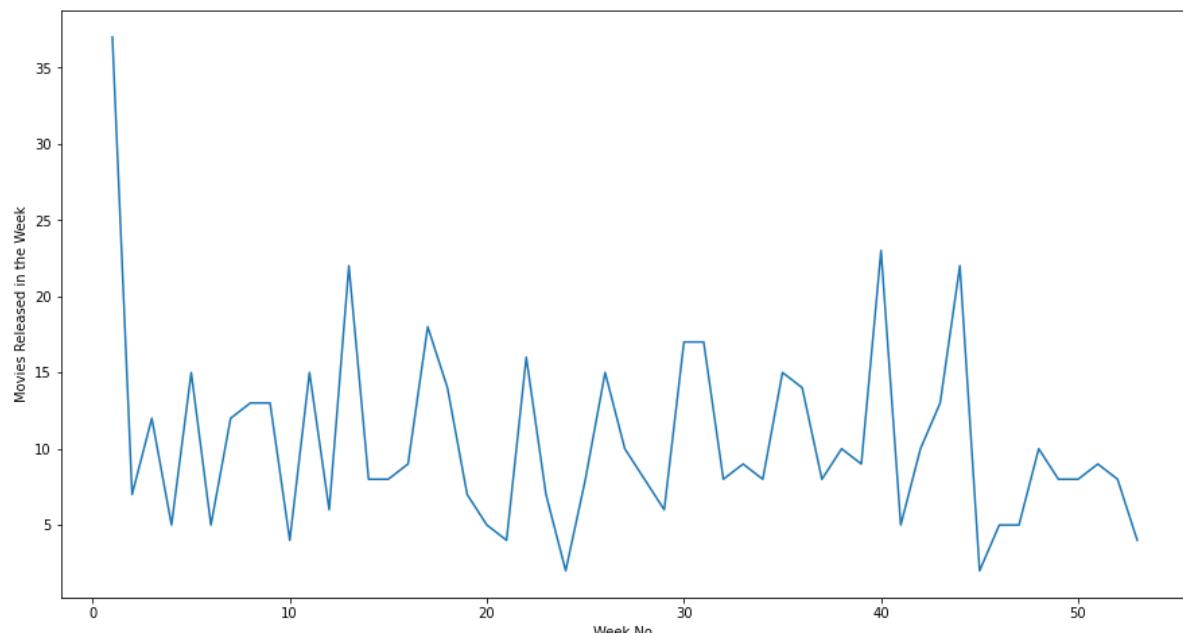
In [164]:

```
df_week=df_uk_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("Shows Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



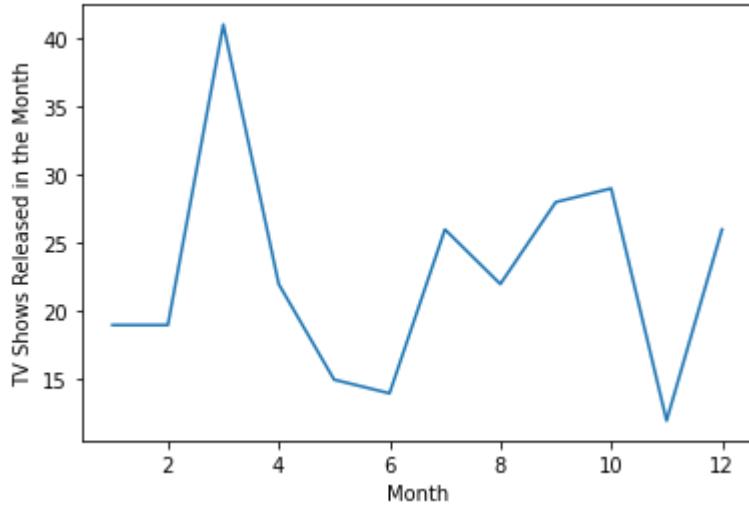
In [165]:

```
df_week=df_uk_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()
```



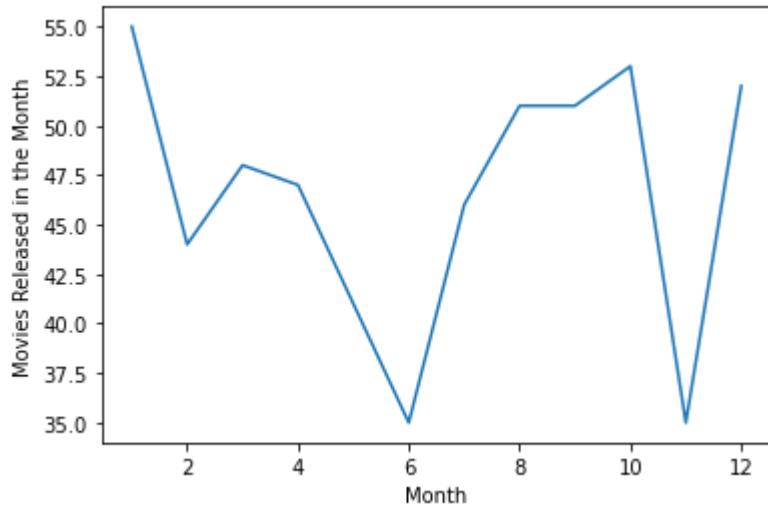
In [166]:

```
df_month=df_uk_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 [167]:

```
df_month=df_uk_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()
```

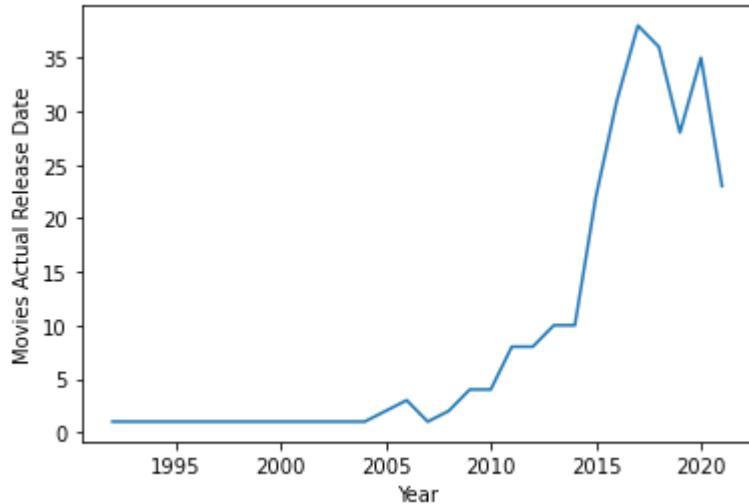


TV Shows are added in Netflix by a tremendous amount in March in UK

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

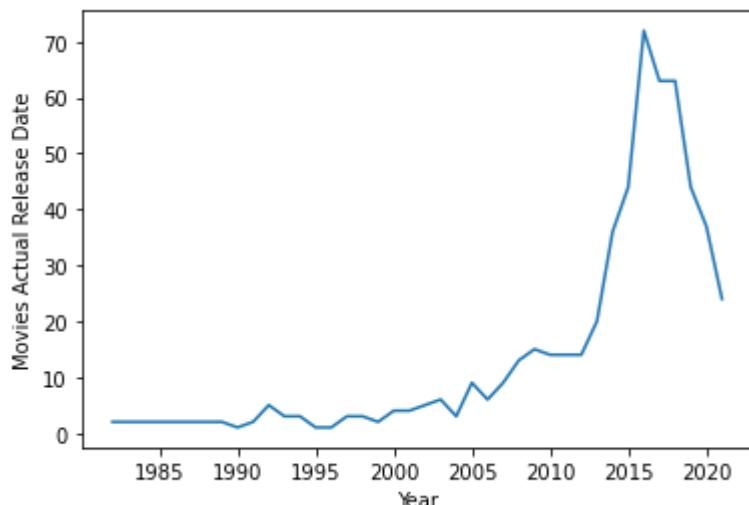
In [168]:

```
df_release_year=df_uk_shows[df_uk_shows['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 [169]:

```
df_release_year=df_uk_movies[df_uk_movies['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()
```



Same trend of reduction in movies and shows after 2020.

In [170]:

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

Out[170]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year
182	Sankofa	Kofi Ghanaba	Haile Gerima	Dramas	United Kingdom	s8	Movie	September 24, 2021	
188	Sankofa	Kofi Ghanaba	Haile Gerima	Independent Movies	United Kingdom	s8	Movie	September 24, 2021	
194	Sankofa	Kofi Ghanaba	Haile Gerima	International Movies	United Kingdom	s8	Movie	September 24, 2021	
200	Sankofa	Oyafunmike Ogunlano	Haile Gerima	Dramas	United Kingdom	s8	Movie	September 24, 2021	
206	Sankofa	Oyafunmike Ogunlano	Haile Gerima	Independent Movies	United Kingdom	s8	Movie	September 24, 2021	

In [171]:

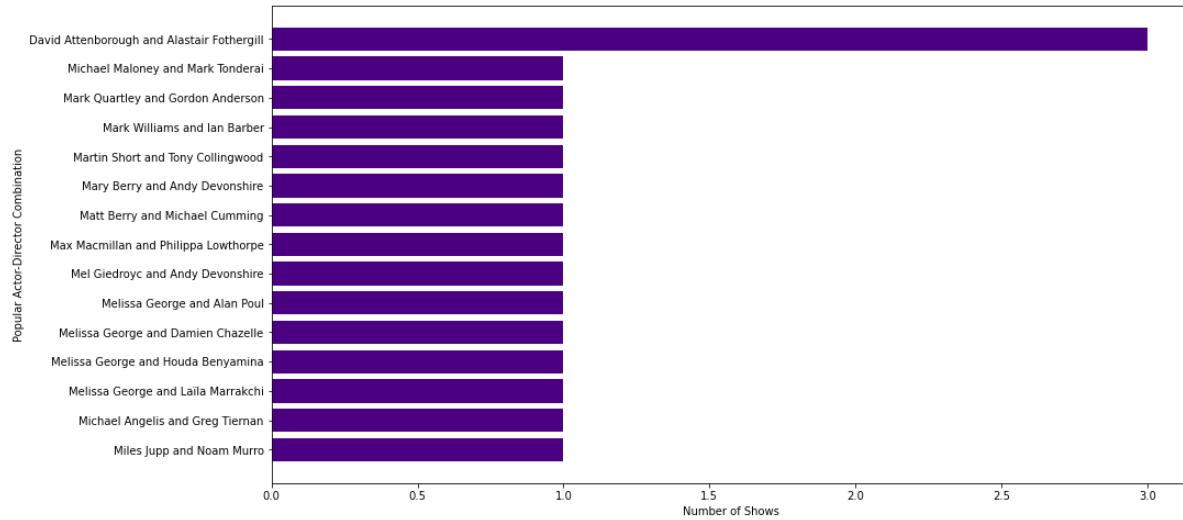
```
df_uk_shows['Actor_Director_Combination'] = df_uk_shows.Actors.str.cat(df_uk_shows.Directors)
df_uk_shows_subset=df_uk_shows[df_uk_shows['Actors']!='Unknown Actor']
df_uk_shows_subset=df_uk_shows_subset[df_uk_shows_subset['Directors']!='Unknown Director']
df_uk_shows_subset.head()
```

Out[171]:

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year
323	The Great British Baking Show	Mel Giedroyc	Andy Devonshire	British TV Shows	United Kingdom	s9	TV Show	September 24, 2021	2021
324	The Great British Baking Show	Mel Giedroyc	Andy Devonshire	Reality TV	United Kingdom	s9	TV Show	September 24, 2021	2021
325	The Great British Baking Show	Sue Perkins	Andy Devonshire	British TV Shows	United Kingdom	s9	TV Show	September 24, 2021	2021
326	The Great British Baking Show	Sue Perkins	Andy Devonshire	Reality TV	United Kingdom	s9	TV Show	September 24, 2021	2021
327	The Great British Baking Show	Mary Berry	Andy Devonshire	British TV Shows	United Kingdom	s9	TV Show	September 24, 2021	2021

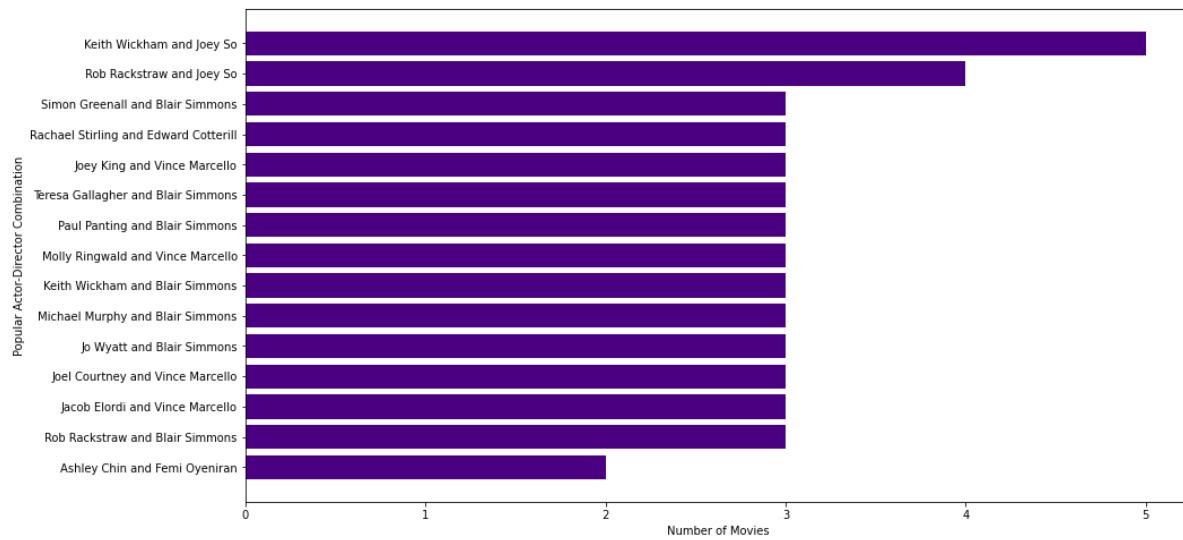
In [172]:

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



In [173]:

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



In [174]:

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

Out[174]:

```
array(['Keith Wickham and Joey So', 'Rob Rackstraw and Joey So',
       'Simon Greenall and Blair Simmons',
       'Rachael Stirling and Edward Cotterill',
       'Joey King and Vince Marcello',
       'Teresa Gallagher and Blair Simmons',
       'Paul Panting and Blair Simmons',
       'Molly Ringwald and Vince Marcello',
       'Keith Wickham and Blair Simmons',
       'Michael Murphy and Blair Simmons', 'Jo Wyatt and Blair Simmon
s',
       'Joel Courtney and Vince Marcello',
       'Jacob Flordi and Vince Marcello',
       'Rob Rackstraw and Blair Simmons', 'Ashley Chin and Femi Oyenir
an'],
      dtype=object)
```

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

'Keith Wickham and Joey So',  
 'Rob Rackstraw and Joey So'

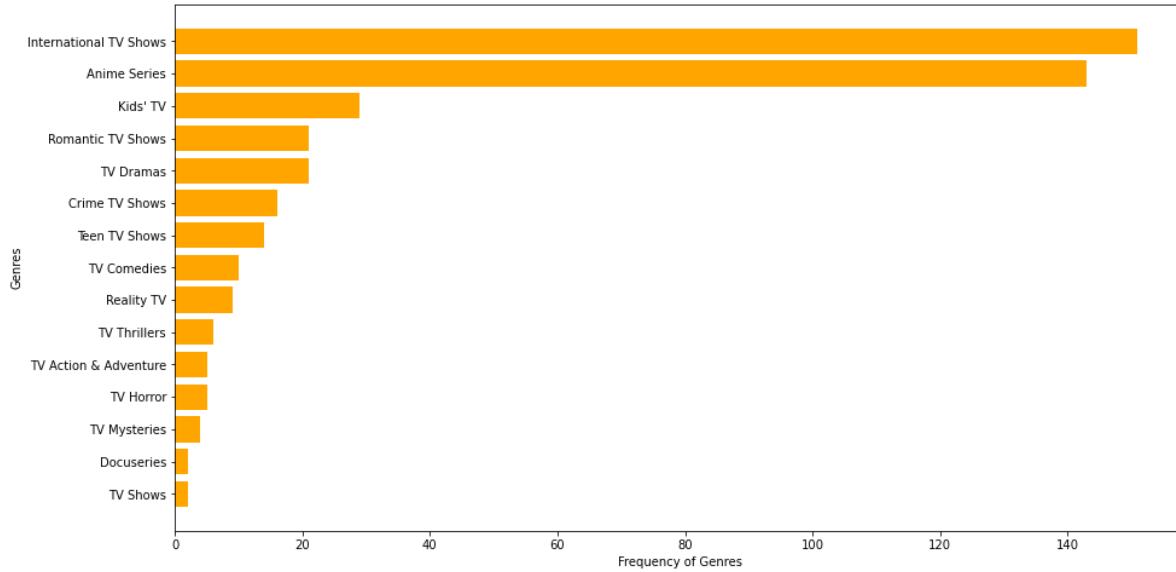
**Univariate Analysis separately for shows in Japan**

In [175]:

```
#Analyzing India for both shows and movies
df_japan_shows=df_final1[df_final1['country']=='Japan'][df_final1[df_final1['country']]
```

In [176]:

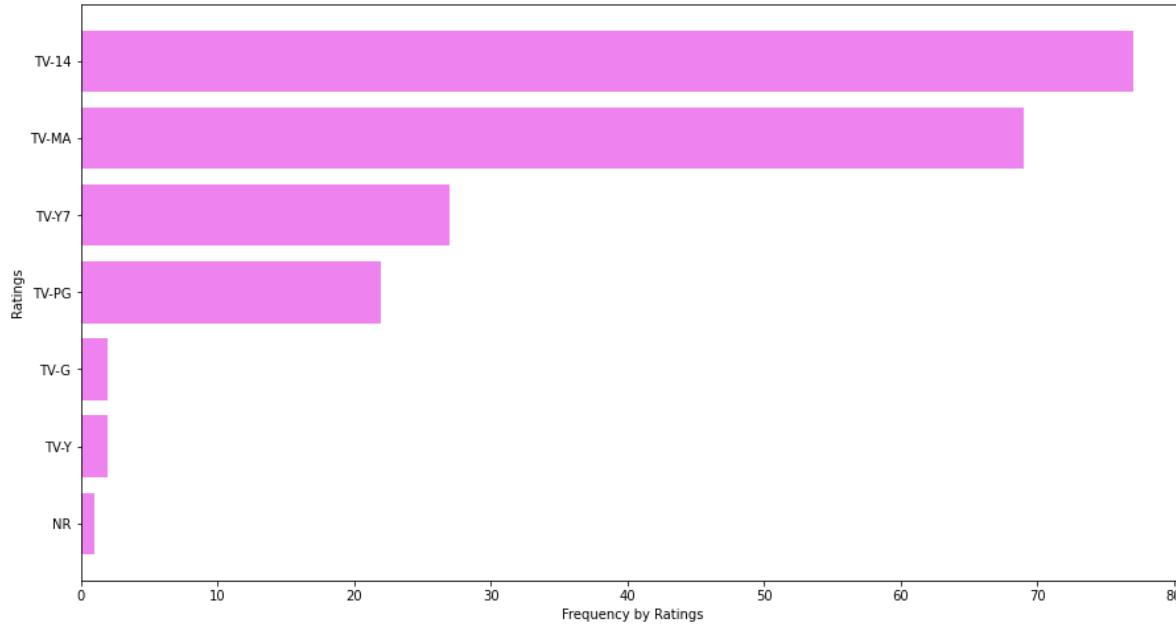
```
df_genre=df_japan_shows.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_genre[:::-1]['Genre'], df_genre[:::-1]['title'], color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



International TV Shows and Anime Genres are popular in TV Shows in Japan

In [177]:

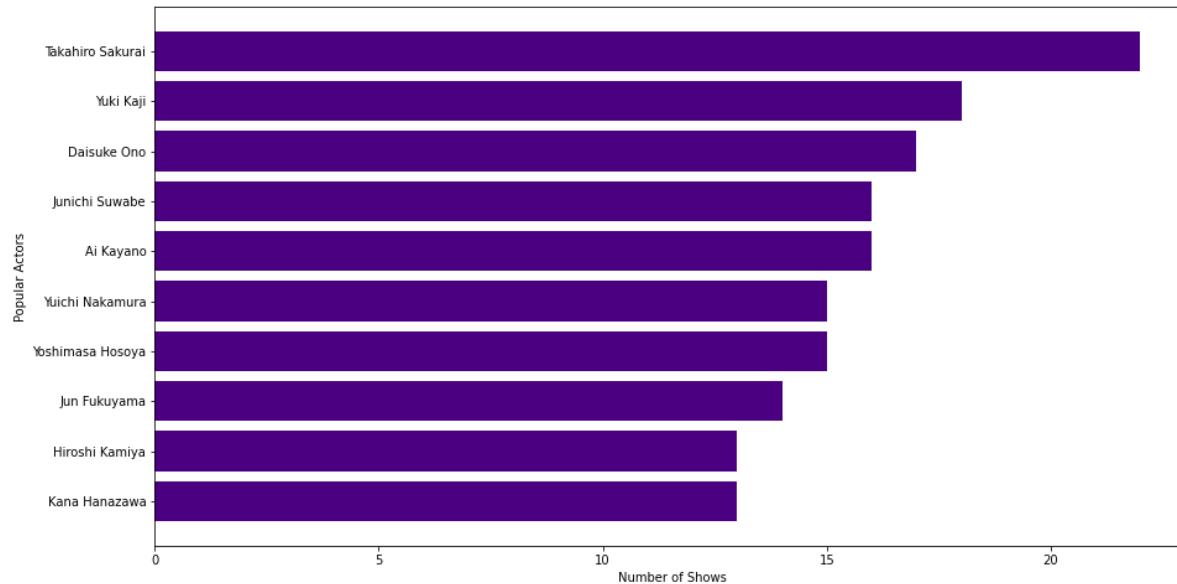
```
df_rating=df_japan_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_rating[:::-1]['rating'], df_rating[:::-1]['title'], color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()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



So it seems plausible to conclude that the popular ratings across Netflix includes TV-14 Mature Audiences in TV Shows

In [178]:

```
df_actors=df_japan_shows.groupby(['Actors']).agg({"title":"nunique"}).reset_index()
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 [179]:

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

Out[179]:

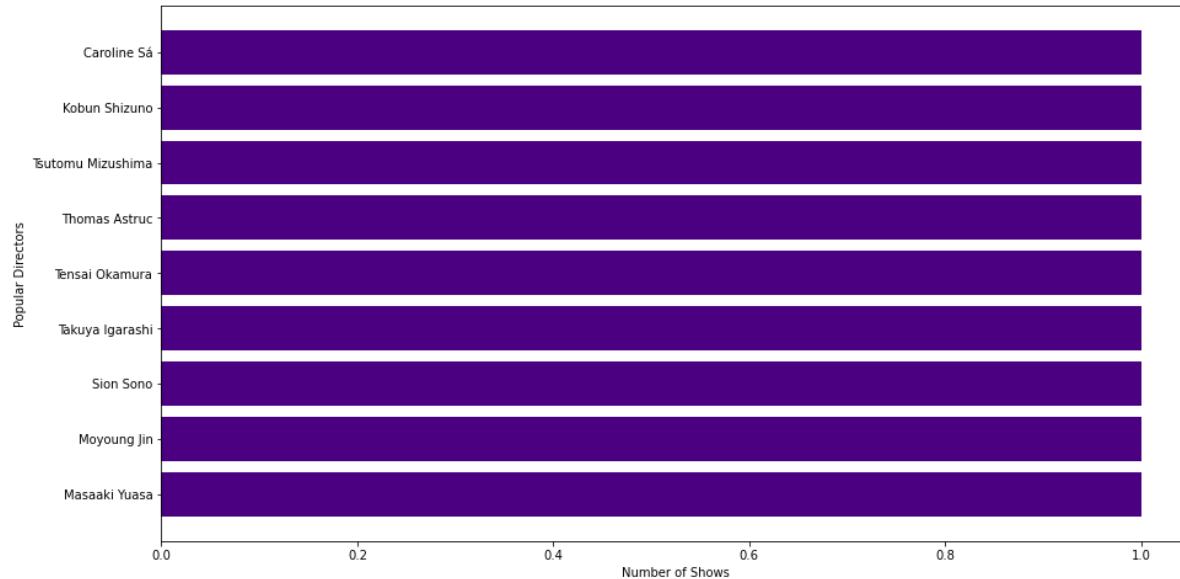
```
array(['Takahiro Sakurai', 'Yuki Kaji', 'Daisuke Ono', 'Junichi Suwabe',
       'Ai Kayano', 'Yuichi Nakamura', 'Yoshimasa Hosoya', 'Jun Fukuyama',
       'Hiroshi Kamiya', 'Kana Hanazawa'], dtype=object)
```

### Popular Actors in TV Shows in Japan are:-

- 'Takahiro Sakurai',
- 'Yuki Kaji',
- 'Daisuke Ono',
- 'Junichi Suwabe',
- 'Ai Kayano',
- 'Yuichi Nakamura',
- 'Yoshimasa Hosoya',
- 'Jun Fukuyama',
- 'Hiroshi Kamiya',
- 'Kana Hanazawa'

In [180]:

```
df_directors=df_japan_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 Shows')
plt.ylabel('Popular Directors')
plt.show()
```



In [181]:

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

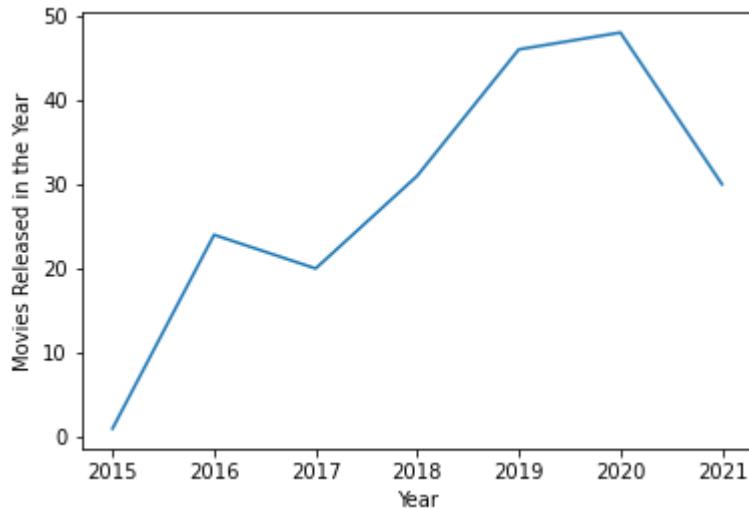
Out[181]:

```
array(['Caroline Sá', 'Kobun Shizuno', 'Tsutomu Mizushima',
       'Thomas Astruc', 'Tensai Okamura', 'Takuya Igarashi', 'Sion Son
o',
       'Moyoung Jin', 'Masaaki Yuasa'], dtype=object)
```

**All Directors are one time directors only**

In [182]:

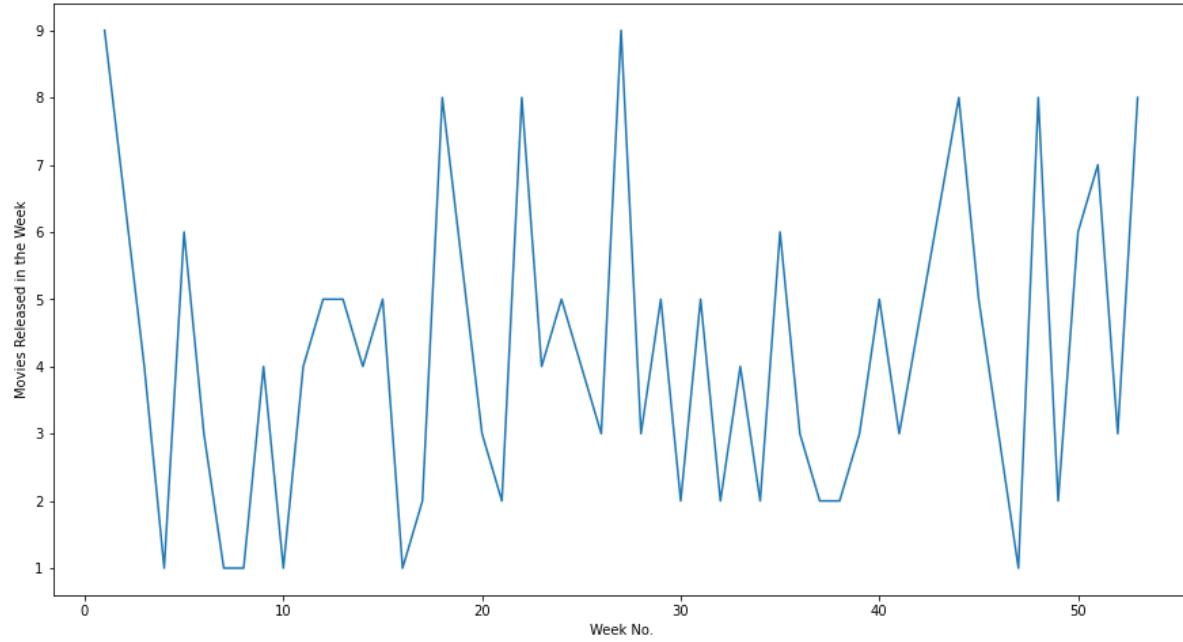
```
df_year=df_japan_shows.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



In Japan, TV Shows have diminished in 2017 from 2016 and then increased till 2020 after which it has reduced in 2021.

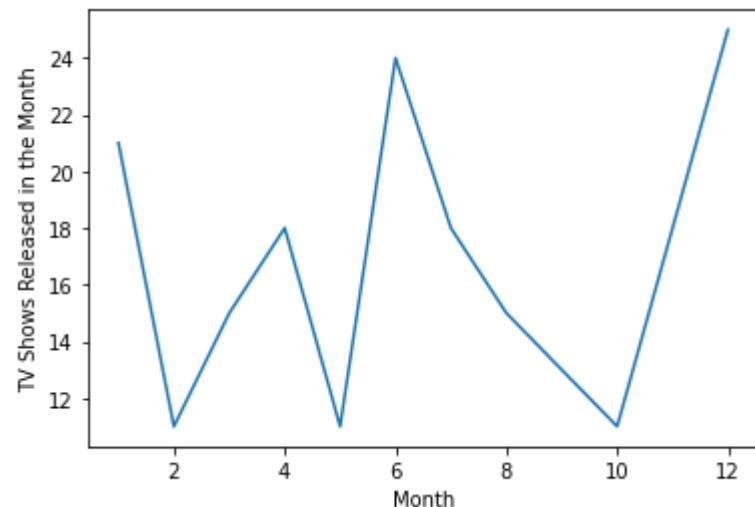
In [183]:

```
df_week=df_japan_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 [184]:

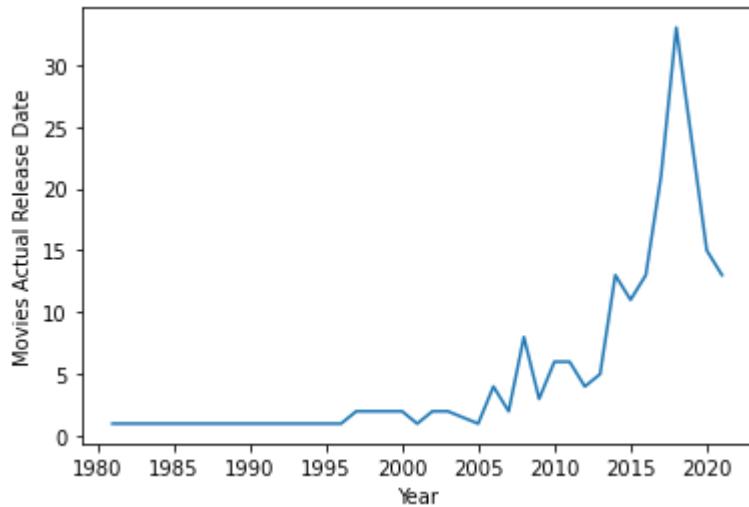
```
df_month=df_japan_shows.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("TV Shows Released in the Month")
plt.xlabel("Month")
plt.show()
```



TV Shows are added in Netflix by significant numbers in April and January in Japan

In [185]:

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



Reduction in TV Shows after 2019 in Japan

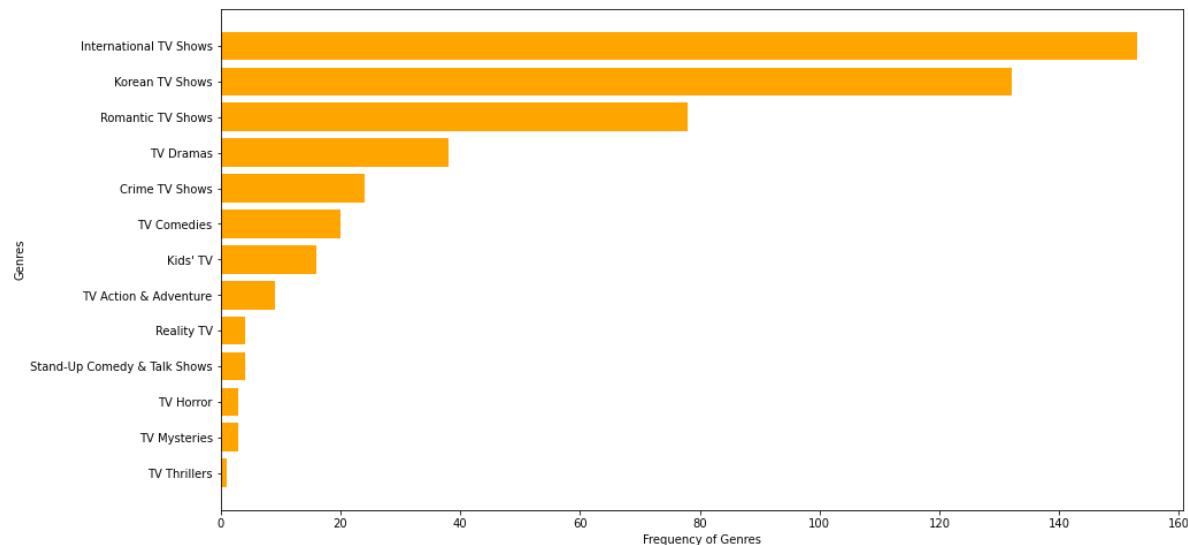
### Univariate Analysis separately for shows in South Korea

In [186]:

```
#Analyzing India for both shows and movies  
df_sk_shows=df_final1[df_final1['country']=='South Korea'][df_final1[df_final1['cour
```

In [187]:

```
df_genre=df_sk_shows.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()
```

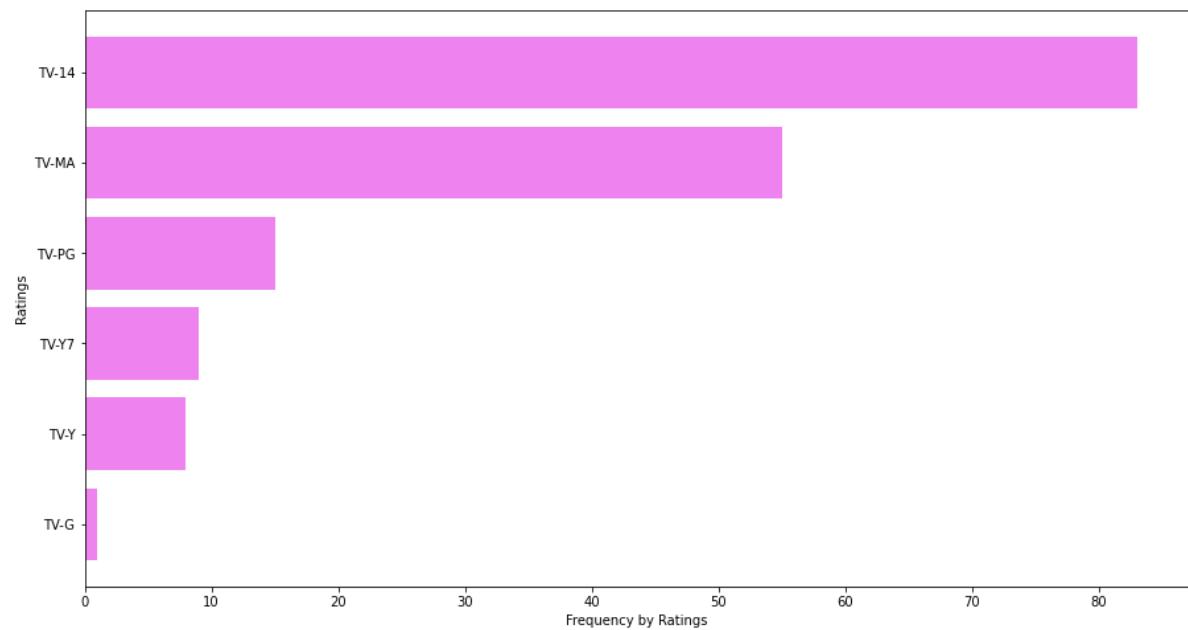


International TV Shows, Romantic TV Shows, Drama, Crime and Comedy Genres are popular in TV Shows in S.Korea.

Only S.Korea has Romance as a top 3 favorable genre which depicts an inclination of their audience

In [188]:

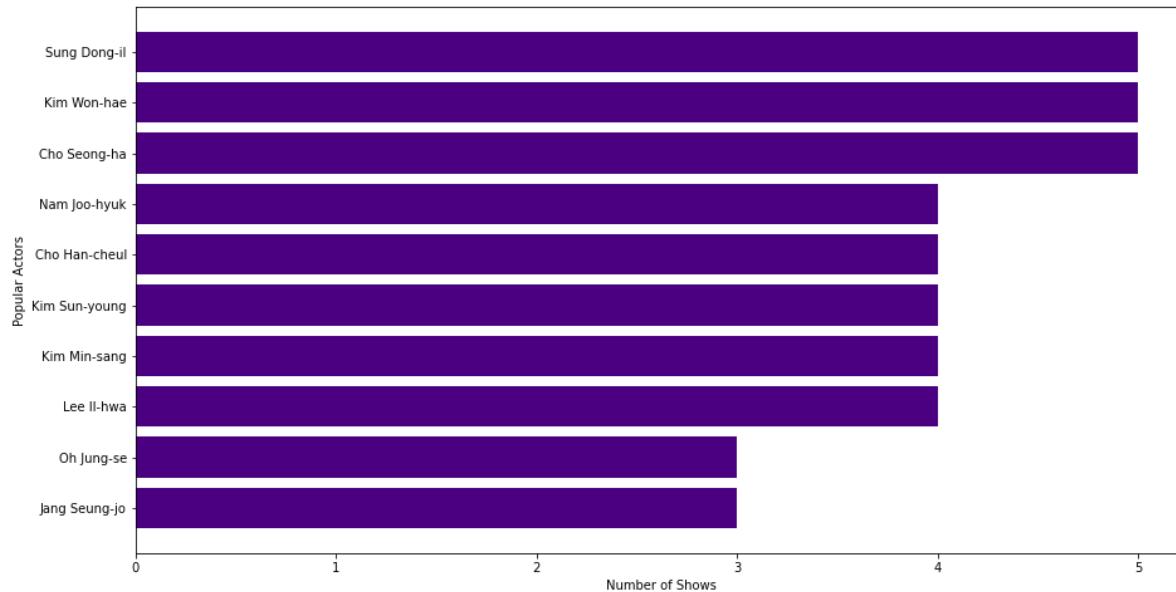
```
df_rating=df_sk_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_
plt.figure(figsize=(15,8))
plt.barh(df_rating[:::-1]['rating'], df_rating[:::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



So it seems plausible to conclude that the popular ratings across Netflix includes TV-14 and Mature Audiences in TV Shows

In [189]:

```
shows.groupby(['Actors']).agg({'title':'nunique'}).reset_index().sort_values(by=['tit
ors[df_actors['Actors']!='Unknown Actor']
se=(15,8))
's[:::-1]['Actors'], df_actors[::-1]['title'],color=['indigo'])
r of Shows')
ar Actors')
```



In [190]:

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

Out[190]:

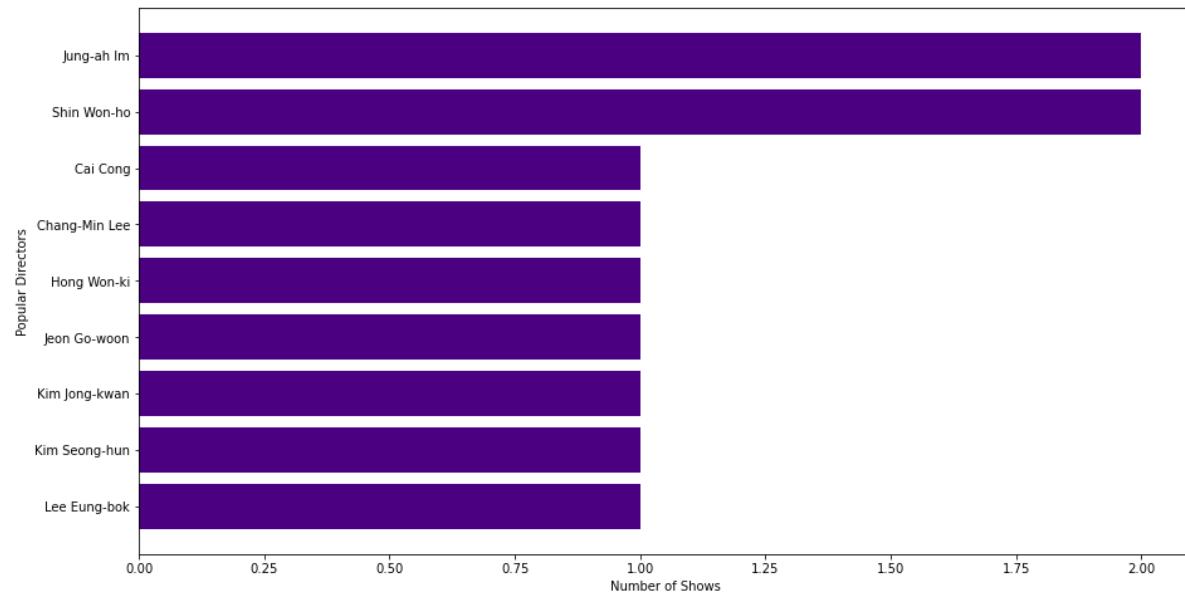
```
array(['Sung Dong-il', 'Kim Won-hae', 'Cho Seong-ha', 'Nam Joo-hyuk',
       'Cho Han-cheul', 'Kim Sun-young', 'Kim Min-sang', 'Lee Il-hwa',
       'Oh Jung-se', 'Jang Seung-jo'], dtype=object)
```

**Popular Actors in TV Shows in South Korea are:-**

- 'Sung Dong-il',
- 'Kim Won-hae',
- 'Cho Seong-ha',
- 'Nam Joo-hyuk'

In [191]:

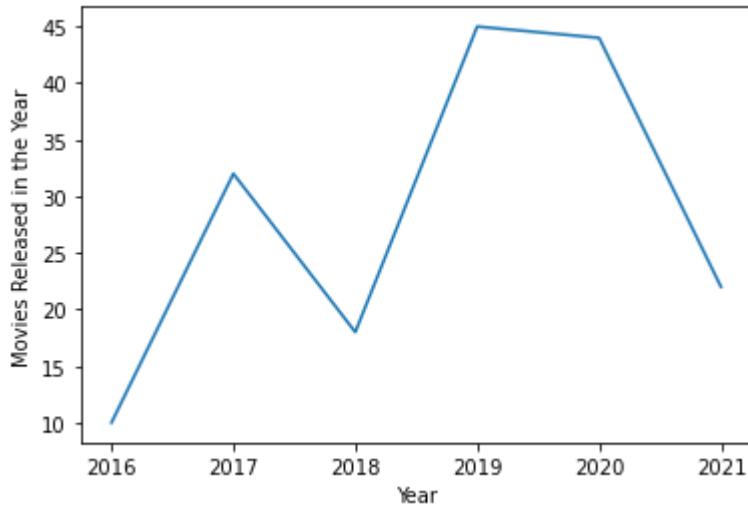
```
df_directors=df_sk_shows.groupby(['Directors']).agg({"title":"nunique"}).reset_index()
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[:::-1]['Directors'], df_directors[:::-1]['title'],color=['indigo'])
plt.xlabel('Number of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



**Two directors have directed 2 shows and rest all Directors are one time directors only**

In [192]:

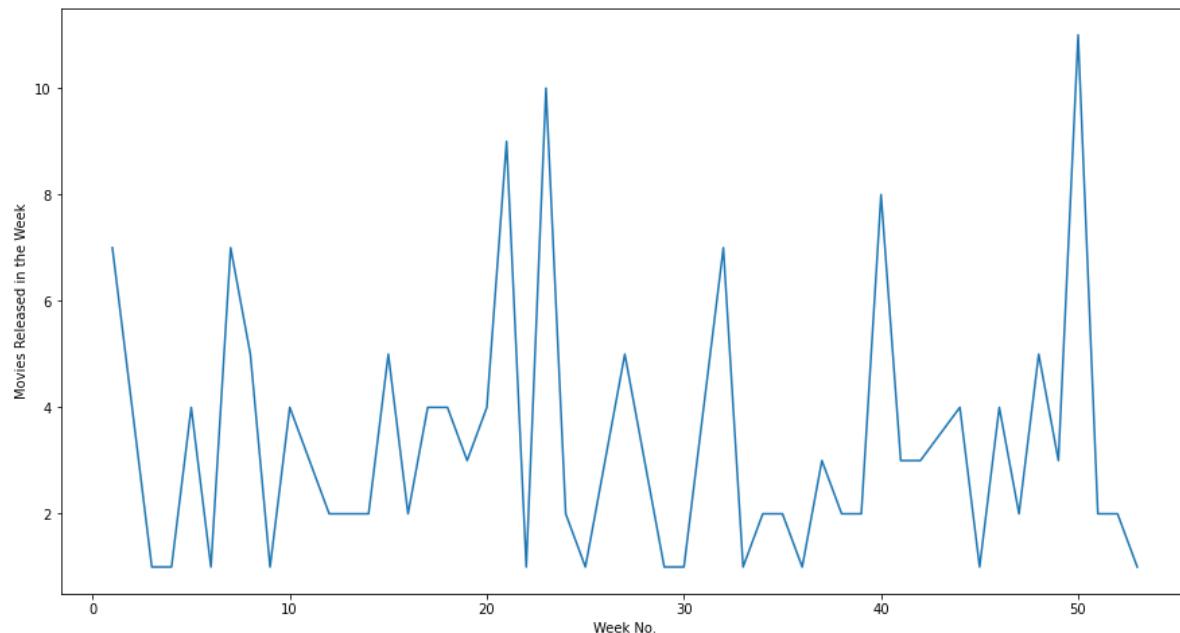
```
df_year=df_sk_shows.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



In South Korea, number of TV Shows reduced in 2018 from 2017, then increased till 2019 but have been on a heavy downfall since then

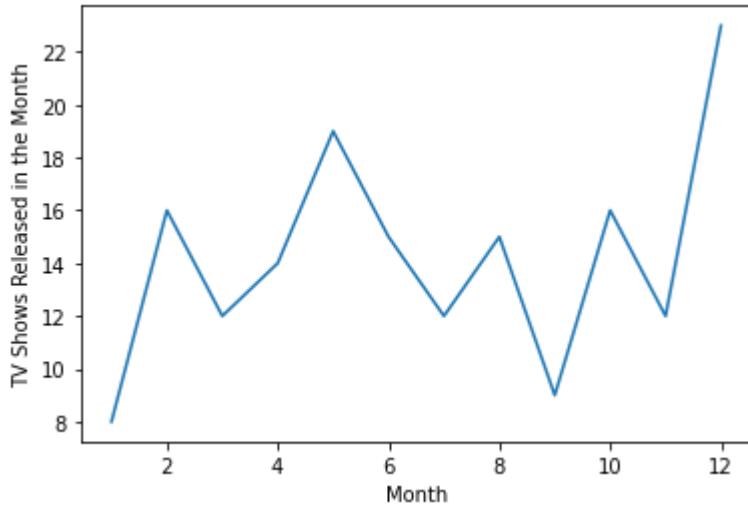
In [193]:

```
df_week=df_sk_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 [194]:

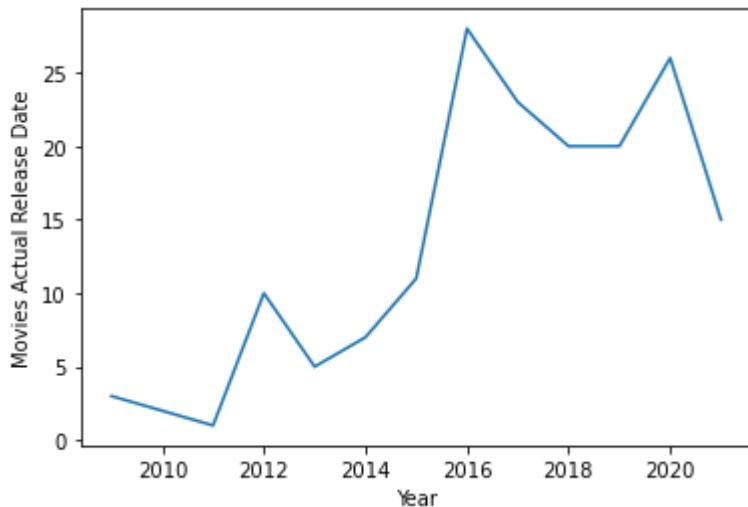
```
df_month=df_sk_shows.groupby(['month_added']).agg({'title':'nunique'}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("TV Shows Released in the Month")
plt.xlabel("Month")
plt.show()
```



TV Shows are added in Netflix by significant numbers in May and January in South Korea

In [195]:

```
df_release_year=df_sk_shows[df_sk_shows['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()
```



The number of TV Shows in S.Korea reached peak in 2016. It then reached a second peak in 2019. It has reduced in 2021 from 2020.

## 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) For UK audience, recommended length for movies is same as that of USA (80-120 mins)
- 5) The target audience in USA and India is recommended to be 14+ and above ratings while for UK, its recommended to be completely Mature/R content .
- 6) Add movies for Indian Audience, it has been declining since 2018.
- 7) Anime Genre for Japan and Romantic Genre in TV Shows for South Korean audiences is recommended.
- 8) 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 [ ]: