

```
In [2]: #importing libraries for our purpose
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: #Uploading Data
```

```
In [4]: df = pd.read_csv(r'C:\Users\Lenovo\Desktop\Scaler\netflix.csv')
```

```
In [5]: # head and tail of data
```

```
In [6]: df.head()
```

```
Out[6]: show_id type title director cast country date_added release_year rating duration
```

0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	Seas
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Seas
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Seas
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	Seas



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

Out[7]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana...	India	March 2, 2019	2015	TV-14



In [8]:

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

Out[8]: 8807

In [9]:

```
#checking datatypes
```

In [10]:

```
df.dtypes
```

Out[10]:

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 [11]:

```
#number of unique values in our data
```

In [12]: `df.nunique()`

```
Out[12]: show_id      8807
type          2
title         8807
director     4528
cast          7692
country       748
date_added   1767
release_year  74
rating         17
duration      220
listed_in     514
description    8775
dtype: int64
```

In [13]: `# Other ways to find 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 [14]: `# checking null values in every column of our data`
`df.isnull().sum()`

```
Out[14]: show_id      0
type          0
title         0
director     2634
cast          825
country       831
date_added   10
release_year  0
rating         4
duration      3
listed_in     0
description    0
dtype: int64
```

In [15]: `#Checking null values in every columns in data % wise`

In [16]: `print(round((df.isnull().sum()/len(df)*100),2))`

```
show_id      0.00
type         0.00
title        0.00
director    29.91
cast          9.37
country      9.44
```

```
date_added      0.11
release_year    0.00
rating          0.05
duration        0.03
listed_in       0.00
description     0.00
dtype: float64
```

In [17]: *# checking the occurrences of each of the ratings*

In [18]: `df['rating'].value_counts()`

Out[18]:

Rating	Count
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
UR	3
NC-17	3
74 min	1
84 min	1
66 min	1

Name: rating, dtype: int64

In [19]: *#unnesting the directors column, i.e- creating separate lines for each director*

In [20]:

```
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[20]:

	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 [21]: *#unnesting the cast column, i.e- creating separate lines for each cast member*

In [22]:

```
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[22]:

	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 [23]:

```
#unnesting the Listed_in column, i.e- creating separate Lines for each genre
```

In [24]:

```
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=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[24]:

	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 [25]:

```
#unnesting the country column, i.e- creating separate Lines for each country
```

In [26]:

```
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=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[26]:

	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 [27]: #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')
```

```
In [28]: #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[28]:

	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 [29]: #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[29]:

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

```
In [30]: # CHECKING NULL VALUES IN ABOVE DATA
```

```
In [31]: df_final.isnull().sum()
```

Out[31]:

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 [32]: `df_final[pd.isnull(df_final['duration'])]`

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

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

In [34]: `df_final.loc[126537]`

```
title           Louis C.K. 2017
Actors          Louis C.K.
Directors       Louis C.K.
Genre            Movies
Country         United States
show_id         s5542
type            Movie
date_added     April 4, 2017
release_year    2017
rating          74 min
duration        74 min
Name: 126537, dtype: object
```

In [35]: `df_final.loc[df_final['rating'].str.contains('min', na=False), 'rating']`

```
126537    74 min
131603    84 min
131737    66 min
Name: rating, dtype: object
```

In [36]: `df_final.loc[df_final['rating'].str.contains('min', na=False), 'rating'] = 'NR'`

In [37]: `df_final.isnull().sum()`

```
Out[37]: 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 [38]: # Analysing null values in date_added columns
```

```
In [39]: df_final[df_final['date_added'].isnull()].head()
```

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year
136893	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	NaN	201
136894	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	TV Comedies	United Kingdom	s6067	TV Show	NaN	201
136895	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	TV Dramas	United Kingdom	s6067	TV Show	NaN	201
136896	A Young Doctor's Notebook and Other Stories	Jon Hamm	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	NaN	201
136897	A Young Doctor's Notebook and Other Stories	Jon Hamm	Unknown Director	TV Comedies	United Kingdom	s6067	TV Show	NaN	201

```
In [40]: #date added column is imputed on the basis of release year, i.e- suppose there's a null value in date added column, then we can check for the mode of release year and impute it in place of nulls
#when release year was 2013. So below piece of code just checks the mode of date added column 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['relea
```

```
In [41]: df_final[df_final['date_added'].isnull()].head()
```

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating	duration
--	-------	--------	-----------	-------	---------	---------	------	------------	--------------	--------	----------



In [42]: `# Analysing null values in country columns`

In [43]: `df_final[df_final['Country'].isnull()].head()`

Out[43]:

		title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year
58	Ganglands	Sami Bouajila	Julien Leclercq	Crime TV Shows	NaN	s3	TV Show	September 24, 2021	2021	
59	Ganglands	Sami Bouajila	Julien Leclercq	International TV Shows	NaN	s3	TV Show	September 24, 2021	2021	
60	Ganglands	Sami Bouajila	Julien Leclercq	TV Action & Adventure	NaN	s3	TV Show	September 24, 2021	2021	
61	Ganglands	Tracy Gotoas	Julien Leclercq	Crime TV Shows	NaN	s3	TV Show	September 24, 2021	2021	
62	Ganglands	Tracy Gotoas	Julien Leclercq	International TV Shows	NaN	s3	TV Show	September 24, 2021	2021	



In [44]: `#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`

In [45]: `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['Directo`

In [46]: `df_final[df_final['Country'].isnull()].head()`

Out[46]:

		title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	Children & Family Movies	NaN	s7	Movie	September 24, 2021	2021	
160	My Little Pony: A New Generation	Vanessa Hudgens	José Luis Ucha	Children & Family Movies	NaN	s7	Movie	September 24, 2021	2021	
161	My Little Pony: A New Generation	Kimiko Glenn	Robert Cullen	Children & Family Movies	NaN	s7	Movie	September 24, 2021	2021	
162	My Little Pony: A New Generation	Kimiko Glenn	José Luis Ucha	Children & Family Movies	NaN	s7	Movie	September 24, 2021	2021	

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year
163	My Little Pony: A New Generation	James Marsden	Robert Cullen	Children & Family Movies	NaN	s7	Movie	September 24, 2021	2021

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 [57]: 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,'Country']
    #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[57]: title      0
Actors      0
Directors   0
Genre       0
Country     0
show_id     0
type        0
date_added  0
release_year 0
rating      67
duration    0
duration_copy 0
dtype: int64
```

```
In [48]: df_final.head()
```

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

```
In [49]: df_final['duration'].value_counts()
```

```
Out[49]: 1 Season      35035
2 Seasons     9559
3 Seasons      5084
94 min        4343
106 min       4040
...
196 min         4
5 min          3
9 min          2
11 min         2
8 min          2
Name: duration, Length: 220, dtype: int64
```

```
In [50]: #removing mins from data
df_final['duration']=df_final['duration'].str.replace(" min","");
df_final.head()
```

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

```
In [51]: df_final['duration'].unique()
```

```
Out[51]: array(['90', '2 Seasons', '1 Season', '91', '125', '9 Seasons', '104',
   '127', '4 Seasons', '67', '94', '5 Seasons', '161', '61', '166',
   '147', '103', '97', '106', '111', '3 Seasons', '110', '105', '96',
   '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', '114',
   '118', '108', '63', '121', '142', '154', '120', '82', '109', '101',
   '86', '229', '76', '89', '156', '112', '107', '129', '135', '136',
   '165', '150', '133', '70', '84', '140', '78', '7 Seasons', '64',
   '59', '139', '69', '148', '189', '141', '130', '138', '81', '132',
   '10 Seasons', '123', '65', '68', '66', '62', '74', '131', '39',
   '46', '38', '8 Seasons', '17 Seasons', '126', '155', '159', '137',
   '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', '173',
   '181', '185', '21', '24', '51', '151', '42', '22', '134', '177',
   '13 Seasons', '52', '14', '53', '8', '57', '28', '50', '9', '26',
   '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', '192',
   '209', '187', '172', '16', '186', '11', '193', '176', '56', '169',
   '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 [52]: df_final['duration_copy']=df_final['duration'].copy()
```

```
df_final1=df_final.copy()
```

In [53]:

```
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[53]:

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

In [54]:

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

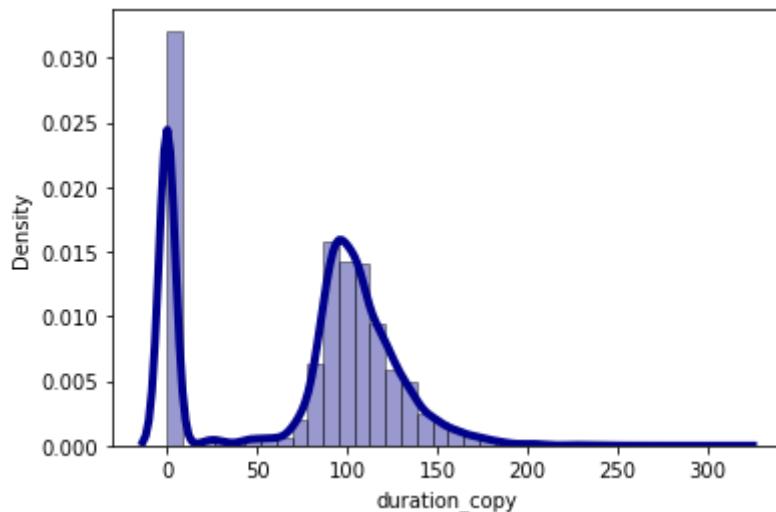
Out[54]:

```
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 [55]:

```
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()
```

C:\Users\Lenovo\anaconda3\lib\site-packages\seaborn\distributions.py:2557: 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 [56]:

```
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[56]:

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



In [58]:

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

Out[58]:

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



In [62]: `df_final1['duration'].value_counts()`

Out[62]:

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
11 Seasons	30
17 Seasons	30

Name: duration, dtype: int64

In [63]:

```
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()
```

<ipython-input-63-2755cd09232c>: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[63]:

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



```
In [64]: df_final1['title']=df_final1['title'].str.replace(r"\(.*\)", "")  
df_final1.head()
```

<ipython-input-64-b3cc70e1b430>:1: 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[64]:

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



```
In [65]: # Univariate Analysis in terms of counts of each column
```

```
In [66]: #number of distinct titles on the basis of genre
```

```
In [68]: df_final1.groupby(['Genre']).agg({'title':'nunique'}).sort_values(by=['title'], ascending=False)
```

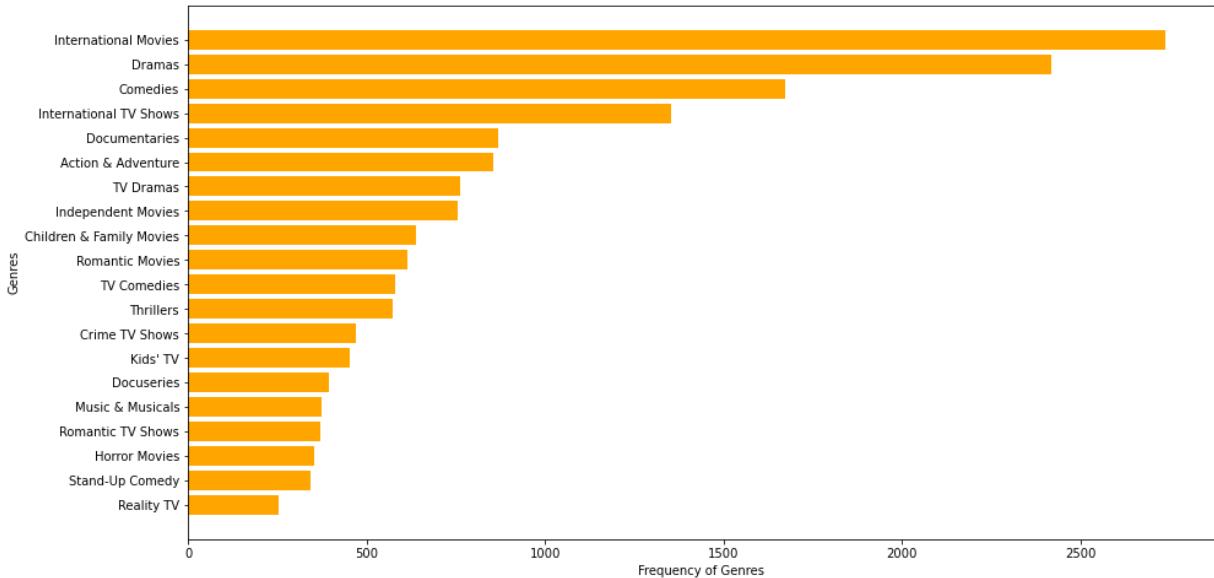
Out[68]:

Genre	title
International Movies	2738
Dramas	2418
Comedies	1673
International TV Shows	1351
Documentaries	869
Action & Adventure	854
TV Dramas	763
Independent Movies	756
Children & Family Movies	639
Romantic Movies	615
TV Comedies	581
Thrillers	573
Crime TV Shows	470
Kids' TV	451

title	
Genre	
Docuseries	395
Music & Musicals	372
Romantic TV Shows	370
Horror Movies	353
Stand-Up Comedy	343
Reality TV	255
British TV Shows	253
Sci-Fi & Fantasy	243
Sports Movies	219
Anime Series	176
Spanish-Language TV Shows	174
TV Action & Adventure	168
Korean TV Shows	151
Classic Movies	116
LGBTQ Movies	102
TV Mysteries	98
Science & Nature TV	92
TV Sci-Fi & Fantasy	84
TV Horror	75
Anime Features	71
Cult Movies	71
Teen TV Shows	69
Faith & Spirituality	65
TV Thrillers	57
Movies	57
Stand-Up Comedy & Talk Shows	56
Classic & Cult TV	28
TV Shows	16

In [87]:

```
df_genre=df_final1.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()
```

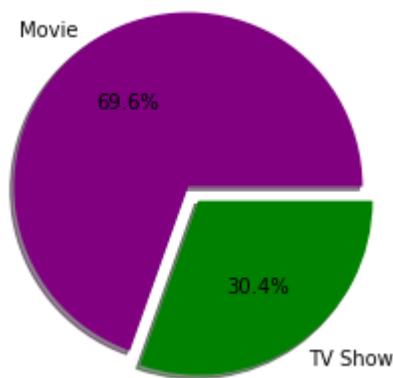


In [73]: *#number of distinct titles on the basis of type*
`df_final1.groupby(['type']).agg({"title":"nunique"})`

Out[73]:

type	title
Movie	6115
TV Show	2676

In [75]: *df_type=df_final1.groupby(['type']).agg({"title":"nunique"}).reset_index()*
`plt.pie(df_type['title'], explode=(0.05,0.05), labels=df_type['type'], colors=['purple', 'green'])`
`plt.show()`



In [76]: *# We have 70:30 ratio of Movies and TV Shows in our data*

In [78]: *#number of distinct titles on the basis of country*
`df_final1.groupby(['Country']).agg({"title":"nunique"})`

#The BELOW dataframe shows a flaw in which we are seeing countries, such as Cambodia and United States and United States, are shown as different countries. They should have

Out[78]:

Country	title
---------	-------

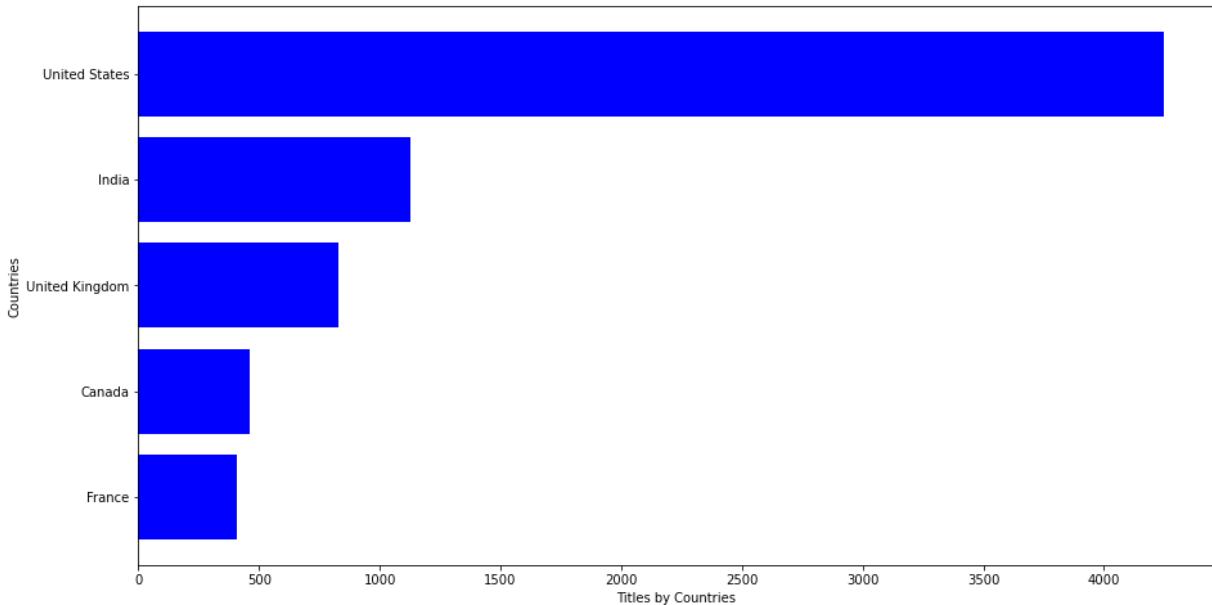
Country title	
Country	
	3
Afghanistan	1
Albania	1
Algeria	3
Angola	2
...	...
Vatican City	1
Venezuela	4
Vietnam	7
West Germany	5
Zimbabwe	3

128 rows × 1 columns

```
In [80]: df_final1['Country'] = df_final1['Country'].str.replace(',', '')
df_final1.head()
```

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

```
In [85]: df_country=df_final1.groupby(['Country']).agg({"title":"nunique"}).reset_index().sort_values('titles', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_country[:::-1]['Country'], df_country[:::-1]['titles'], color=['blue'])
plt.xlabel('Titles by Countries')
plt.ylabel('Countries')
plt.show()
```



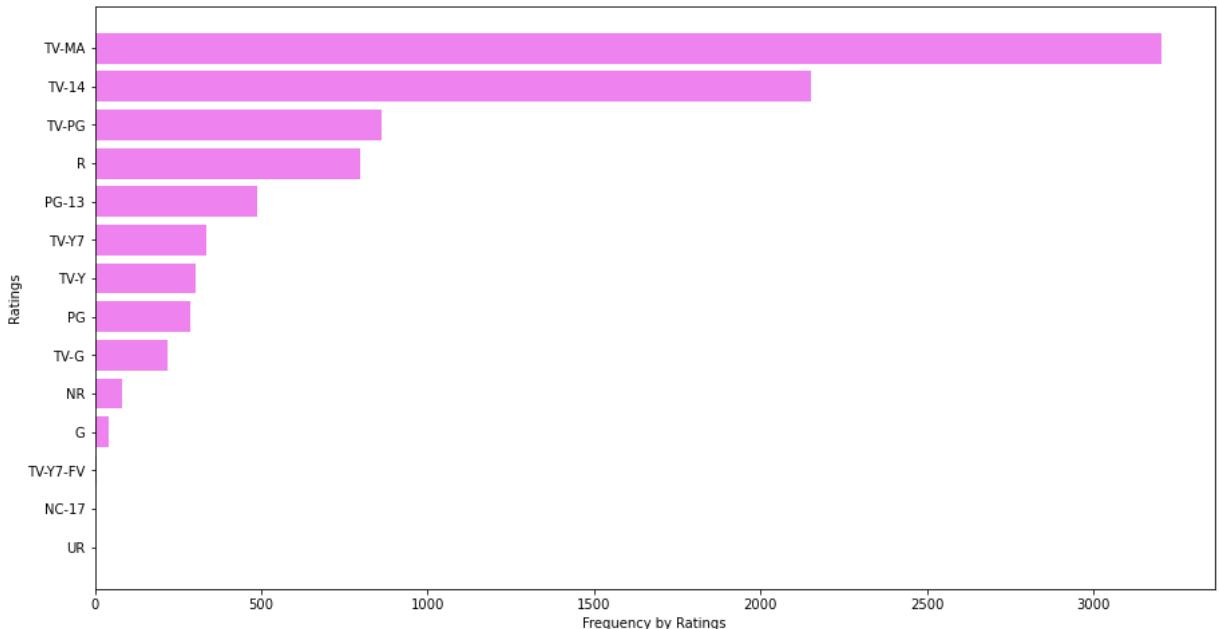
```
In [ ]: # US, India, UK, Canada and France are leading countries in Content Creation on Netflix
```

```
In [88]: #number of distinct titles on the basis of rating
df_final1.groupby(['rating']).agg({"title":"nunique"})
```

Out[88]:

	title
rating	
G	41
NC-17	3
NR	83
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 [90]: 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()
```



```
In [ ]: #Most of the highly rated content on Netflix is intended for Mature Audiences, R Rating audience under 14 and those which require Parental Guidance
```

```
In [91]: #number of distinct titles on the basis of duration
df_final1.groupby(['duration']).agg({'title':'nunique'})
```

```
Out[91]:
```

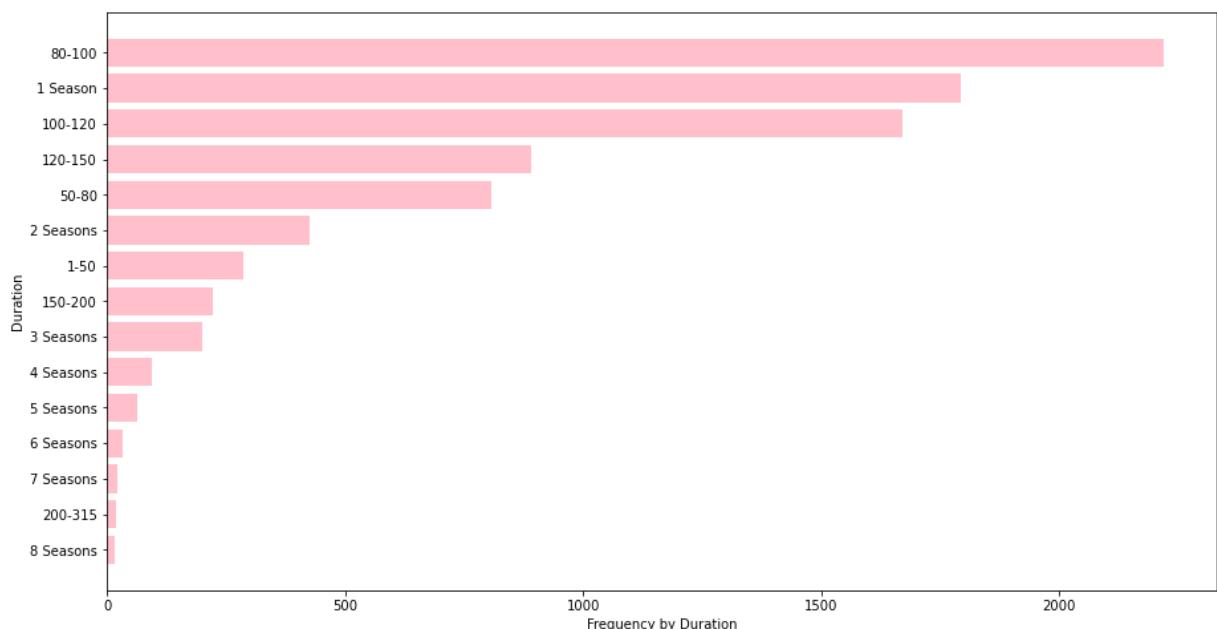
duration	title
1 Season	1793
1-50	287
10 Seasons	7
100-120	1671
11 Seasons	2
12 Seasons	2
120-150	891
13 Seasons	3
15 Seasons	2
150-200	222
17 Seasons	1
2 Seasons	425
200-315	19
3 Seasons	199
4 Seasons	95
5 Seasons	65
50-80	808
6 Seasons	33

title**duration**

7 Seasons	23
8 Seasons	17
80-100	2220
9 Seasons	9

In [92]:

```
df_duration=df_final1.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()
```



In []:

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

In [93]:

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

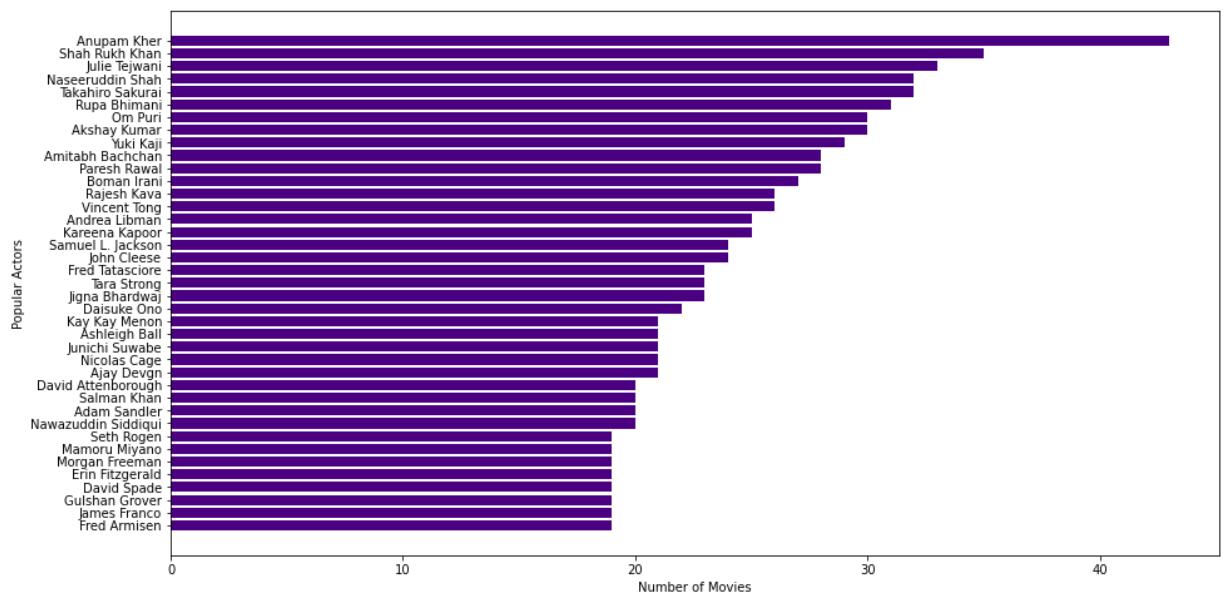
Out[93]:

title	
Actors	
Jr.	2
"Riley" Lakdhar Dridi	1
'Najite Dede	2
2 Chainz	1
2Mex	1
...	...
Şevket Çoruh	1

title	
Actors	
Şinasi Yurtsever	3
Şükran Ovalı	1
Şükrü Özyıldız	2
Şopé Dirísù	1

36440 rows × 1 columns

```
In [95]: 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()
```



```
In [ ]: # Anupam Kher, SRK, Julie Tejwani, Naseeruddin Shah and Takahiro Sakurai occupy the top
#watched content.
```

```
In [96]: #number of distinct titles on the basis of Directors
df_final1.groupby(['Directors']).agg({"title":"nunique"})
```

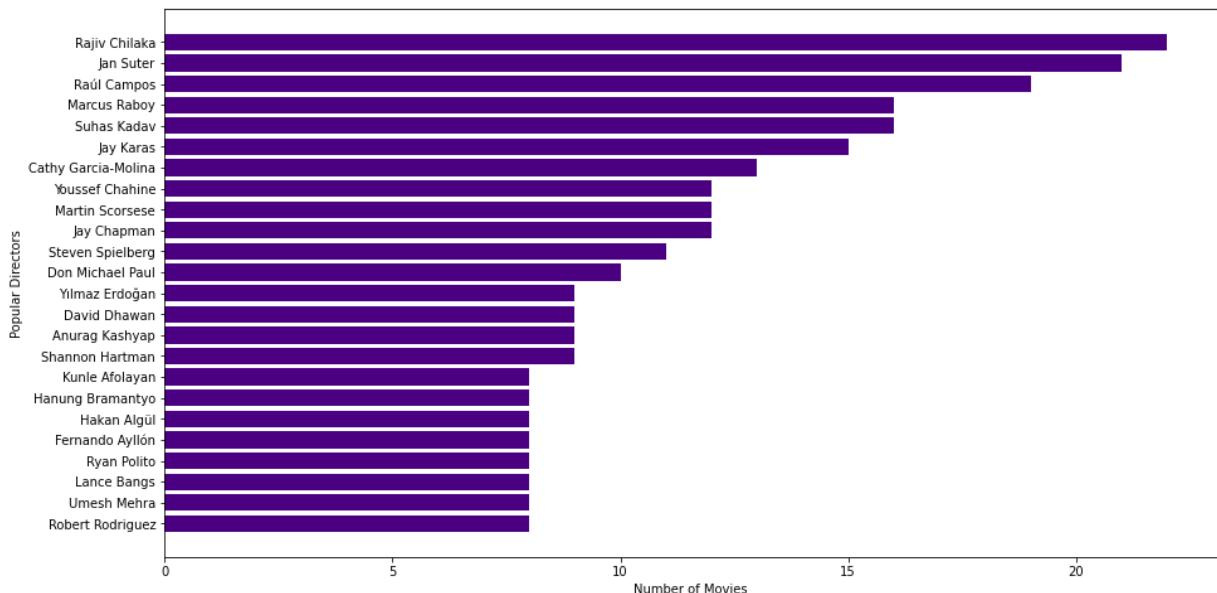
title	
Directors	
A. L. Vijay	2
A. Raajdheep	1
A. Salaam	1
A.R. Murugadoss	2
Aadish Keluskar	1

title	
Directors	
...	...
Éric Warin	1
Ísold Uggadóttir	1
Óskar Thór Axelsson	1
Ömer Faruk Sorak	3
Şenol Sönmez	2

4994 rows × 1 columns

In [103...]

```
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 [104...]

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

Out[104...]

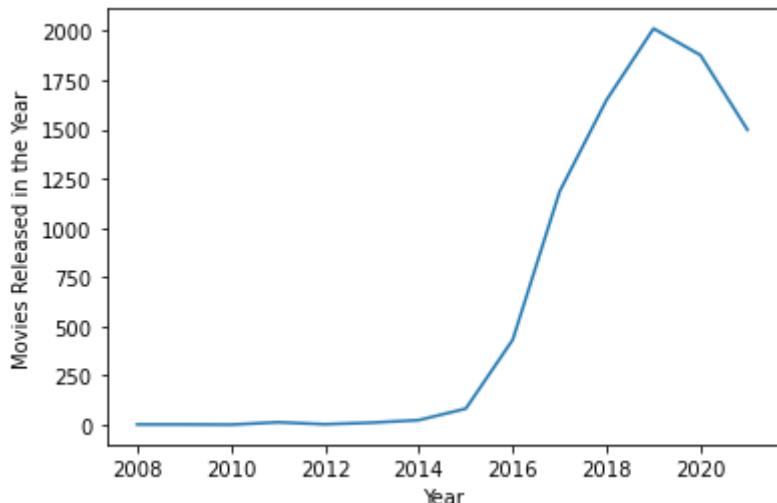
title	
year	
2008	2
2009	2
2010	1
2011	13
2012	3

title

year	
2013	11
2014	24
2015	82
2016	432
2017	1185
2018	1650
2019	2012
2020	1877
2021	1498

In [105...]

```
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 [106...]

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

Out[106...]

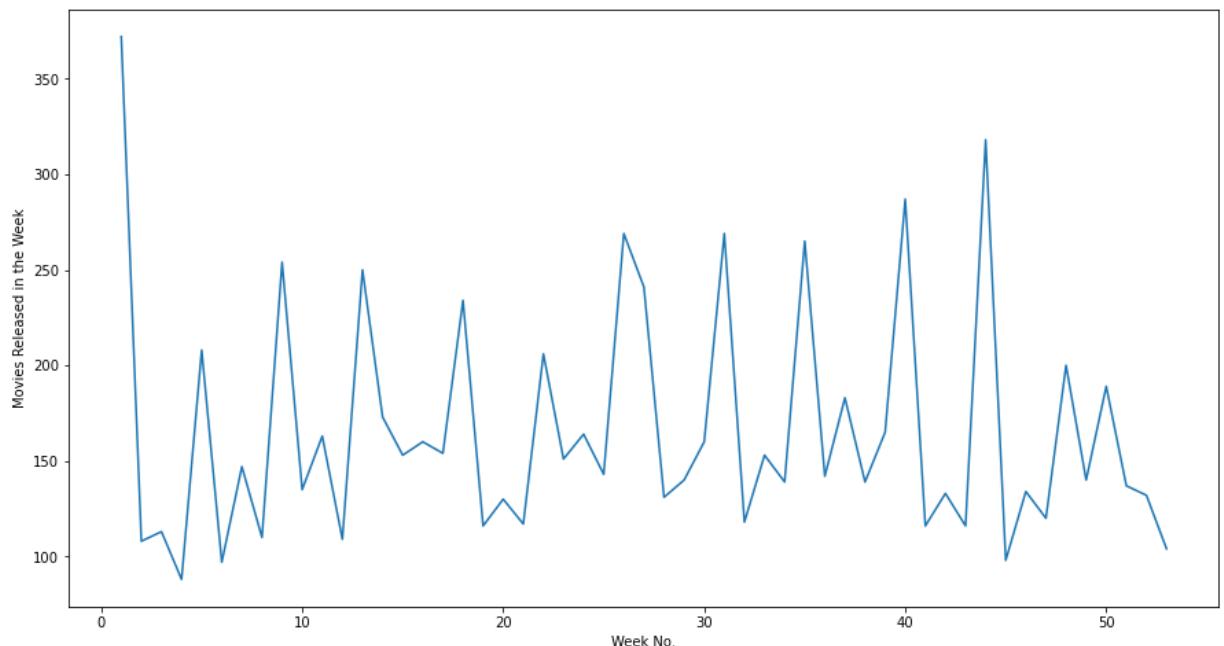
title**week_Added**

1	372
2	108
3	113
4	88
5	208

title**week_Added****6** 97**7** 147**8** 110**9** 254**10** 135**11** 163**12** 109**13** 250**14** 173**15** 153**16** 160**17** 154**18** 234**19** 116**20** 130**21** 117**22** 206**23** 151**24** 164**25** 143**26** 269**27** 241**28** 131**29** 140**30** 160**31** 269**32** 118**33** 153**34** 139**35** 265**36** 142**37** 183**38** 139**39** 165**40** 287

title	
week_Added	
41	116
42	133
43	116
44	318
45	98
46	134
47	120
48	200
49	140
50	189
51	137
52	132
53	104

```
In [107...]: df_week=df_final1.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



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

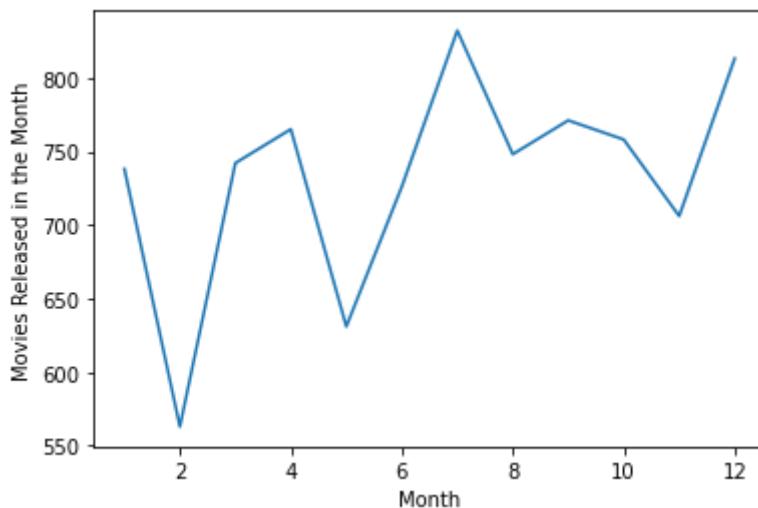
```
In [108...]: #number of distinct titles on the basis of Month
df_final1.groupby(['month_added']).agg({"title":"nunique"})
```

Out[108...]

title	
month_added	
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 [109...]

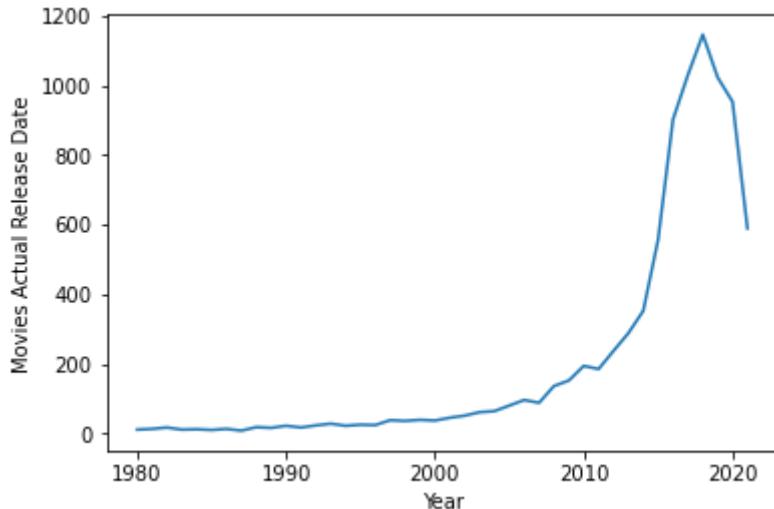
```
df_month=df_final1.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("Movies Released in the Month")
plt.xlabel("Month")
plt.show()
```



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

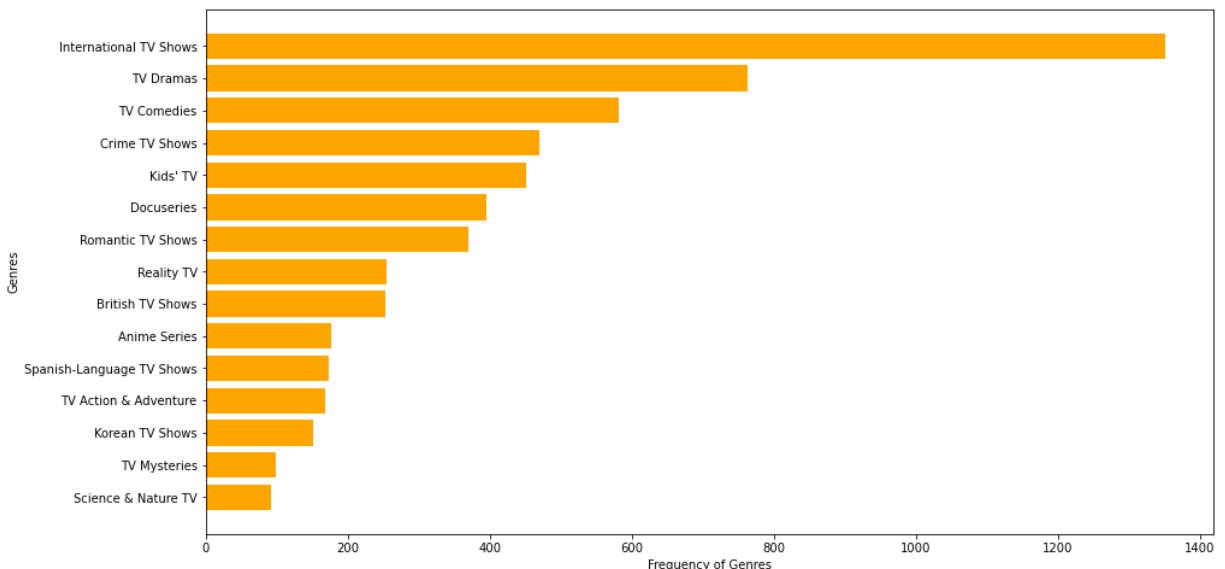
In [111...]

```
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()
```



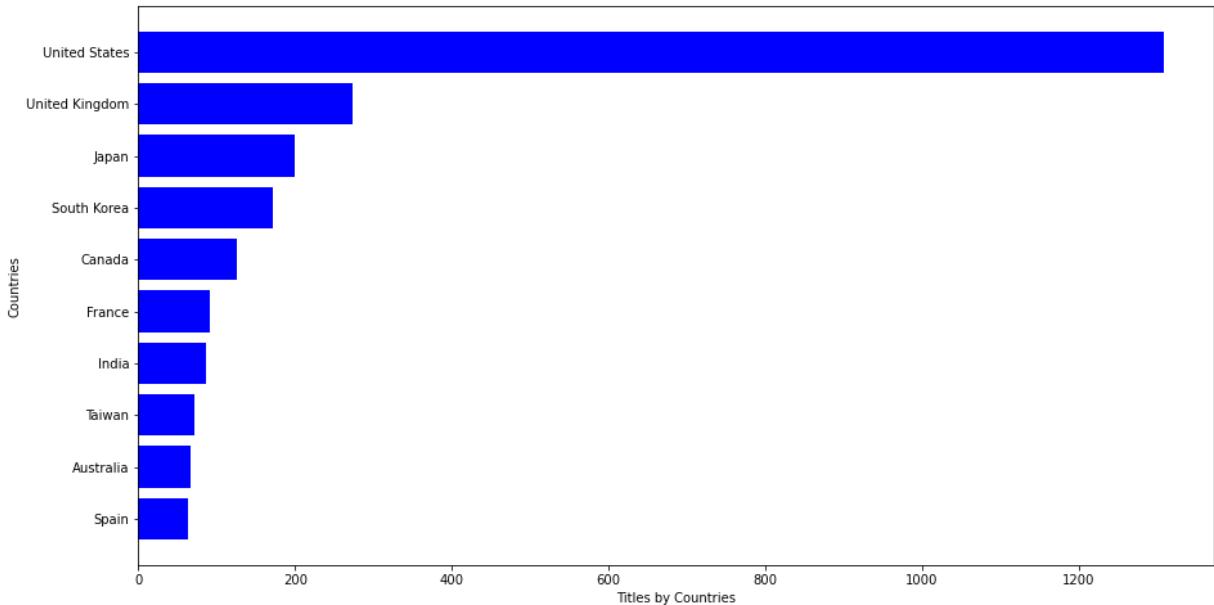
Univariate Analysis separately for shows and movies

```
In [112...]: df_shows=df_final1[df_final1['type']=='TV Show']
df_movies=df_final1[df_final1['type']=='Movie']
df_genre=df_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()
```

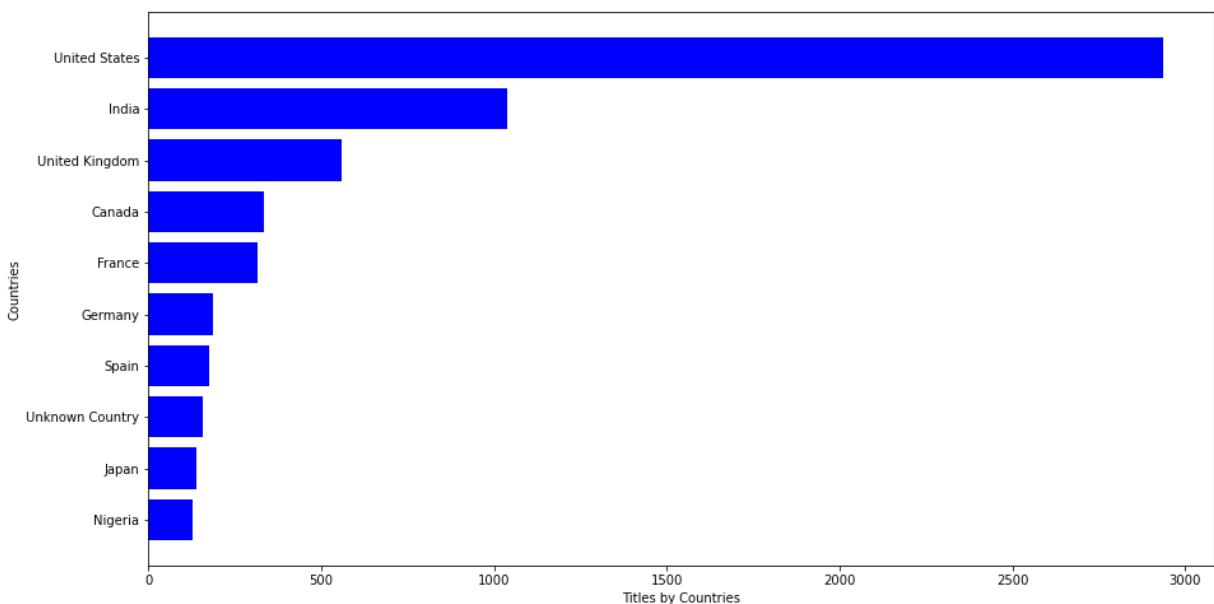


International TV Shows, Dramas and Comedy Genres are popular across TV Shows in Netflix

```
In [114...]: df_country=df_shows.groupby(['Country']).agg({"title":"nunique"}).reset_index().sort_values('nunique', 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()
```



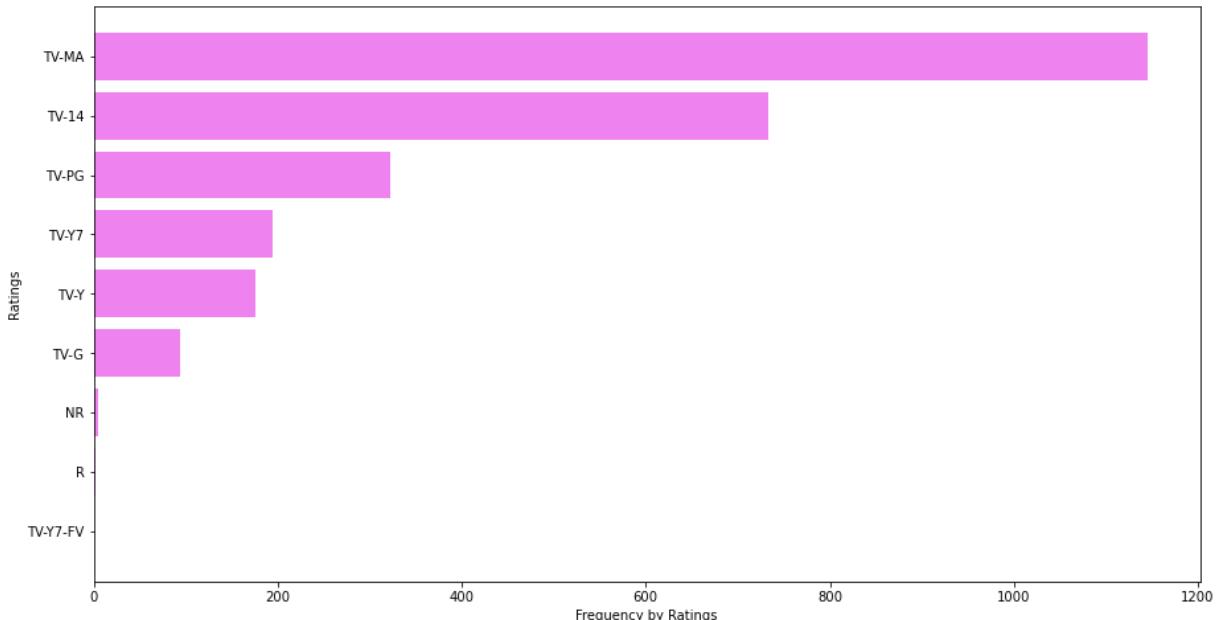
```
In [115...]: df_country=df_movies.groupby(['Country']).agg({"title":"nunique"}).reset_index().sort_values('titles', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_country['Country'], df_country['titles'], color=['blue'])
plt.xlabel('Titles by Countries')
plt.ylabel('Countries')
plt.show()
```



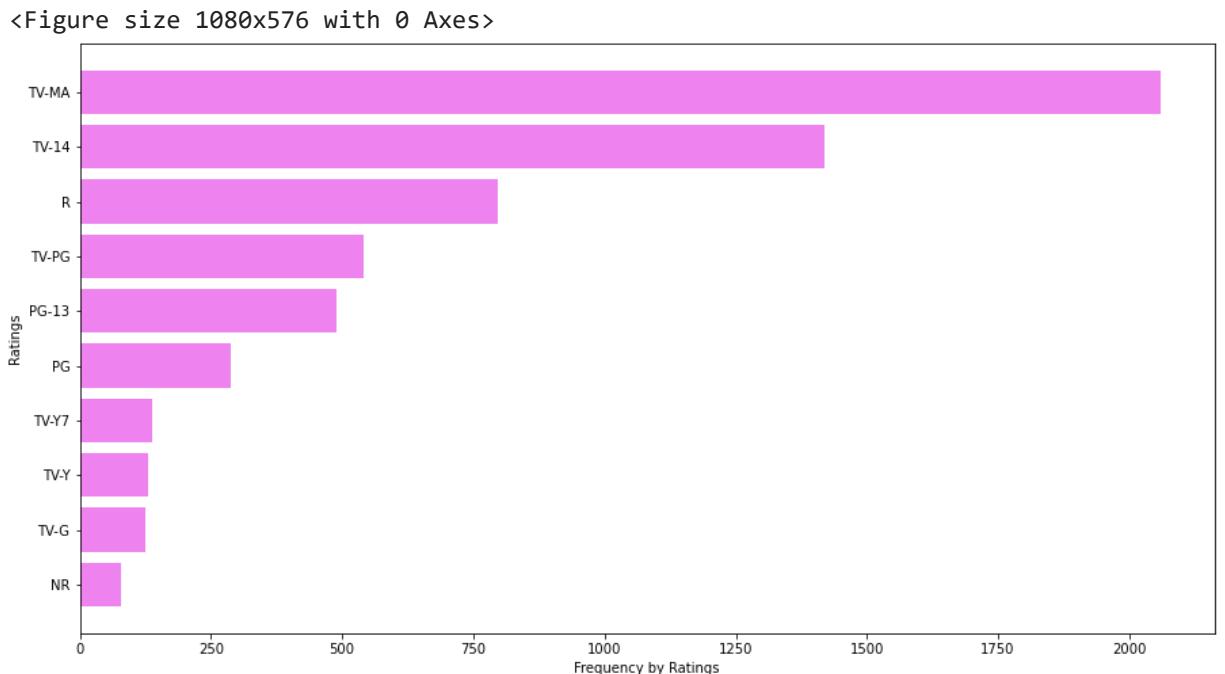
United States is leading across both TV Shows and Movies, UK also provides great content across TV Shows and Movies. Surprisingly India is much more prevalent in Movies as compared to TV Shows. Moreover the number of Movies created in India outweigh the sum of TV Shows and Movies across UK since India was rated as second in net sum of whole content across Netflix.

```
In [116...]: df_rating=df_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values('titles', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_rating['rating'], df_rating['titles'], color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```

<Figure size 1080x576 with 0 Axes>

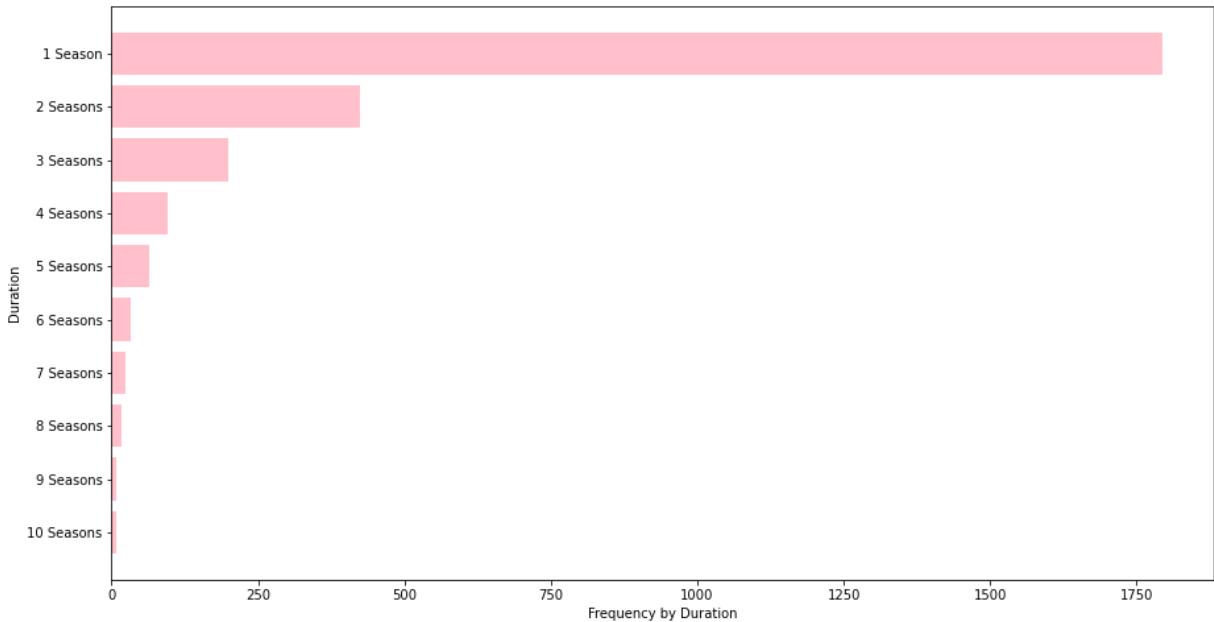


```
In [117...]: df_rating=df_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_
plt.figure(figsize=(15,8))
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



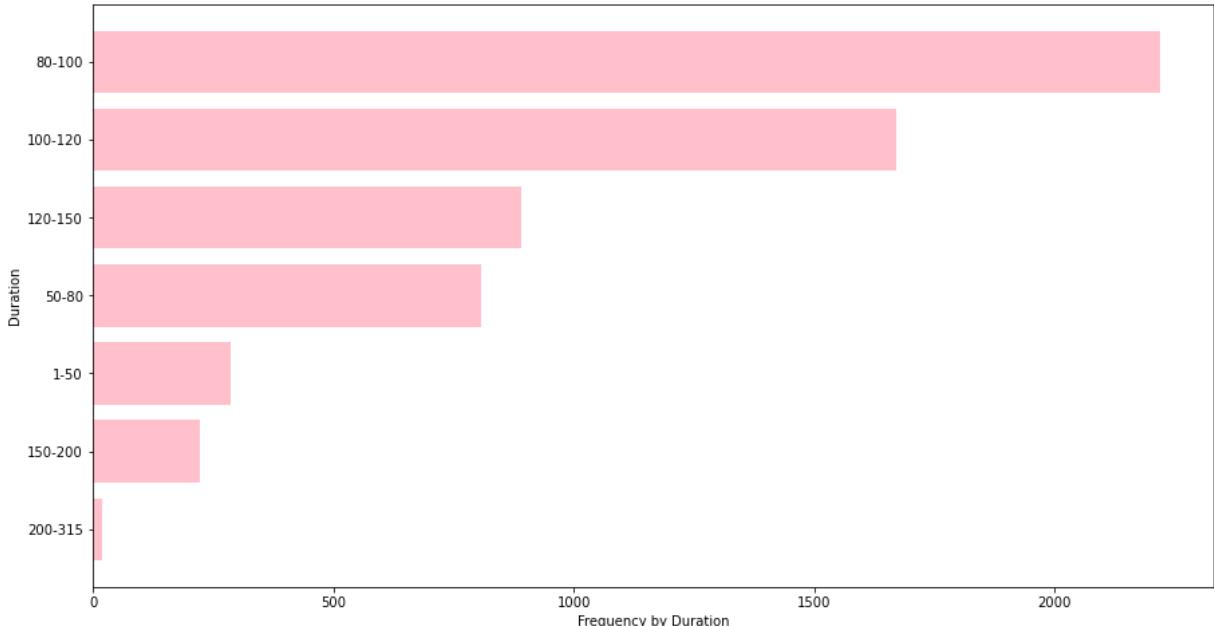
So it seems plausible to conclude that the popular ratings across Netflix includes Mature Audiences and those appropriate for over 14/over 17 ages. Moreover there are no TV Shows having a rating of R

```
In [118...]: df_duration=df_shows.groupby(['duration']).agg({"title":"nunique"}).reset_index().so
plt.figure(figsize=(15,8))
plt.barh(df_duration[::-1]['duration'], df_duration[::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



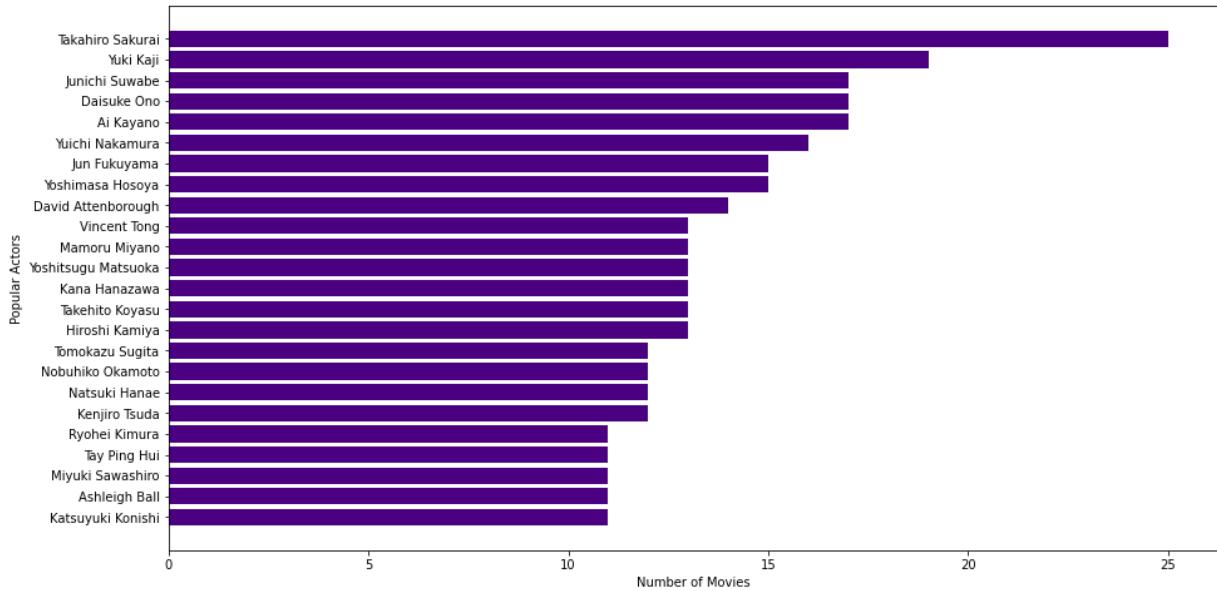
Across TV Shows, shows having only 1 Season are common as soon as the season length increases, the number of shows decrease and this definitely sounds as expected

```
In [119]: 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()
```



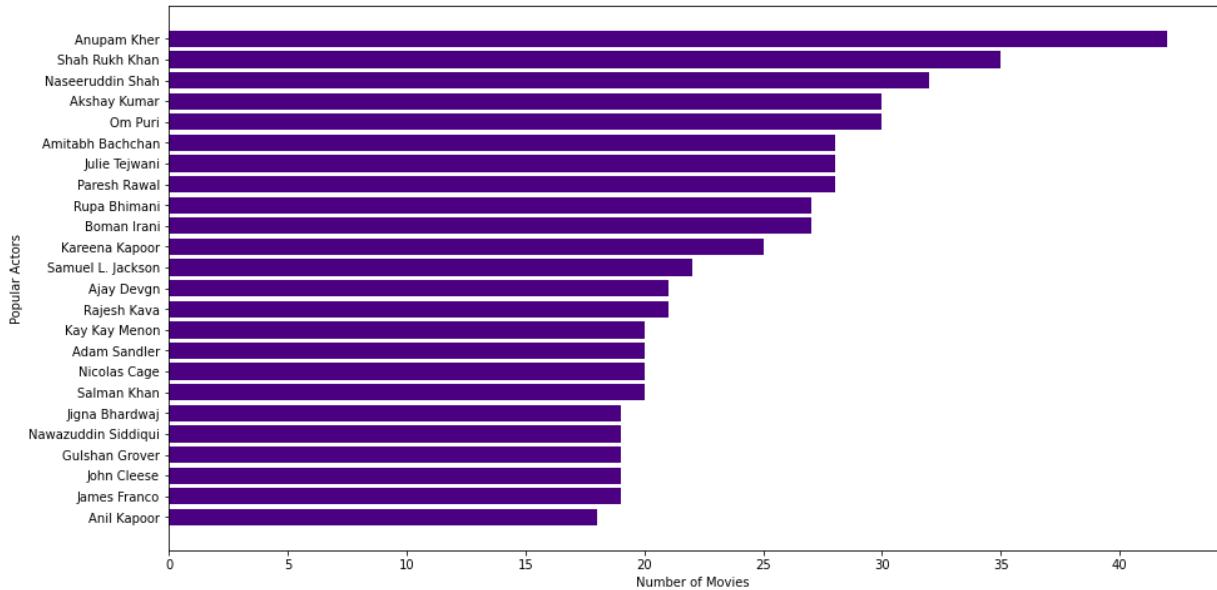
Across movies 80-100, 100-120 and 120-150 is the ranges of minutes for which most movies lie. So quite possibly 80-150 mins is the sweet spot we would be wanting for movies.

```
In [121]: 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()
```



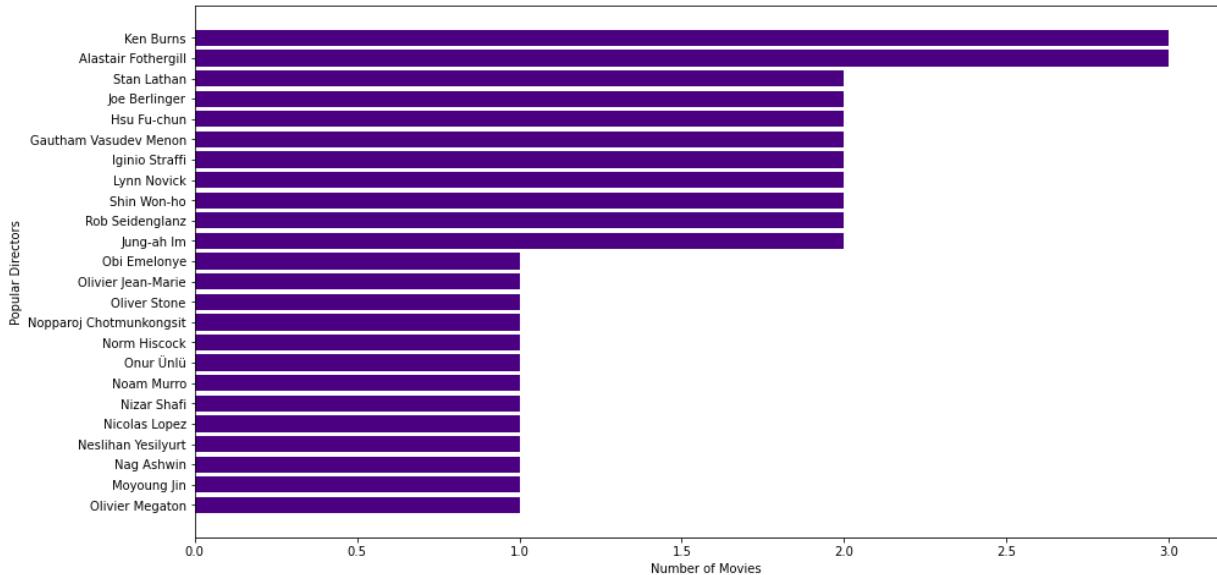
Takahiro Sakurai, Yuki Kaji and other South Korean/Japanese actors are the most popular actors across TV Shows

```
In [122]: 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()
```



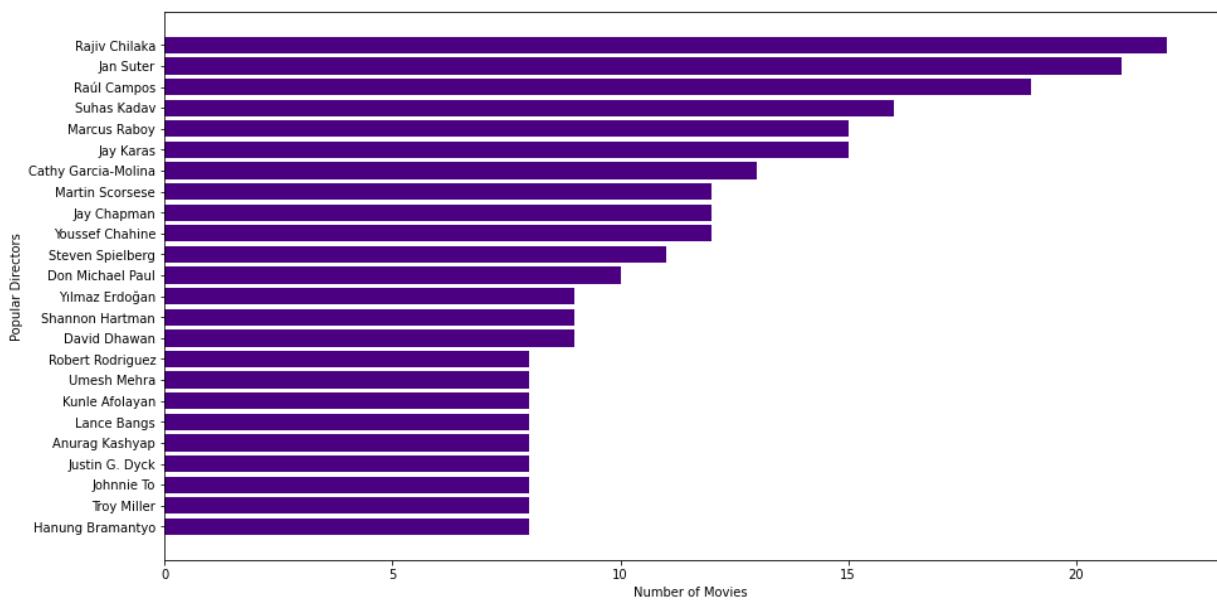
```
In [ ]: Our bollywood actors such as Anupam Kher, SRK, Naseeruddin Shah are very much popula
Netflix
```

```
In [124]: 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()
```



In [125...]

```
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()
```

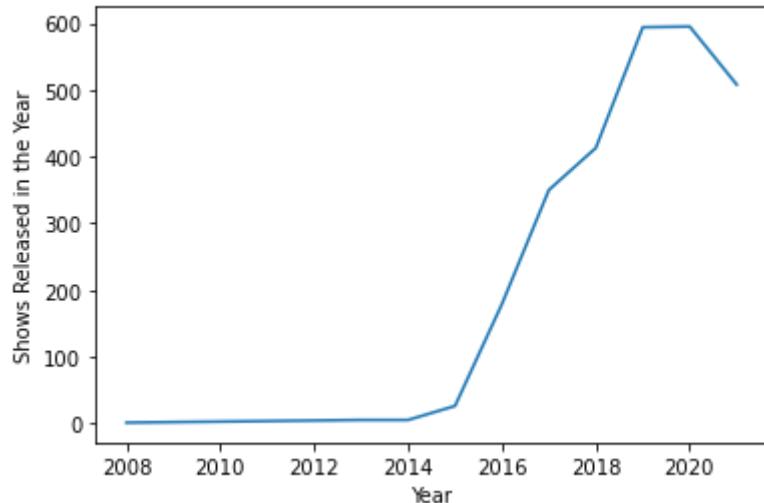


In [126...]

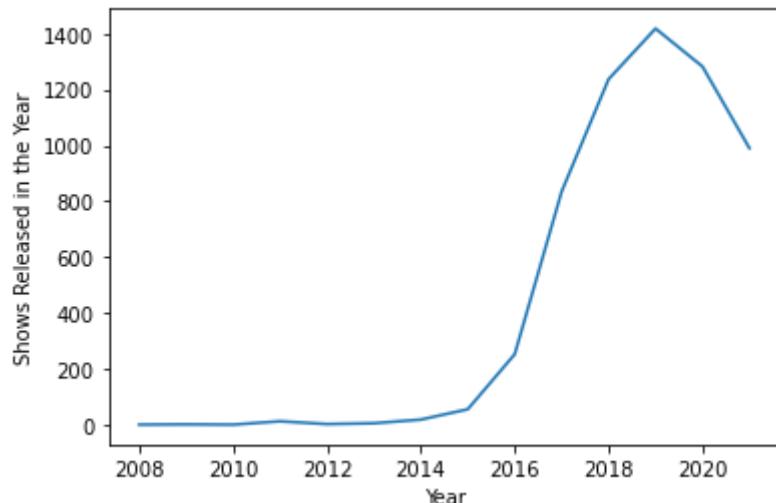
```
# Ken Burns, Alastair Fothergill, Stan Lathan, Joe Barlinger are popular directors across movies
# Rajiv Chilka, Jan Suter, Raul Campos, Suhas Kadav are popular directors across movies
```

In [127...]

```
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()
```

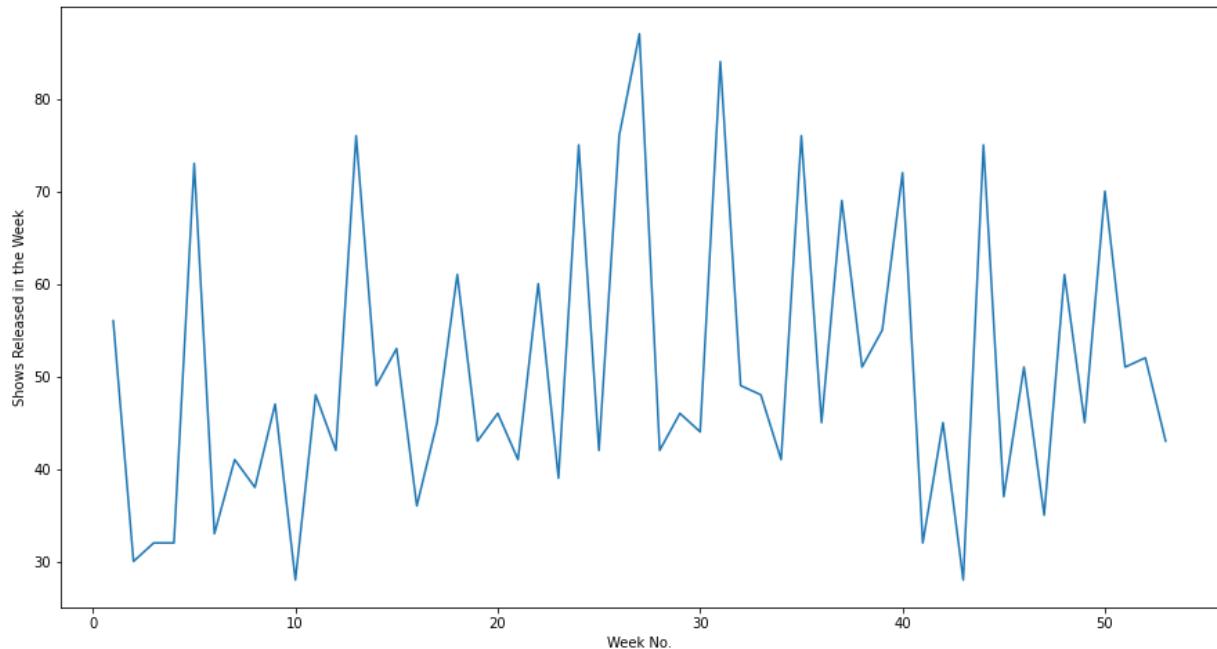


```
In [128...]: df_year=df_movies.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Shows Released in the Year")
plt.xlabel("Year")
plt.show()
```



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

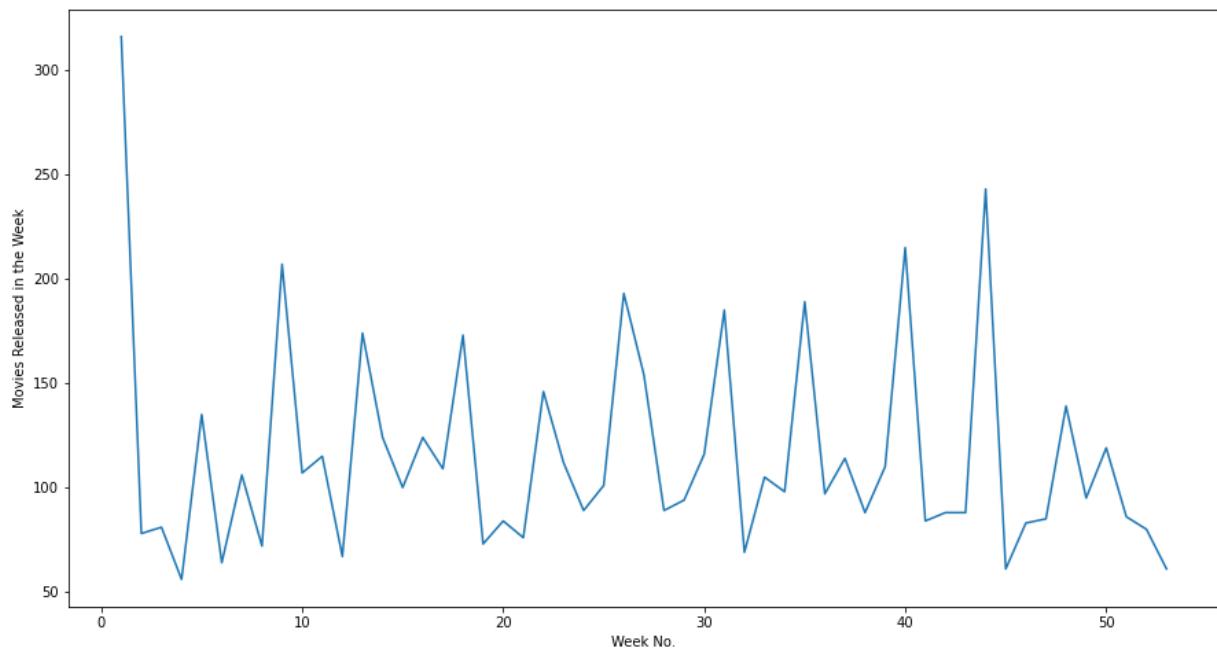
```
In [131...]: df_week=df_shows.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Shows Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



In []:

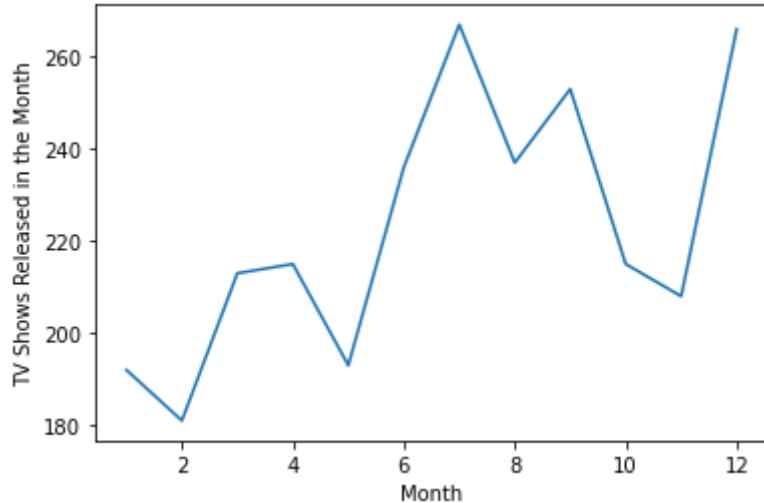
In [130...]

```
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()
```



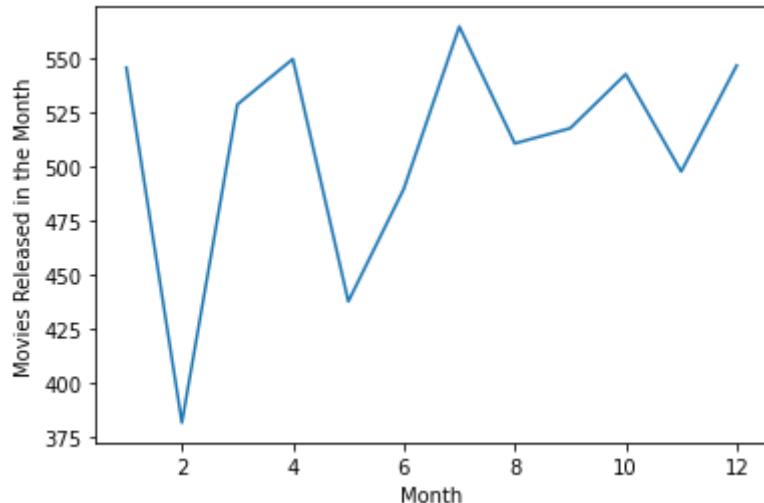
In [132...]

```
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 [133...]

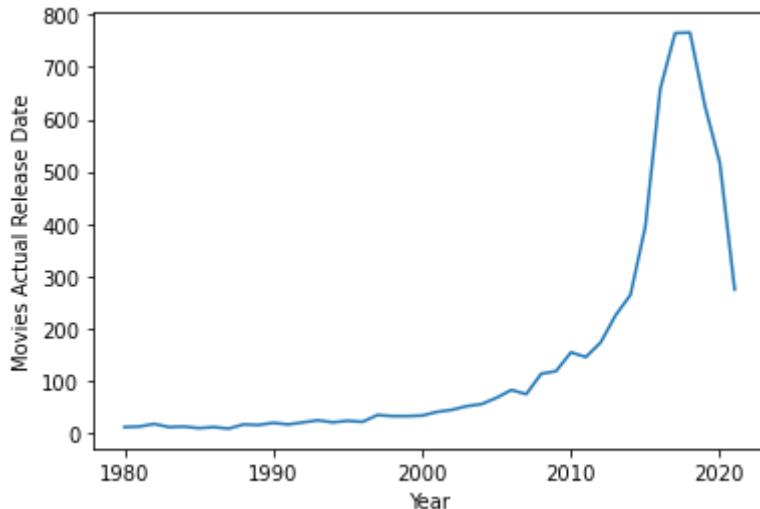
```
df_month=df_movies.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("Movies Released in the Month")
plt.xlabel("Month")
plt.show()
```



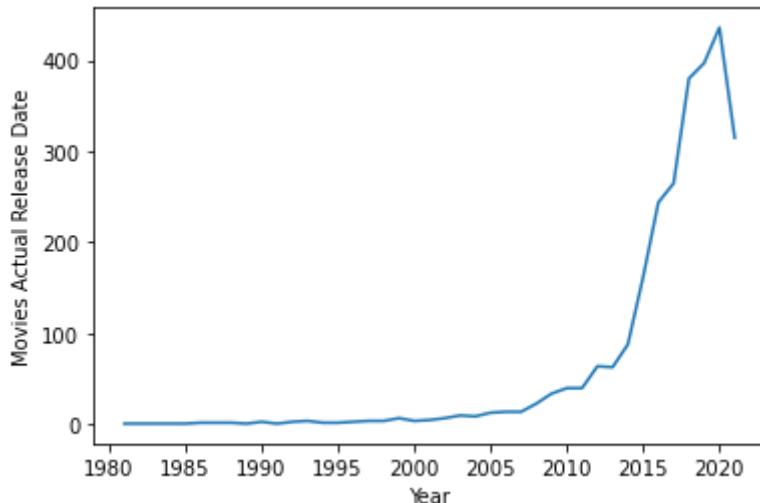
TV Shows are added in Netflix by a tremendous amount in mid weeks/months of the year, i.e-
 July
 Movies are added in Netflix by a tremendous amount in first week/last month of current year and first month of next year

In [135...]

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



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

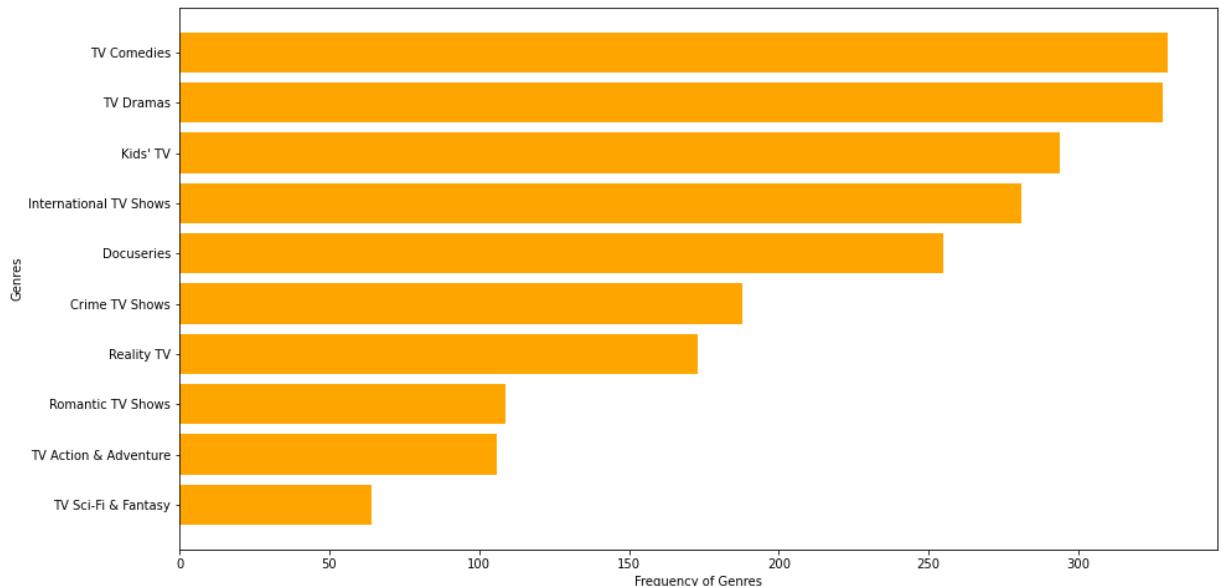
```
In [138...]: #below countries will be analyzed for both shows and movies  
shows_and_movies=['United States','India','United Kingdom']
```

```
#below countries will be only analyzed on basis of shows
only_shows=['Japan','South Korea']
```

Univariate Analysis separately for shows and movies in USA

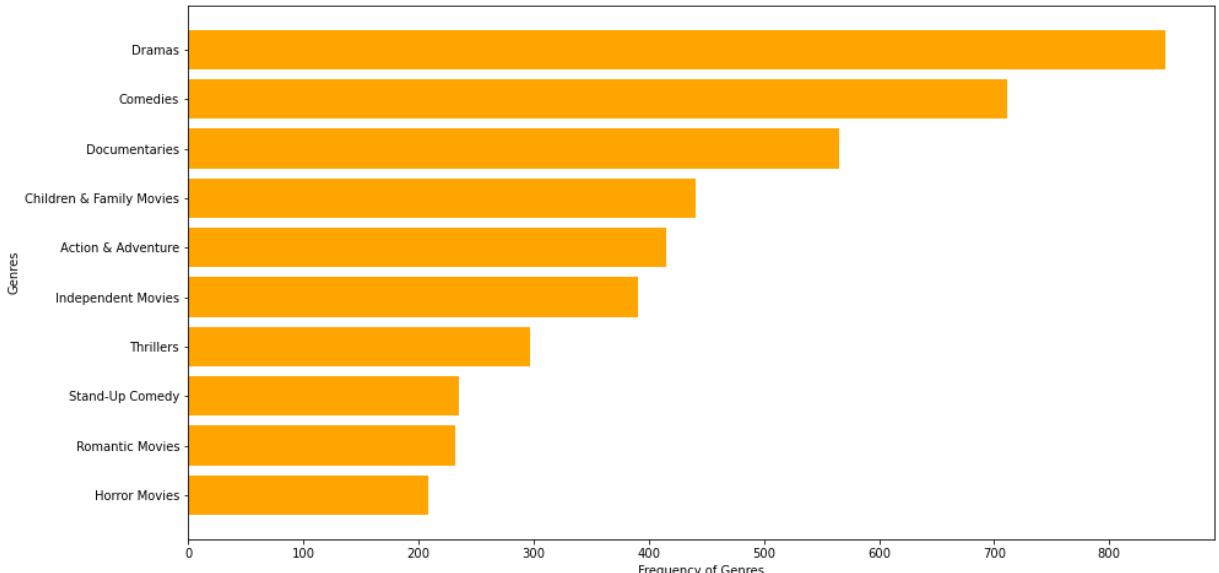
```
In [143...]: #Analyzing USA for both shows and movies
df_usa_shows=df_final1[df_final1['Country']=='United States'][df_final1[df_final1['C
df_usa_movies=df_final1[df_final1['Country']=='United States'][df_final1[df_final1['C
df_usa_shows=df_final1[df_final1['Country']=='United States'][df_final1[df_final1['C
df_usa_movies=df_final1[df_final1['Country']=='United States'][df_final1[df_final1['C
```

```
In [144...]: df_genre=df_usa_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()
```



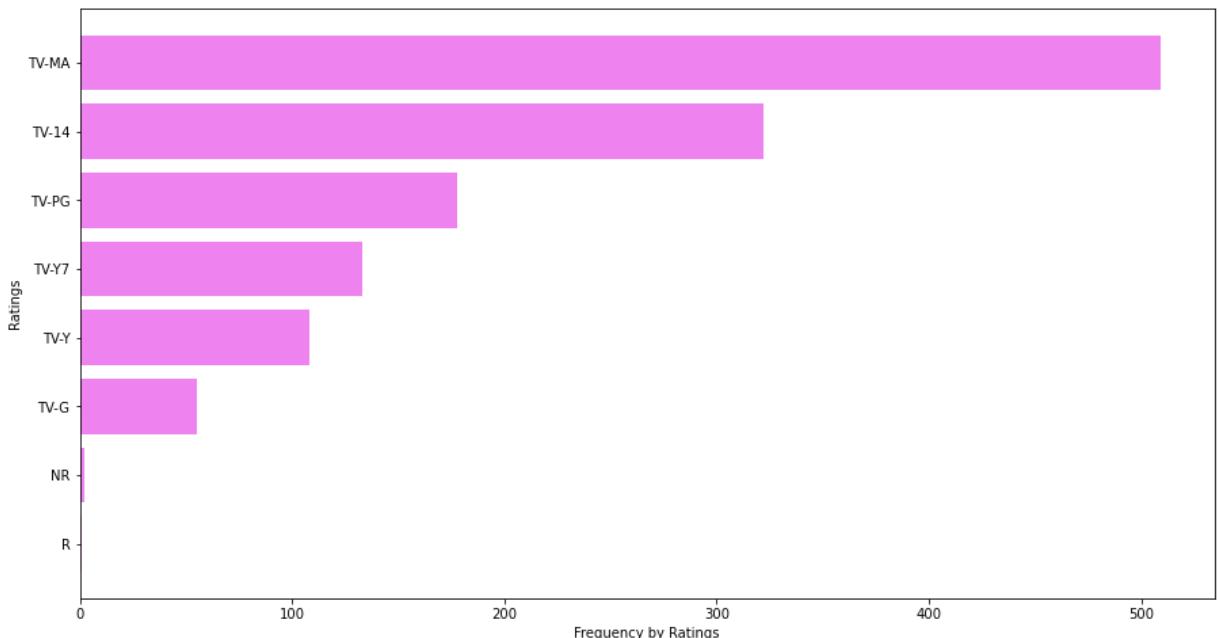
Dramas, Comedy, Kids 'TV Shows, International TV Shows and Docuseries, Genres are popular in TV Series in USA

```
In [145...]: df_genre=df_usa_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```

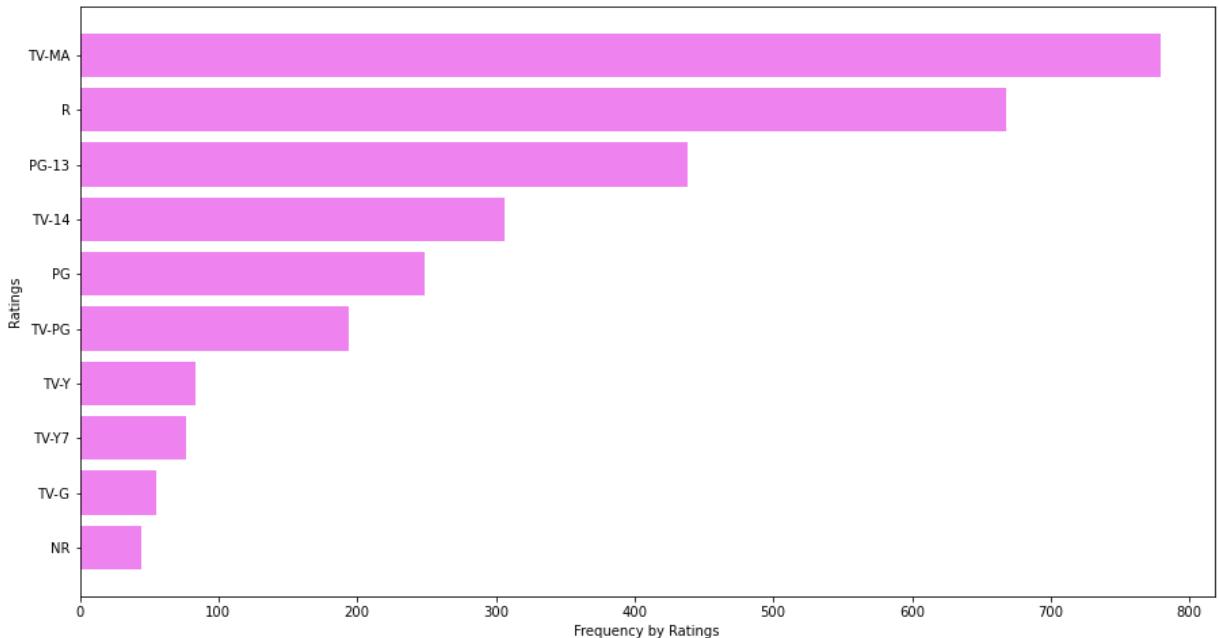


Dramas, Comedy, Documentaries, Family Movies and Action Genres in Movies are popular in USA

```
In [146]: 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()
```

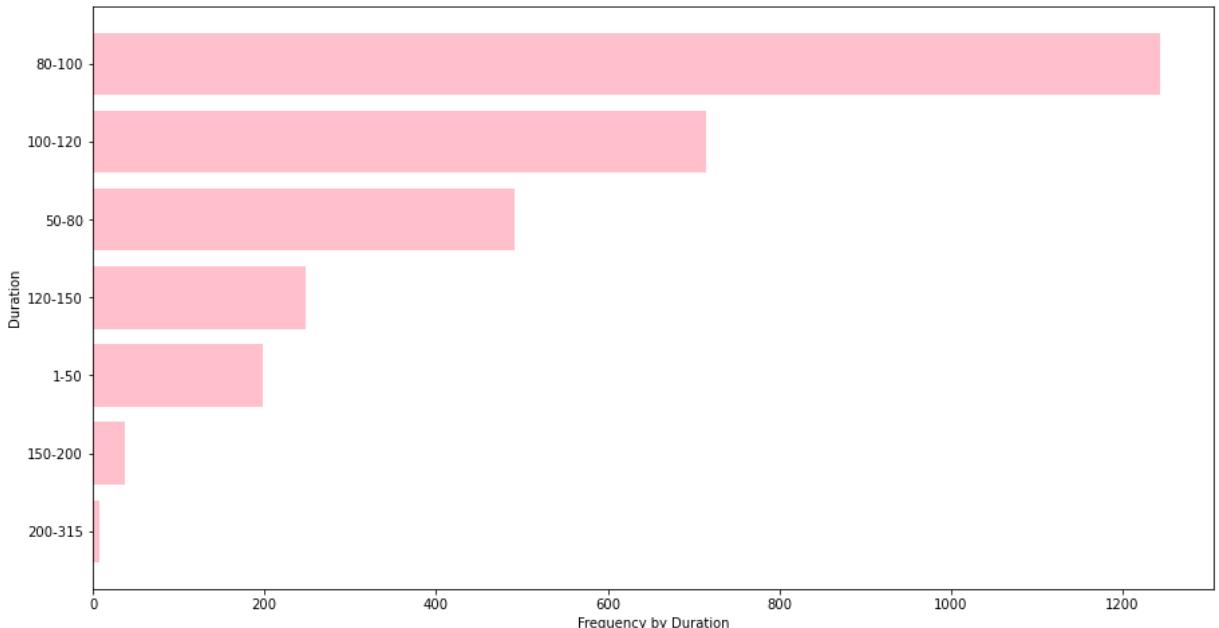


```
In [147]: df_rating=df_usa_movies.groupby(['rating']).agg({'title':'nunique'}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'], color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



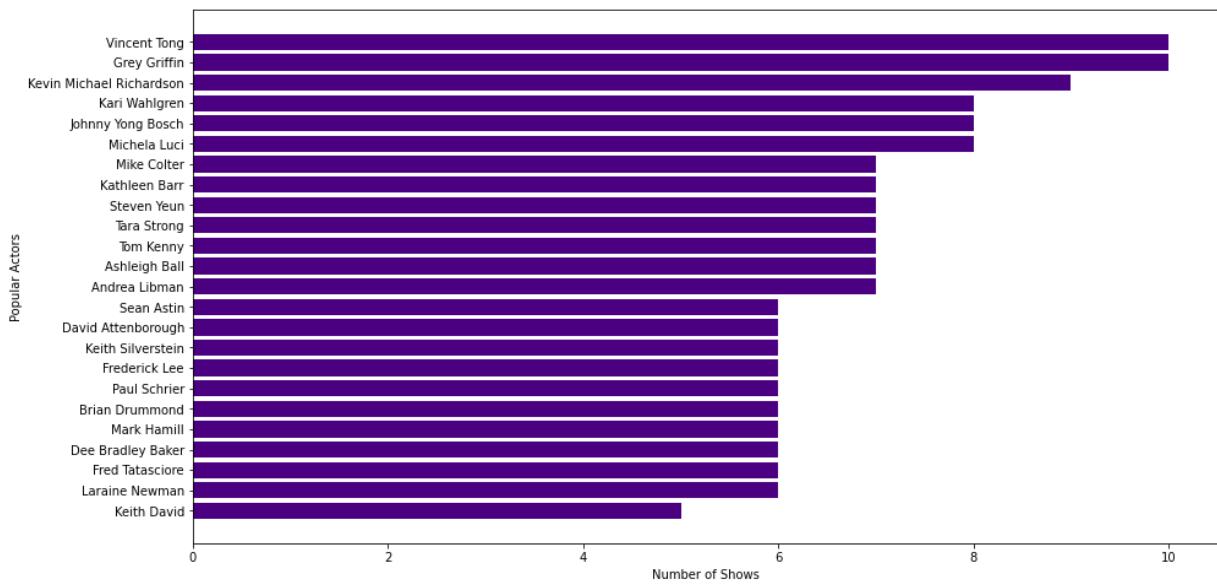
So it seems plausible to conclude that the popular ratings across Netflix includes Mature Audiences and those appropriate for over 14/over 17 ages in both Movies and TV Shows in USA

```
In [148...]: 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

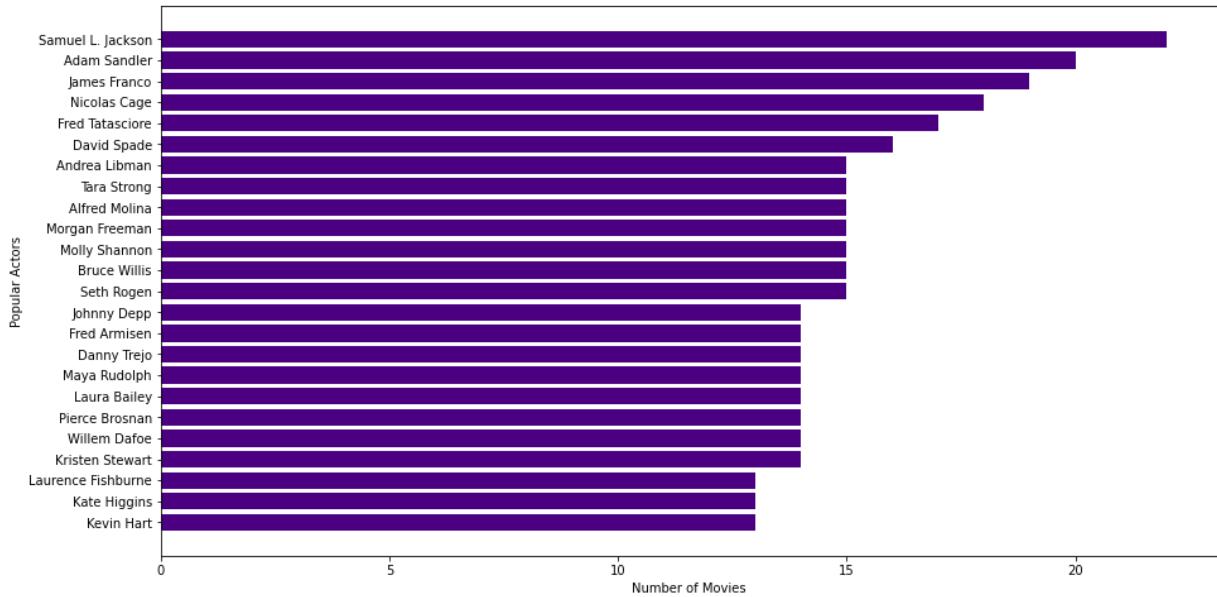
```
In [149...]: df_actors=df_usa_shows.groupby(['Actors']).agg({"title":"nunique"}).reset_index()
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[:::-1]['Actors'], df_actors[:::-1]['title'], color=['indigo'])
plt.xlabel('Number of Shows')
plt.ylabel('Popular Actors')
plt.show()
```



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

In [150...]

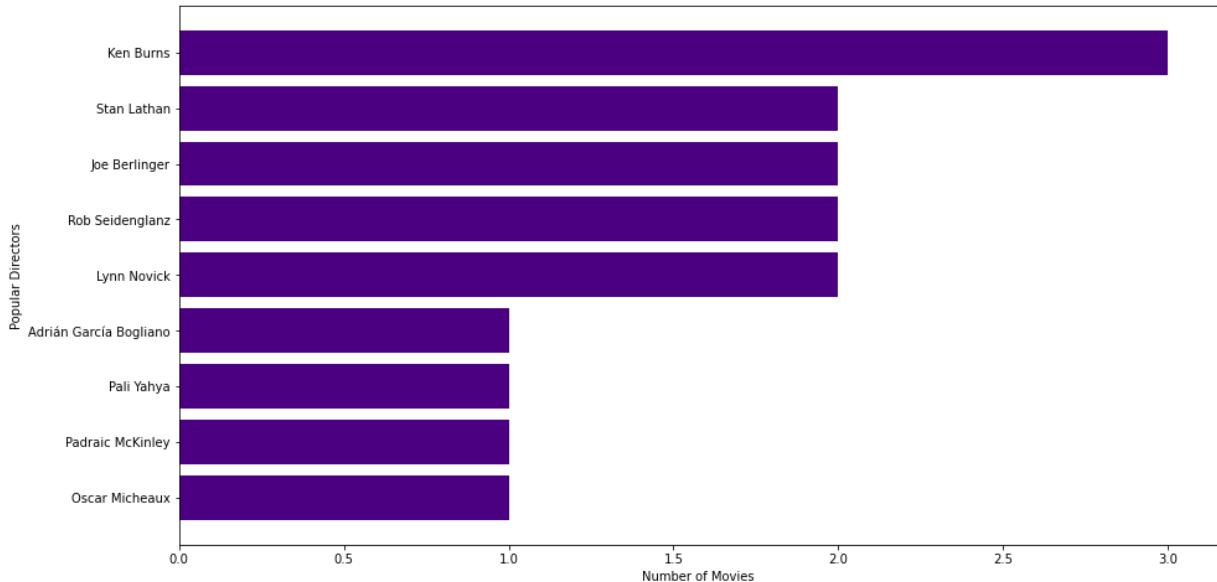
```
df_actors=df_usa_movies.groupby(['Actors']).agg({"title":"nunique"}).reset_index()
df_actors=df_actors[df_actors['Actors']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['Actors'], df_actors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular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



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

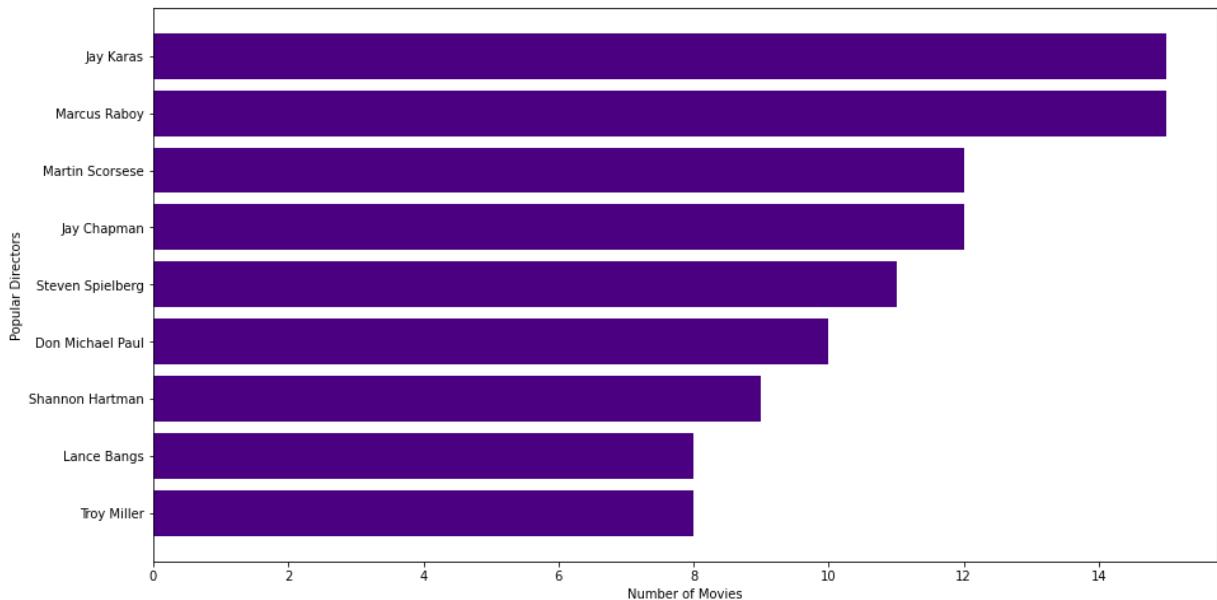
In [154...]

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



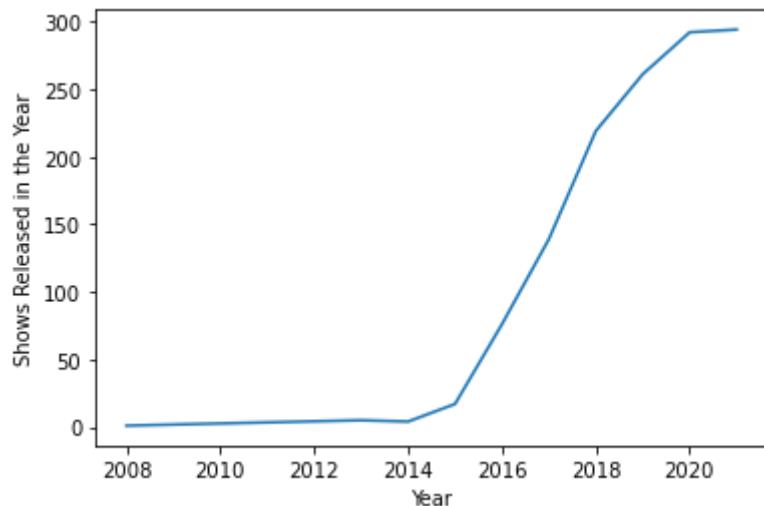
Ken Burns, Stan Lathan, Joe Barlinger are popular directors across TV Shows on Netflix in USA

```
In [156...]: 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()
```

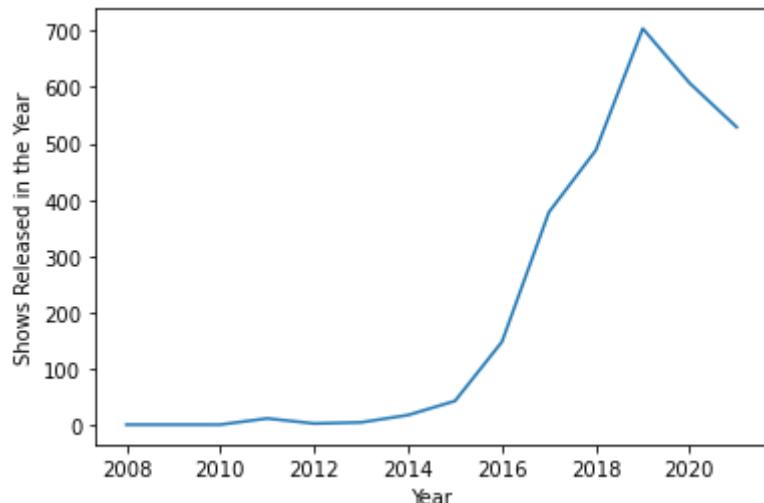


Jay Karas, Marcus Raboy, Martin Scorsese and Jay Chapman are popular directors across movies in USA

```
In [157...]: 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()
```

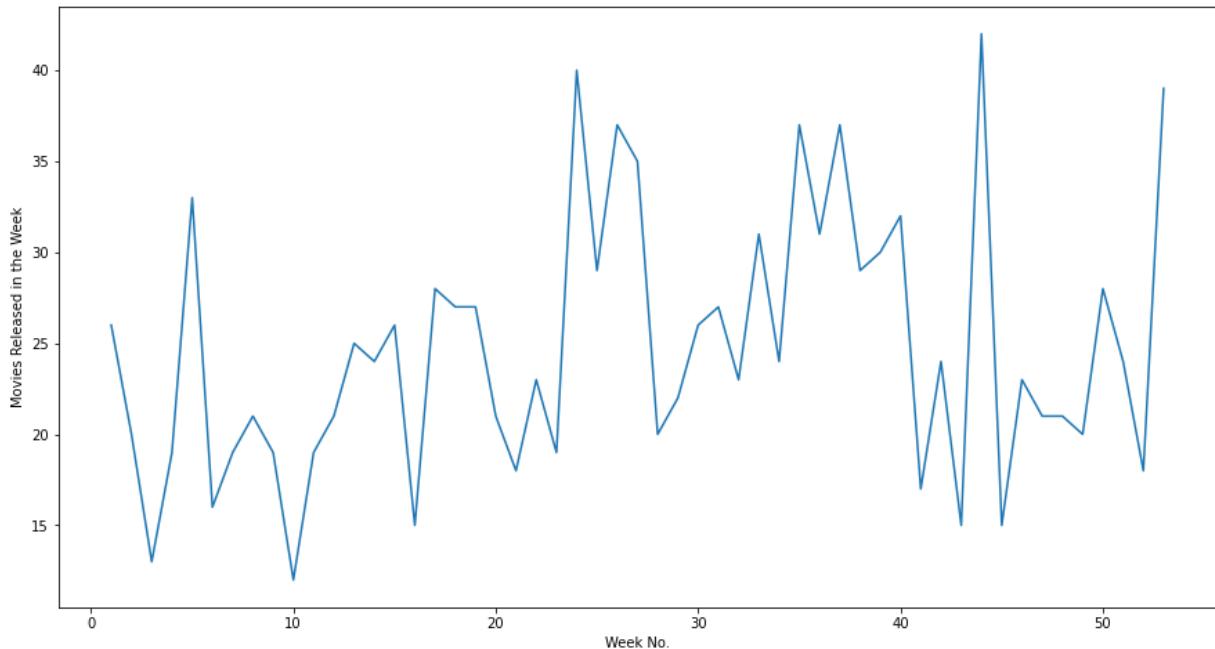


```
In [158...]: df_year=df_usa_movies.groupby(['year']).agg({'title':'nunique'}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Shows Released in the Year")
plt.xlabel("Year")
plt.show()
```



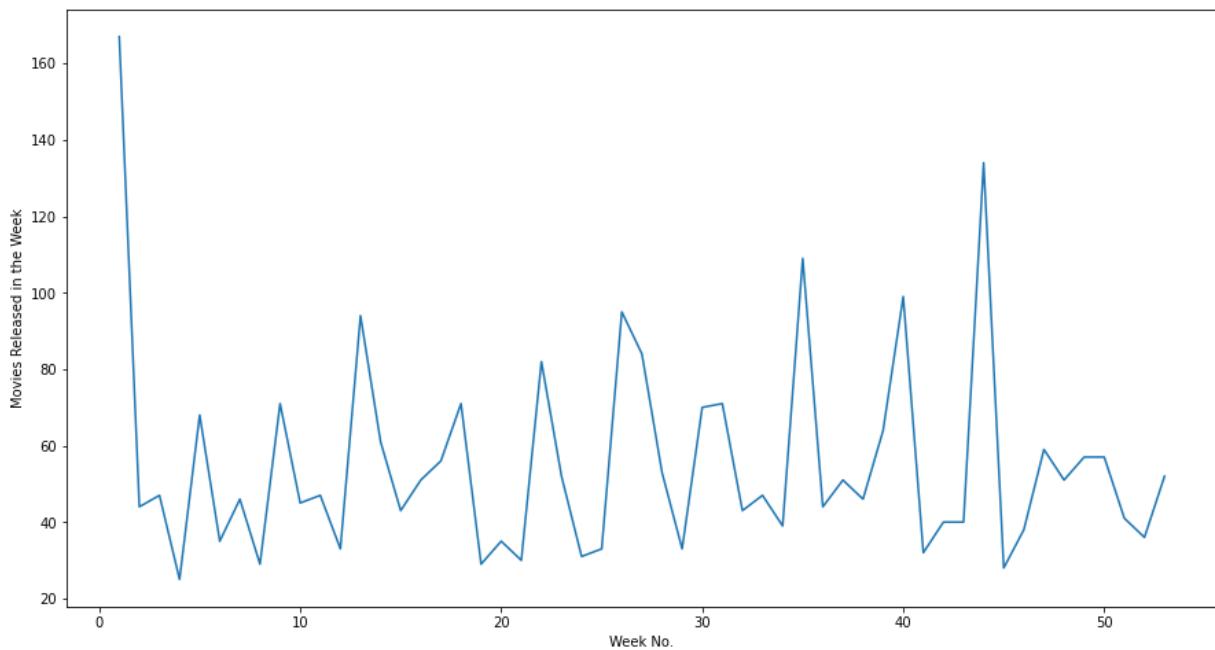
In USA, number of shows remained the same in 2021 as they were in 2020 while number of movies declined:

```
In [159...]: 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()
```



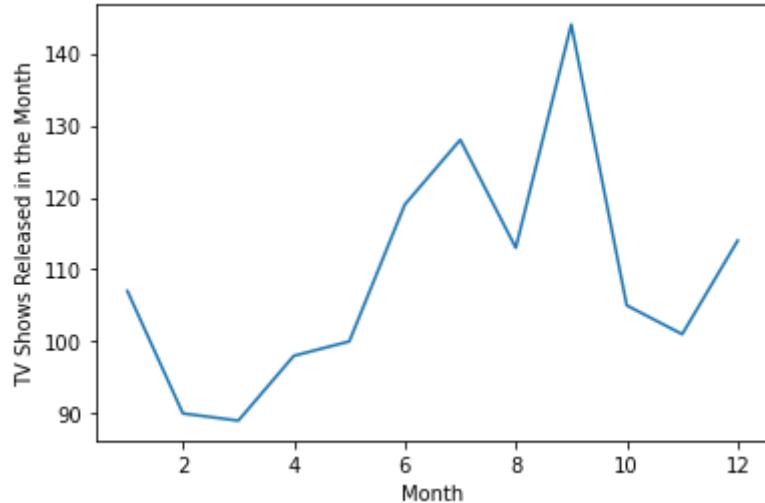
In [160...]

```
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()
```

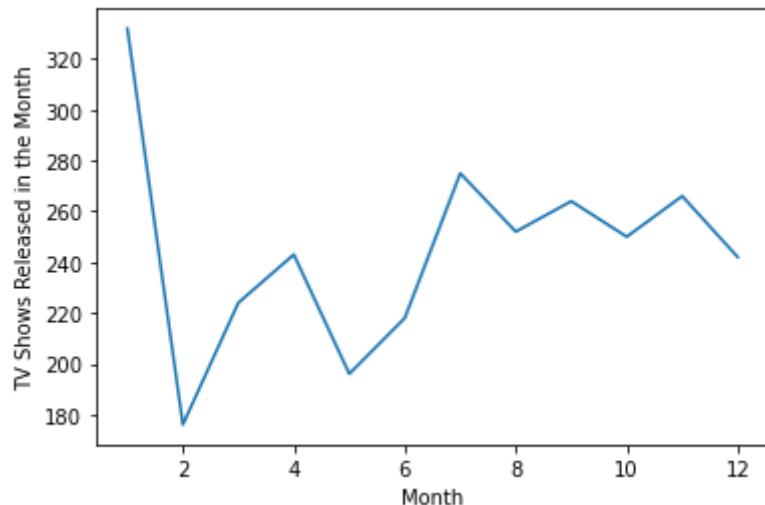


In [163...]

```
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()
```

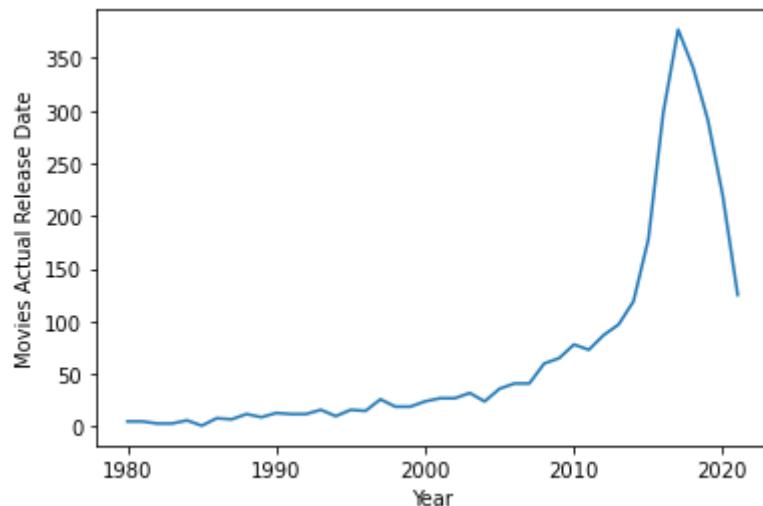


```
In [164...]: df_month=df_usa_movies.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("TV Shows Released in the Month")
plt.xlabel("Month")
plt.show()
```

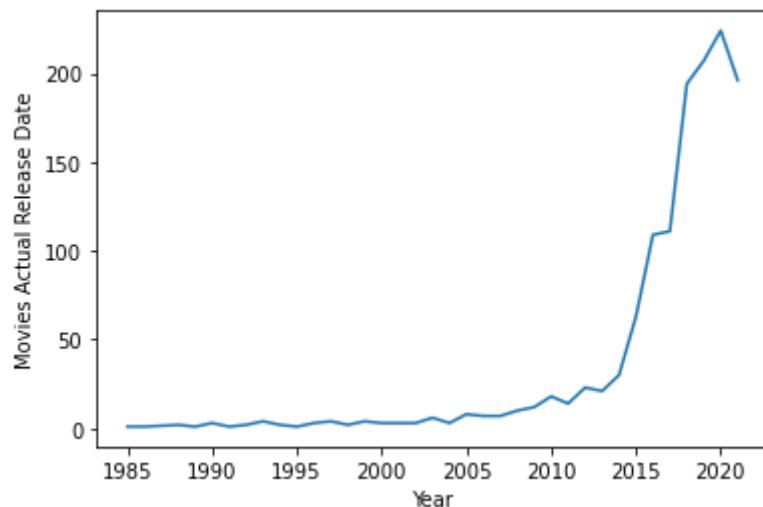


TV Shows are added in Netflix by a tremendous amount in July and September in USA Movies are added in Netflix in USA by a tremendous amount in first week/last month of current year and first month of next year

```
In [165...]: 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 [167...]: df_release_year=df_usa_shows[df_usa_shows['release_year']>=1980].groupby(['release_ye...  
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 [168...]: df_usa_movies.head()
```

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_
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	

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_
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 [170...]

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

```
df_usa_movies['Actor_Director_Combination'] = df_usa_movies.Actors.str.cat(df_usa_mo
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
df_usa_movies_subset.head()
```

Out[170...]

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



In [171...]

```
df_usa_shows['Actor_Director_Combination'] = df_usa_shows.Actors.str.cat(df_usa_sho
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 D
df_usa_shows_subset.head()
```

Out[171...]

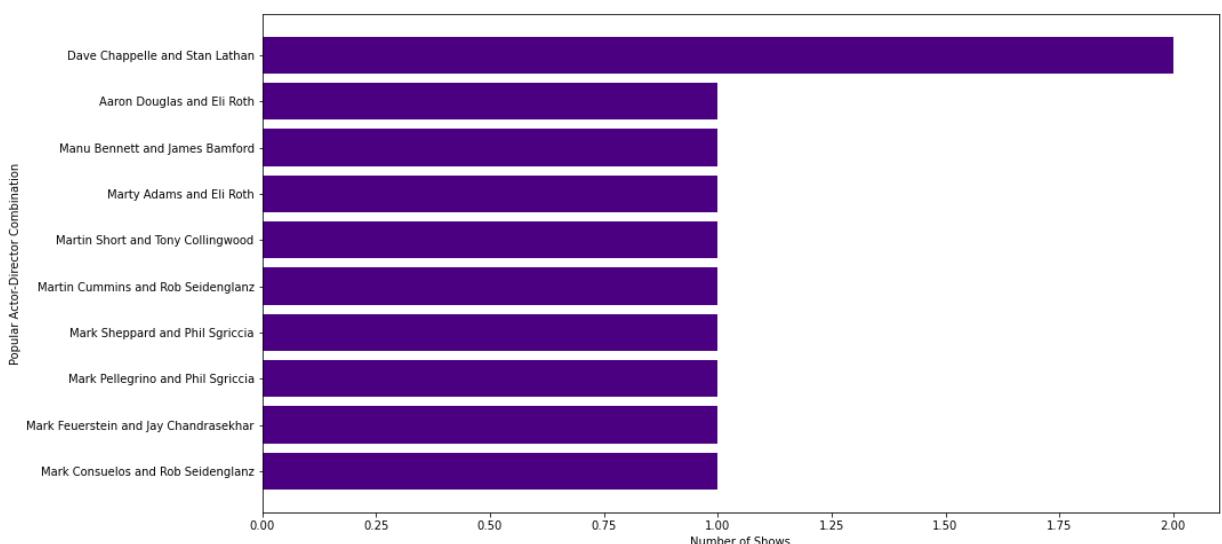
	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	ratio
111	Midnight Mass	Kate Siegel	Mike Flanagan	TV Dramas	United States	s6	TV Show	September 24, 2021	2021	1
112	Midnight Mass	Kate Siegel	Mike Flanagan	TV Horror	United States	s6	TV Show	September 24, 2021	2021	1

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	ratio
113	Midnight Mass	Kate Siegel	Mike Flanagan	TV Mysteries	United States	s6	TV Show	September 24, 2021	2021	1
114	Midnight Mass	Zach Gilford	Mike Flanagan	TV Dramas	United States	s6	TV Show	September 24, 2021	2021	1
115	Midnight Mass	Zach Gilford	Mike Flanagan	TV Horror	United States	s6	TV Show	September 24, 2021	2021	1



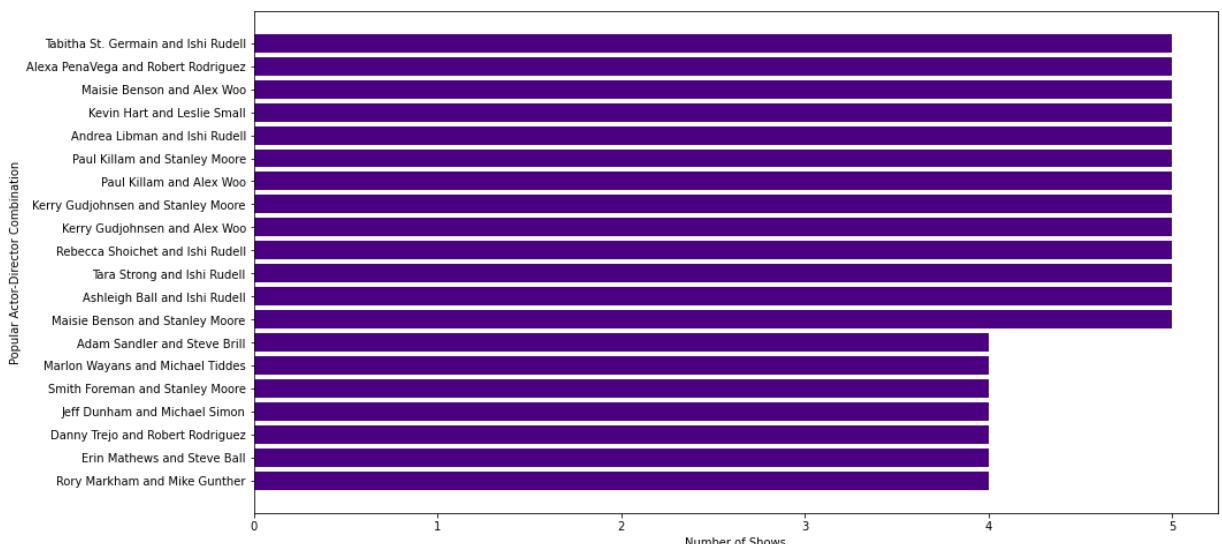
In [173...]

```
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 [175...]

```
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 Shows')  
plt.ylabel('Popular Actor-Director Combination')  
plt.show()
```



```
In [176... df_actors_directors[::-1]['Actor_Director_Combination'].values
```

```
Out[176... 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'

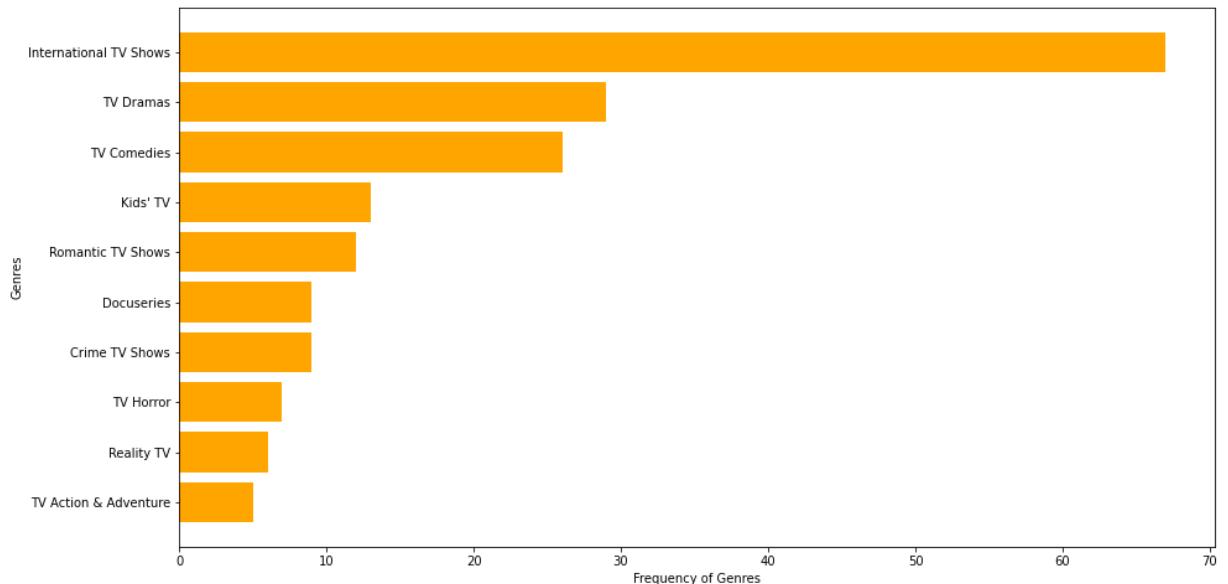
The Second Most Popular Actor Director Combination in Movies Across USA are:-

'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

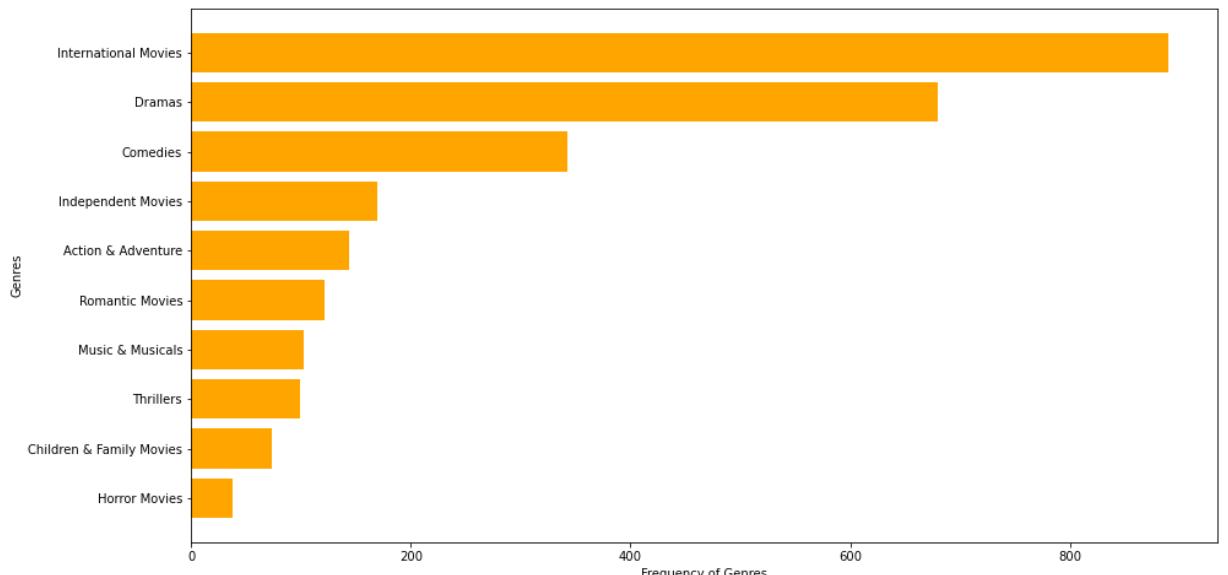
```
In [177... #Analyzing India for both shows and movies
df_india_shows=df_final1[df_final1['Country']=='India'][df_final1[df_final1['Country'
df_india_movies=df_final1[df_final1['Country']=='India'][df_final1[df_final1['Country'
df_final1['Country']=='India']]
```

```
In [178... df_genre=df_india_shows.groupby(['Genre']).agg({"title":"nunique"}).reset_index()
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()
```



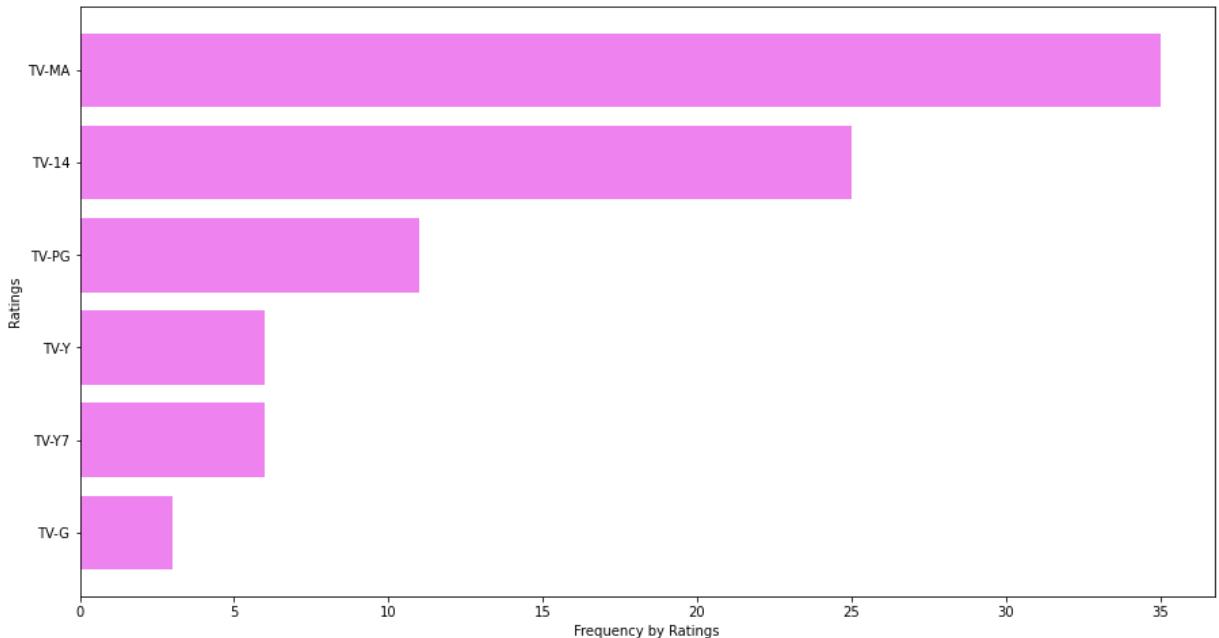
Dramas, Comedy, Kids' TV Shows and International TV Shows Genres are popular in TV Series in India

```
In [179...]: df_genre=df_india_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()
```



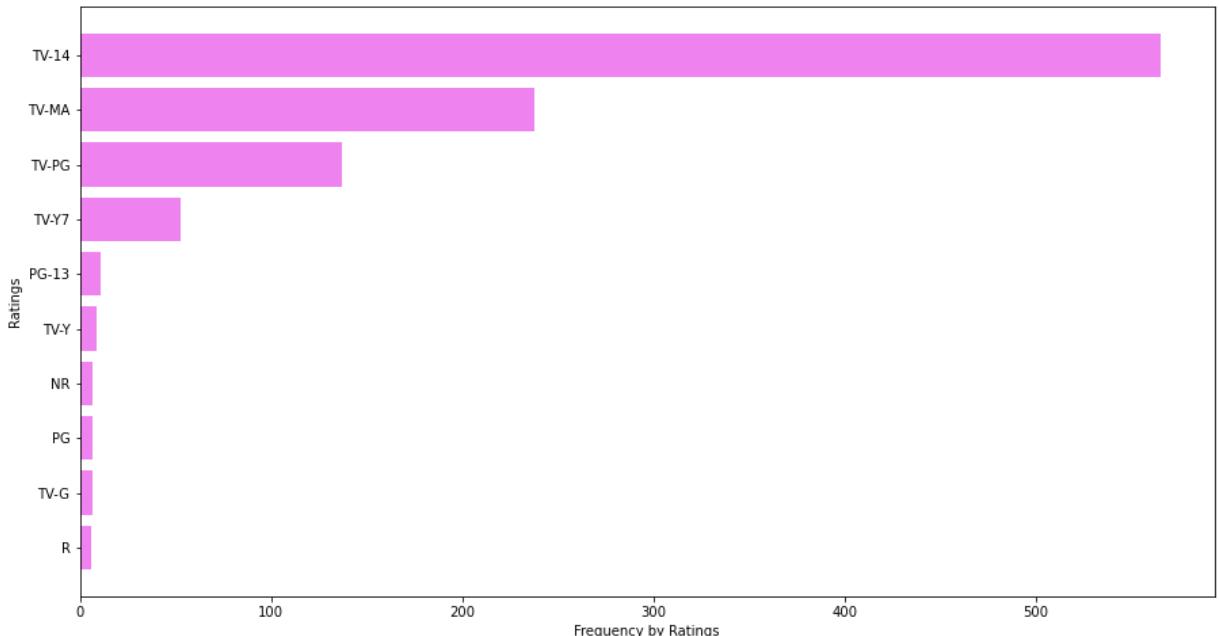
Dramas, Comedy, International movies, Independent movies, and Action & Adventure Genres are popular in TV Series in India

```
In [180...]: 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 [181...]

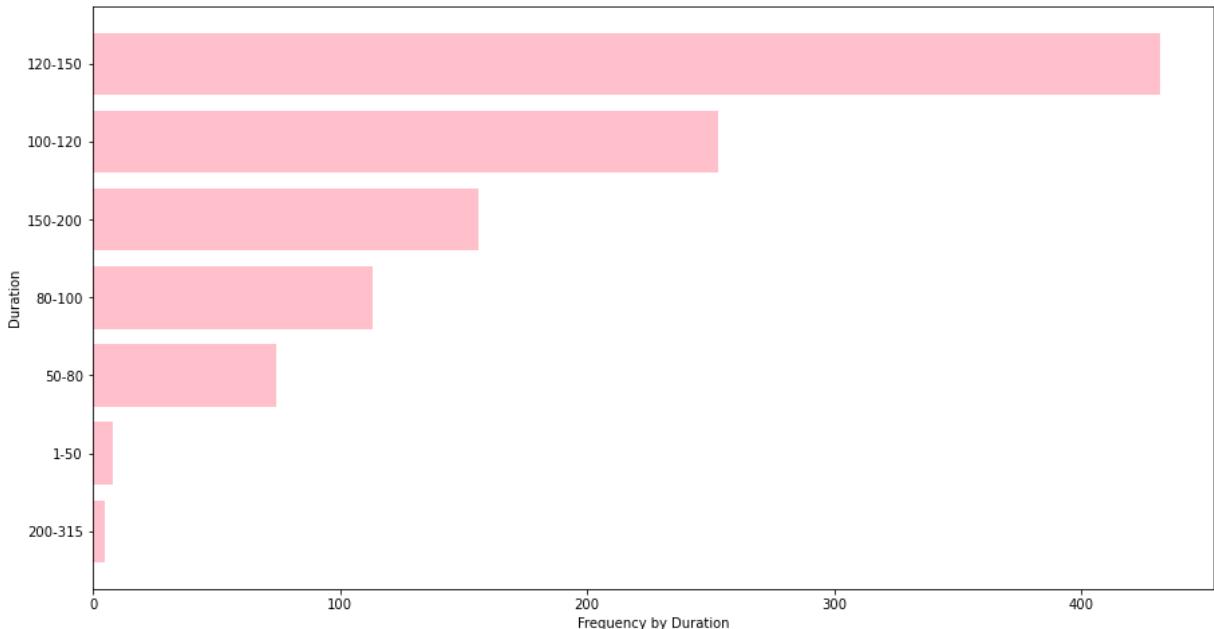
```
df_rating=df_india_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'], color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



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.

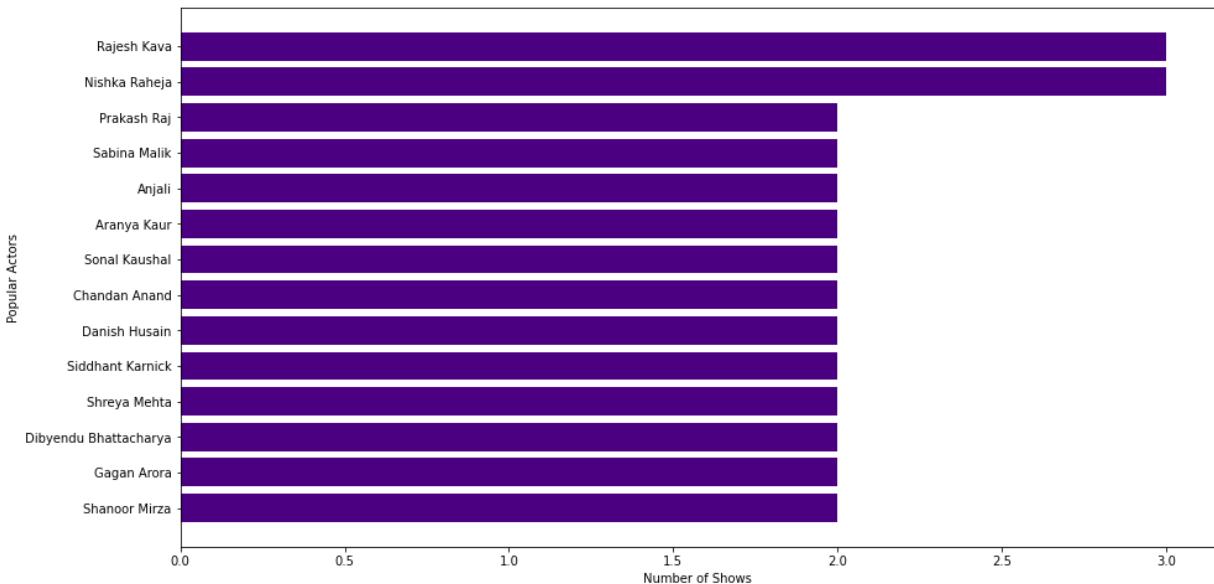
In [182...]

```
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 [183...]: 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()
```

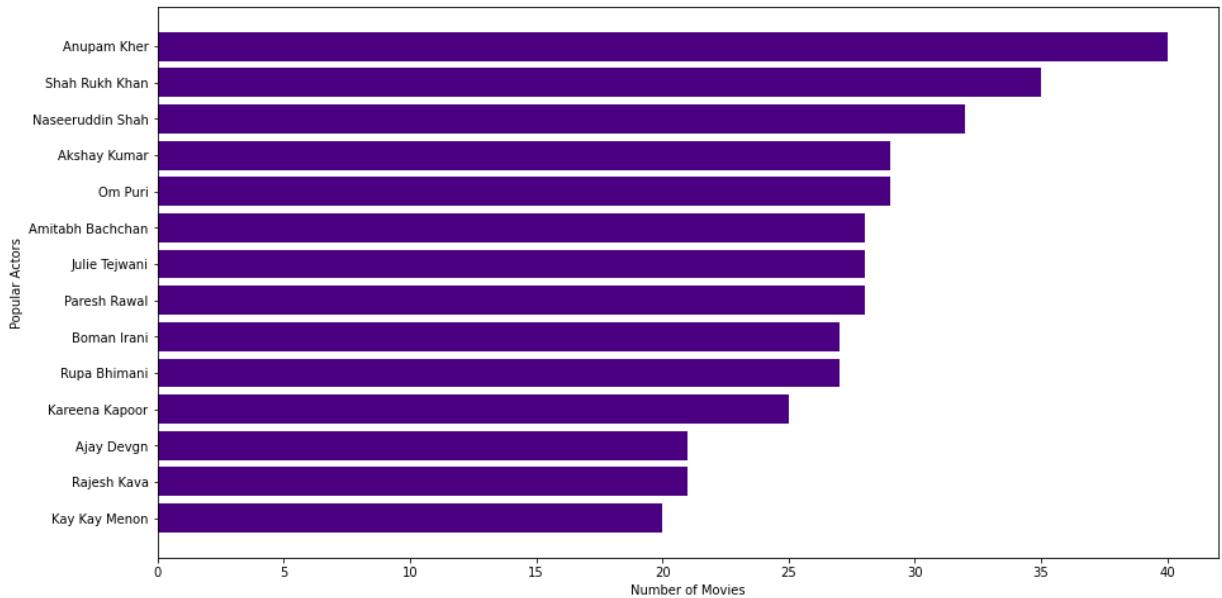


```
In [184...]: df_actors['Actors'].values
```

```
Out[184...]: array(['Rajesh Kava', 'Nishka Raheja', 'Prakash Raj', 'Sabina Malik',
       'Anjali', 'Aranya Kaur', 'Sonal Kaushal', 'Chandan Anand',
       'Danish Husain', 'Siddhant Karnick', 'Shreya Mehta',
       'Dibyendu Bhattacharya', 'Gagan Arora', 'Shanoor Mirza'],
      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 [185... 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()
```

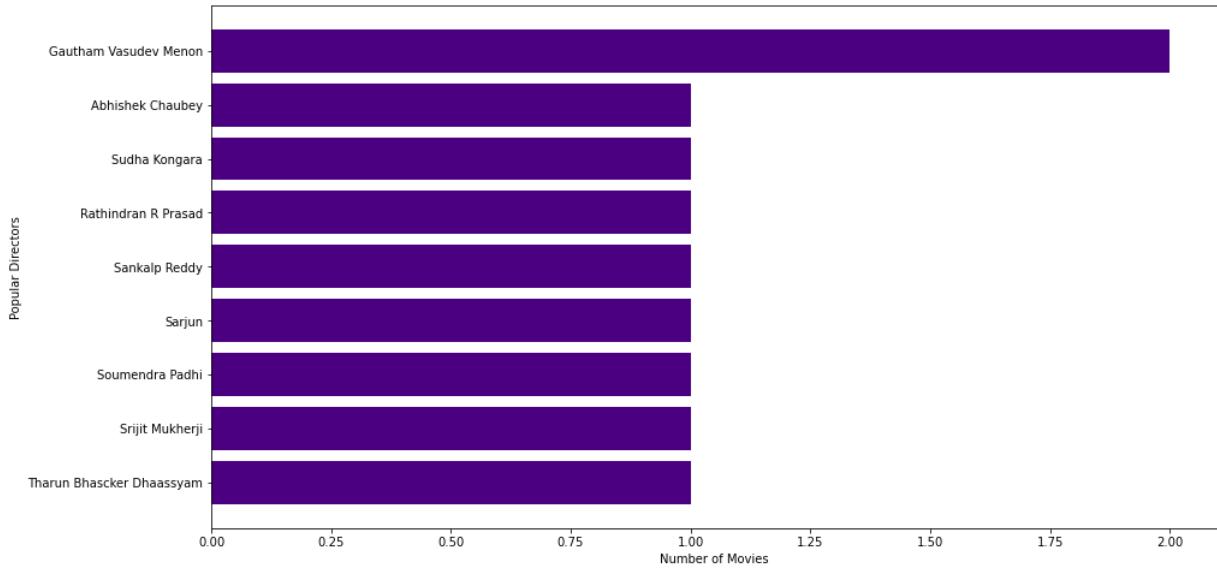


```
In [186... df_actors['Actors'].values
```

```
Out[186... 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 [188... 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()
```



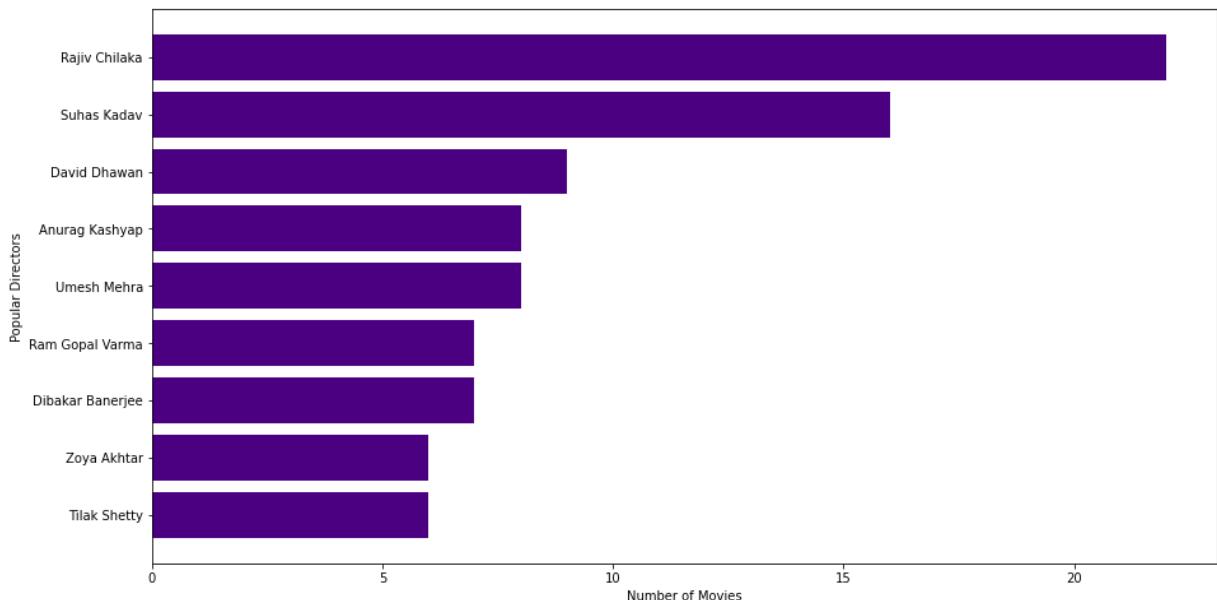
```
In [189]: df_directors['Directors'].values
```

```
Out[189]: array(['Gautham Vasudev Menon', 'Abhishek Chaubey', 'Sudha Kongara',
   'Rathindran R Prasad', 'Sankalp Reddy', 'Sarjun',
   'Soumendra Padhi', 'Srijiit 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', 'Srijiit Mukherji', 'Tharun Bhascker Dhaassyam'

```
In [190]: df_directors=df_india_movies.groupby(['Directors']).agg({"title":"nunique"}).reset_index()
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[:,-1]['Directors'], df_directors[:,-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Directors')
plt.show()
```



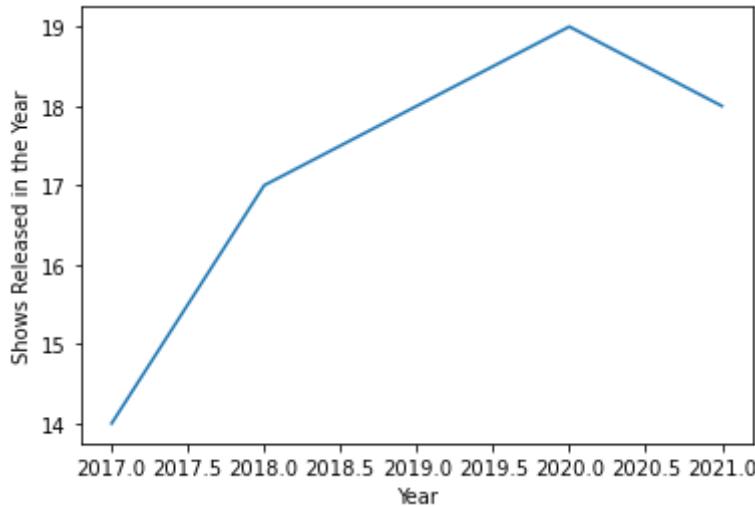
```
In [191]: df_directors['Directors'].values
```

```
Out[191... array(['Rajiv Chilaka', 'Suhas Kadav', 'David Dhawan', 'Anurag Kashyap',
   'Umesh Mehra', 'Ram Gopal Varma', 'Dibakar Banerjee',
   'Zoya Akhtar', 'Tilak Shetty'], 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'

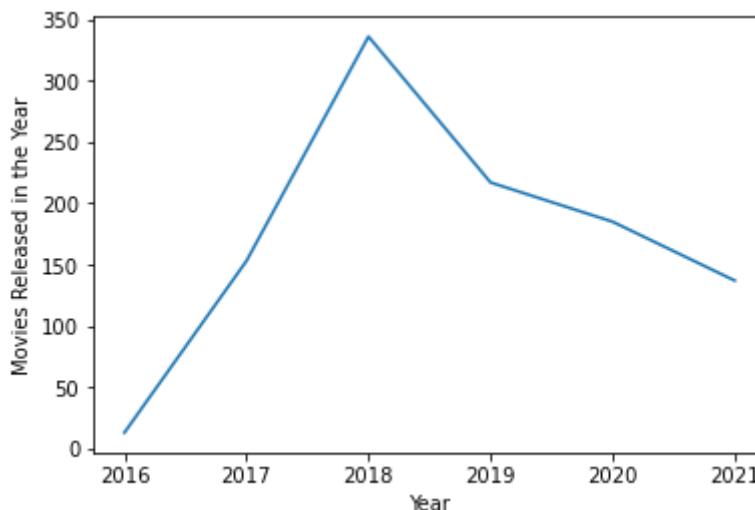
In [192...]

```
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 [193...]

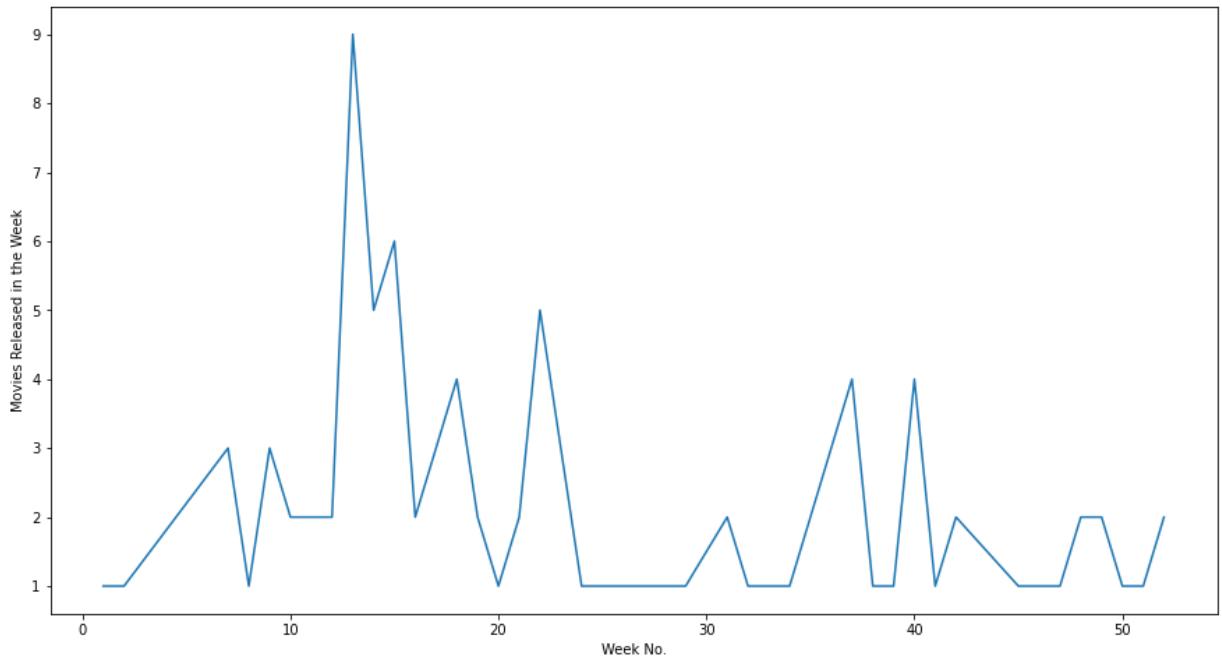
```
df_year=df_india_movies.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



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.

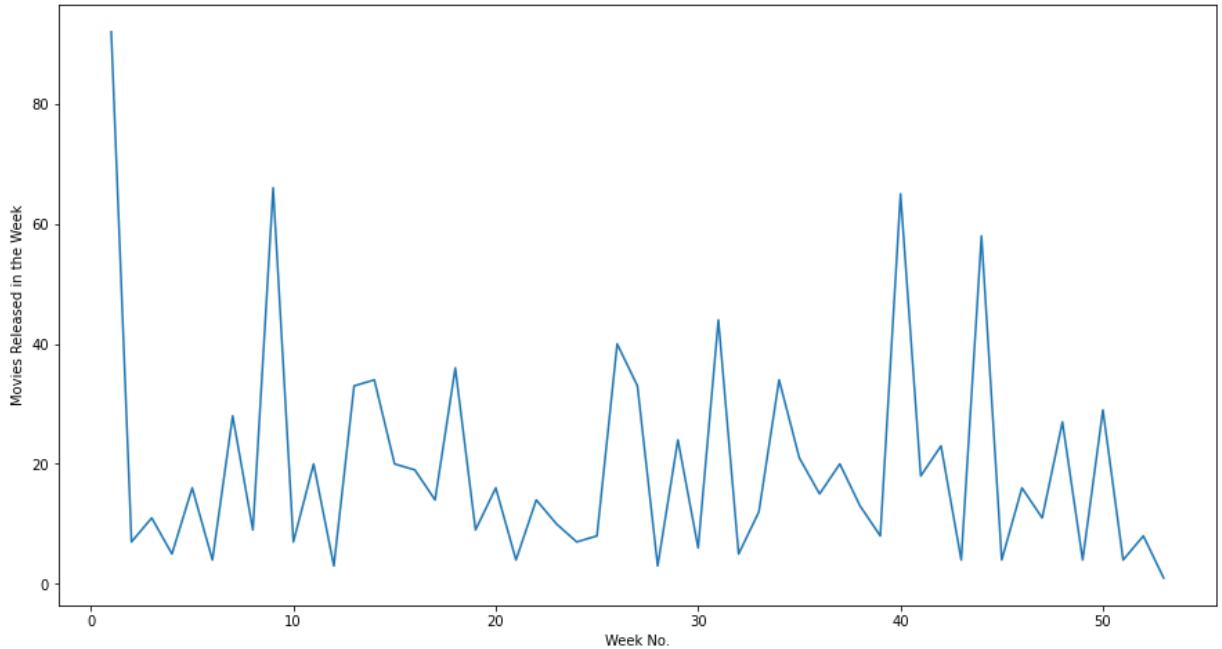
In [194...]

```
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()
```



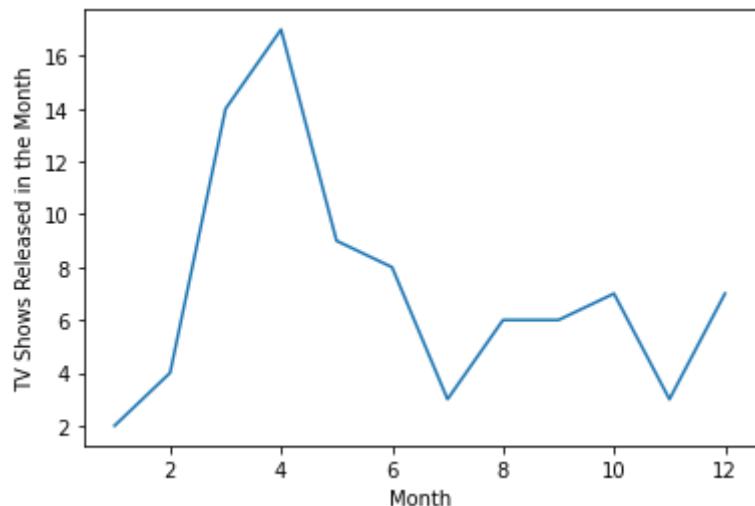
In [195...]

```
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()
```



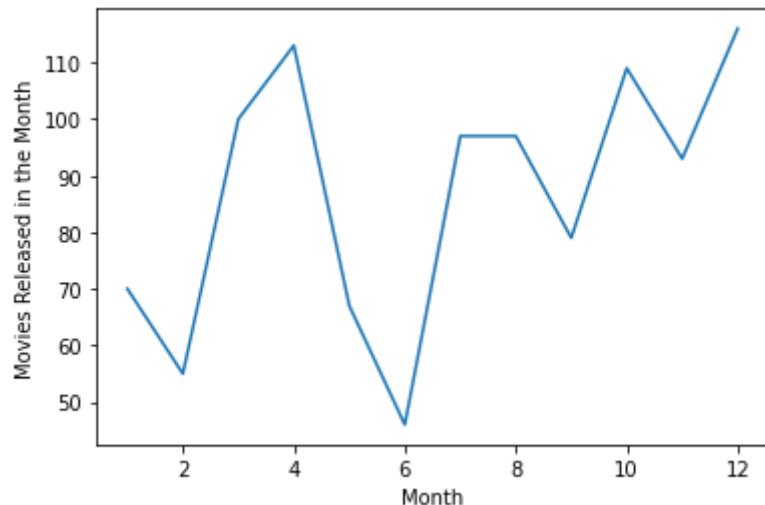
In [196...]

```
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 [197]:

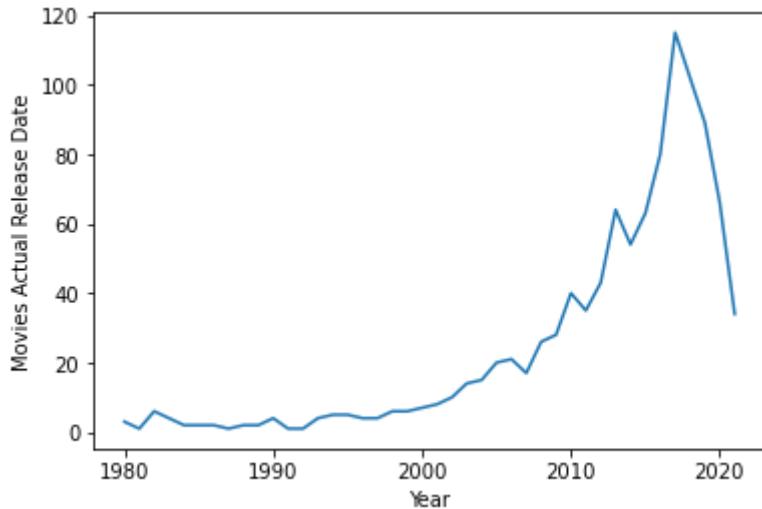
```
df_month=df_india_movies.groupby(['month_added']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_month, x='month_added', y='title')
plt.ylabel("Movies Released in the Month")
plt.xlabel("Month")
plt.show()
```



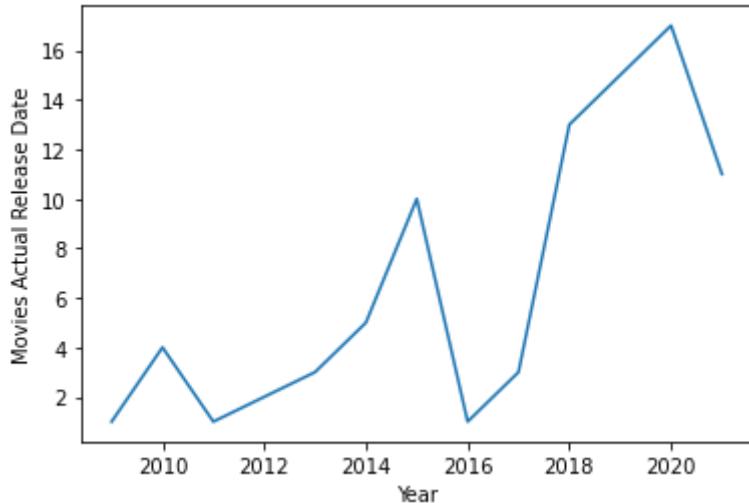
TV Shows are added in Netflix by a tremendous amount in April in India Movies are added in Netflix in India by a tremendous amount in first week/last month of current year and first month of next year

In [198]:

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



```
In [200...]: df_release_year=df_india_shows[df_india_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()
```



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

```
In [201...]: #Analysing a combination of actors and directors
df_india_movies['Actor_Director_Combination'] = df_india_movies.Actors.str.cat(df_in
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()
```

	title	Actors	Directors	Genre	Country	show_id	type	date_added
621	Avvai Shanmughi	Kamal Hassan	K.S. Ravikumar	Comedies	India	s23	Movie	September 21, 2021
622	Avvai Shanmughi	Kamal Hassan	K.S. Ravikumar	International Movies	India	s23	Movie	September 21, 2021
629	Avvai Shanmughi	Nassar	K.S. Ravikumar	Comedies	India	s23	Movie	September 21, 2021
630	Avvai Shanmughi	Nassar	K.S. Ravikumar	International Movies	India	s23	Movie	September 21, 2021

	title	Actors	Directors	Genre	Country	show_id	type	date_added
631	Avvai Shanmughi	S.P. Balasubrahmanyam	K.S. Ravikumar	Comedies	India	s23	Movie	September 21, 2021

◀ ▶

In [202...]

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

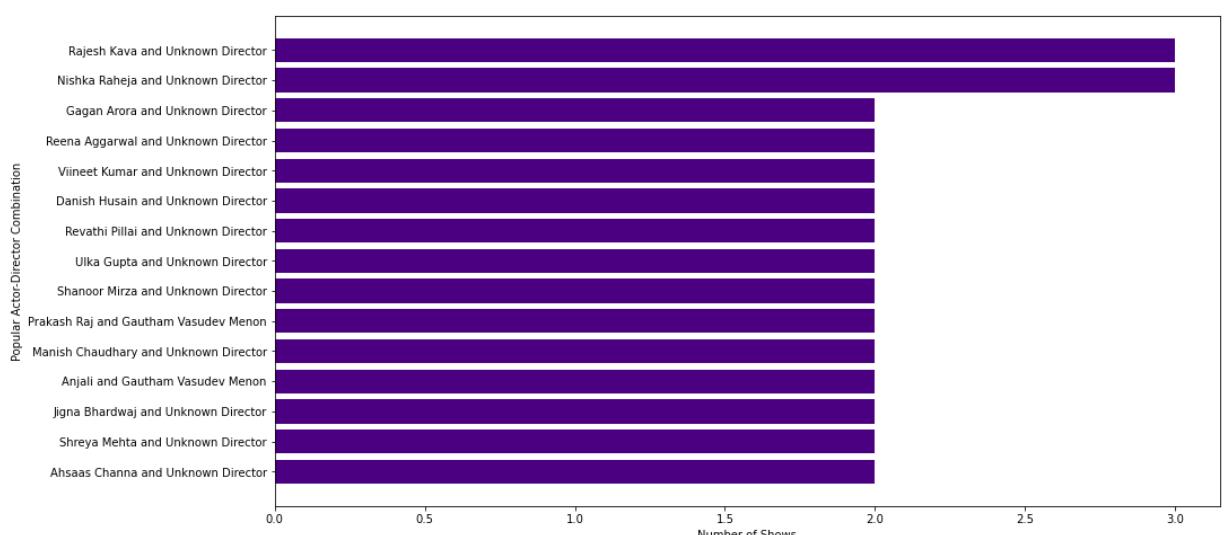
Out[202...]

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating
87	Kota Factory	Mayur More	Unknown Director	International TV Shows	India	s5	TV Show	September 24, 2021	2021	8.5
88	Kota Factory	Mayur More	Unknown Director	Romantic TV Shows	India	s5	TV Show	September 24, 2021	2021	8.5
89	Kota Factory	Mayur More	Unknown Director	TV Comedies	India	s5	TV Show	September 24, 2021	2021	8.5
90	Kota Factory	Jitendra Kumar	Unknown Director	International TV Shows	India	s5	TV Show	September 24, 2021	2021	8.5
91	Kota Factory	Jitendra Kumar	Unknown Director	Romantic TV Shows	India	s5	TV Show	September 24, 2021	2021	8.5

◀ ▶

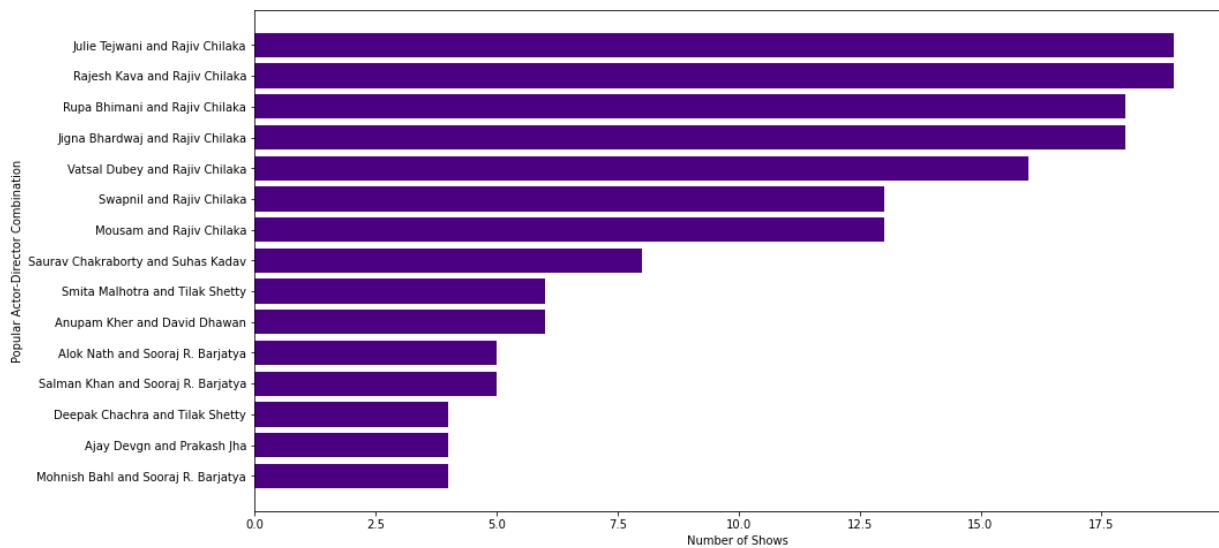
In [204...]

```
df_actors_directors=df_india_shows_subset.groupby(['Actor_Director_Combination']).agg({'show_id': 'count'})
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 [205...]

```
df_actors_directors=df_india_movies_subset.groupby(['Actor_Director_Combination']).agg({'show_id': 'count'})
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 [207]:

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

Out[207]:

		title	Actors	Directors	Genre	Country	show_id	type	date_added	release_yr
10058		Chhota Bheem - Neeli Pahaadi	Vatsal Dubey	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	2021
10059		Chhota Bheem - Neeli Pahaadi	Julie Tejwani	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	2021
10060		Chhota Bheem - Neeli Pahaadi	Rupa Bhimani	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	2021
10061		Chhota Bheem - Neeli Pahaadi	Jigna Bhardwaj	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	2021
10062		Chhota Bheem - Neeli Pahaadi	Rajesh Kava	Rajiv Chilaka	Children & Family Movies	India	s407	Movie	July 22, 2021	2021
...	
145810		Chhota Bheem Kungfu Dhamaka	Vaibhav Thakkar	Rajiv Chilaka	Children & Family Movies	India	s6465	Movie	August 15, 2019	2019
145812		Chhota Bheem Kungfu Dhamaka	Samriddhi Shuklaa	Rajiv Chilaka	Children & Family Movies	India	s6465	Movie	August 15, 2019	2019
145814		Chhota Bheem Kungfu Dhamaka	Aditya Raj Sharma	Rajiv Chilaka	Children & Family Movies	India	s6465	Movie	August 15, 2019	2019

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_yr
145816	Chhota Bheem Kungfu Dhamaka	Vinod Kulkarni	Rajiv Chilaka	Children & Family Movies	India	s6465	Movie	August 15, 2019	2019
150769	Dragonkala Ka Rahasya	Unknown Actor	Rajiv Chilaka	Children & Family Movies	India	s6646	Movie	June 18, 2019	2019

147 rows × 16 columns



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

```
In [208...]: df_actors_directors['Actor_Director_Combination'].values
```

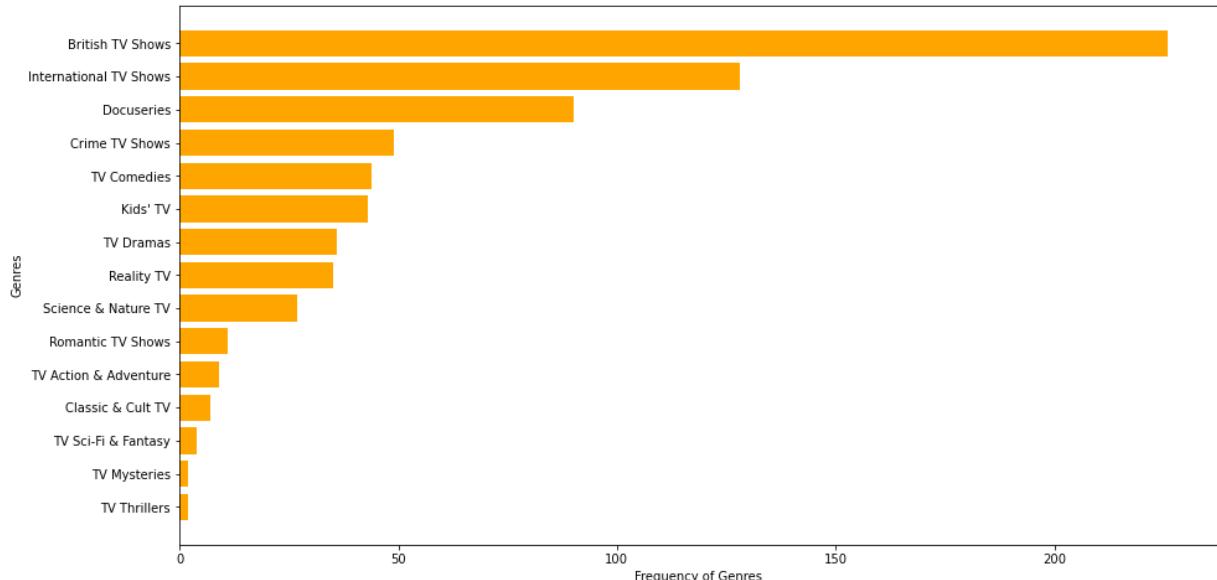
```
Out[208...]: array(['Julie Tejwani and Rajiv Chilaka', 'Rajesh Kava and Rajiv Chilaka', 'Rupa Bhimani and Rajiv Chilaka', 'Jigna Bhardwaj and Rajiv Chilaka', 'Vatsal Dubey and Rajiv Chilaka', 'Swapnil and Rajiv Chilaka', 'Mousam and Rajiv Chilaka', 'Saurav Chakraborty and Suhas Kadav', 'Smita Malhotra and Tilak Shetty', 'Anupam Kher and David Dhawan', 'Alok Nath and Sooraj R. Barjatya', 'Salman Khan and Sooraj R. Barjatya', 'Deepak Chachra and Tilak Shetty', 'Ajay Devgn and Prakash Jha', 'Mohnish Bahl and Sooraj R. Barjatya'], 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

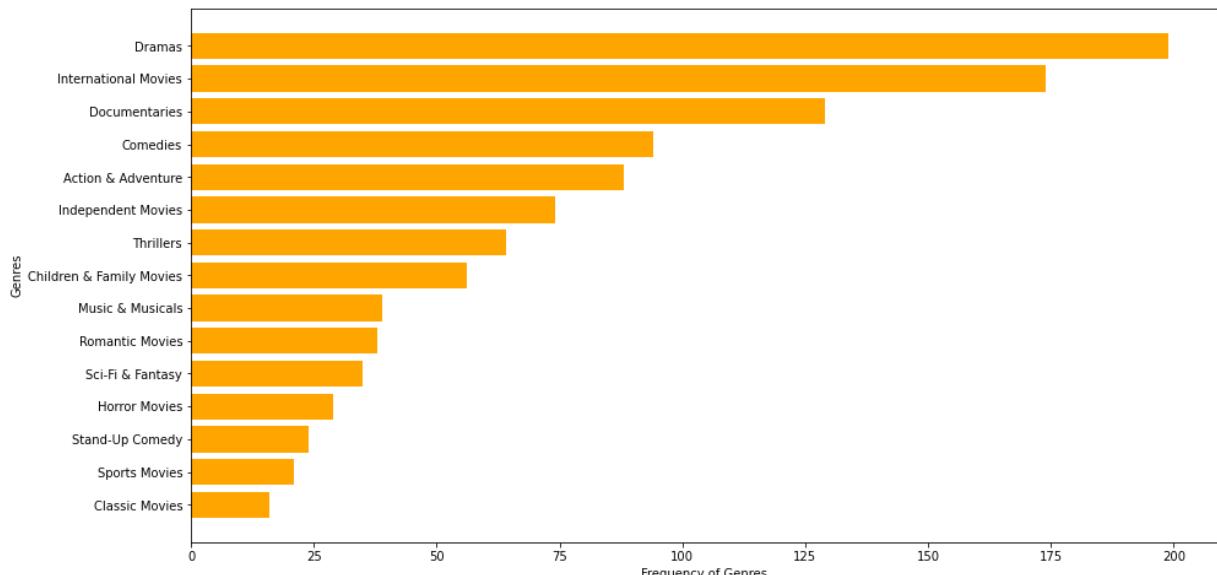
```
In [209...]: #Analyzing India for both shows and movies
df_uk_shows=df_final1[df_final1['Country']=='United Kingdom'][df_final1[df_final1['Country']=='United Kingdom'][df_final1['Country']=='United Kingdom']]
```

```
In [210...]: 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()
```



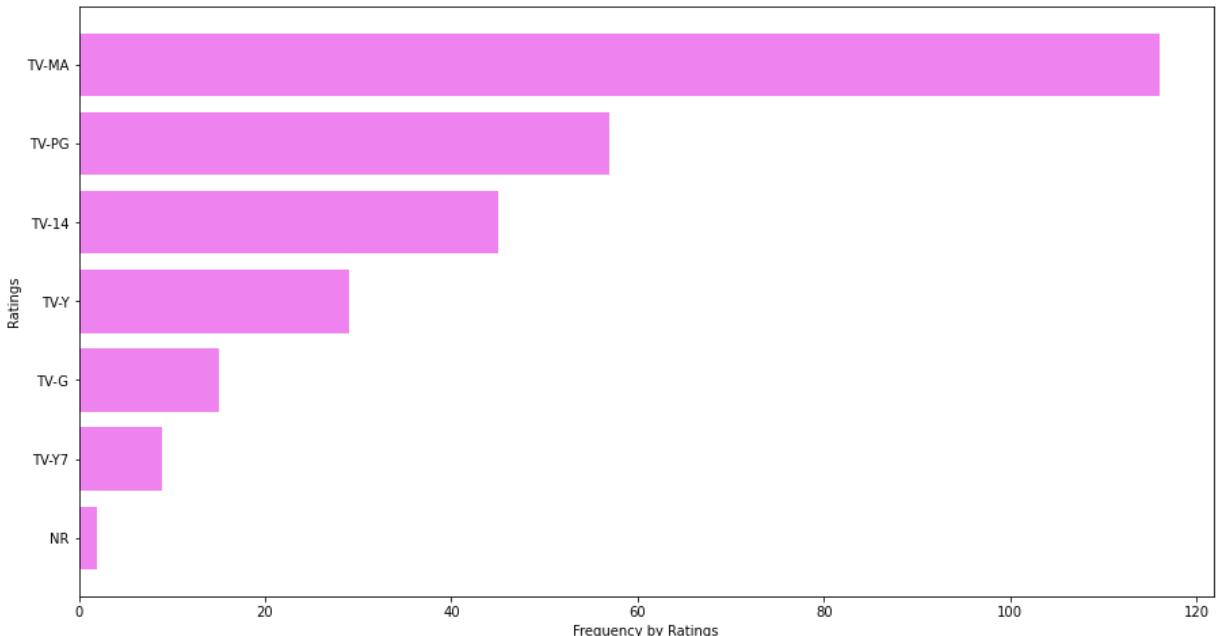
British TV Shows, International TV Shows, Docuseries, Crime, Comedy are widely watched Genres in TV Shows in UK

```
In [212...]: df_genre=df_uk_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```

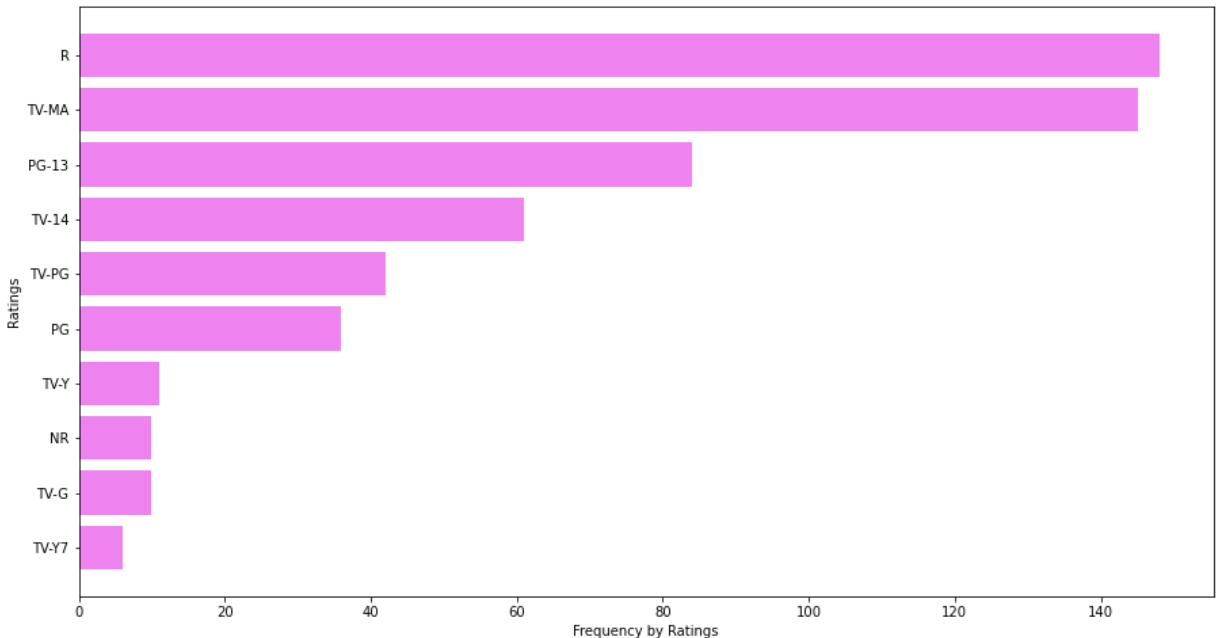


International Movies, Drama, Comedy, Indpeendent Movies and Action, Romance Genres in Movies are prevalent in UK

```
In [213...]: df_rating=df_uk_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```

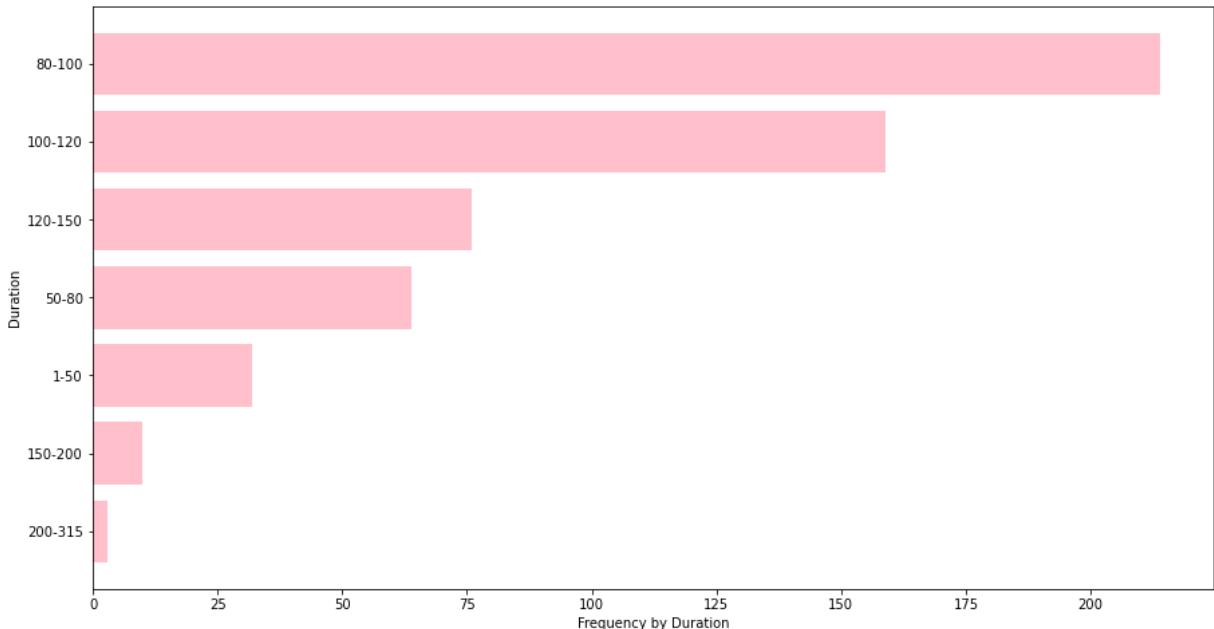


```
In [214...]: df_rating=df_uk_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'], color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



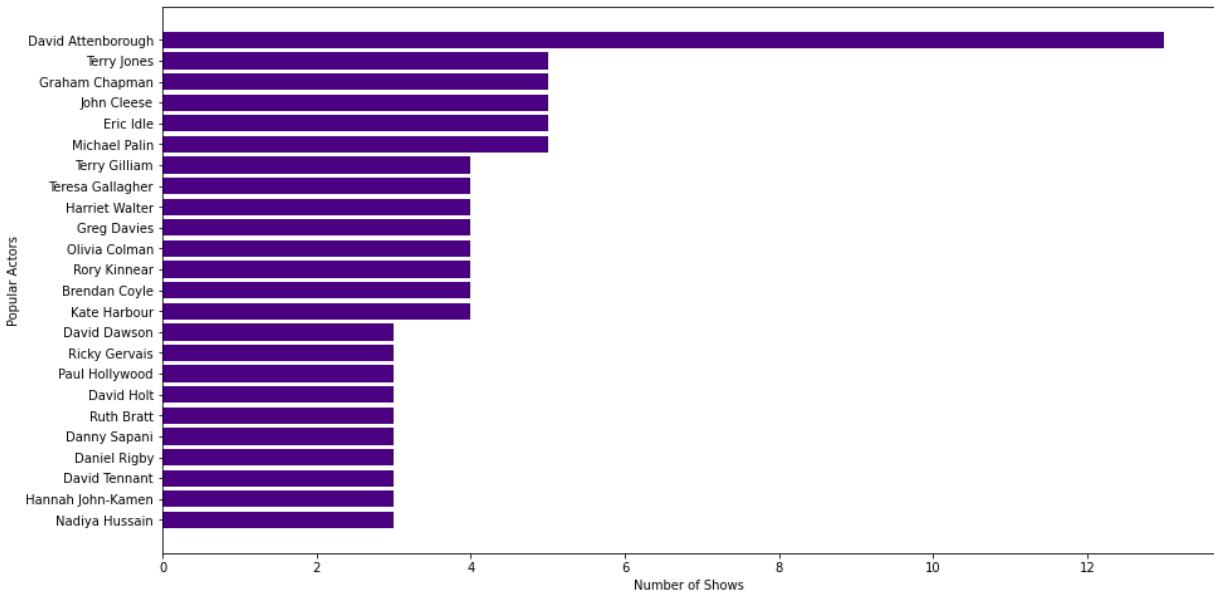
So it seems plausible to conclude that the popular ratings across Netflix includes Parental Guidance and Mature Audiences in TV Shows and R Rated+MA Rated in Movies in UK

```
In [215...]: 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 [217...]: df_actors=df_uk_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 Shows')
plt.ylabel('Popular Actors')
plt.show()
```



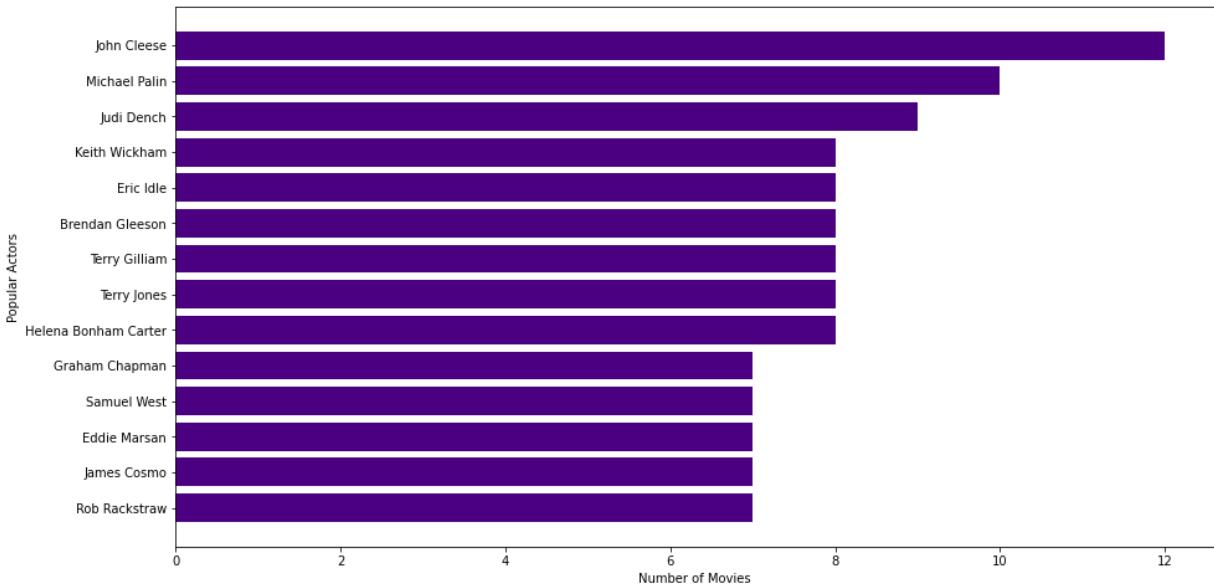
```
In [218...]: df_actors['Actors'].values
```

```
Out[218...]: array(['David Attenborough', 'Terry Jones', 'Graham Chapman',
       'John Cleese', 'Eric Idle', 'Michael Palin', 'Terry Gilliam',
       'Teresa Gallagher', 'Harriet Walter', 'Greg Davies',
       'Olivia Colman', 'Rory Kinnear', 'Brendan Coyle', 'Kate Harbour',
       'David Dawson', 'Ricky Gervais', 'Paul Hollywood', 'David Holt',
       'Ruth Bratt', 'Danny Sapani', 'Daniel Rigby', 'David Tennant',
       'Hannah John-Kamen', 'Nadiya Hussain'], 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 [219...]

```
df_actors=df_uk_movies.groupby(['Actors']).agg({"title":"nunique"}).reset_index().so
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 [220...]

df_actors['Actors'].values

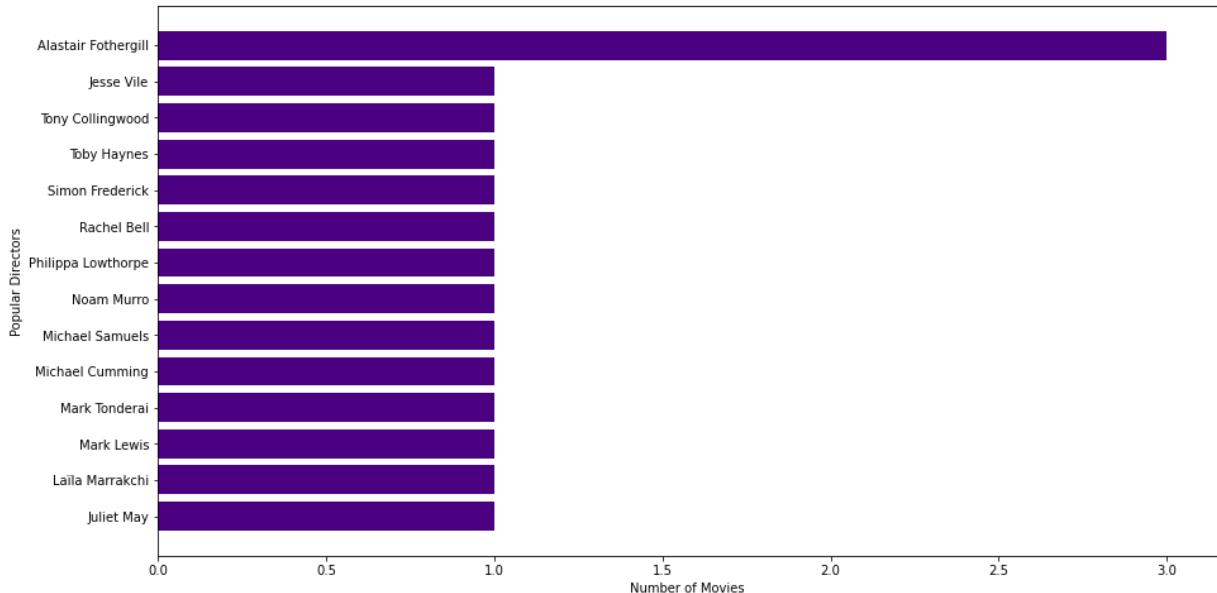
Out[220...]

```
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 [222...]

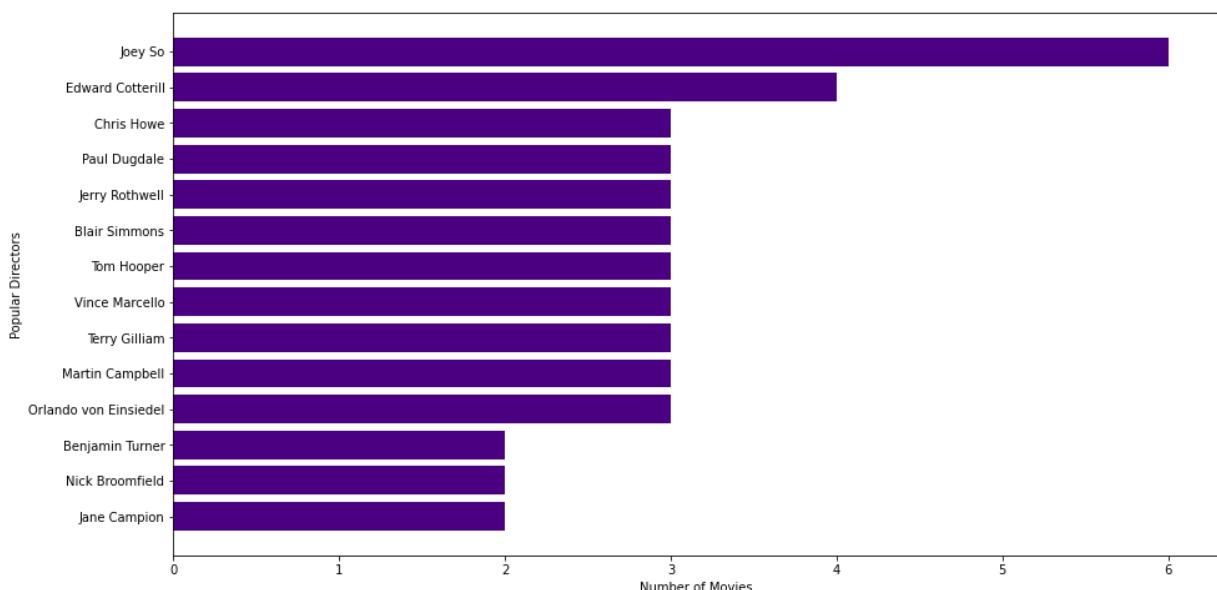
```
df_directors=df_uk_shows.groupby(['Directors']).agg({"title":"nunique"}).reset_index()
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[::-1]['Directors'], df_directors[::-1]['title'],color=['indigo'])
plt.xlabel('Number of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



```
In [223...]: df_directors['Directors'].values
```

```
Out[223...]: array(['Alastair Fothergill', 'Jesse Vile', 'Tony Collingwood',
       'Toby Haynes', 'Simon Frederick', 'Rachel Bell',
       'Philippa Lowthorpe', 'Noam Murro', 'Michael Samuels',
       'Michael Cumming', 'Mark Tonderai', 'Mark Lewis',
       'Laila Marrakchi', 'Juliet May'], dtype=object)
```

```
In [224...]: 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()
```



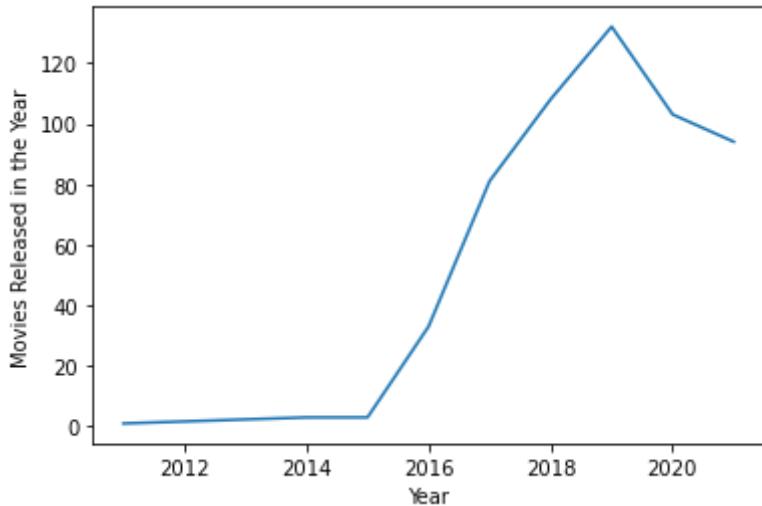
```
In [225...]: df_directors['Directors'].values
```

```
Out[225...]: array(['Joey So', 'Edward Cotterill', 'Chris Howe', 'Paul Dugdale',
       'Jerry Rothwell', 'Blair Simmons', 'Tom Hooper', 'Vince Marcello',
       'Terry Gilliam', 'Martin Campbell', 'Orlando von Einsiedel',
       'Benjamin Turner', 'Nick Broomfield', 'Jane Campion'], dtype=object)
```

Popular directors across movies in UK:- 'Joey So', 'Edward Cotterill'

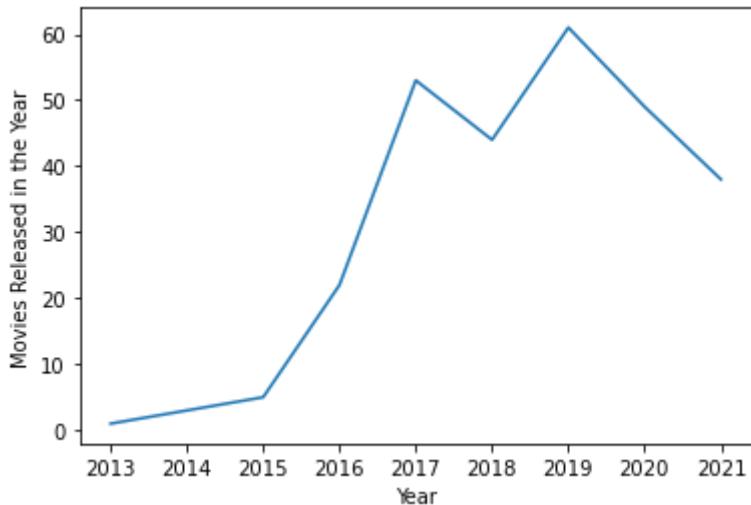
In [226...]

```
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 [227...]

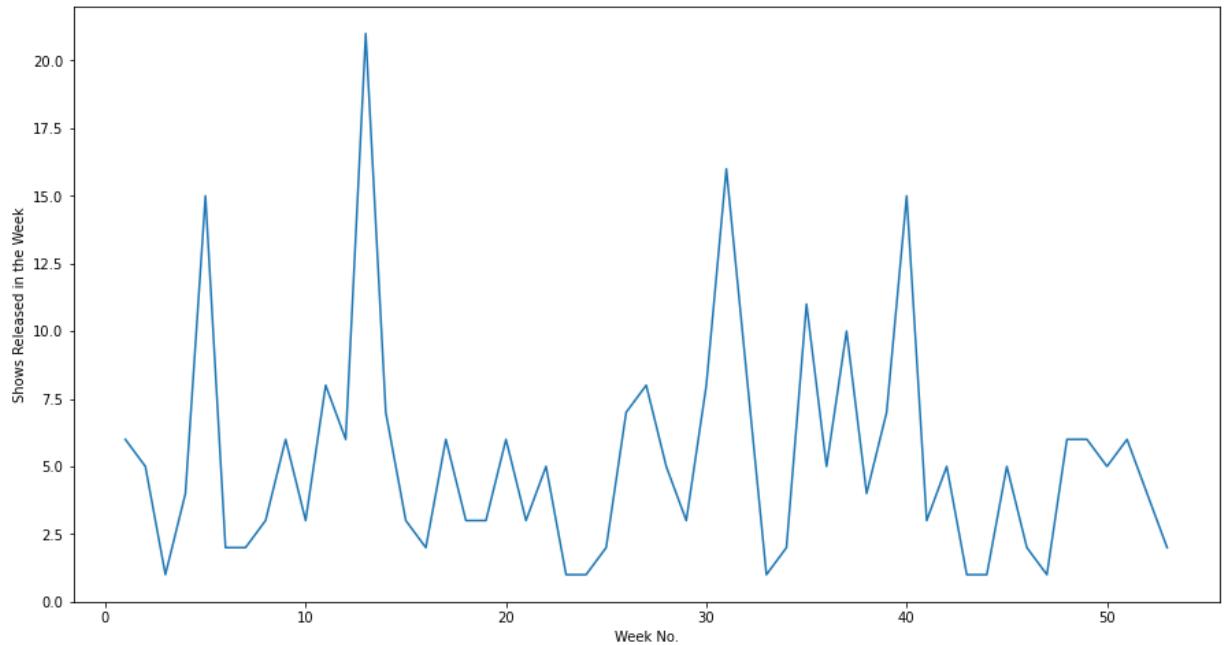
```
df_year=df_uk_shows.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



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.

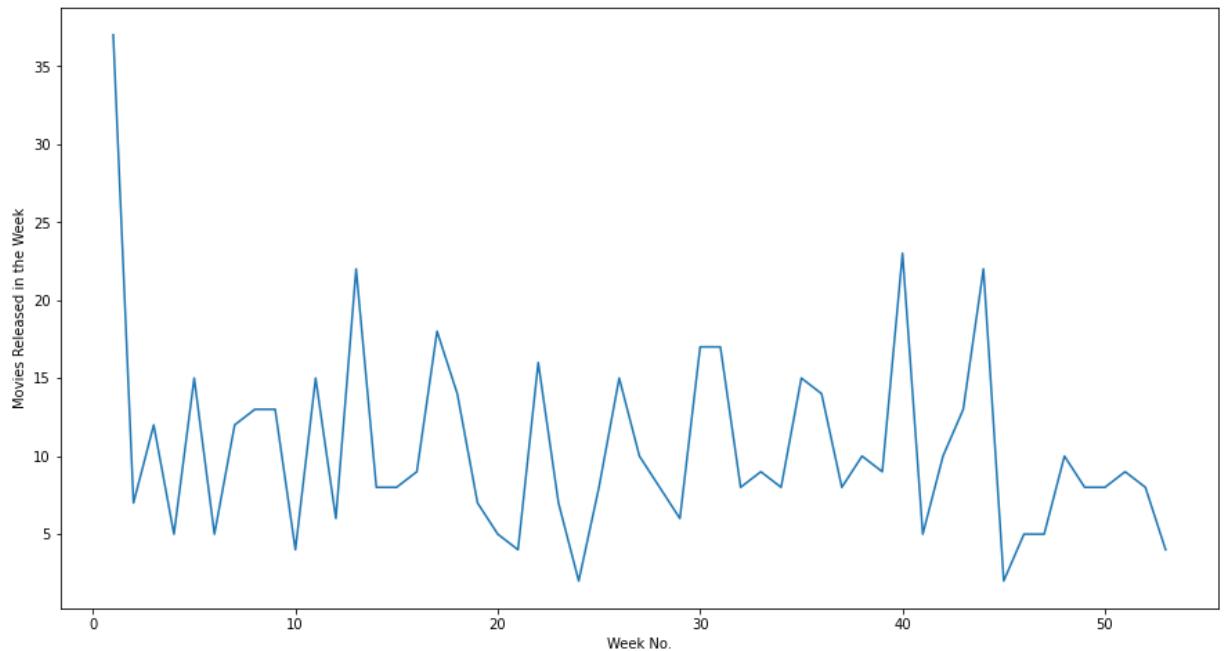
In [228...]

```
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()
```



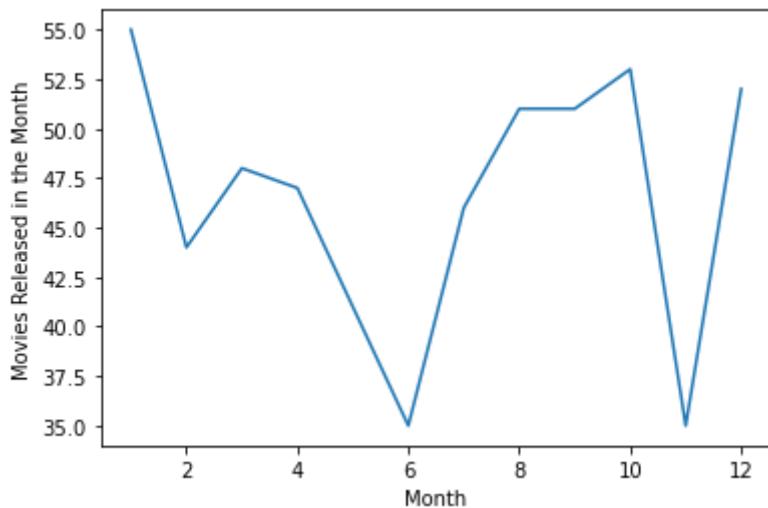
In [229...]

```
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()
```



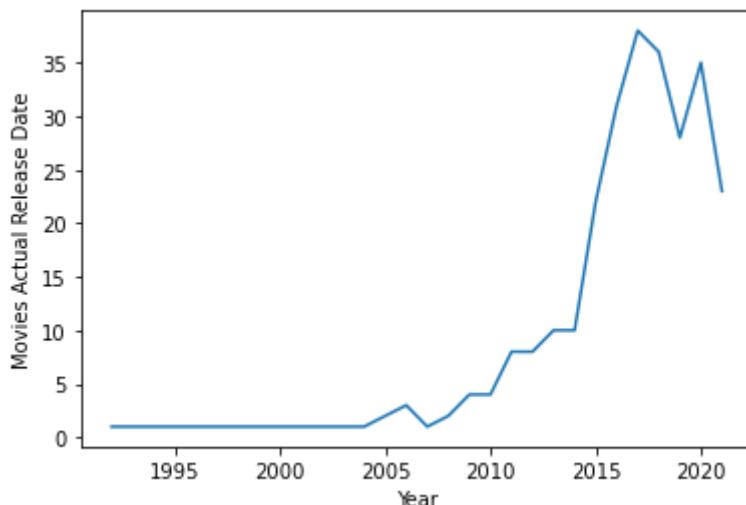
In [230...]

```
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()
```

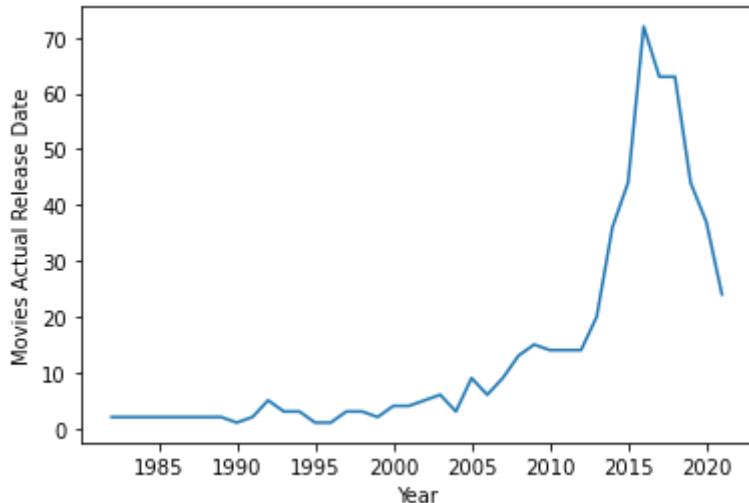


TV Shows are added in Netflix by a tremendous amount in March in UK Movies are added in Netflix in India by a tremendous amount in first week/last month of current year and first month of next year

```
In [233...]: 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 [235...]: df_release_year=df_uk_movies[df_uk_movies['release_year']>=1980].groupby(['release_yea
sns.lineplot(data=df_release_year, x='release_year', y='title')
plt.ylabel("Movies Actual Release Date")
plt.xlabel("Year")
plt.show()
```



In [238...]

```
#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[238...]

	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	19
188	Sankofa	Kofi Ghanaba	Haile Gerima	Independent Movies	United Kingdom	s8	Movie	September 24, 2021	19
194	Sankofa	Kofi Ghanaba	Haile Gerima	International Movies	United Kingdom	s8	Movie	September 24, 2021	19
200	Sankofa	Oyafunmike Ogunlano	Haile Gerima	Dramas	United Kingdom	s8	Movie	September 24, 2021	19
206	Sankofa	Oyafunmike Ogunlano	Haile Gerima	Independent Movies	United Kingdom	s8	Movie	September 24, 2021	19



In [239...]

```
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[239...]

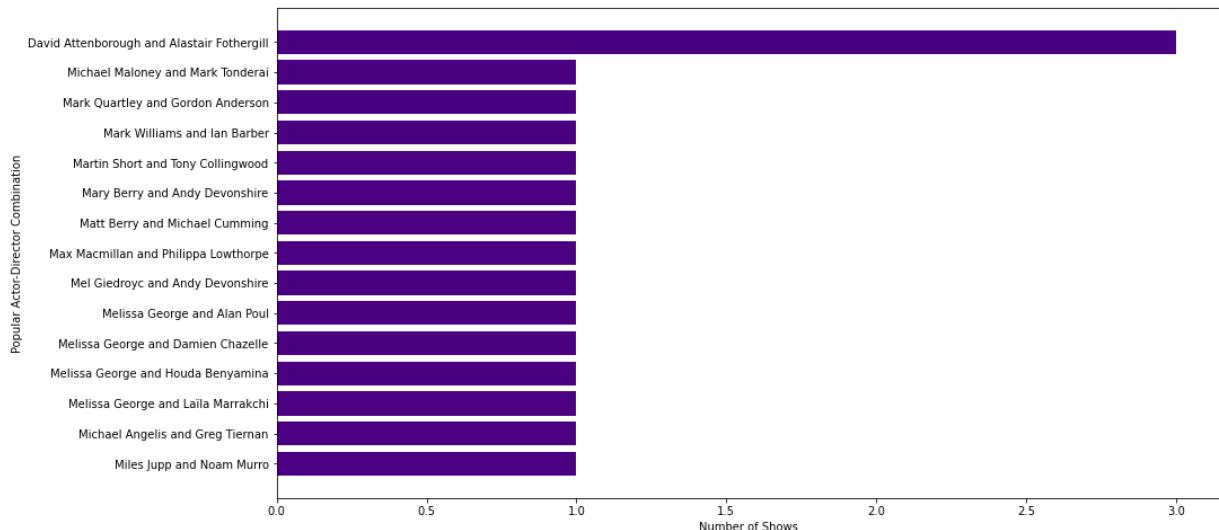
	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating
323	The Great British Baking Show	Mel Giedroyc	Andy Devonshire	British TV Shows	United Kingdom	s9	TV Show	September 24, 2021	2021	TV-1
324	The Great British Baking Show	Mel Giedroyc	Andy Devonshire	Reality TV	United Kingdom	s9	TV Show	September 24, 2021	2021	TV-1

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating
325	The Great British Baking Show	Sue Perkins	Andy Devonshire	British TV Shows	United Kingdom	s9	TV Show	September 24, 2021	2021	TV-1
326	The Great British Baking Show	Sue Perkins	Andy Devonshire	Reality TV	United Kingdom	s9	TV Show	September 24, 2021	2021	TV-1
327	The Great British Baking Show	Mary Berry	Andy Devonshire	British TV Shows	United Kingdom	s9	TV Show	September 24, 2021	2021	TV-1



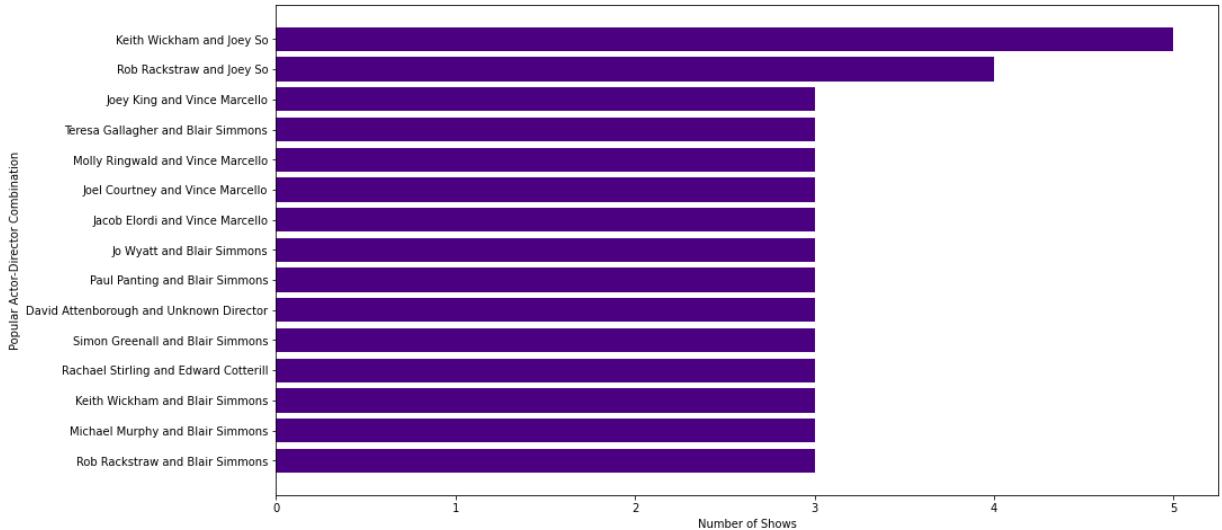
In [244...]

```
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 [245...]

```
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 Shows')
    plt.ylabel('Popular Actor-Director Combination')
    plt.show()
```



In [246...]: df_actors_directors['Actor_Director_Combination'].values

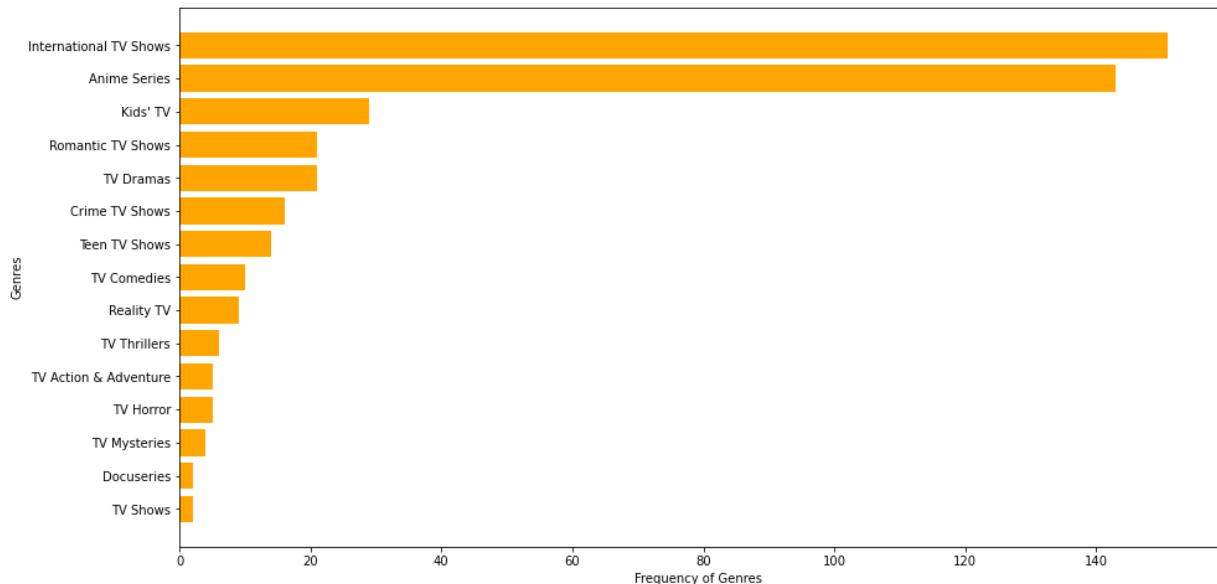
Out[246...]: array(['Keith Wickham and Joey So', 'Rob Rackstraw and Joey So', 'Joey King and Vince Marcello', 'Teresa Gallagher and Blair Simmons', 'Molly Ringwald and Vince Marcello', 'Joel Courtney and Vince Marcello', 'Jacob Elordi and Vince Marcello', 'Jo Wyatt and Blair Simmons', 'Paul Panting and Blair Simmons', 'David Attenborough and Unknown Director', 'Simon Greenall and Blair Simmons', 'Rachael Stirling and Edward Cotterill', 'Keith Wickham and Blair Simmons', 'Michael Murphy and Blair Simmons', 'Rob Rackstraw and Blair Simmons'], 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

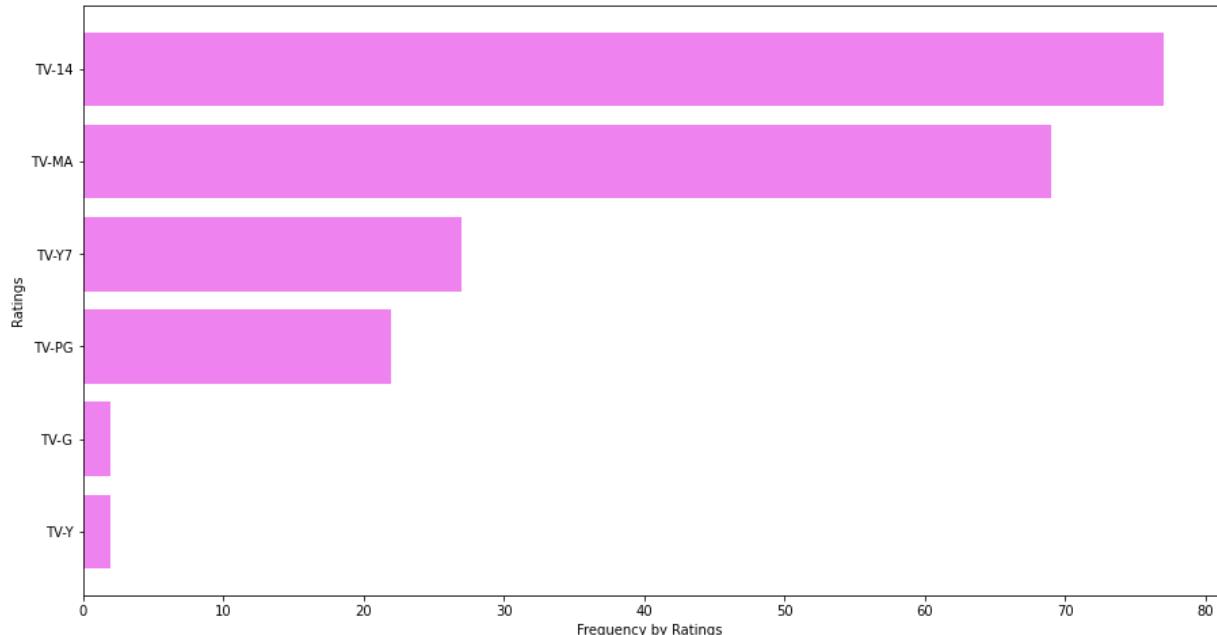
In [247...]: #Analyzing India for both shows and movies
df_japan_shows=df_final1[df_final1['Country']=='Japan'][df_final1['Country']]

In [249...]: 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['Genre'], df_genre['nunique'], color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()



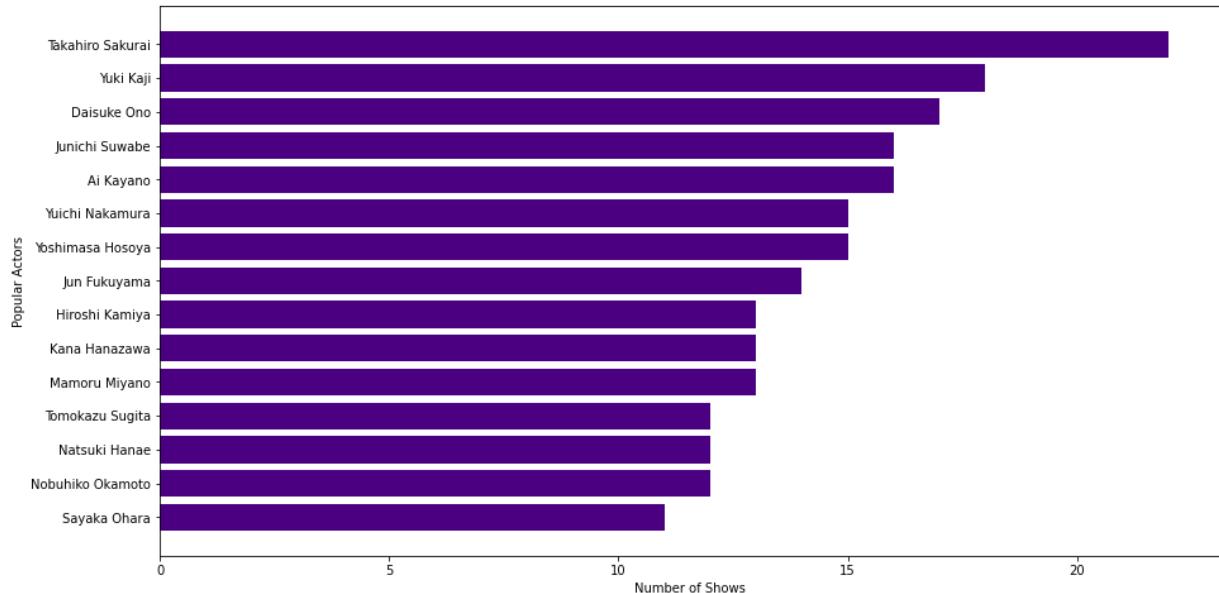
International TV Shows and Anime Genres are popular in TV Shows in Japan

```
In [250...]: df_rating=df_japan_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()
```



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

```
In [252...]: 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()
```

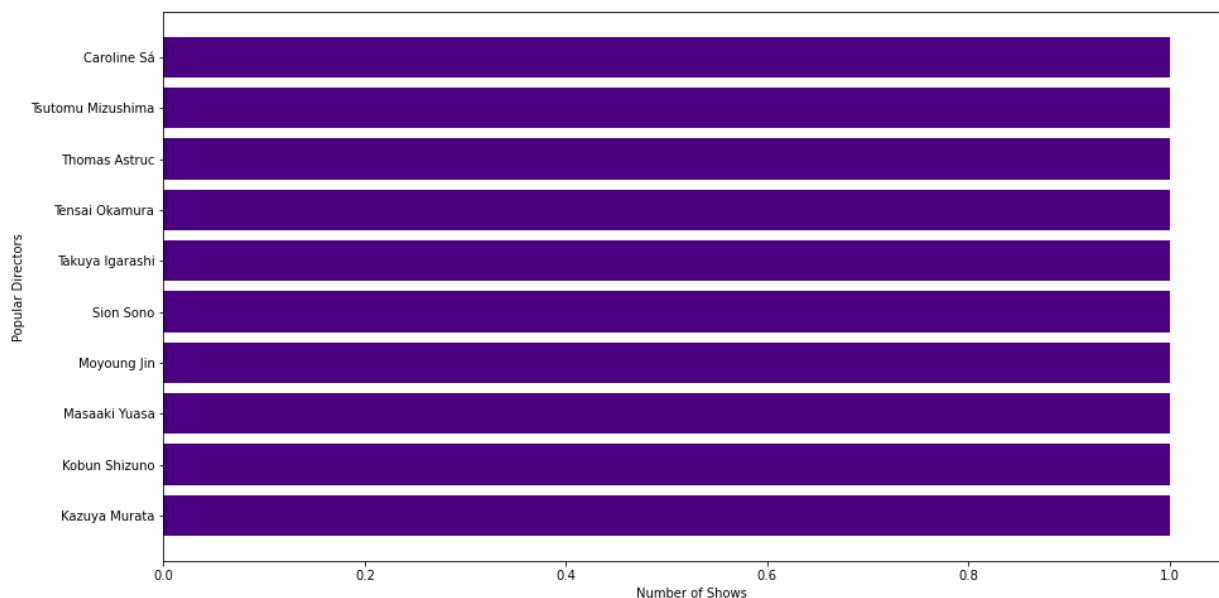


```
In [253...]: df_actors['Actors'].values
```

```
Out[253...]: array(['Takahiro Sakurai', 'Yuki Kaji', 'Daisuke Ono', 'Junichi Suwabe',
       'Ai Kayano', 'Yuichi Nakamura', 'Yoshimasa Hosoya', 'Jun Fukuyama',
       'Hiroshi Kamiya', 'Kana Hanazawa', 'Mamoru Miyano',
       'Tomokazu Sugita', 'Natsuki Hanae', 'Nobuhiko Okamoto',
       'Sayaka Ohara'], dtype=object)
```

Popular Actors in TV Shows in Japan are:- 'Takahiro Sakurai', 'Yuki Kaji', 'Daisuke Ono', 'Junichi Suwabe', 'Ai Kayano',

```
In [255...]: 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 [256...]: df_directors['Directors'].values
```

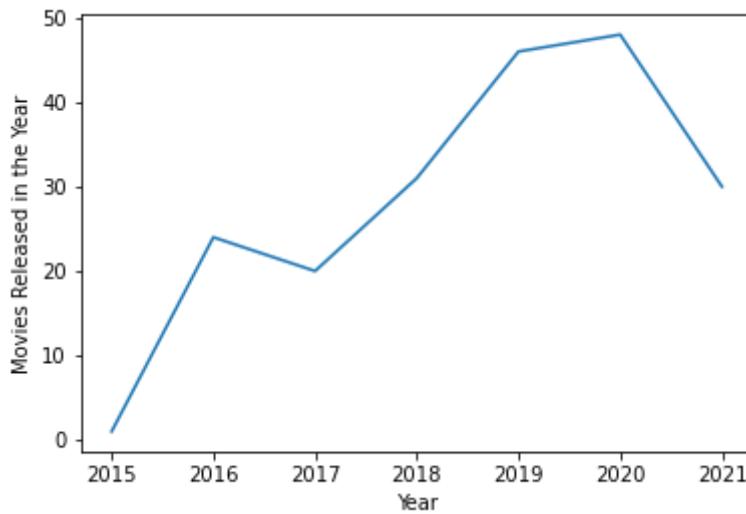
```
Out[256...]: array(['Caroline Sá', 'Tsutomu Mizushima', 'Thomas Astruc',
```

```
'Tensai Okamura', 'Takuya Igarashi', 'Sion Sono', 'Moyoung Jin',
'Masaaki Yuasa', 'Kobun Shizuno', 'Kazuya Murata'], dtype=object)
```

All Directors are one time directors only

In [258...]

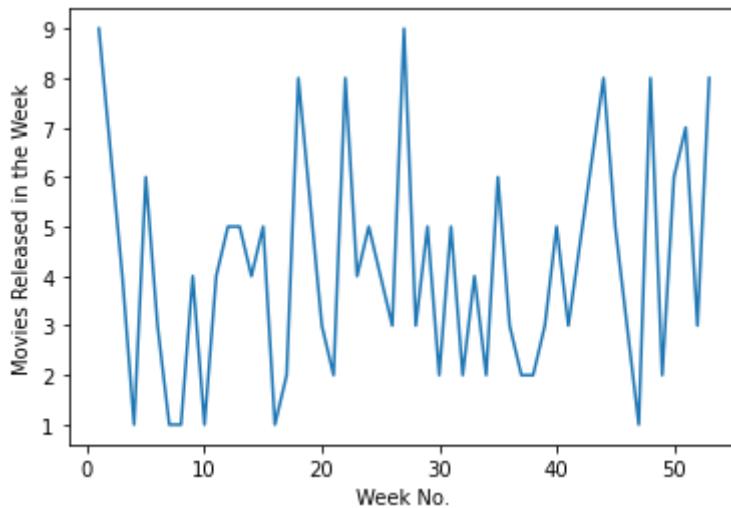
```
df_year=df_japan_shows.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



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

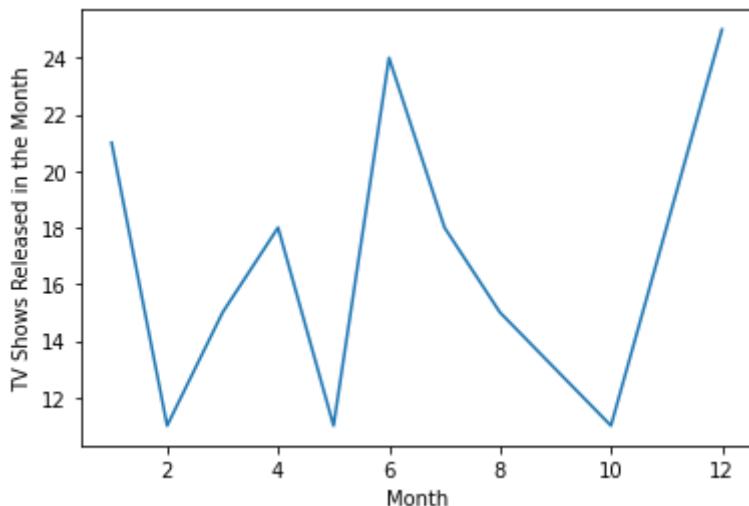
In [260...]

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



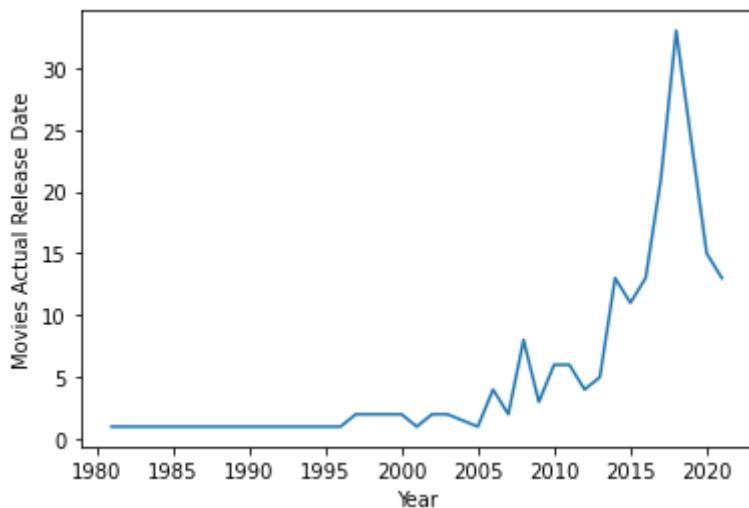
In [261...]

```
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 [262...]: 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()
```

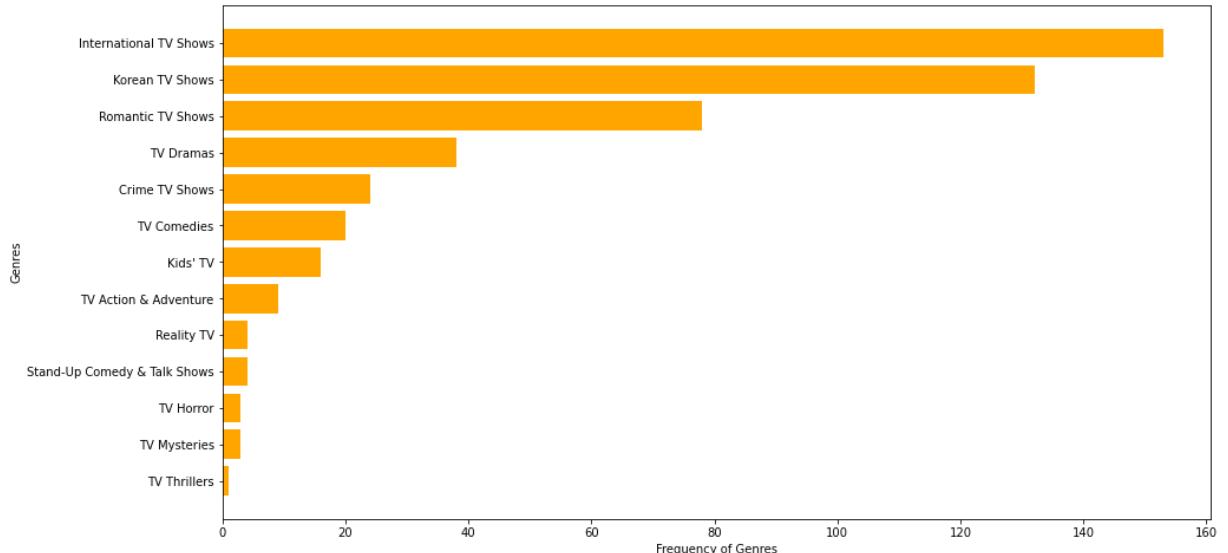


Reduction in TV Shows after 2019 in Japan

Univariate Analysis separately for shows in South Korea

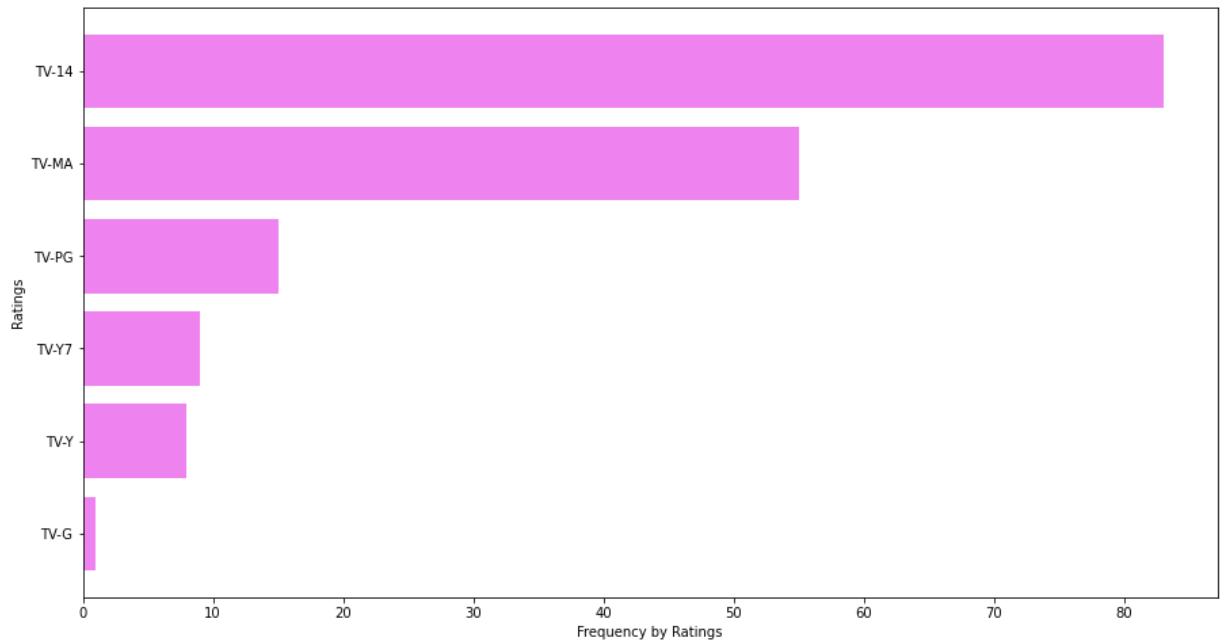
```
In [263...]: #Analyzing soth Korea for both shows and movies
df_sk_shows=df_final1[df_final1['Country']=='South Korea'][df_final1['Country']=='South Korea'].groupby(['Genre']).agg({'title':'nunique'}).reset_index().sort_values('nunique', ascending=False)
```

```
In [264...]: df_genre=df_sk_shows.groupby(['Genre']).agg({'title':'nunique'}).reset_index().sort_values('nunique', ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_genre['Genre'], df_genre['nunique'], color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```



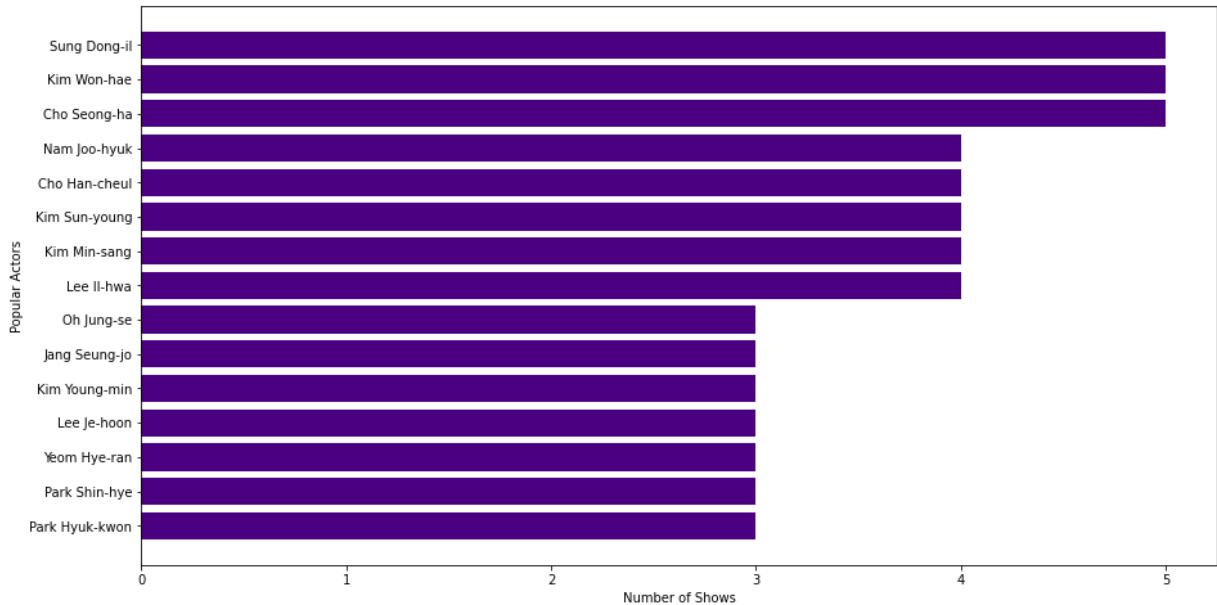
International TV Shows, Romantic TV Shows, Drama, Crime and Comedy Genres are popular in TV Shows in S.Korea. Only S.Korea has Romance as a top 3 favorable genre which depicts an inclination of their audience

```
In [265...]: df_rating=df_sk_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 and Mature Audiences in TV Shows

```
In [266...]: df_actors=df_sk_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 Shows')
plt.ylabel('Popular Actors')
plt.show()
```

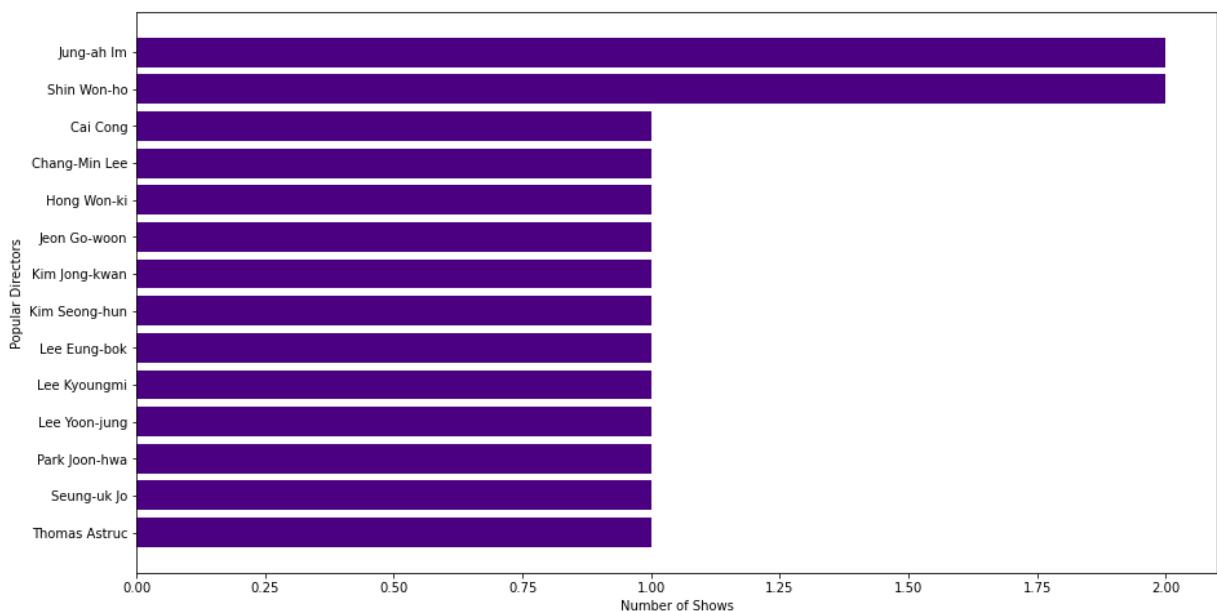


```
In [267]: df_actors['Actors'].values
```

```
Out[267]: 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', 'Kim Young-min', 'Lee Je-hoon',
       'Yeom Hye-ran', 'Park Shin-hye', 'Park Hyuk-kwon'], dtype=object)
```

Popular Actors in TV Shows in South Korea are:- 'Sung Dong-il', 'Kim Won-hae', 'Cho Seong-ha', 'Nam Joo-hyuk'

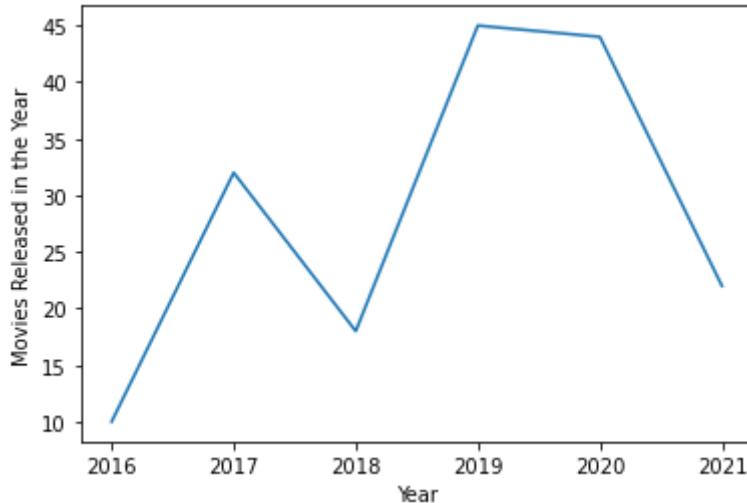
```
In [269]: df_directors=df_sk_shows.groupby(['Directors']).agg({"title":"nunique"}).reset_index()
df_directors=df_directors[df_directors['Directors']!='Unknown Director']
plt.figure(figsize=(15,8))
plt.barh(df_directors[:::-1]['Directors'], df_directors[:::-1]['title'],color=['indigo'])
plt.xlabel('Number of Shows')
plt.ylabel('Popular Directors')
plt.show()
```



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

```
In [270]: df_year=df_sk_shows.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
```

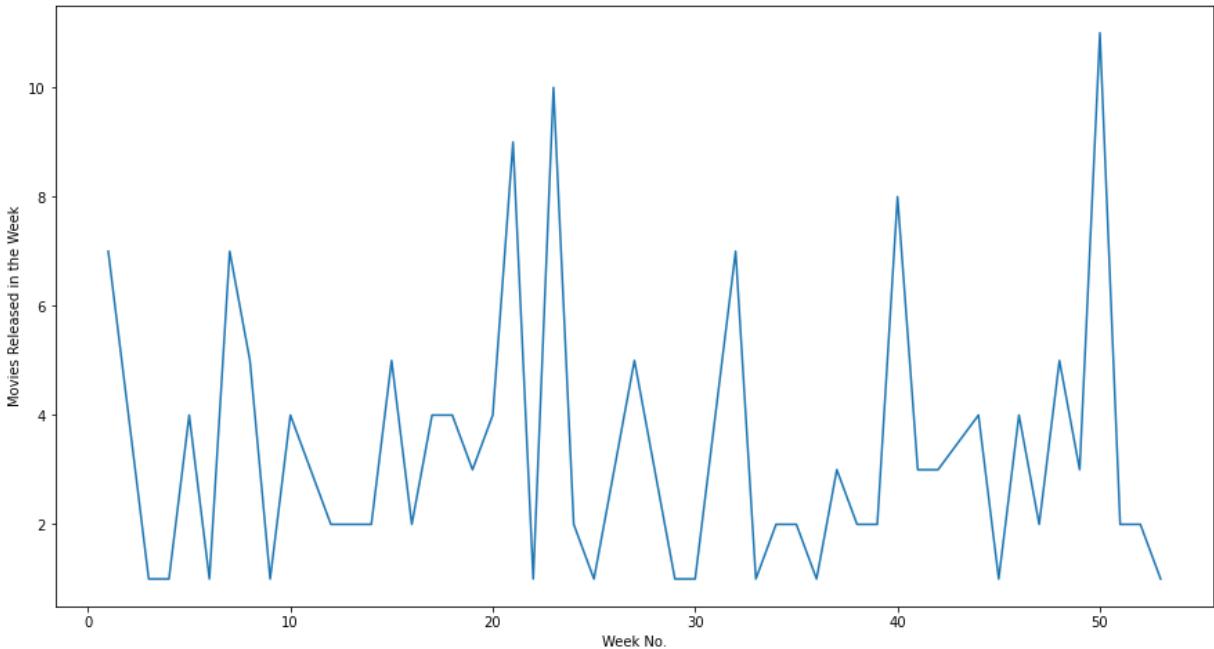
```
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



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

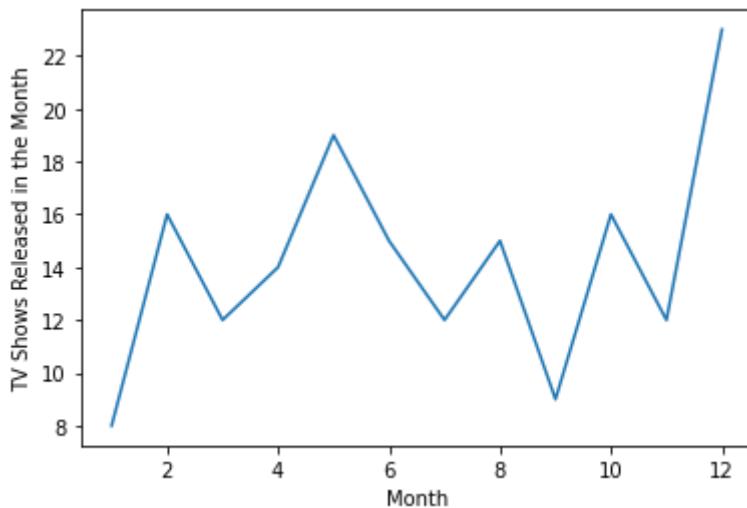
In [272...]

```
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()
```



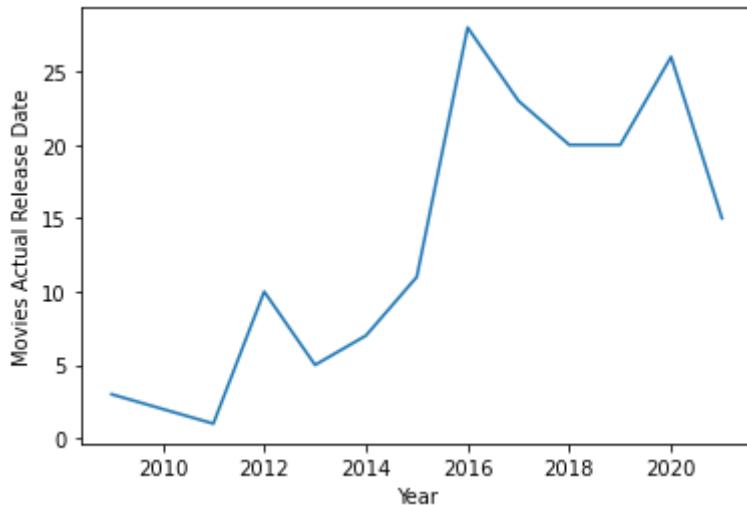
In [273...]

```
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 [274...]: 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 []:

In []: