```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns


data=pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/


data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	r
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	

Analyzing Basic Metrics

```
data.shape
```

```
(8807, 12)
```

```
data.ndim
```

```
2
```

```
data.isna().sum()
```

	0
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

```
data.isna().sum().sum()
```

```
4307
```

```
len(data)
```

```
8807
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64 
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
data.dtypes
```

	0
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

```
data.isnull().sum()
```

	0
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

```
data.isnull().sum().sum()
```

```
4307
```

```
data.nunique()
```

	0
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

Unique Attributes & Value Counts

```
data['rating'].value_counts()
```

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

dtype: int64

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

duration	count
1 Season	1793
2 Seasons	425
3 Seasons	199
90 min	152
94 min	146
...	...
16 min	1
186 min	1
193 min	1
189 min	1
191 min	1

220 rows × 1 columns

dtype: int64

Pivoting Data

```
data[data.director.apply(lambda x: "," in str(x))]
```

	show_id	type	title	director	cast	country	date_added	relea
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	NaN	September 24, 2021	
16	s17	Movie	Europe's Most Dangerous Man: Otto Skorzeny in ...	Pedro de Echave García, Pablo Azorín Williams	NaN	NaN	September 22, 2021	
23	s24	Movie	Go! Go! Cory Carson: Chrissy Takes the Wheel	Alex Woo, Stanley Moore	Maisie Benson, Paul Killam, Kerry Gudjohnsen, ...	NaN	September 21, 2021	
30	s31	Movie	Ankahi Kahaniya	Ashwiny Iyer Tiwari, Abhishek Chaubey, Saket C...	Abhishek Banerjee, Rinku Rajguru, Delzad Hiwal...	NaN	September 17, 2021	
68	s69	Movie	Schumacher	Hanns-Bruno Kammertöns, Vanessa	Michael Schumacher	NaN	September 15, 2021	

```
def split_my_str(inp):
    return str(inp).split(', ')
```

```
unnest1 = data[['title', 'director']]
unnest1.head()
```

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	NaN
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	NaN
4	Kota Factory	NaN

```
unnest1['director']=unnest1.director.apply(split_my_str)
unnest1.shape
```

```
<ipython-input-20-4e3ca9d25569>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  unnest1['director']=unnest1.director.apply(split_my_str)
(8807, 2)
```

```
unnest1=unnest1.explode('director')
unnest1
```

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan
...	...	...
8802	Zodiac	David Fincher
8803	Zombie Dumb	nan
8804	Zombieland	Ruben Fleischer
8805	Zoom	Peter Hewitt
8806	Zubaan	Mozez Singh

9612 rows × 2 columns

```
data[data.cast.apply(lambda x: "," in str(x))]
```

	show_id	type	title	director	cast	country	date_added	release_y
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2(
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2(
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2(
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	September 24, 2021	2(

```
unnest2 = data[['title', 'cast']]
unnest2['cast']=unnest2.cast.apply(split_my_str)
unnest2=unnest2.explode('cast',ignore_index=True)
```

```
<ipython-input-23-f7825a25112e>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  unnest2['cast']=unnest2.cast.apply(split_my_str)
```

```
data[data.country.apply(lambda x: "," in str(x))]
```

	show_id	type	title	director	cast	country	date_added	release
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	
12	s13	Movie	Je Suis Karl	Christian Schwochow	Luna Wedler, Jannis Niewöhner, Milan Peschel, ...	Germany, Czech Republic	September 23, 2021	
29	s30	Movie	Paranoia	Robert Luketic	Liam Hemsworth, Gary Oldman, Amber Heard, Harr...	United States, India, France	September 19, 2021	
38	s39	Movie	Birth of the Dragon	George Nolfi	Billy Magnussen, Ron Yuan, Qu Jingjing, Terry ...	China, Canada, United States	September 16, 2021	
46	s47	Movie	Safe House	Daniel Espinosa	Denzel Washington, Ryan Reynolds	South Africa, United	September 16, 2021	

```
unnest3 = data[['title', 'country']]
unnest3['country']=unnest3.country.apply(split_my_str)
unnest3=unnest3.explode('country',ignore_index=True)
```

```
<ipython-input-25-a5194101093e>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  unnest3['country']=unnest3.country.apply(split_my_str)
```

```
data[data.listed_in.apply(lambda x: "," in str(x))]
```

	show_id	type	title	director	cast	country	date_added	release_ye
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	20
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	20
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	20
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	20
					Kate			

```
unnest4 = data[['title', 'listed_in']]
unnest4['listed_in']=unnest4.listed_in.apply(split_my_str)
unnest4=unnest4.explode('listed_in',ignore_index=True)
```

```
<ipython-input-27-d56e76cb5e61>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  unnest4['listed_in']=unnest4.listed_in.apply(split_my_str)
```

```
unnest1.head()
```

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan

```
unnest2.head()
```

	title	cast
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

```
#merging the unnested director data with unnested actors data
data_new1=unnest2.merge(unnest1,on=['title'],how='inner')

data_new2=data_new1.merge(unnest3,on=['title'],how='inner')

data_new3=data_new2.merge(unnest4,on=['title'],how='inner')
data_new3.head()
```

	title	cast	director	country	listed_in
0	Dick Johnson Is Dead	nan	Kirsten Johnson	United States	Documentaries
1	Blood & Water	Ama Qamata	nan	South Africa	International TV Shows
2	Blood & Water	Ama Qamata	nan	South Africa	TV Dramas
3	Blood & Water	Ama Qamata	nan	South Africa	TV Mysteries
4	Blood & Water	Khosi Ngema	nan	South Africa	International TV Shows

Missing Values and Outlier Treatment

```
data_new3.isna().sum()
```

	0
title	0
cast	0
director	0
country	0
listed_in	0

dtype: int64

```
data_new3.shape
```

```
(201991, 5)
```

```
data_new3['director'].replace('nan','Unknown Director',inplace=True)
data_new3['cast'].replace('nan','Unknown Cast',inplace=True)
data_new3['country'].replace(['nan'],[np.nan],inplace=True)
```

```
data_new3.shape
```

```
<ipython-input-33-4ad82945e4ab>:1: FutureWarning: A value is trying to be set on a co
The behavior will change in pandas 3.0. This inplace method will never work because t

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({


  data_new3['director'].replace('nan','Unknown Director',inplace=True)
<ipython-input-33-4ad82945e4ab>:2: FutureWarning: A value is trying to be set on a co
The behavior will change in pandas 3.0. This inplace method will never work because t

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({


  data_new3['cast'].replace('nan','Unknown Cast',inplace=True)
<ipython-input-33-4ad82945e4ab>:3: FutureWarning: A value is trying to be set on a co
The behavior will change in pandas 3.0. This inplace method will never work because t

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({


  data_new3['country'].replace(['nan'],[np.nan],inplace=True)
(201991, 5)
```

```
data_new3.head()
```

	title	cast	director	country	listed_in
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas
			Unknown		

Merging with original dataset

```
data_final=data_new3.merge(data[['show_id','type','title','date_added','release_year','ra
data_final.head()

#df_final=df_new.merge(df[['show_id', 'type', 'title', 'date_added',
 #      'release_year', 'rating', 'duration']],on=['title'],how='left')
#df_final.head()
```

	title	cast	director	country	listed_in	show_id	type	date_added	rel
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021	
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas	s2	TV Show	September 24, 2021	

```
data_final1=data_final.copy()
```

```
data_final.isna().sum()
```

	0
title	0
cast	0
director	0
country	11897
listed_in	0
show_id	0
type	0
date_added	158
release_year	0
rating	67
duration	3

dtype: int64

```
data_final.loc[data_final['duration'].isnull(),['duration','rating']]
```

	duration	rating
126537	NaN	74 min
131603	NaN	84 min
131737	NaN	66 min

```
data_final.loc[data_final['duration'].isnull(),'duration']=data_final.loc[data_final['dur
#data_final.isnull().sum()


#data_final.loc[data_final['duration'].isnull(),'duration']=data_final.loc[data_final['du
data_final.isnull().sum()
```

	0
title	0
cast	0
director	0
country	11897
listed_in	0
show_id	0
type	0
date_added	158
release_year	0
rating	67
duration	0

dtype: int64

```
data_final.loc[data_final['rating'].str.contains('min', na=False),'rating']='Unknown Rati
#data_final.loc[data_final['rating'].str.contains('min', na=False),'rating']='Unknown_Rat


data_final.isnull().sum()
```

	0
title	0
cast	0
director	0
country	11897
listed_in	0
show_id	0
type	0
date_added	158
release_year	0
rating	67
duration	0

dtype: int64

```
data_final.loc[data_final['rating'].str.contains('min', na=False),'rating']
```

	rating

dtype: object

```
data_final['rating'].fillna('Unknown Rating',inplace=True)
data_final.isnull().sum()
```

	0
title	0
cast	0
director	0
country	11897
listed_in	0
show_id	0
type	0
date_added	158
release_year	0
rating	0
duration	0

dtype: int64

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

	title	cast	director	country	listed_in	show_id	type	date_added
136893	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	United Kingdom	British TV Shows	s6067	TV Show	NaN
	A Young Doctor's	Daniel	Unknown	United	TV		TV	

```
#data_final[data_final['date_added'].isnull()]['release_year'].unique()
a=data_final[data_final['date_added'].isnull()]['release_year'].unique()
a
```

```
array([2013, 2018, 2003, 2008, 2010, 2012, 2016, 2015])
```

```
for i in a:
  temp=data_final[data_final['release_year']==i]['date_added'].mode().values[0]
```

```
  data_final.loc[data_final['release_year']==i,'date_added']=data_final.loc[data_final['r
```

```
data_final.isnull().sum()
```

	0
title	0
cast	0
director	0
country	11897
listed_in	0
show_id	0
type	0
date_added	0
release_year	0
rating	0
duration	0

dtype: int64

```
for i in data_final[data_final['country'].isnull()]['director'].unique(): # all the place

    if i in data_final[~data_final['country'].isnull()]['director'].unique():
      temp=data_final[data_final['director']==i]['country'].mode().values[0]
      data_final.loc[data_final['director']==i,'country']=data_final.loc[data_final['dire
```

```
for i in data_final[data_final['country'].isnull()]['cast'].unique():
  if i in data_final[~data_final['country'].isnull()]['cast'].unique():
    temp=data_final[data_final['cast']==i]['country'].mode().values[0]
    data_final.loc[data_final['cast']==i,'country']=data_final.loc[data_final['cast']==i,
```

```
data_final.isnull().sum()
```

	0
title	0
cast	0
director	0
country	2069
listed_in	0
show_id	0
type	0
date_added	0
release_year	0
rating	0
duration	0

dtype: int64

```
data_final['country'].fillna('Unknown Country',inplace=True)
```

```
<ipython-input-52-210e7522e83f>:1: FutureWarning: A value is trying to be set on a cc
The behavior will change in pandas 3.0. This inplace method will never work because t

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({


  data_final['country'].fillna('Unknown Country',inplace=True)
```

```
data_final.isnull().sum()
```

	0
title	0
cast	0
director	0
country	0
listed_in	0
show_id	0
type	0
date_added	0
release_year	0
rating	0
duration	0

dtype: int64

Graphical and Non-Graphical Ananlysis

```
data_final.head()
```

	title	cast	director	country	listed_in	show_id	type	date_added	rel
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021	
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	September 24, 2021	
2	Blood &	Ama	Unknown	South	TV Dramas	s2	TV	September	

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

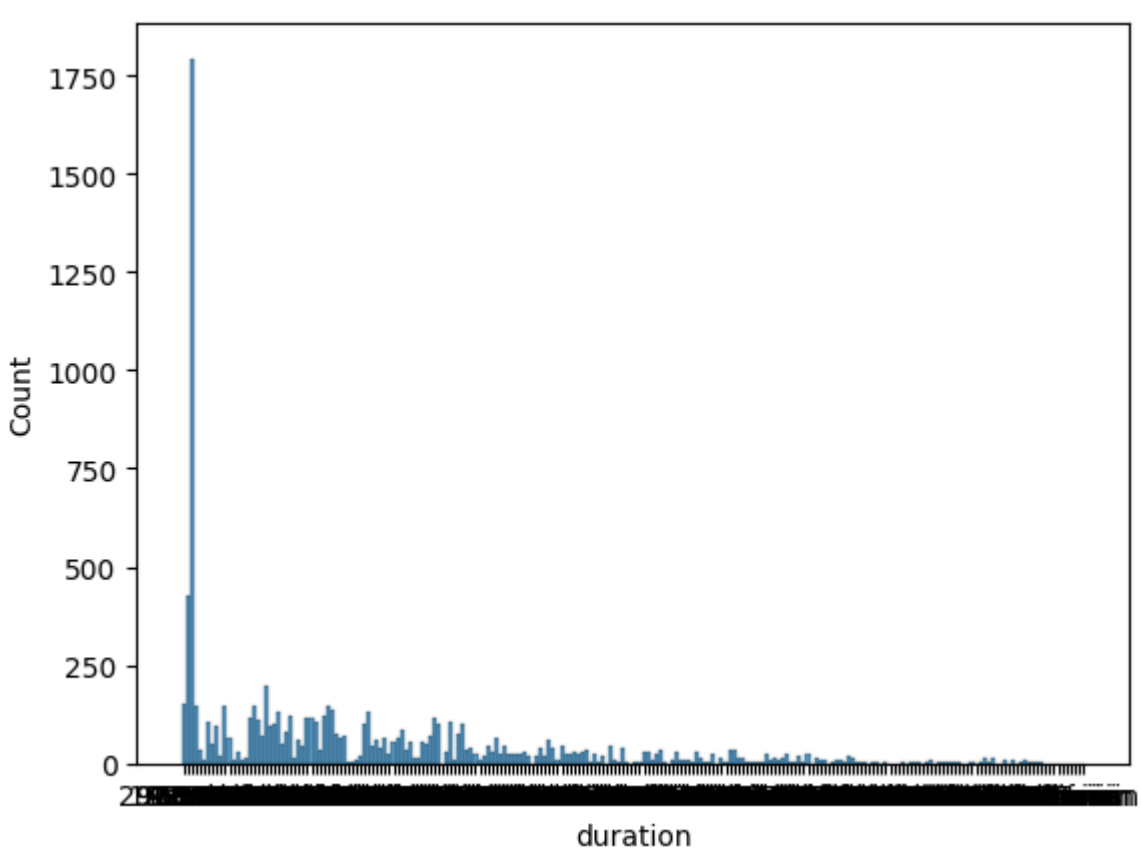
duration	count
1 Season	35035
2 Seasons	9559
3 Seasons	5084
94 min	4343
106 min	4040
...	...
3 min	4
5 min	3
11 min	2
8 min	2
9 min	2

220 rows × 1 columns

dtype: int64

The Duration column has both minutes for movies and seasons for shows merged.

```
sns.histplot(data['duration'], bins=10)
plt.show()
```




```
data_min=data_final.loc[data_final['duration'].str.contains('min')]
data_min
```

	title	cast	director	country	listed_in	show_id	type	date_
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	Septe 25
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	United States	Children & Family Movies	s7	Movie	Septe 24
160	My Little Pony: A New Generation	Vanessa Hudgens	José Luis Ucha	United States	Children & Family Movies	s7	Movie	Septe 24
161	My Little Pony: A New Generation	Kimiko Glenn	Robert Cullen	United States	Children & Family Movies	s7	Movie	Septe 24
162	My Little Pony: A	Kimiko	José Luis	United	Children &	s7	Movie	Septe

```
data_min['duration']=data_min['duration'].str.replace(" min","")
data_min.head()
```

```
<ipython-input-58-4b0c2acf6eef>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  data_min['duration']=data_min['duration'].str.replace(" min","")
```

	title	cast	director	country	listed_in	show_id	type	date_added
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	United States	Children & Family Movies	s7	Movie	September 24, 2021

```
data_min['duration']=data_min['duration'].astype(int)
data_min['duration'].describe()
```

```
<ipython-input-59-cd7b9a4675b4>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  data_min['duration']=data_min['duration'].astype(int)
```

	duration
count	145843.00000
mean	106.85579
std	24.69672
min	3.00000
25%	93.00000
50%	104.00000
75%	119.00000
max	312.00000

dtype: float64

```
bins = [1,20,40,60,80,100,120,140,160,180,200,250,315]
labels = ['1-20','20-40','40-60','60-80','80-100','100-120','120-140','140-160','160-180'
data_min['duration'] = pd.cut(data_min['duration'],bins=bins,labels=labels)
data_min.head()
```

```
<ipython-input-60-c4e774028024>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  data_min['duration'] = pd.cut(data_min['duration'],bins=bins,labels=labels)
```

	title	cast	director	country	listed_in	show_id	type	date_added
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	United States	Children & Family Movies	s7	Movie	September 24, 2021

```
data_season=data_final.loc[data_final['duration'].str.contains('Season')]
data_season.dtypes
```

	0
title	object
cast	object
director	object
country	object
listed_in	object
show_id	object
type	object
date_added	object
release_year	int64
rating	object
duration	object

dtype: object

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

duration	count
1 Season	35035
2 Seasons	9559
3 Seasons	5084
4 Seasons	2134
5 Seasons	1698
7 Seasons	843
6 Seasons	633
8 Seasons	286
9 Seasons	257
10 Seasons	220
13 Seasons	132
12 Seasons	111
15 Seasons	96
17 Seasons	30
11 Seasons	30

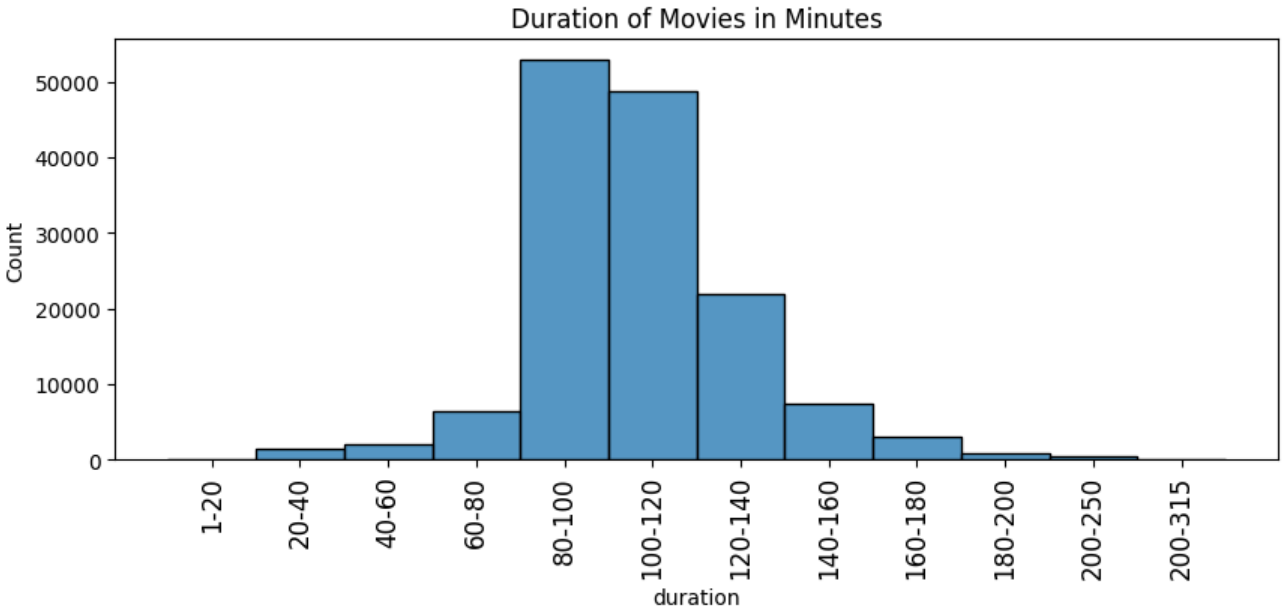
dtype: int64

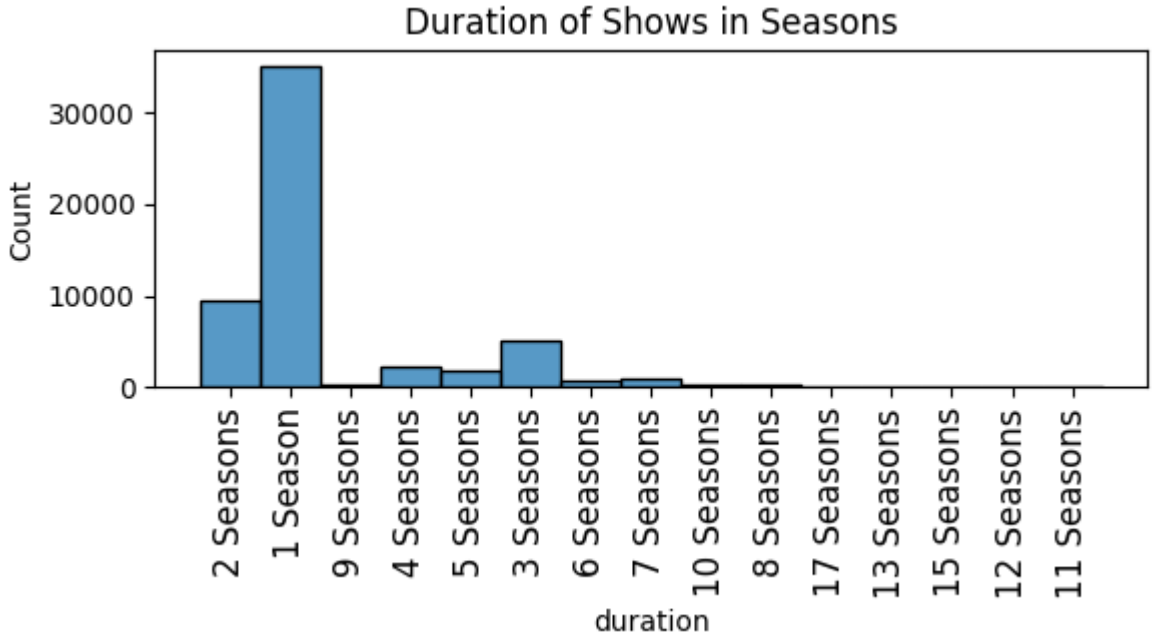
```
data_min['duration'].value_counts()
```

duration	count
80-100	52937
100-120	48724
120-140	21898
140-160	7528
60-80	6498
160-180	3136
40-60	2155
20-40	1455
180-200	866
200-250	481
1-20	122
200-315	43

dtype: int64

```
plt.figure(figsize=(10,8))
plt.subplot(2,1,1)
plt.title('Duration of Movies in Minutes')
sns.histplot(data_min['duration'], bins=10)
plt.xticks(rotation=90,fontsize=12)
plt.show()

plt.subplot(2,1,2)
plt.title('Duration of Shows in Seasons')
sns.histplot(data_season['duration'], bins=10)
plt.xticks(rotation=90,fontsize=12)
plt.show()
```





The below are the mostly found content duration in netflix.

1. 80-100 min/100-120 min
2. 1Season

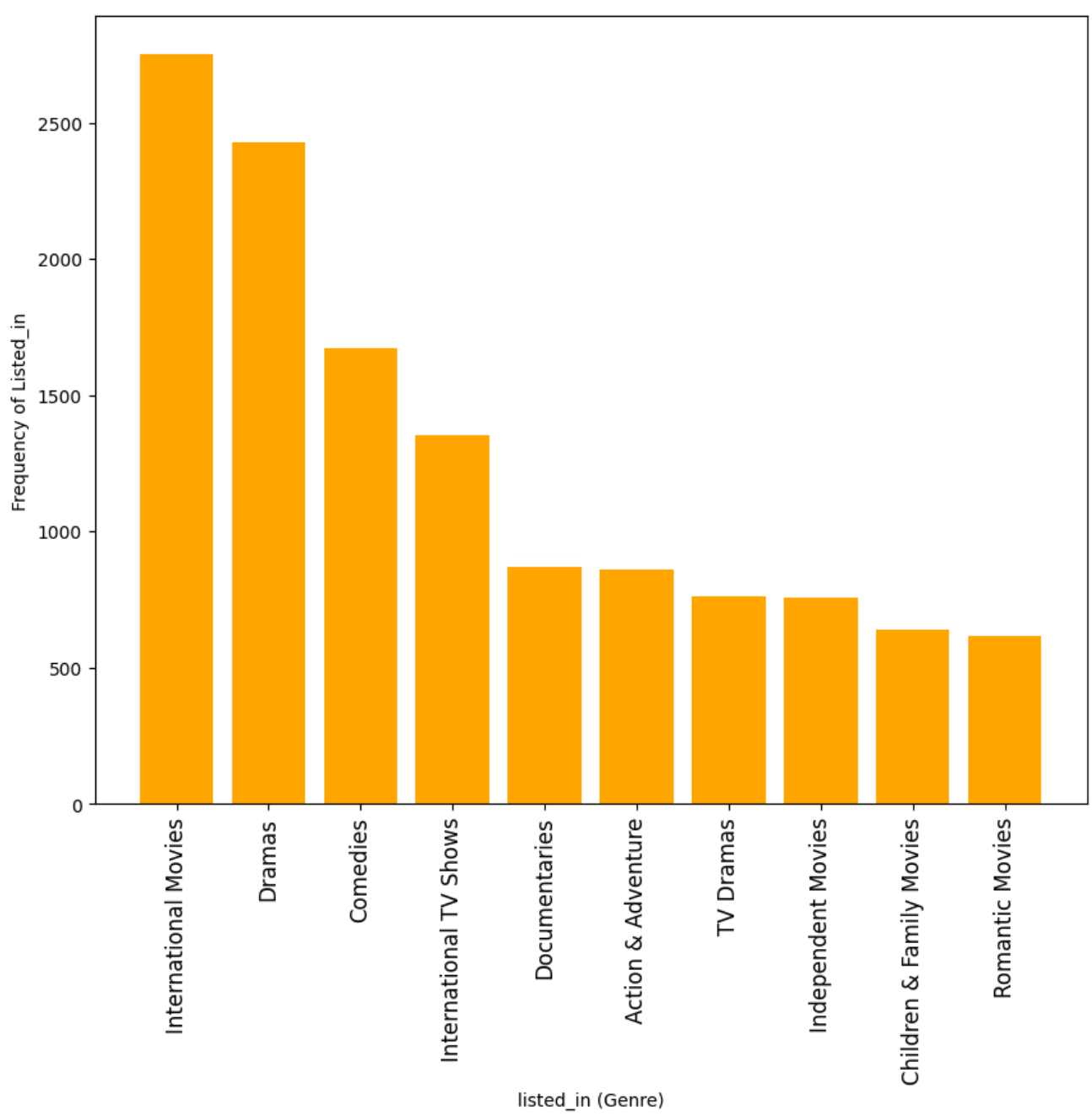
```
data_final
```

	title	cast	director	country	listed_in	show_id	type	date_ad
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	Septem 25, 2
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	Septem 24, 2
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas	s2	TV Show	Septem 24, 2
3	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Mysteries	s2	TV Show	Septem 24, 2
4	Blood & Water	Khosi Ngema	Unknown Director	South Africa	International TV Shows	s2	TV Show	Septem 24, 2
...	...	...	...	...	...	...	...	
201986	Zubaan	Anita Shabdish	Mozez Singh	India	International Movies	s8807	Movie	Marc 2
201987	Zubaan	Anita	Mozez	India	Music &	s8807	Movie	Marc

```
pd.set_option('display.max_rows',None)
pd.set_option('display.max_columns',None)
data_listed_in_val=data_final.groupby(['listed_in'])['title'].nunique().sort_values(ascen
data_listed_in_val
```

	listed_in	title
0	International Movies	2752
1	Dramas	2427
2	Comedies	1674
3	International TV Shows	1351
4	Documentaries	869
5	Action & Adventure	859
6	TV Dramas	763
7	Independent Movies	756
8	Children & Family Movies	641
9	Romantic Movies	616

```
plt.figure(figsize=(10,8))
plt.bar(data_listed_in_val['listed_in'], data_listed_in_val['title'],color=['orange'])
plt.xlabel('listed_in (Genre)')
plt.ylabel('Frequency of Listed_in')
plt.xticks(rotation=90,fontsize=12)
plt.show()
```



The Top3 content in netflix are International Movies, Dramas and Comedies

```
data_country_val=data_final.groupby(['country'])['title'].nunique().reset_index()
data_country_val
```

	country	title
0		3
1	Afghanistan	1
2	Albania	1
3	Algeria	3
4	Angola	2
5	Argentina	94
6	Armenia	1
7	Australia	162
8	Austria	12
9	Azerbaijan	1
10	Bahamas	1
11	Bangladesh	4
12	Belarus	1
13	Belgium	94
14	Bermuda	1
15	Botswana	1
16	Brazil	103
17	Bulgaria	10
18	Burkina Faso	1
19	Cambodia	5
20	Cambodia,	1
21	Cameroon	2
22	Canada	460
23	Cayman Islands	2
24	Chile	30
25	China	166
26	Colombia	54
27	Croatia	4
28	Cuba	2
29	Cyprus	1
30	Czech Republic	23
31	Denmark	50

32	Dominican Republic	1
33	East Germany	1
34	Ecuador	1
35	Egypt	134
36	Ethiopia	1
37	Finland	12
38	France	409
39	Georgia	2
40	Germany	231
41	Ghana	8
42	Greece	11
43	Guatemala	2
44	Hong Kong	110
45	Hungary	11
46	Iceland	11
47	India	1138
48	Indonesia	97
49	Iran	4
50	Iraq	2
51	Ireland	46
52	Israel	30
53	Italy	102
54	Jamaica	1
55	Japan	338
56	Jordan	10
57	Kazakhstan	1
58	Kenya	6
59	Kuwait	9
60	Latvia	1
61	Lebanon	33
62	Liechtenstein	1
63	Lithuania	1
64	Luxembourg	12
65	Malawi	1

66	Malaysia	26
67	Malta	3
68	Mauritius	3
69	Mexico	175
70	Mongolia	1
71	Montenegro	1
72	Morocco	6
73	Mozambique	1
74	Namibia	2
75	Nepal	2
76	Netherlands	50
77	New Zealand	33
78	Nicaragua	1
79	Nigeria	140
80	Norway	30
81	Pakistan	24
82	Palestine	1
83	Panama	1
84	Paraguay	1
85	Peru	11
86	Philippines	90
87	Poland	41
88	Poland,	1
89	Portugal	6
90	Puerto Rico	1
91	Qatar	10
92	Romania	14
93	Russia	27
94	Samoa	1
95	Saudi Arabia	14
96	Senegal	3
97	Serbia	7
98	Singapore	41

99	Slovakia	1
100	Slovenia	3
101	Somalia	1
102	South Africa	65
103	South Korea	235
104	Soviet Union	3
105	Spain	239
106	Sri Lanka	1
107	Sudan	1
108	Sweden	44
109	Switzerland	19
110	Syria	3
111	Taiwan	94
112	Thailand	74
113	Turkey	115
114	Uganda	1
115	Ukraine	3
116	United Arab Emirates	38
117	United Kingdom	829
118	United Kingdom,	2
119	United States	4248
120	United States,	1
121	Unknown Country	175
122	Uruguay	14
123	Vatican City	1
124	Venezuela	4
125	Vietnam	7
126	West Germany	5
127	Zimbabwe	3

```
data_country_val[data_country_val['country'].str.contains(',')]
```

	country	title
20	Cambodia,	1
88	Poland,	1
118	United Kingdom,	2
120	United States,	1

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

```
data_country_val=data_final.groupby(['country'])['title'].nunique().reset_index()
data_country_val
```

	country	title
0		3
1	Afghanistan	1
2	Albania	1
3	Algeria	3
4	Angola	2
5	Argentina	94
6	Armenia	1
7	Australia	162
8	Austria	12
9	Azerbaijan	1
10	Bahamas	1
11	Bangladesh	4
12	Belarus	1
13	Belgium	94
14	Bermuda	1
15	Botswana	1
16	Brazil	103
17	Bulgaria	10
18	Burkina Faso	1
19	Cambodia	6
20	Cameroon	2
21	Canada	460
22	Cayman Islands	2
23	Chile	30
24	China	166
25	Colombia	54
26	Croatia	4
27	Cuba	2
28	Cyprus	1
29	Czech Republic	23
30	Denmark	50
31	Dominican Republic	1

32	East Germany	1
33	Ecuador	1
34	Egypt	134
35	Ethiopia	1
36	Finland	12
37	France	409
38	Georgia	2
39	Germany	231
40	Ghana	8
41	Greece	11
42	Guatemala	2
43	Hong Kong	110
44	Hungary	11
45	Iceland	11
46	India	1138
47	Indonesia	97
48	Iran	4
49	Iraq	2
50	Ireland	46
51	Israel	30
52	Italy	102
53	Jamaica	1
54	Japan	338
55	Jordan	10
56	Kazakhstan	1
57	Kenya	6
58	Kuwait	9
59	Latvia	1
60	Lebanon	33
61	Liechtenstein	1
62	Lithuania	1
63	Luxembourg	12
64	Malawi	1
65	Malaysia	26

66	Malta	3
67	Mauritius	3
68	Mexico	175
69	Mongolia	1
70	Montenegro	1
71	Morocco	6
72	Mozambique	1
73	Namibia	2
74	Nepal	2
75	Netherlands	50
76	New Zealand	33
77	Nicaragua	1
78	Nigeria	140
79	Norway	30
80	Pakistan	24
81	Palestine	1
82	Panama	1
83	Paraguay	1
84	Peru	11
85	Philippines	90
86	Poland	42
87	Portugal	6
88	Puerto Rico	1
89	Qatar	10
90	Romania	14
91	Russia	27
92	Samoa	1
93	Saudi Arabia	14
94	Senegal	3
95	Serbia	7
96	Singapore	41
97	Slovakia	1
98	Slovenia	3

99	Somalia	1
100	South Africa	65
101	South Korea	235
102	Soviet Union	3
103	Spain	239
104	Sri Lanka	1
105	Sudan	1
106	Sweden	44
107	Switzerland	19
108	Syria	3
109	Taiwan	94
110	Thailand	74
111	Turkey	115
112	Uganda	1
113	Ukraine	3
114	United Arab Emirates	38
115	United Kingdom	831
116	United States	4249
117	Unknown Country	175
118	Uruguay	14
119	Vatican City	1
120	Venezuela	4
121	Vietnam	7
122	West Germany	5
123	Zimbabwe	3

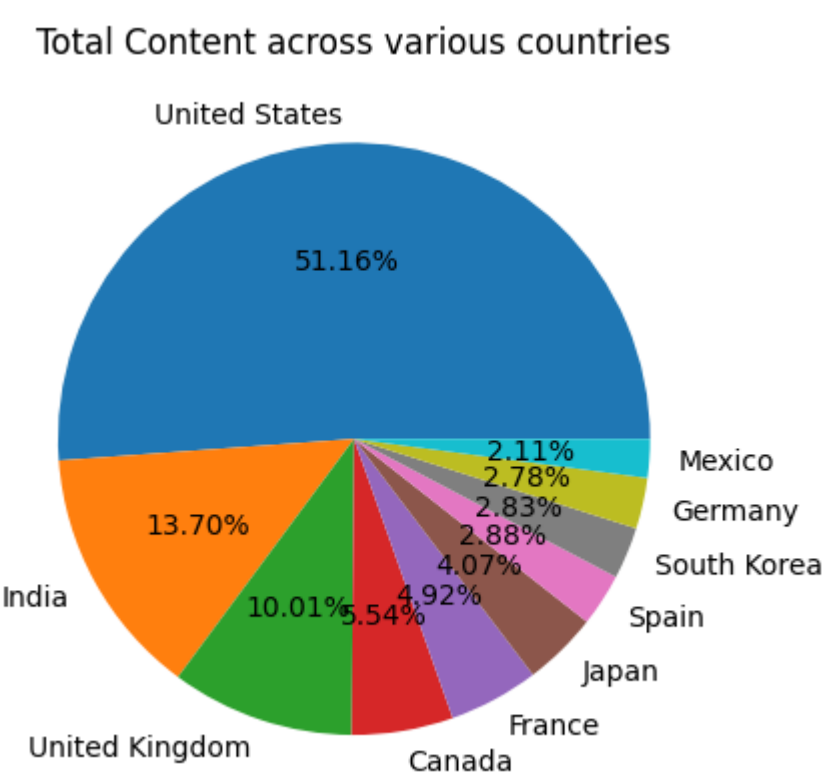
```
data_country_val[data_country_val['country'].str.contains(',')]
```

	country	title

```
data_country_val=data_country_val.sort_values(by='title',ascending=False)[:10]
data_country_val
```

	country	title
116	United States	4249
46	India	1138
115	United Kingdom	831
21	Canada	460
37	France	409
54	Japan	338
103	Spain	239
101	South Korea	235
39	Germany	231
68	Mexico	175

```
plt.pie(data_country_val['title'],labels=data_country_val['country'],autopct='%.2f%%')
plt.title('Total Content across various countries')
plt.show()
```

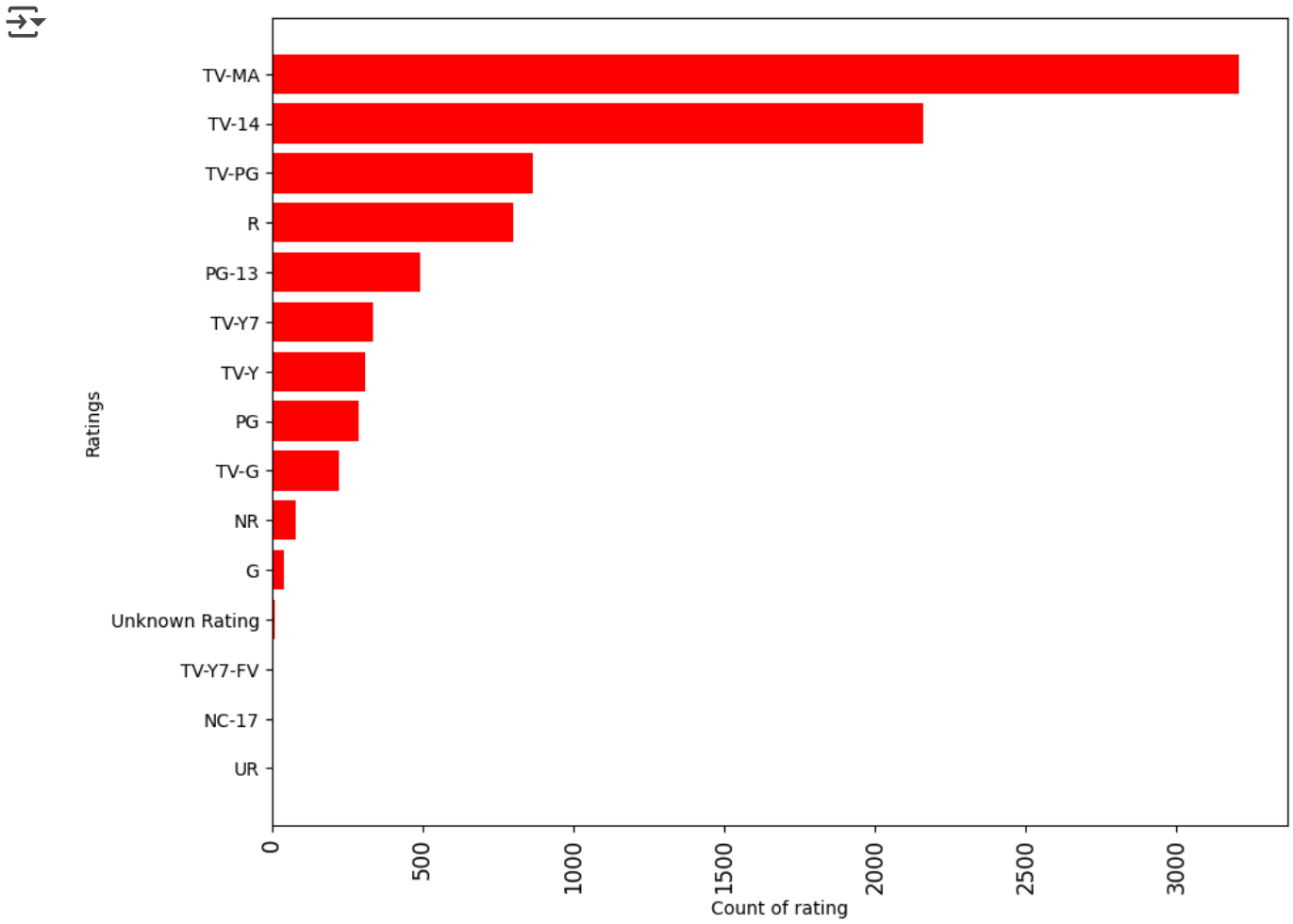


United States, India and United Kingdom are the top 3 countries whos contents were published.

```
data_rating=data_final.groupby(['rating'])['title'].nunique().sort_values(ascending=False
data_rating
```

	rating	title
0	TV-MA	3207
1	TV-14	2160
2	TV-PG	863
3	R	799
4	PG-13	490
5	TV-Y7	334
6	TV-Y	307
7	PG	287
8	TV-G	220
9	NR	80
10	G	41
11	Unknown Rating	7
12	TV-Y7-FV	6
13	NC-17	3
14	UR	3

```
plt.figure(figsize=(10,8))
plt.barh(data_rating['rating'][::-1], data_rating['title'][::-1],color=['red'])
plt.xlabel('Count of rating')
plt.ylabel('Ratings')
plt.xticks(rotation=90,fontsize=12)
plt.show()
```



Most of the contents are rated as TV-MA(Mature Audience), TV-14(restricted for age under 14), TV-PG(Parental Guidance), R(Restricted)

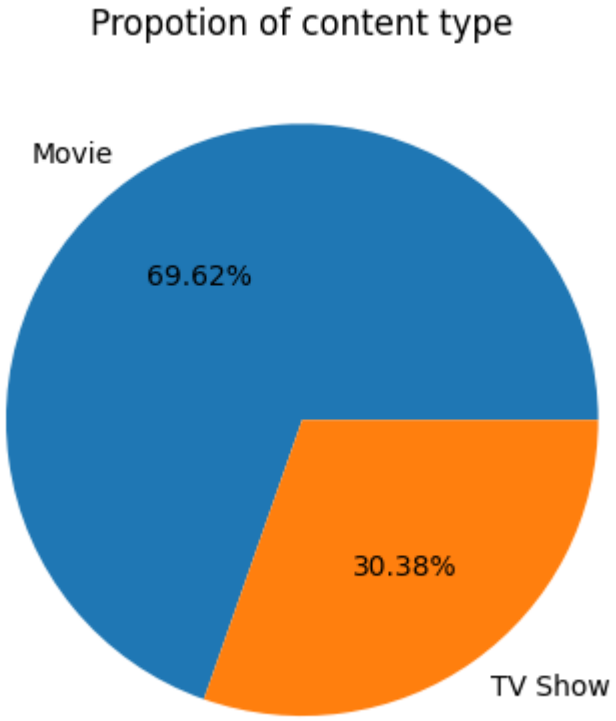
```
data_final.head()
```

	title	cast	director	country	listed_in	show_id	type	date_added	rel
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021	
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas	s2	TV Show	September 24, 2021	

```
data_final.groupby(['type']).agg({"title":"nunique"})
data_type=data_final.groupby(['type']).agg({"title":"nunique"}).reset_index()
data_type
```

	type	title
0	Movie	6131
1	TV Show	2676

```
plt.pie(data_type['title'],
       labels=data_type['type'],
       autopct = '%.2f%%') # label the wedges with their numeric value

plt.title('Propotion of content type')
plt.show()
```



On the overall content available in the netflix, almost 70% are movies and remaining 30% are shows.

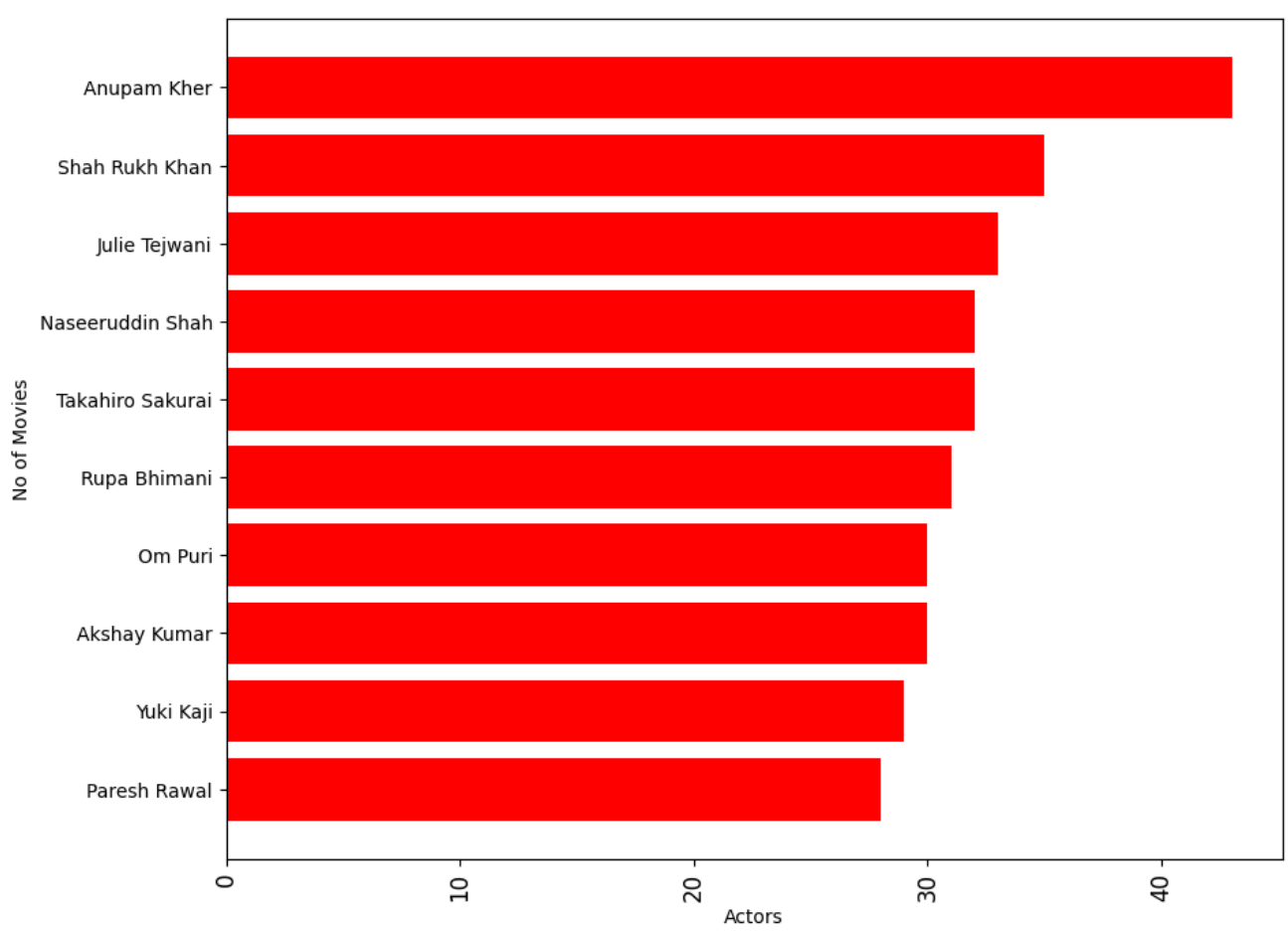
```
#number of distinct titles on the basis of Actors
data_cast=data_final.groupby(['cast']).agg({"title":"nunique"}).sort_values(by=['title'],
data_cast
```

	cast	title
0	Unknown Cast	825
1	Anupam Kher	43
2	Shah Rukh Khan	35
3	Julie Tejwani	33
4	Naseeruddin Shah	32
5	Takahiro Sakurai	32
6	Rupa Bhimani	31
7	Om Puri	30
8	Akshay Kumar	30
9	Yuki Kaji	29
10	Paresh Rawal	28

```
data_cast=data_cast[data_cast['cast']!='Unknown Cast']
data_cast
```

	cast	title
1	Anupam Kher	43
2	Shah Rukh Khan	35
3	Julie Tejwani	33
4	Naseeruddin Shah	32
5	Takahiro Sakurai	32
6	Rupa Bhimani	31
7	Om Puri	30
8	Akshay Kumar	30
9	Yuki Kaji	29
10	Paresh Rawal	28

```
plt.figure(figsize=(10,8))
plt.barh(data_cast['cast'][::-1], data_cast['title'][::-1],color=['red'])
plt.xlabel('Actors')
plt.ylabel('No of Movies')
plt.xticks(rotation=90,fontsize=12)
plt.show()
```



The overall top three most popular or most prolific actors are Anupam Kher, Shah Rukh Khan, and Julie Tejwani

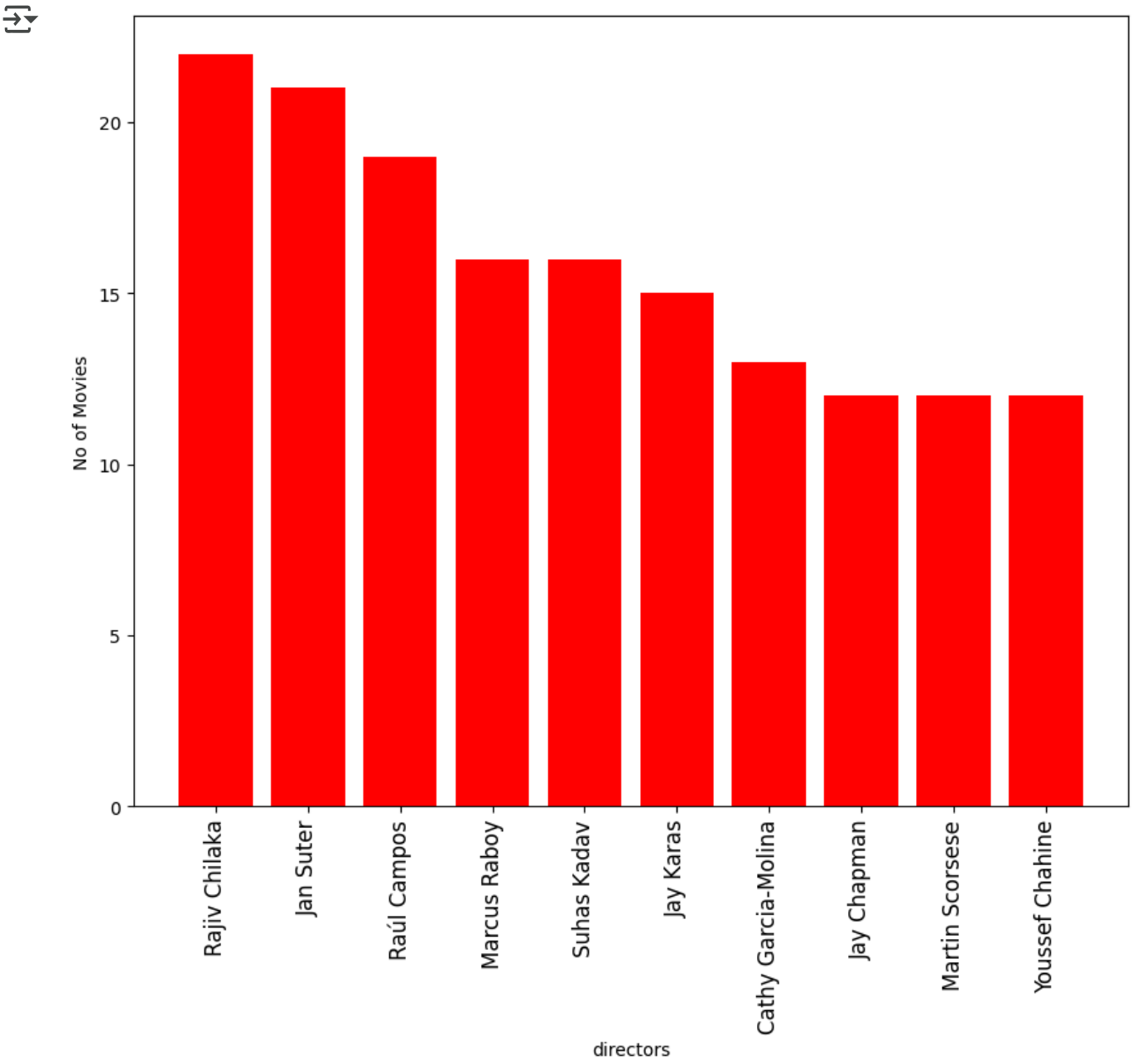
```
data_dir=data_final.groupby(['director']).agg({"title":"nunique"}).sort_values(by=['title
data_dir
```

	director	title
0	Unknown Director	2634
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Marcus Raboy	16
5	Suhas Kadav	16
6	Jay Karas	15
7	Cathy Garcia-Molina	13
8	Jay Chapman	12
9	Martin Scorsese	12
10	Youssef Chahine	12

```
data_dir=data_dir[data_dir['director']!='Unknown Director']
data_dir
```

	director	title
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Marcus Raboy	16
5	Suhas Kadav	16
6	Jay Karas	15
7	Cathy Garcia-Molina	13
8	Jay Chapman	12
9	Martin Scorsese	12
10	Youssef Chahine	12

```
plt.figure(figsize=(10,8))
plt.bar(data_dir['director'], data_dir['title'],color=['red'])
plt.xlabel('directors')
plt.ylabel('No of Movies')
plt.xticks(rotation=90,fontsize=12)
plt.show()
```



The overall top three most popular or most prolific actors are Rajiv Chilaka, Jan Suter , and Raúl Campos

```
data_final.head()
```

	title	cast	director	country	listed_in	show_id	type	date_added	rel
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021	
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas	s2	TV Show	September 24, 2021	

```
from datetime import datetime
arr=[]
for i in data_final['date_added'].values:
  arr.append(pd.to_datetime(i).strftime('%Y-%m-%d'))
data_final['Content_Added_Date'] =arr
```

```
data_final.head()
```

	title	cast	director	country	listed_in	show_id	type	date_added	rel
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021	
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas	s2	TV Show	September 24, 2021	

```
data_final.dtypes
```

	0
title	object
cast	object
director	object
country	object
listed_in	object
show_id	object
type	object
date_added	object
release_year	int64
rating	object
duration	object
Content_Added_Date	object

dtype: object

```
data_final['Content_Added_Date'] =pd.to_datetime(data_final['Content_Added_Date'])
```

```
data_final.dtypes
```

	0
title	object
cast	object
director	object
country	object
listed_in	object
show_id	object
type	object
date_added	object
release_year	int64
rating	object
duration	object
Content_Added_Date	datetime64[ns]

dtype: object

```
data_final['Content_Added_Month']=data_final['Content_Added_Date'].dt.month
data_final['Content_Added_Week']=data_final['Content_Added_Date'].dt.isocalendar().week
data_final['Content_Added_Year']=data_final['Content_Added_Date'].dt.year
```

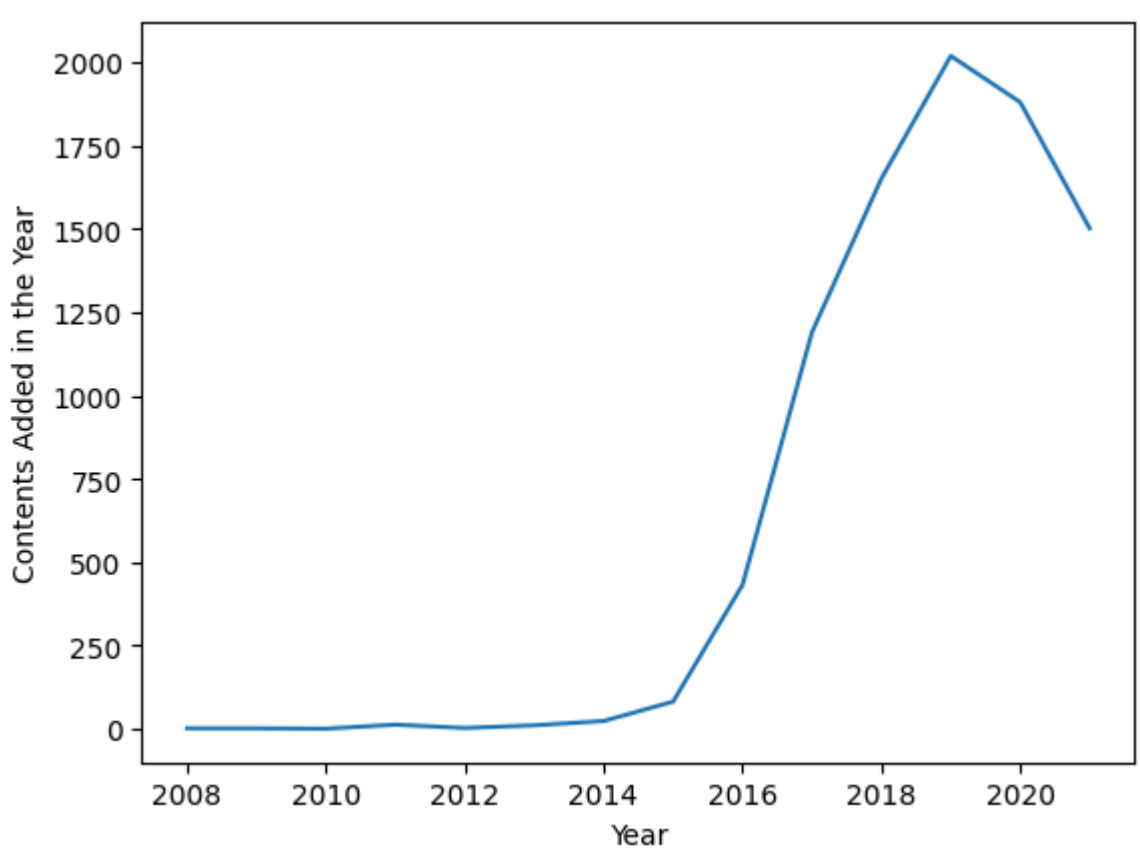
```
data_final.head()
```

	title	cast	director	country	listed_in	show_id	type	date_added	rel
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021	
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas	s2	TV Show	September 24, 2021	
3	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Mysteries	s2	TV Show	September 24, 2021	
4	Blood &	Khosi	Unknown	South	International		TV	September	

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

	Content_Added_Year	title
0	2008	2
1	2009	2
2	2010	1
3	2011	13
4	2012	3
5	2013	11
6	2014	24
7	2015	82
8	2016	432
9	2017	1189
10	2018	1650
11	2019	2018
12	2020	1879
13	2021	1501

```
sns.lineplot(data=data_year, x='Content_Added_Year', y='title')
plt.ylabel("Contents Added in the Year")
plt.xlabel("Year")
plt.show()
```

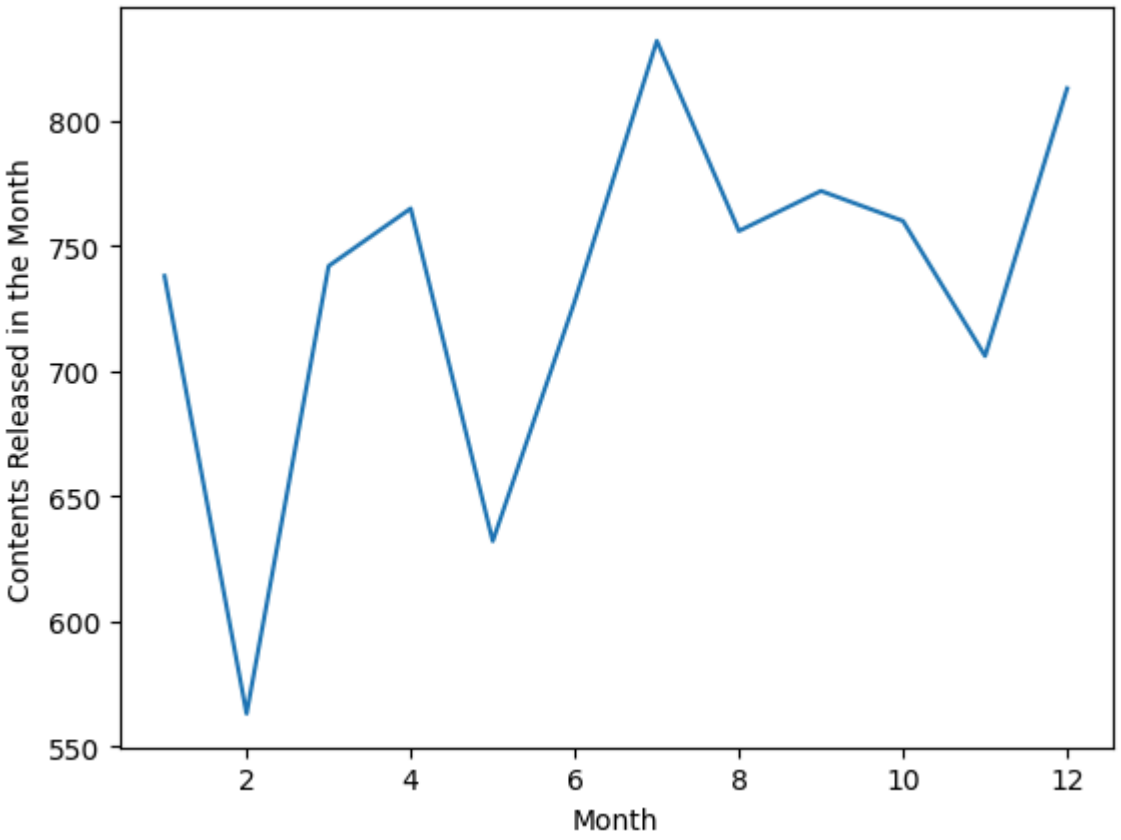


The Contents addition had gradual increase until around the year 2019 where it found its peak. After 2019 the content addition has started to decrease.

```
data_final.groupby(['Content_Added_Month']).agg({"title":"nunique"}).sort_values(by=['tit
```

	Content_Added_Month	title
0	7	832
1	12	813
2	9	772
3	4	765
4	10	760
5	8	756
6	3	742
7	1	738
8	6	728
9	11	706
10	5	632
11	2	563

```
#number of distinct titles on the basis of year
data_month=data_final.groupby(['Content_Added_Month']).agg({"title":"nunique"}).reset_ind
sns.lineplot(data=data_month, x='Content_Added_Month', y='title')
plt.ylabel("Contents Released in the Month")
plt.xlabel("Month")
plt.show()
```



Most of the contents were added in the seventh, twelfth and ninth months of the year.

```
data_final.groupby(['Content_Added_Week']).agg({"title":"nunique"}).sort_values(by=['titl
```

	Content_Added_Week	title
0	1	372
1	44	318
2	40	287
3	26	271
4	31	269
5	35	265
6	9	254
7	13	250
8	27	241
9	18	234
10	5	208
11	22	206
12	48	200
13	50	189
14	37	183
15	14	173
16	39	166
17	24	164
18	11	163
19	16	160
20	30	160
21	17	154
22	15	153
23	33	153
24	23	151
25	7	147
26	34	143
27	25	143
28	36	142
29	29	140
30	49	140
31	38	139

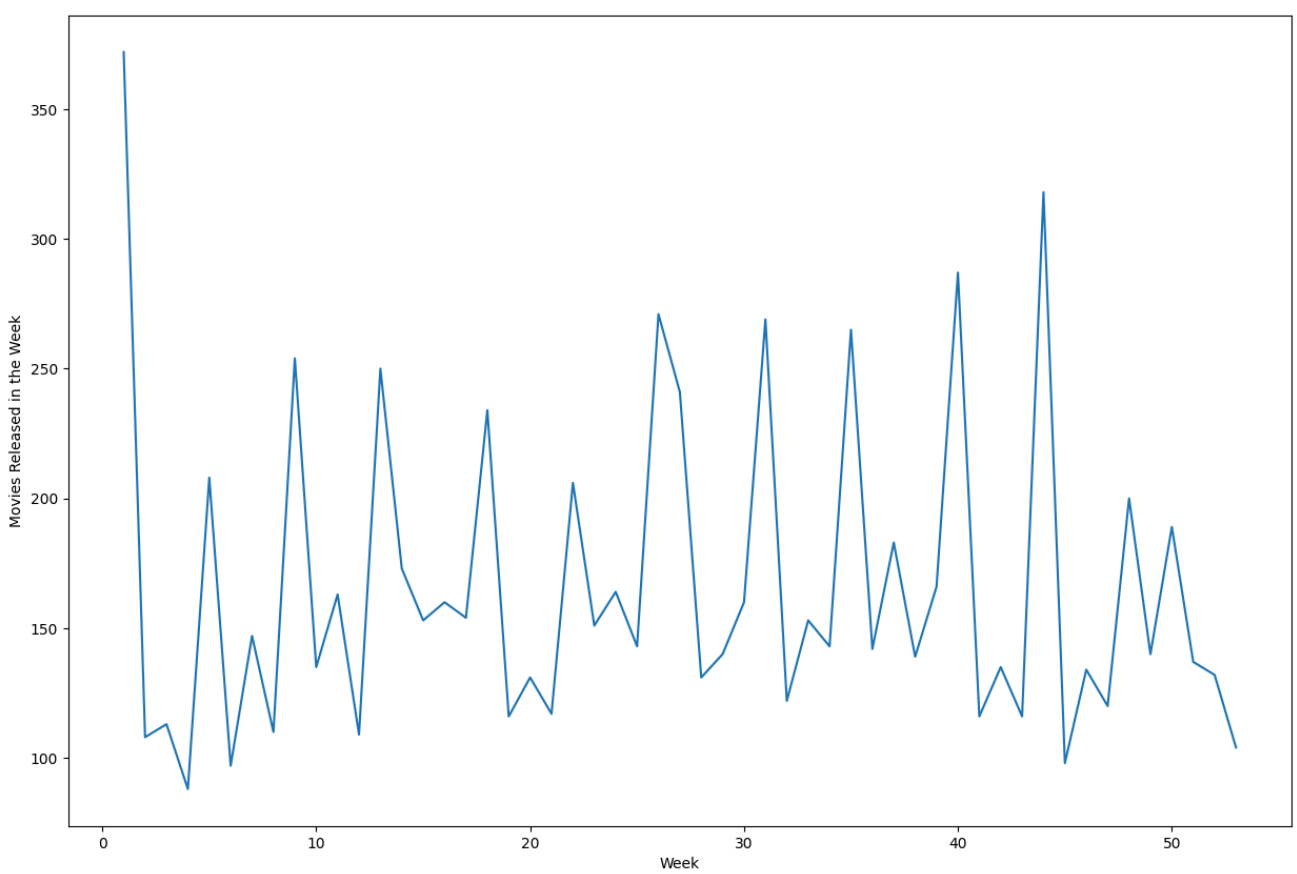
32	51	137
33	42	135
34	10	135
35	46	134
36	52	132
37	28	131
38	20	131
39	32	122
40	47	120
41	21	117
42	41	116
43	43	116
44	19	116
45	3	113
46	8	110
47	12	109
48	2	108
49	53	104
50	45	98
51	6	97
52	4	88

```
data_final.groupby(['Content_Added_Week']).agg({"title":"nunique"}).sort_values(by=['titl
```

	Content_Added_Week	title
0	1	372
1	44	318
2	40	287
3	26	271
4	31	269
5	35	265
6	9	254
7	13	250
8	27	241
9	18	234
10	5	208
11	22	206
12	48	200
13	50	189
14	37	183
15	14	173
16	39	166
17	24	164
18	11	163
19	16	160
20	30	160
21	17	154
22	15	153
23	33	153
24	23	151
25	7	147
26	34	143
27	25	143
28	36	142
29	29	140
30	49	140
31	38	139

32	51	137
33	42	135
34	10	135
35	46	134
36	52	132
37	28	131
38	20	131
39	32	122
40	47	120
41	21	117
42	41	116
43	43	116
44	19	116
45	3	113
46	8	110
47	12	109
48	2	108
49	53	104
50	45	98
51	6	97
52	4	88

```
#number of distinct titles on the basis of year
plt.figure(figsize=(15,10))
data_week=data_final.groupby(['Content_Added_Week']).agg({"title":"nunique"}).reset_index
sns.lineplot(data=data_week, x='Content_Added_Week', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week")
plt.show()
```



Most of the contents were added in the 1st and 44th week of the year.

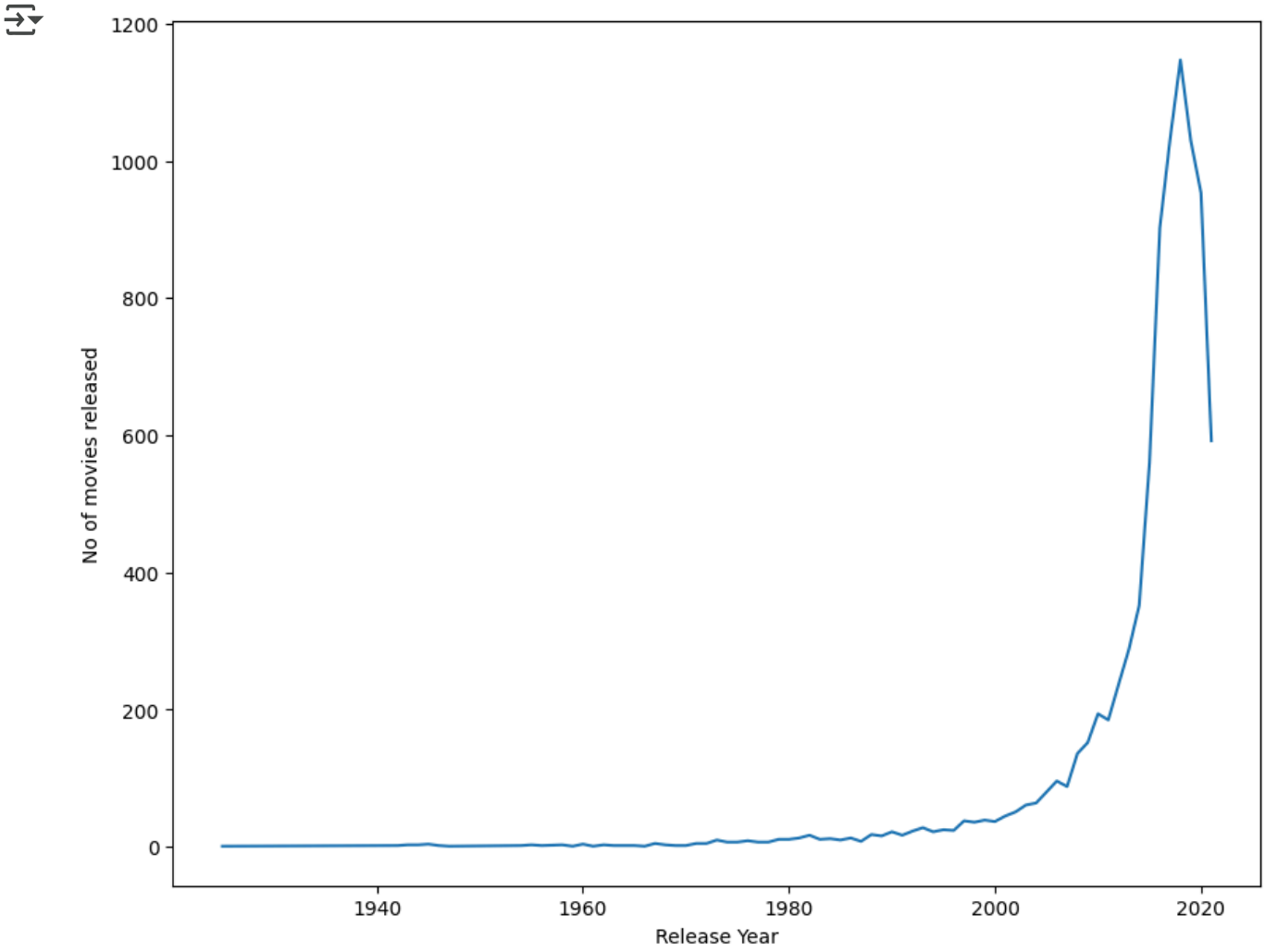
```
data_final.groupby(['release_year']).agg({"title":"nunique"}).sort_values(by=['title'],as
```

	release_year	title
0	2018	1147
1	2017	1032
2	2019	1030
3	2020	953
4	2016	902
5	2021	592
6	2015	560
7	2014	352
8	2013	288
9	2012	237
10	2010	194
11	2011	185
12	2009	152
13	2008	136
14	2006	96
15	2007	88
16	2005	80
17	2004	64
18	2003	61
19	2002	51
20	2001	45
21	1999	39
22	1997	38
23	2000	37
24	1998	36
25	1993	28
26	1995	25
27	1996	24
28	1992	23
29	1990	22
30	1994	22
31	1988	18

32	1982	17
33	1991	17
34	1989	16
35	1981	13
36	1986	13
37	1984	12
38	1980	11
39	1983	11
40	1979	11
41	1985	10
42	1973	10
43	1976	9
44	1987	8
45	1978	7
46	1977	7
47	1975	7
48	1974	7
49	1967	5
50	1972	5
51	1971	5
52	1945	4
53	1960	4
54	1968	3
55	1944	3
56	1958	3
57	1955	3
58	1962	3
59	1943	3
60	1946	2
61	1954	2
62	1956	2
63	1964	2
64	1963	2
65	1942	2

65	1942	2
66	1965	2
67	1969	2
68	1970	2
69	1959	1
70	1961	1
71	1947	1
72	1966	1
73	1925	1

```
#number of distinct titles on the basis of year
plt.figure(figsize=(10,8))
data_release_year=data_final.groupby(['release_year']).agg({"title":"nunique"}).reset_ind
sns.lineplot(data=data_release_year, x='release_year', y='title')
plt.ylabel("No of movies released")
plt.xlabel("Release Year")
plt.show()
```



Most of the movies were released in the year 2018 followed by 2017 &2019.

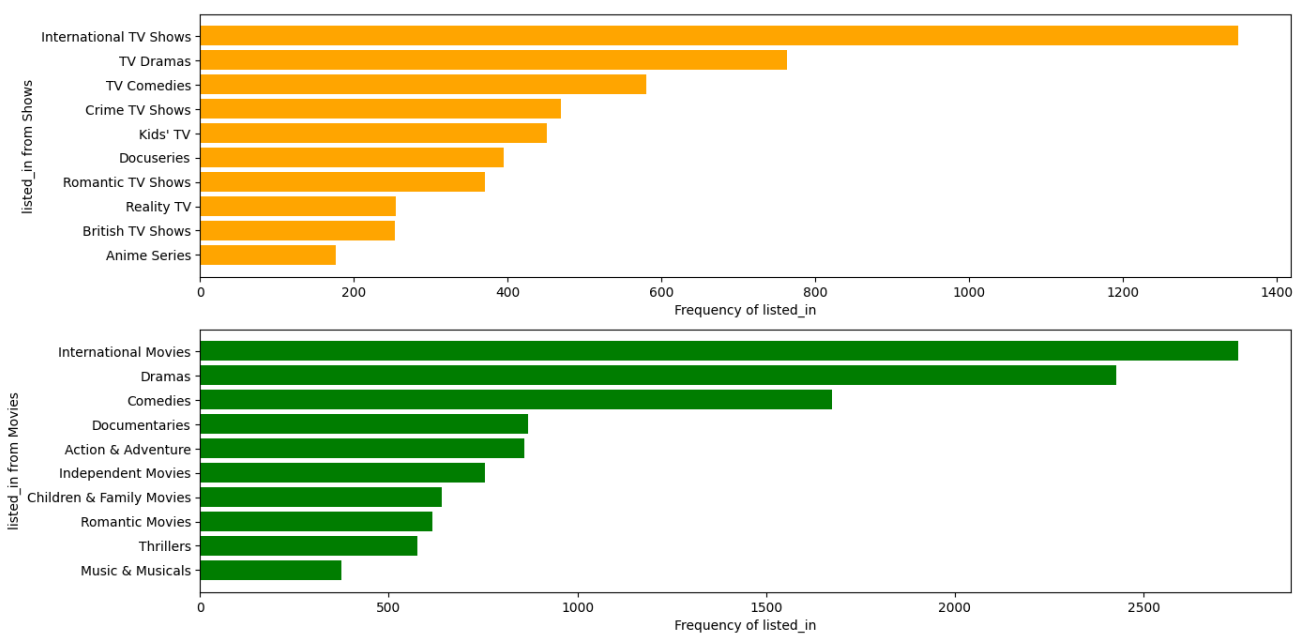
There has been decline in the recently released movies available in the platform, this may be due to the reason the content were bit added yet.

```
data_final.head()
```

	title	cast	director	country	listed_in	show_id	type	date_added	rel
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021	
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas	s2	TV Show	September 24, 2021	
3	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Mysteries	s2	TV Show	September 24, 2021	
4	Blood &	Khosi	Unknown	South	International		TV	September	

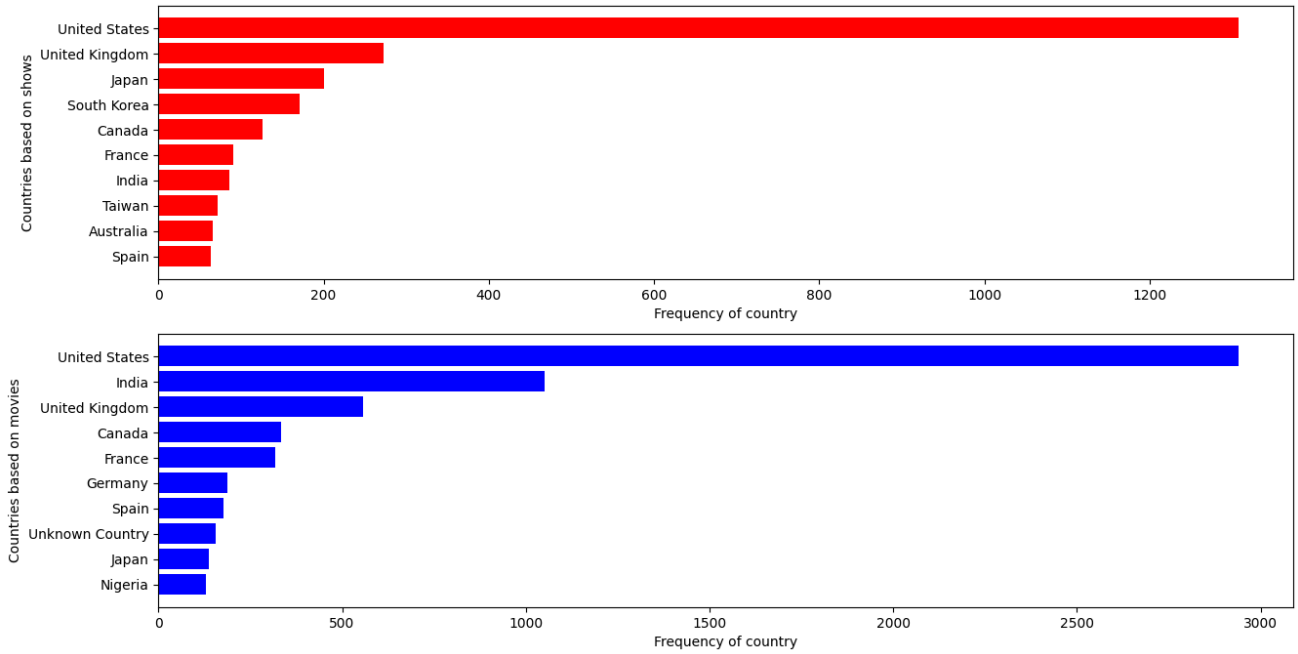
Seperate Analysis Based on Movies/TV Shows

```
data_show=data_final[data_final['type']=='TV Show']
data_movies=data_final[data_final['type']=='Movie']


plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_listed_in=data_show.groupby(['listed_in']).agg({"title":"nunique"}).reset_index().so
plt.barh(data_listed_in[::-1]['listed_in'], data_listed_in[::-1]['title'],color=['orange'
plt.xlabel('Frequency of listed_in')
plt.ylabel('listed_in from Shows')


plt.subplot(2,1,2)
data_listed_in=data_movies.groupby(['listed_in']).agg({"title":"nunique"}).reset_index().
plt.barh(data_listed_in[::-1]['listed_in'], data_listed_in[::-1]['title'],color=['Green']
plt.xlabel('Frequency of listed_in')
plt.ylabel('listed_in from Movies')
plt.show()
```



On the overall International Movies/TV Shows, Dramas , Comedies are the Top 3 contents from Movies and TV Show commonly.

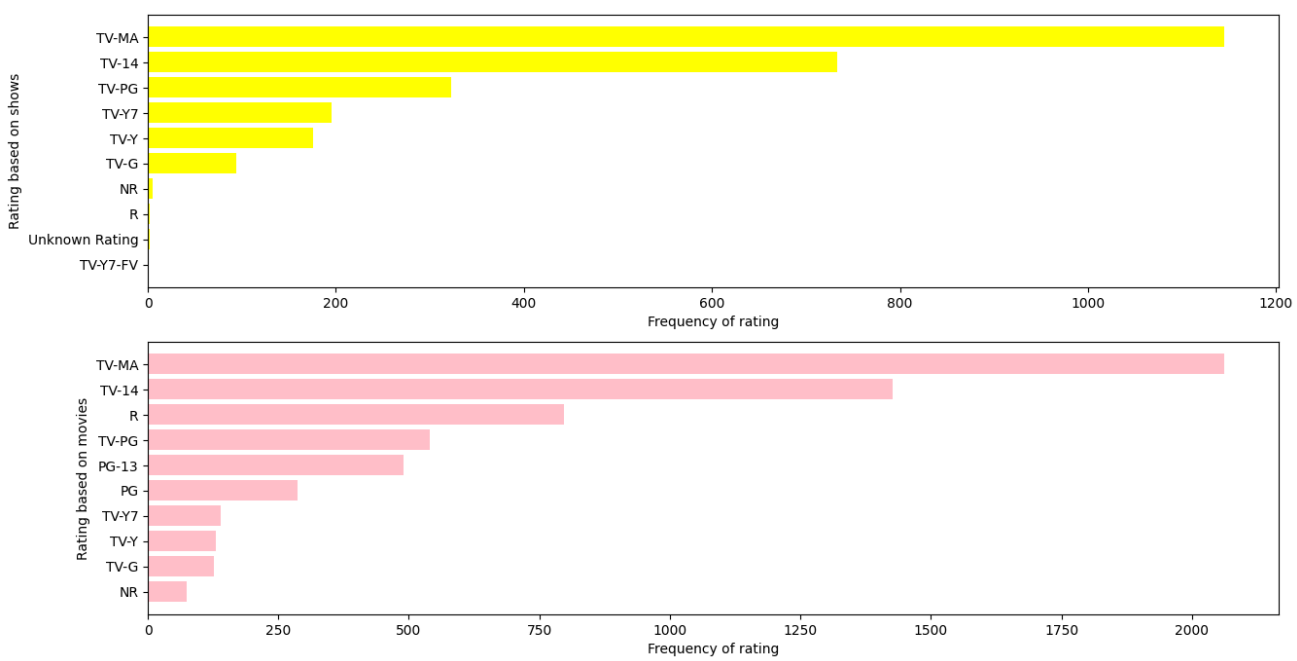
```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_country=data_show.groupby(['country']).agg({"title":"nunique"}).reset_index().sort_v
plt.barh(data_country[::-1]['country'], data_country[::-1]['title'],color=['Red'])
plt.xlabel('Frequency of country')
plt.ylabel('Countries based on shows')


plt.subplot(2,1,2)
data_country=data_movies.groupby(['country']).agg({"title":"nunique"}).reset_index().sort
plt.barh(data_country[::-1]['country'], data_country[::-1]['title'],color=['Blue'])
plt.xlabel('Frequency of country')
plt.ylabel('Countries based on movies')
plt.show()
```



Top 3 Countries based on content type shows are United States, United Kingdom and Japan.

Top 3 Countries based on content type movies are United States, India and. United Kingdom.

```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_rating=data_show.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_val
plt.barh(data_rating[::-1]['rating'], data_rating[::-1]['title'],color=['yellow'])
plt.xlabel('Frequency of rating')
plt.ylabel('Rating based on shows')


plt.subplot(2,1,2)
data_rating=data_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_v
plt.barh(data_rating[::-1]['rating'], data_rating[::-1]['title'],color=['pink'])
plt.xlabel('Frequency of rating')
plt.ylabel('Rating based on movies')
plt.show()
```



Most of the Shows were rated as TV-MA, TV-14, TV-PG

Most of the Movies were rated as TV-MA, TV-14, R

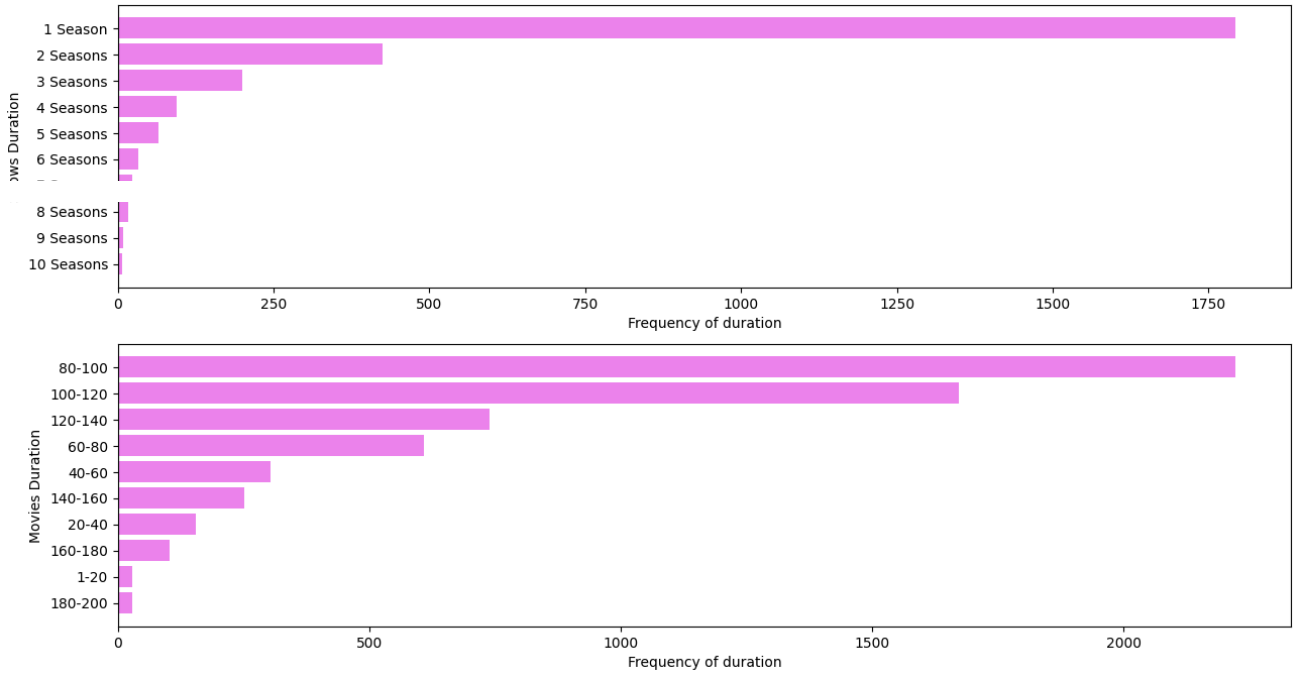
```
import matplotlib.cm as cm

# Get the default colormap
print(plt.colormaps())
```

```
['magma', 'inferno', 'plasma', 'viridis', 'cividis', 'twilight', 'twilight_shifted',
```

```
data_min_movies=data_min[data_min['type']=='Movie']


plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_duration=data_show.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort
plt.barh(data_duration[::-1]['duration'], data_duration[::-1]['title'],color=['violet'])
plt.xlabel('Frequency of duration')
plt.ylabel('Shows Duration')
```

```
plt.subplot(2,1,2)
data_duration=data_min_movies.groupby(['duration']).agg({"title":"nunique"}).reset_index(
plt.barh(data_duration[::-1]['duration'], data_duration[::-1]['title'],color=['violet'])
plt.xlabel('Frequency of duration')
plt.ylabel('Movies Duration')
plt.show()
```

```
<ipython-input-110-e88fb1648575>:11: FutureWarning: The default of observed=False is
  data_duration=data_min_movies.groupby(['duration']).agg({"title":"nunique"}).reset_
```



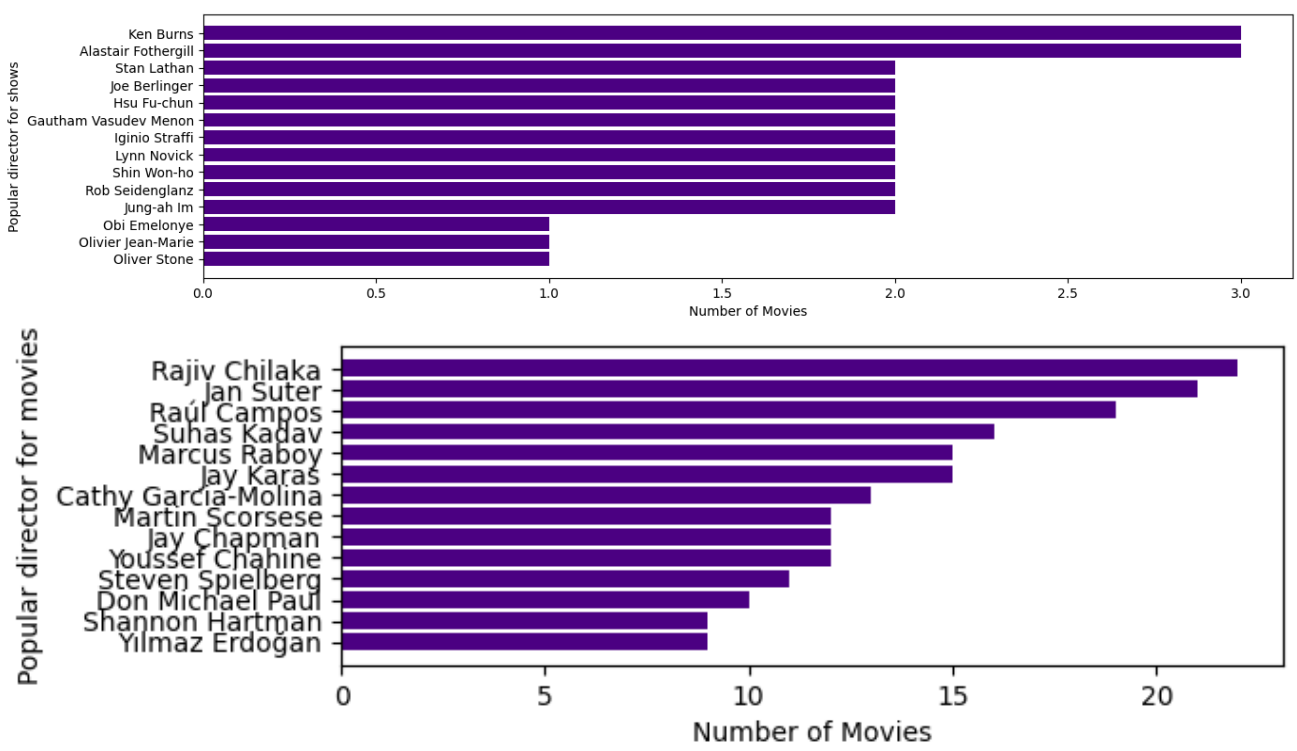
Most of the Show contents are 1 Season duration.

Most of the Movies contents are 80-100 & 100-120 duration.

```
data_show.head()
```

	title	cast	director	country	listed_in	show_id	type	date_added	release_
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas	s2	TV Show	September 24, 2021	
3	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Mysteries	s2	TV Show	September 24, 2021	
4	Blood &	Khosi	Unknown	South	International	s2	TV	September	

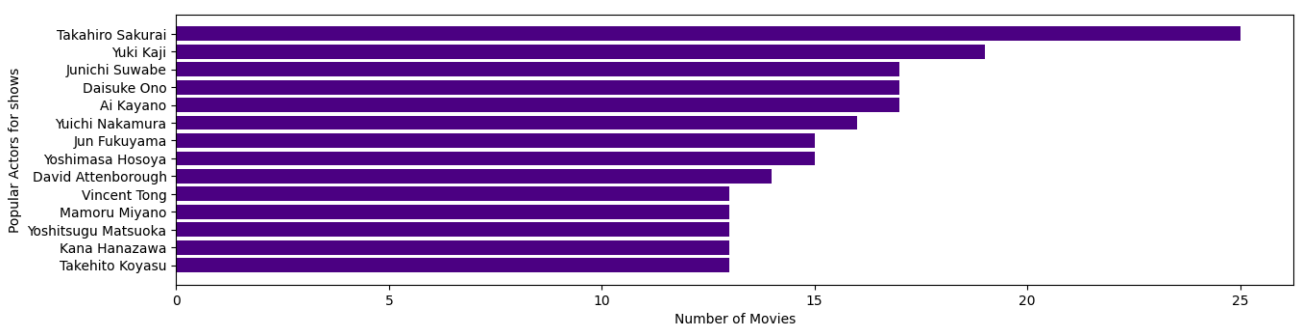
```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_dir=data_show.groupby(['director']).agg({"title":"nunique"}).reset_index().sort_valu
data_dir=data_dir[data_dir['director']!='Unknown Director']
plt.barh(data_dir[::-1]['director'], data_dir[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular director for shows')
plt.show()


plt.subplot(2,1,2)
data_dir=data_movies.groupby(['director']).agg({"title":"nunique"}).reset_index().sort_va
data_dir=data_dir[data_dir['director']!='Unknown Director']
plt.barh(data_dir[::-1]['director'], data_dir[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular director for movies')
plt.show()
```



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

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

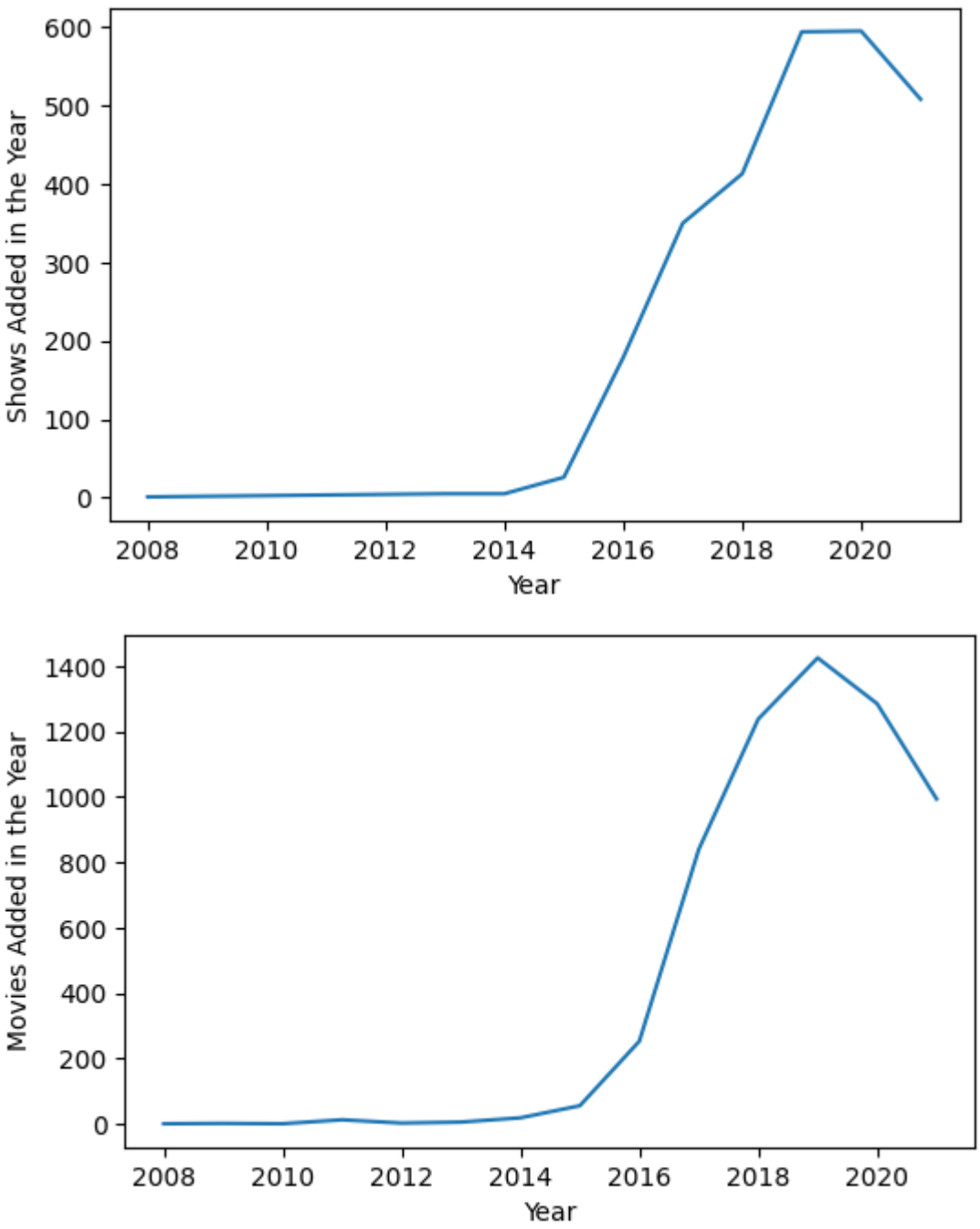
```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_cast=data_show.groupby(['cast']).agg({"title":"nunique"}).reset_index().sort_values(
data_cast=data_cast[data_cast['cast']!='Unknown Cast']
plt.barh(data_cast[::-1]['cast'], data_cast[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actors for shows')
plt.show()


plt.subplot(2,1,2)
data_cast=data_movies.groupby(['cast']).agg({"title":"nunique"}).reset_index().sort_value
data_cast=data_cast[data_cast['cast']!='Unknown Cast']
plt.barh(data_cast[::-1]['cast'], data_cast[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actors for movies')
plt.show()
```



akahiro Sakurai,Yuki Kaji and Junichi Suwabe actors are the most popular actors across TV Shows

Anupam Kher, Shah Rukh Khan, Om Puri are the most popular actors across movies.

```
plt.figure(figsize=(6,8))

plt.subplot(2,1,1)
data_year=data_show.groupby(['Content_Added_Year']).agg({"title":"nunique"}).reset_index(
sns.lineplot(data=data_year, x='Content_Added_Year', y='title')
plt.ylabel("Shows Added in the Year")
plt.xlabel("Year")
plt.show()

plt.figure(figsize=(6,8))
plt.subplot(2,1,2)
data_year=data_movies.groupby(['Content_Added_Year']).agg({"title":"nunique"}).reset_inde
sns.lineplot(data=data_year, x='Content_Added_Year', y='title')
plt.ylabel("Movies Added in the Year")
plt.xlabel("Year")
plt.show()
```

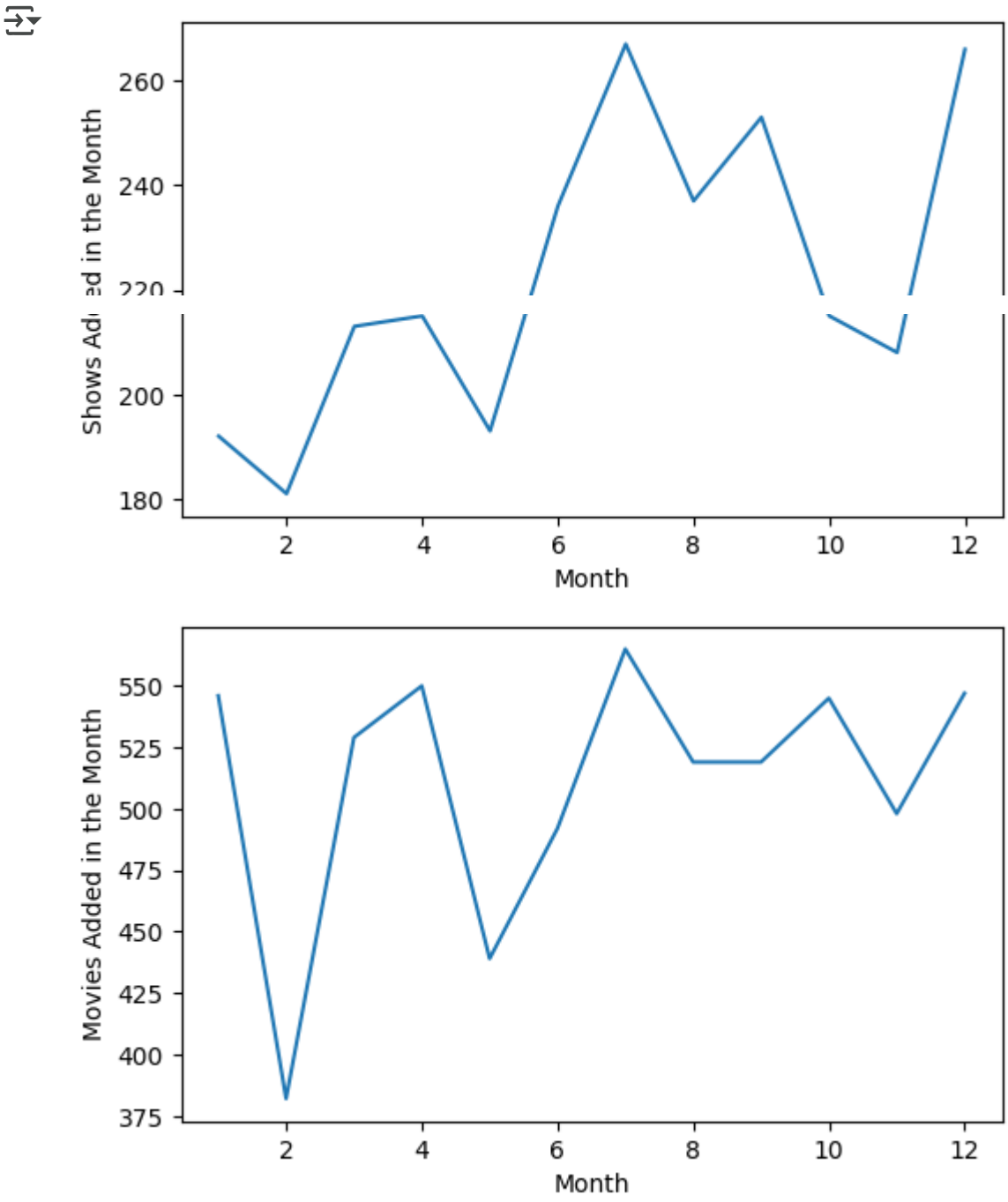


The Tv Show content on netflix was increasing until 2019, after which it started to decline for the recent years.

The Movies content on netflix was increasing until 2019 and constant for 2 years till 2021, after which it started to decline for the recent years.

```
#number of distinct titles on the basis of year
plt.figure(figsize=(6,8))
plt.subplot(2,1,1)
data_month=data_show.groupby(['Content_Added_Month']).agg({"title":"nunique"}).reset_inde
sns.lineplot(data=data_month, x='Content_Added_Month', y='title')
plt.ylabel("Shows Added in the Month")
plt.xlabel("Month")
plt.show()
```

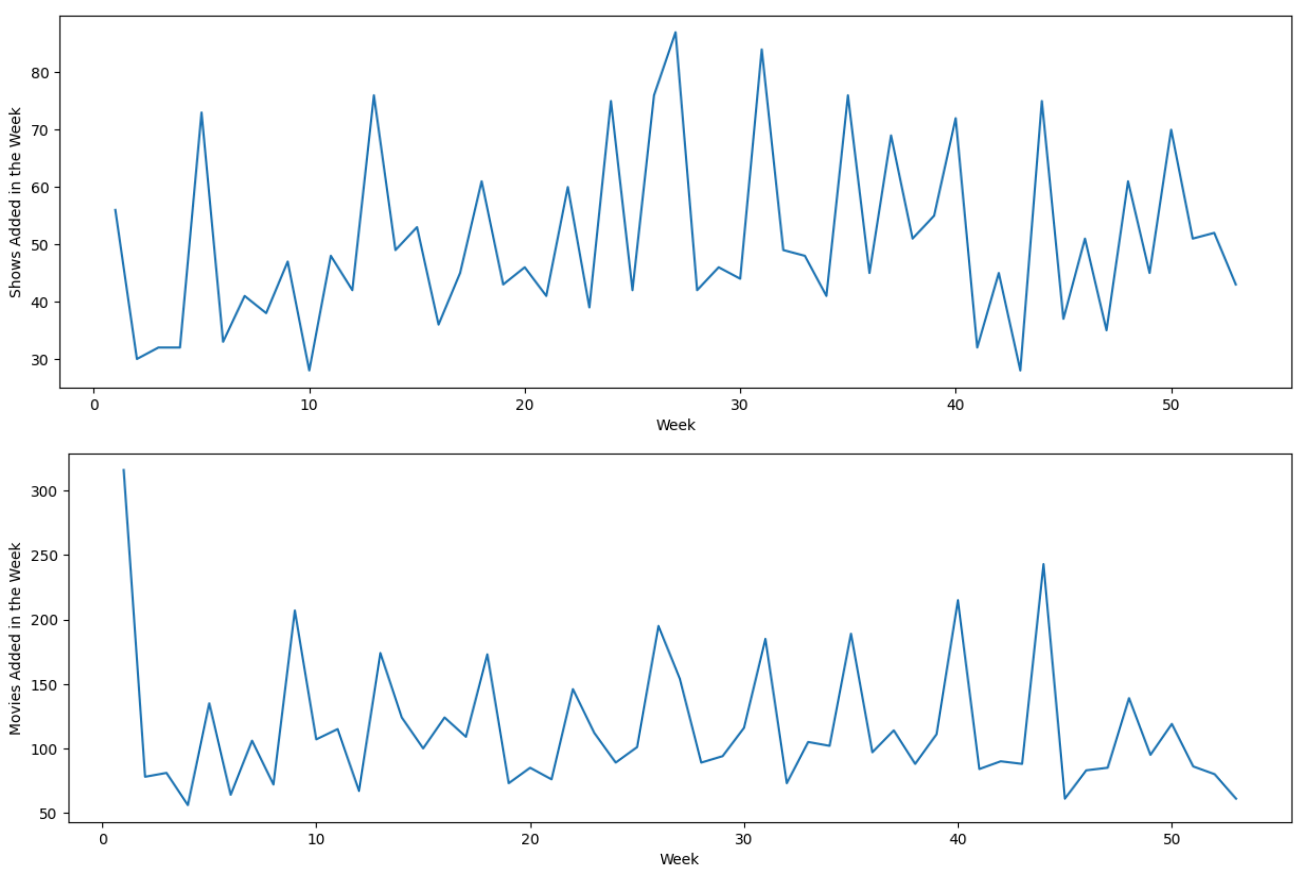
```
plt.figure(figsize=(6,8))
plt.subplot(2,1,2)
data_month=data_movies.groupby(['Content_Added_Month']).agg({"title":"nunique"}).reset_in
sns.lineplot(data=data_month, x='Content_Added_Month', y='title')
plt.ylabel("Movies Added in the Month")
plt.xlabel("Month")
plt.show()
```



Tv Show content in netflix were mostly addded in the mid and last months of the year i.e July & December

Movies content in netflix were mostly addded in the first,last and mid months of the year i.e January, December & July

```
plt.figure(figsize=(15,10))
plt.subplot(2,1,1)
data_week=data_show.groupby(['Content_Added_Week']).agg({"title":"nunique"}).reset_index(
sns.lineplot(data=data_week, x='Content_Added_Week', y='title')
plt.ylabel("Shows Added in the Week")
plt.xlabel("Week")
plt.show()

plt.figure(figsize=(15,10))
plt.subplot(2,1,2)
data_week=data_movies.groupby(['Content_Added_Week']).agg({"title":"nunique"}).reset_inde
sns.lineplot(data=data_week, x='Content_Added_Week', y='title')
plt.ylabel("Movies Added in the Week")
plt.xlabel("Week")
plt.show()
```

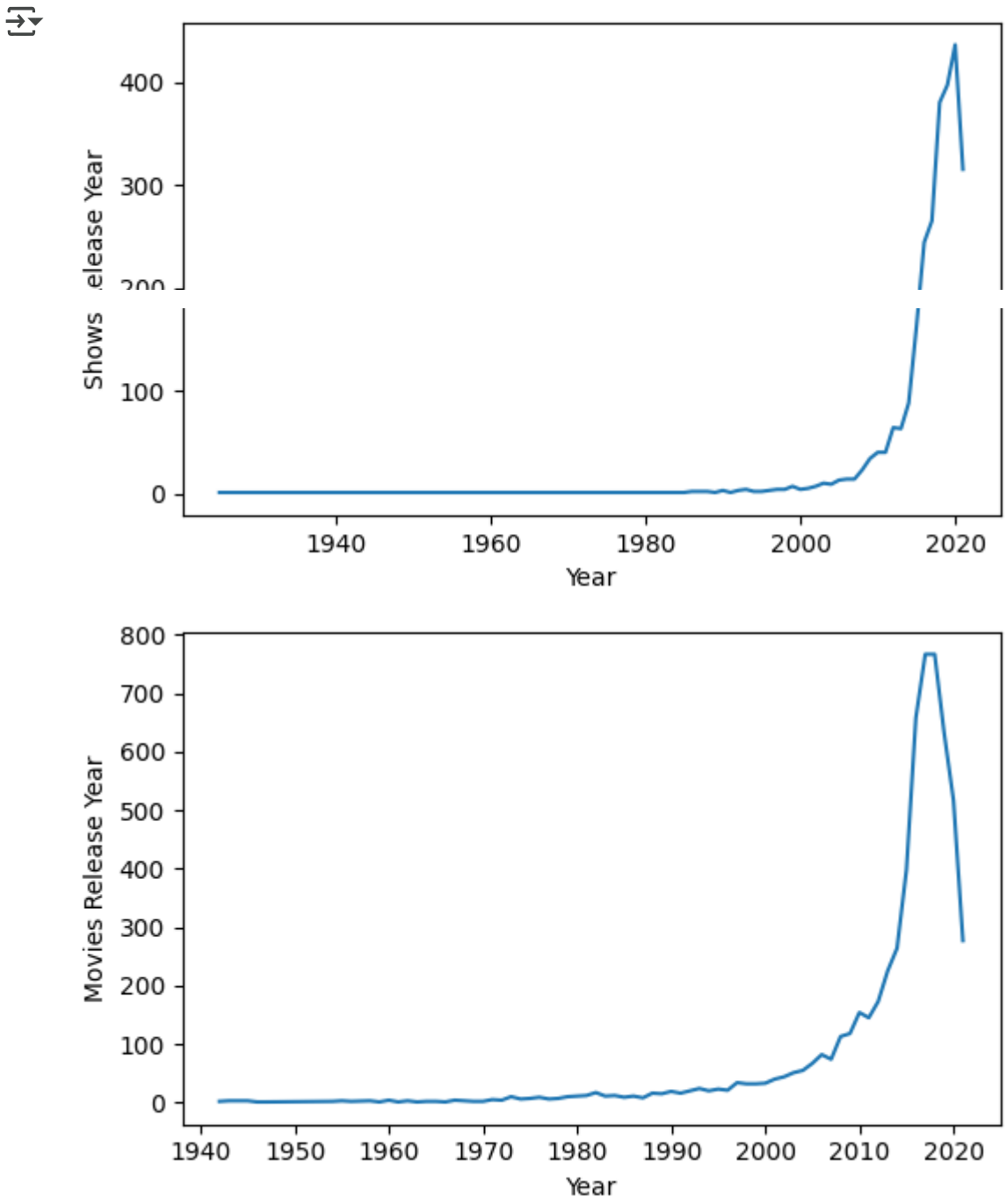


Most of the Tv Shows were added between the 26-32 week of the year

Most of the Movies were added between the first week and 44th week of the year

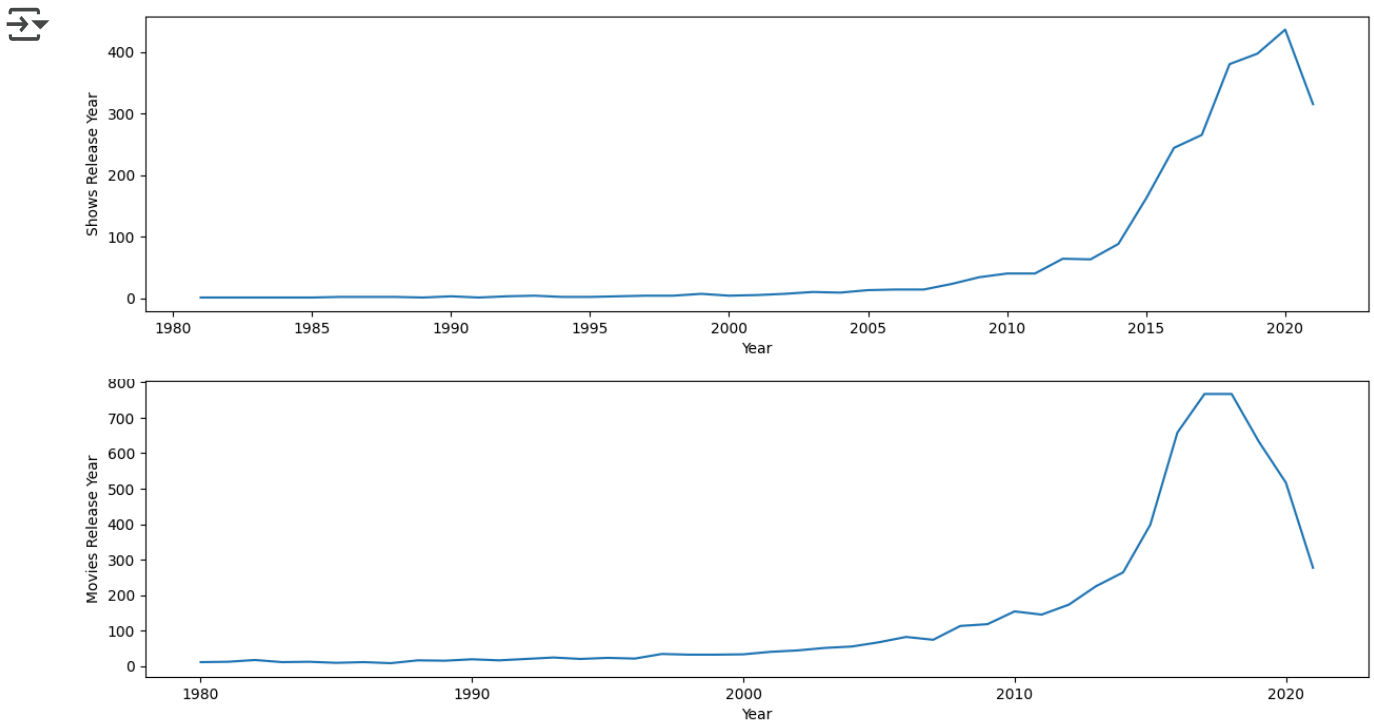
```
plt.figure(figsize=(6,8))
plt.subplot(2,1,1)
data_release_year=data_show.groupby(['release_year']).agg({"title":"nunique"}).reset_inde
sns.lineplot(data=data_release_year, x='release_year', y='title')
plt.ylabel("Shows Release Year")
plt.xlabel("Year")
plt.show()

plt.figure(figsize=(6,8))
```

```
plt.subplot(2,1,2)
data_release_year=data_movies.groupby(['release_year']).agg({"title":"nunique"}).reset_in
sns.lineplot(data=data_release_year, x='release_year', y='title')
plt.ylabel("Movies Release Year")
plt.xlabel("Year")
plt.show()
```




```
plt.figure(figsize=(15,8))
plt.subplot(2,1,1)
data_release_year=data_show[data_show['release_year']>=1980].groupby(['release_year']).ag
sns.lineplot(data=data_release_year, x='release_year', y='title')
plt.ylabel("Shows Release Year")
plt.xlabel("Year")
plt.show()

plt.figure(figsize=(15,8))
plt.subplot(2,1,2)
```

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



Most of the TV shows actual released year were 2019

Most of the Movies actual released year were between 2017-2018

```
data_final.head()
```

	title	cast	director	country	listed_in	show_id	type	date_added	rel
0	Dick Johnson Is Dead	Unknown Cast	Kirsten Johnson	United States	Documentaries	s1	Movie	September 25, 2021	
1	Blood & Water	Ama Qamata	Unknown Director	South Africa	International TV Shows	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Dramas	s2	TV Show	September 24, 2021	
3	Blood & Water	Ama Qamata	Unknown Director	South Africa	TV Mysteries	s2	TV Show	September 24, 2021	
4	Blood &	Khosi	Unknown	South	International		TV	September	

Country-Wise Analysis

USA

```
data_usa_shows=data_final[(data_final['country']=='United States') & (data_final['type']=
data_usa_shows.nunique()
```

	0
title	1308
cast	6854
director	108
country	1
listed_in	22
show_id	1308
type	1
date_added	720
release_year	41
rating	8
duration	15
Content_Added_Date	704
Content_Added_Month	12
Content_Added_Week	53
Content_Added_Year	10

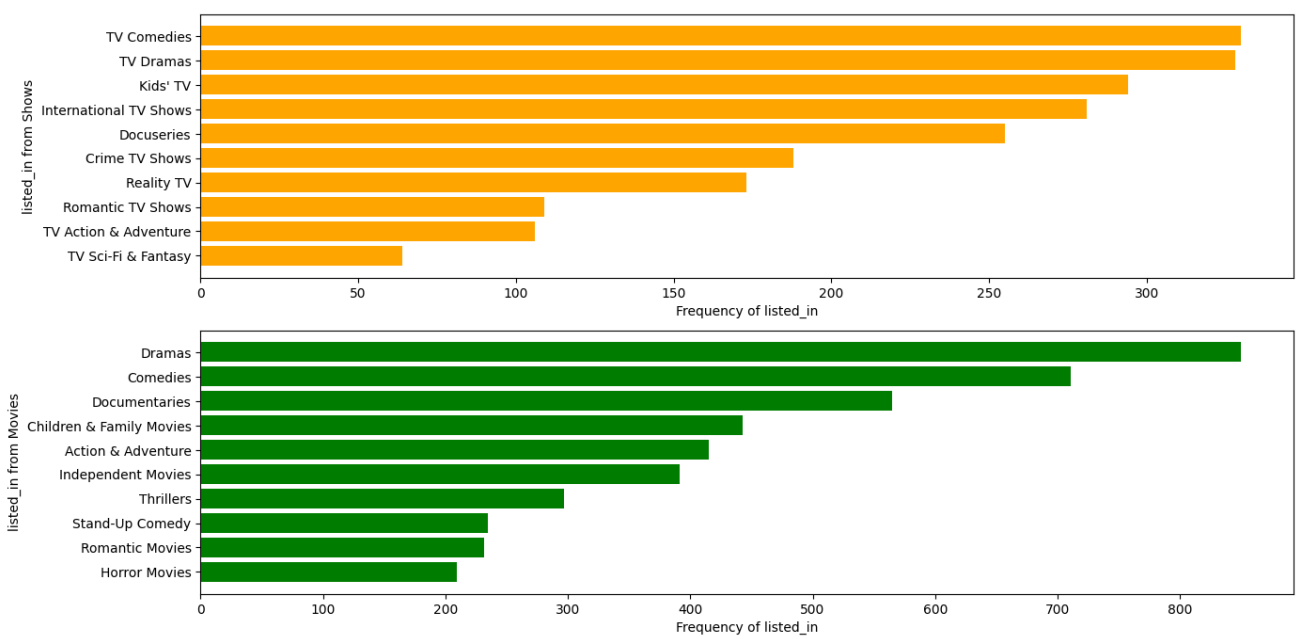
dtype: int64

```
data_usa_movies=data_final[(data_final['country']=='United States') & (data_final['type']
data_usa_movies.nunique()
```

	0
title	2941
cast	11964
director	2347
country	1
listed_in	20
show_id	2941
type	1
date_added	1168
release_year	69
rating	15
duration	174
Content_Added_Date	1168
Content_Added_Month	12
Content_Added_Week	53
Content_Added_Year	14

dtype: int64

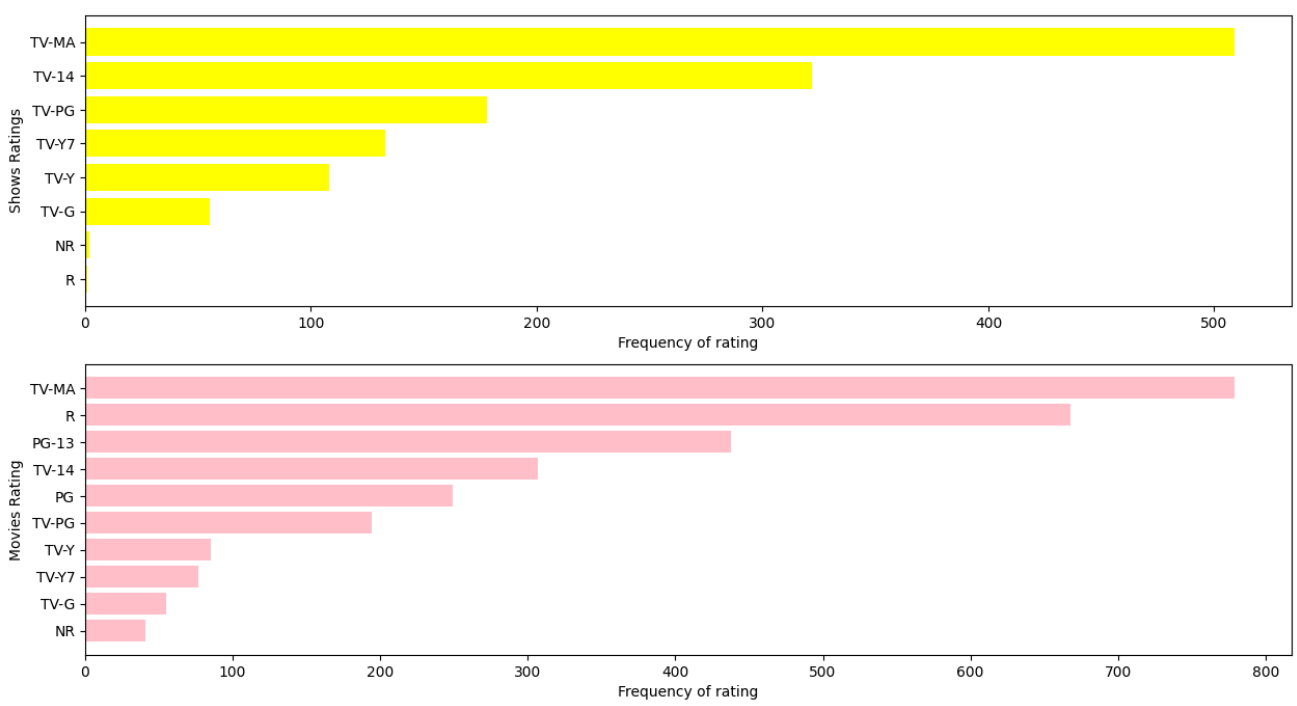
```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_listed_in=data_usa_shows.groupby(['listed_in']).agg({"title":"nunique"}).reset_index
plt.barh(data_listed_in[::-1]['listed_in'], data_listed_in[::-1]['title'],color=['orange'
plt.xlabel('Frequency of listed_in')
plt.ylabel('listed_in from Shows')


plt.subplot(2,1,2)
data_listed_in=data_usa_movies.groupby(['listed_in']).agg({"title":"nunique"}).reset_inde
plt.barh(data_listed_in[::-1]['listed_in'], data_listed_in[::-1]['title'],color=['Green']
plt.xlabel('Frequency of listed_in')
plt.ylabel('listed_in from Movies')
plt.show()
```



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

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

```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_rating=data_usa_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sor
plt.barh(data_rating[::-1]['rating'], data_rating[::-1]['title'],color=['yellow'])
plt.xlabel('Frequency of rating')
plt.ylabel('Shows Ratings')


plt.subplot(2,1,2)
data_rating=data_usa_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().so
plt.barh(data_rating[::-1]['rating'], data_rating[::-1]['title'],color=['pink'])
plt.xlabel('Frequency of rating')
plt.ylabel('Movies Rating')
plt.show()
```



In USA, TV Show ratings were TV-MA, TV-14, TV-PG and TV-Y7. Movies were TV-MA, R, PG-13 and TV-14

```
data_min_usa_movies=data_min[(data_final['country']=='United States')]
data_min_usa_movies['country'] = data_min_usa_movies['country'].str.replace(',', '')
data_min_usa_movies['country'].value_counts()
```

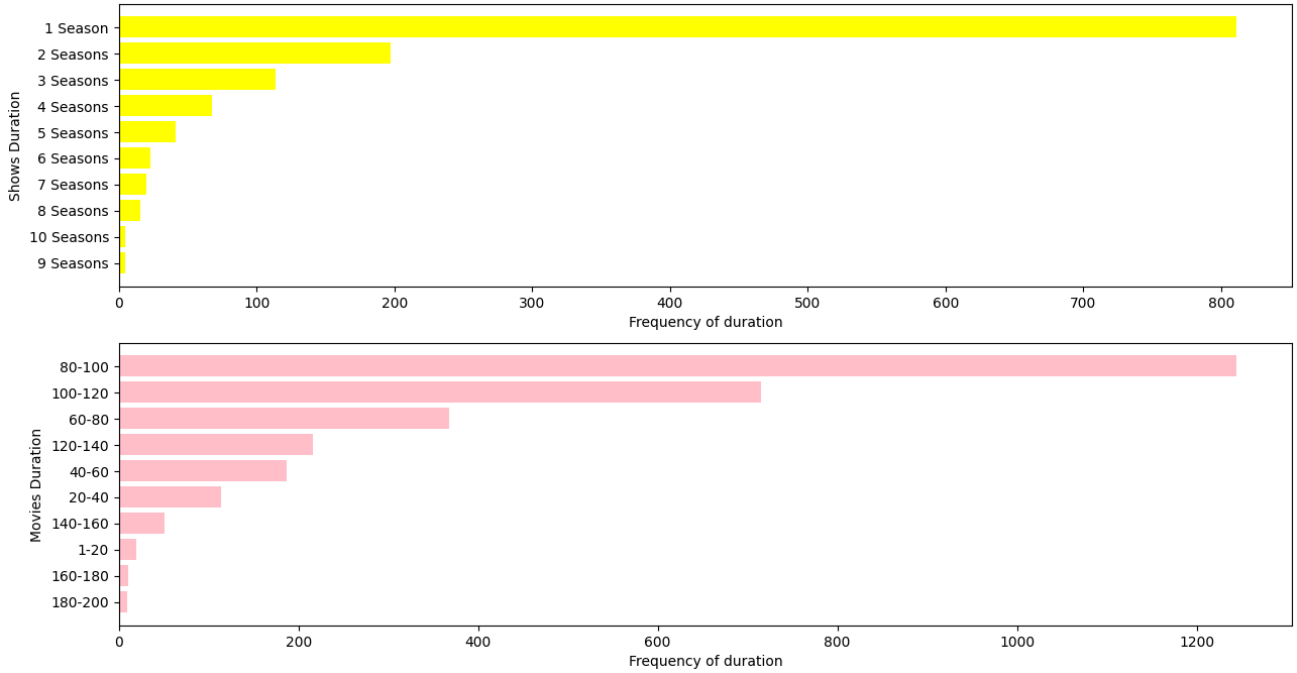
```
<ipython-input-124-0da6da5f2fce>:1: UserWarning: Boolean Series key will be reindexed
  data_min_usa_movies=data_min[(data_final['country']=='United States')]
<ipython-input-124-0da6da5f2fce>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  data_min_usa_movies['country'] = data_min_usa_movies['country'].str.replace(',', ''
```

country	count
United States	46928

dtype: int64

```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_rating=data_usa_shows.groupby(['duration']).agg({"title":"nunique"}).reset_index().s
plt.barh(data_rating[::-1]['duration'], data_rating[::-1]['title'],color=['yellow'])
plt.xlabel('Frequency of duration')
plt.ylabel('Shows Duration')


plt.subplot(2,1,2)
data_rating=data_min_usa_movies.groupby(['duration']).agg({"title":"nunique"}).reset_inde
plt.barh(data_rating[::-1]['duration'], data_rating[::-1]['title'],color=['pink'])
plt.xlabel('Frequency of duration')
plt.ylabel('Movies Duration')
plt.show()
```

```
<ipython-input-125-dc110c738696>:11: FutureWarning: The default of observed=False is
  data_rating=data_min_usa_movies.groupby(['duration']).agg({"title":"nunique"}).rese
```

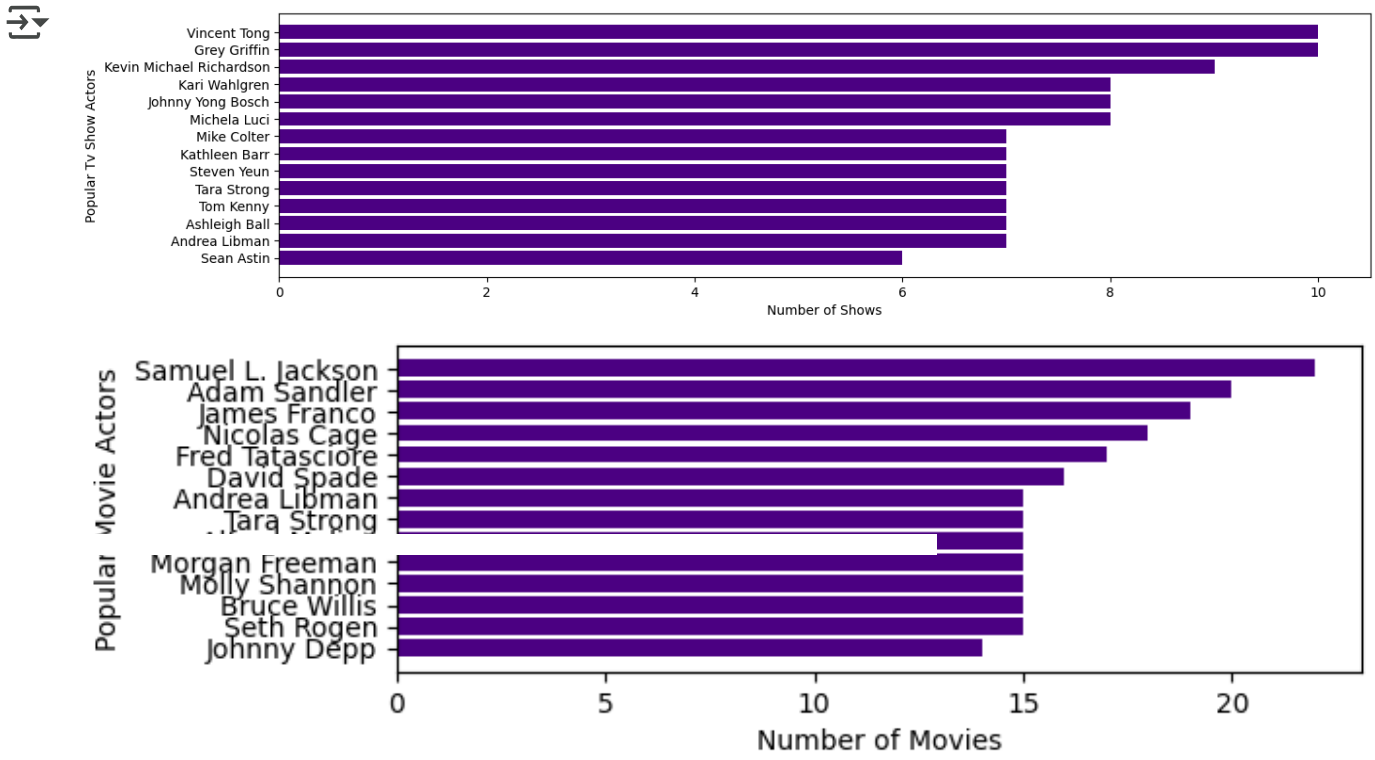


Across movies 80-100,100-120 is the ranges of minutes for which most movies lie.

Across Tv Shows 1 Season shows were dominated followed by 2 season shows.

```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_cast=data_usa_shows.groupby(['cast']).agg({"title":"nunique"}).reset_index().sort_va
data_cast=data_cast[data_cast['cast']!='Unknown Cast']
plt.barh(data_cast[::-1]['cast'], data_cast[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Shows')
plt.ylabel('Popular Tv Show Actors')
plt.show()


plt.subplot(2,1,2)
data_cast=data_usa_movies.groupby(['cast']).agg({"title":"nunique"}).reset_index().sort_v
data_cast=data_cast[data_cast['cast']!='Unknown Cast']
```

```
plt.barh(data_cast[::-1]['cast'], data_cast[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Movie Actors')
plt.show()
```

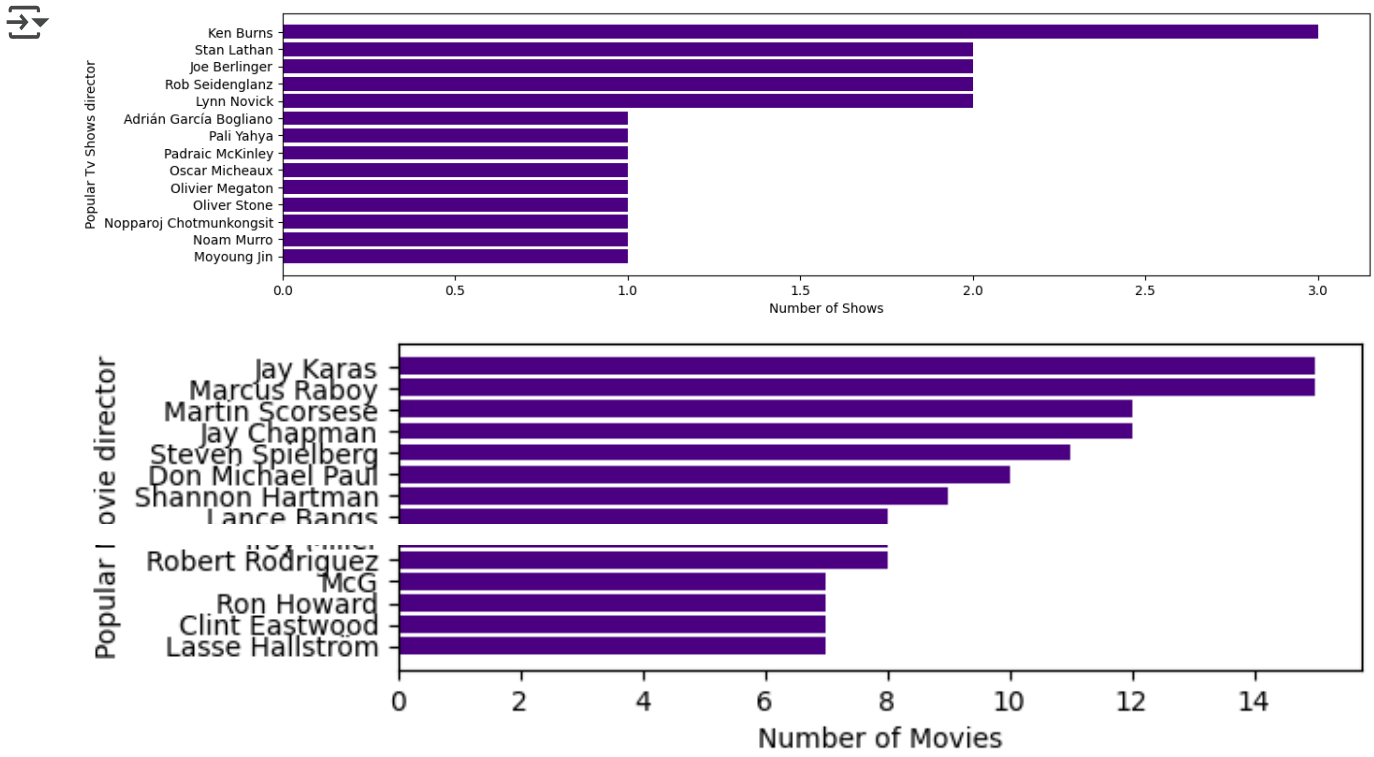


Vincent Tong, Grey Griffin and Kevin Michael Richardson are the popular actors across TV Shows in USA.

Samuel Jackson, Adam Sandler and James Franco are the popular actors across movies on Netflix in USA.

```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_cast=data_usa_shows.groupby(['director']).agg({"title":"nunique"}).reset_index().sor
data_cast=data_cast[data_cast['director']!='Unknown Director']
plt.barh(data_cast[::-1]['director'], data_cast[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Shows')
plt.ylabel('Popular Tv Shows director')
plt.show()


plt.subplot(2,1,2)
data_cast=data_usa_movies.groupby(['director']).agg({"title":"nunique"}).reset_index().so
data_cast=data_cast[data_cast['director']!='Unknown Director']
```

```
plt.barh(data_cast[::-1]['director'], data_cast[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Movie director')
plt.show()
```

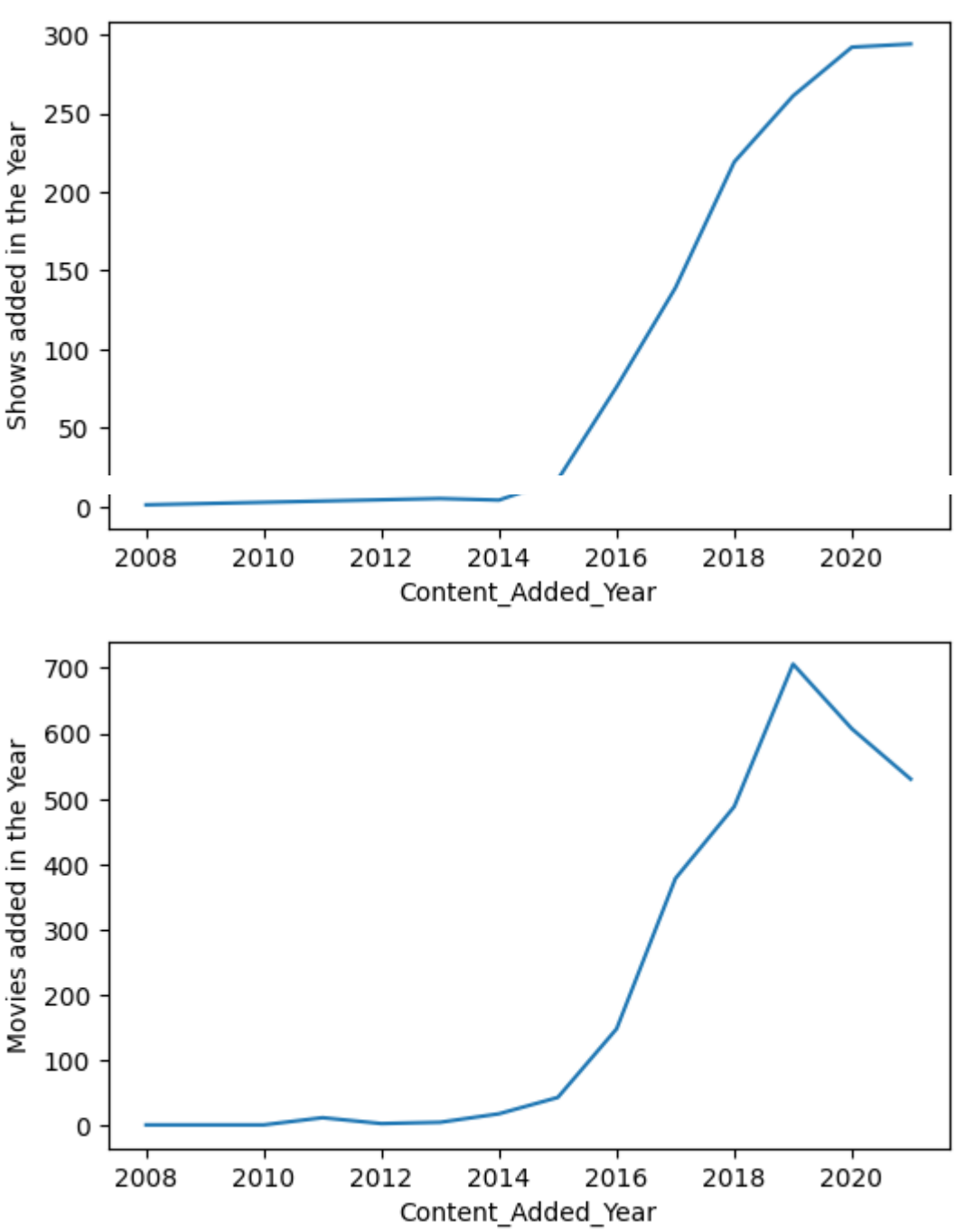


Ken Burns,Stan Lathan, Joe Barlinger, Rob Seidenglanz and Lynn Novick are the popular directors across TV Shows on Netflix in USA.

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

```
plt.figure(figsize=(6,8))

plt.subplot(2,1,1)
data_year=data_usa_shows.groupby(['Content_Added_Year']).agg({"title":"nunique"}).reset_i
sns.lineplot(data=data_year, x='Content_Added_Year', y='title')
plt.ylabel("Shows added in the Year")
plt.xlabel("Content_Added_Year")
plt.show()

plt.figure(figsize=(6,8))
plt.subplot(2,1,2)
data_year=data_usa_movies.groupby(['Content_Added_Year']).agg({"title":"nunique"}).reset_
sns.lineplot(data=data_year, x='Content_Added_Year', y='title')
plt.ylabel("Movies added in the Year")
```

```
plt.xlabel("Content_Added_Year")
plt.show()
```

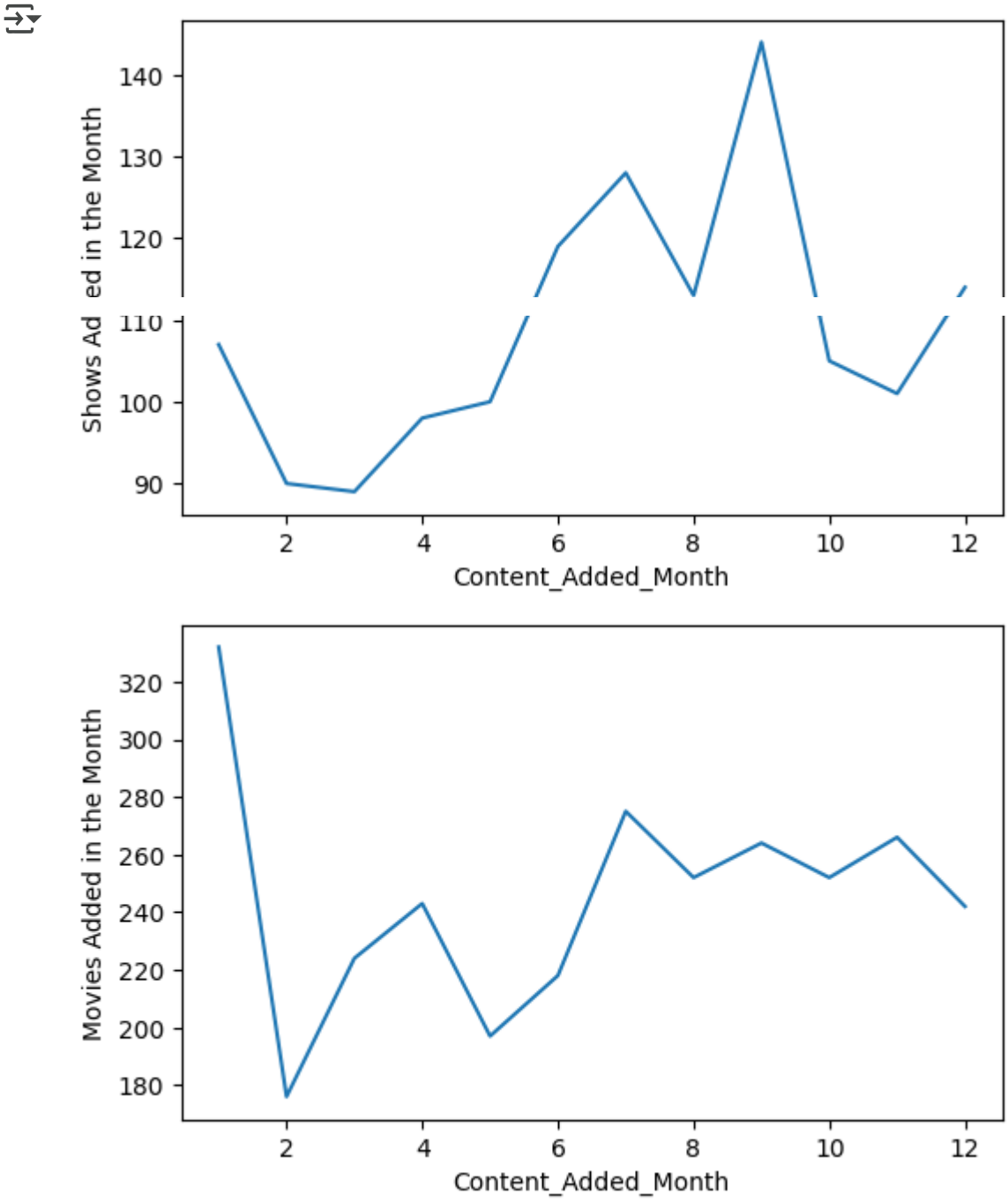


In USA, Tv shows addition remains constant for the recent years.

In USA, Movies addition in USA started to decline after 2019.

```
plt.figure(figsize=(6,8))
plt.subplot(2,1,1)
data_month=data_usa_shows.groupby(['Content_Added_Month']).agg({"title":"nunique"}).reset
sns.lineplot(data=data_month, x='Content_Added_Month', y='title')
plt.ylabel("Shows Added in the Month")
plt.xlabel("Content_Added_Month")
plt.show()
```

```
plt.figure(figsize=(6,8))
plt.subplot(2,1,2)
data_month=data_usa_movies.groupby(['Content_Added_Month']).agg({"title":"nunique"}).rese
sns.lineplot(data=data_month, x='Content_Added_Month', y='title')
plt.ylabel("Movies Added in the Month")
plt.xlabel("Content_Added_Month")
plt.show()
```

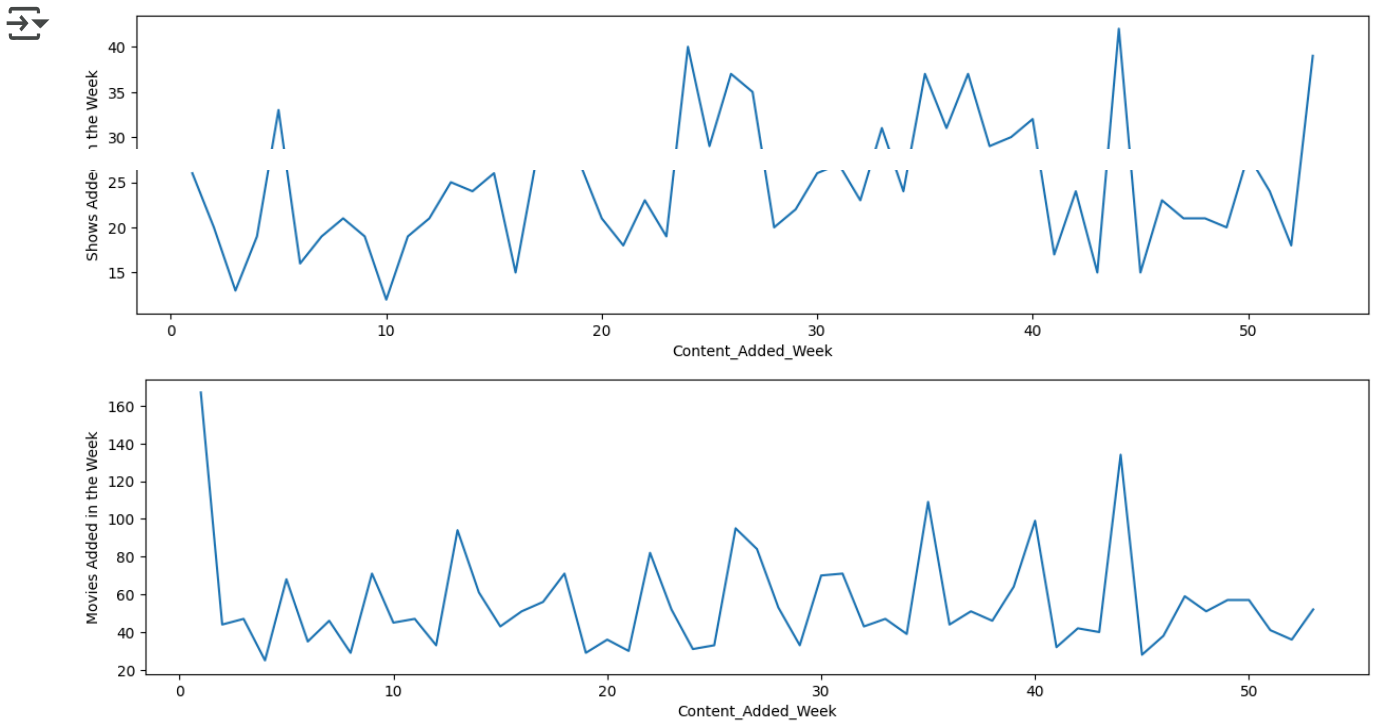


In USA, Tv shows were added mostly on September and July months of the year

In USA, Movies were added mostly on January and July months of the year.

```
plt.figure(figsize=(15,8))
plt.subplot(2,1,1)
data_week=data_usa_shows.groupby(['Content_Added_Week']).agg({"title":"nunique"}).reset_i
sns.lineplot(data=data_week, x='Content_Added_Week', y='title')
```

```
plt.ylabel("Shows Added in the Week")
plt.xlabel("Content_Added_Week")
plt.show()

plt.figure(figsize=(15,8))
plt.subplot(2,1,2)
data_week=data_usa_movies.groupby(['Content_Added_Week']).agg({"title":"nunique"}).reset_
sns.lineplot(data=data_week, x='Content_Added_Week', y='title')
plt.ylabel("Movies Added in the Week")
plt.xlabel("Content_Added_Week")
plt.show()
```

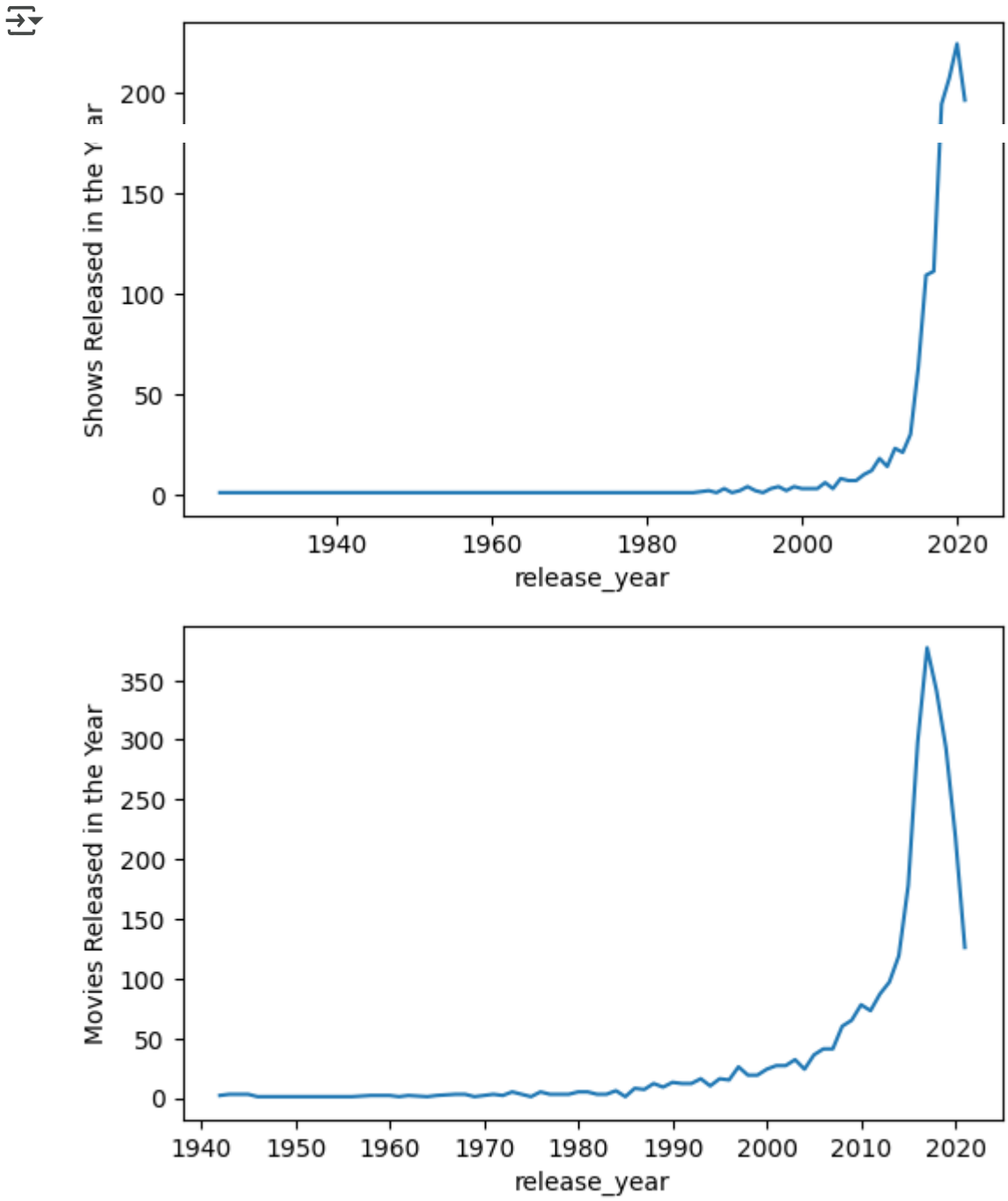


In USA, Tv shows were added between these ranges 42-44 & 22-25 of week in the year

In USA, Movies were added between these ranges 0-2 & 42-44 of week in the year

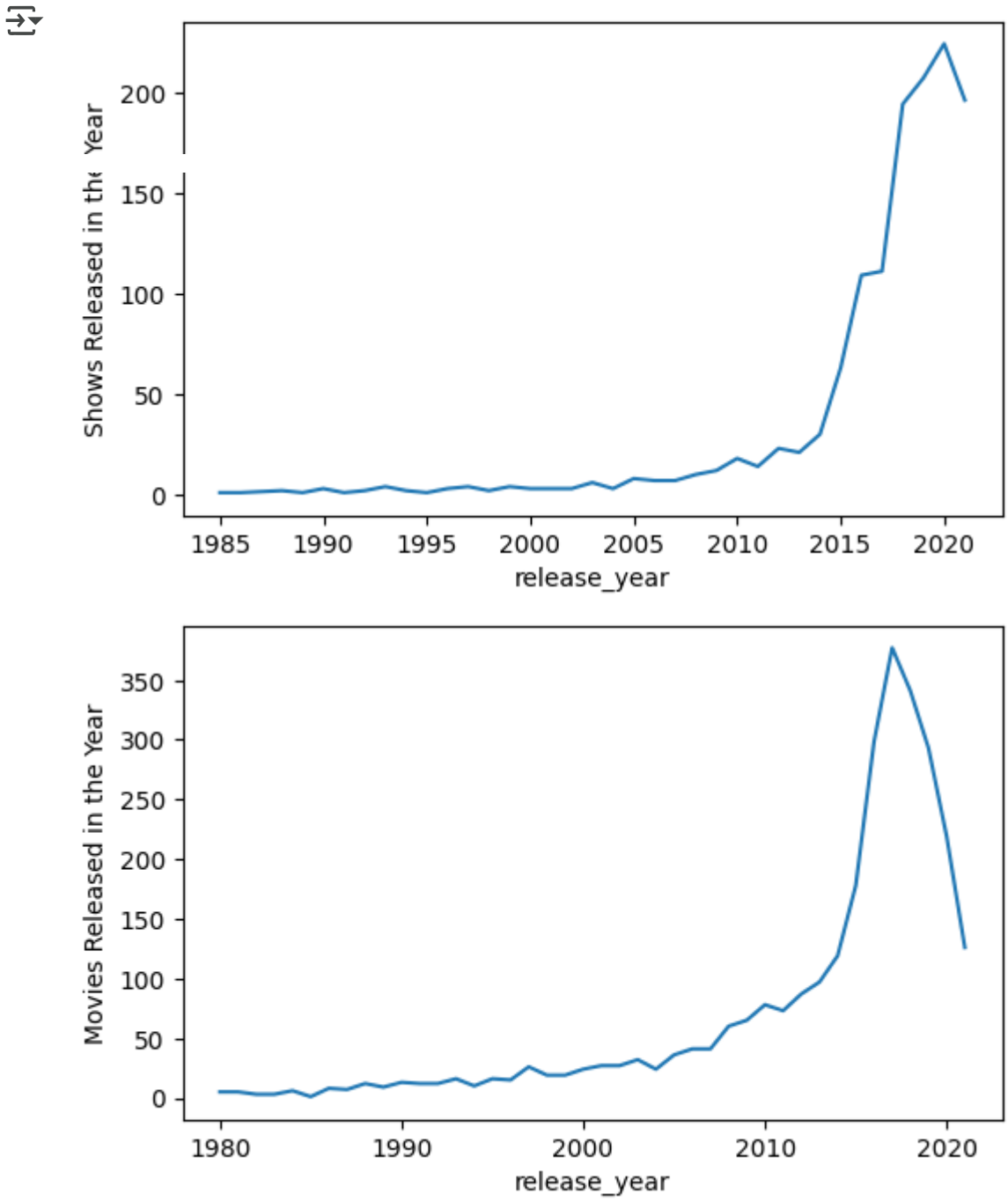
```
plt.figure(figsize=(6,8))
plt.subplot(2,1,1)
data_release_year=data_usa_shows.groupby(['release_year']).agg({"title":"nunique"}).reset
sns.lineplot(data=data_release_year, x='release_year', y='title')
```

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

plt.figure(figsize=(6,8))
plt.subplot(2,1,2)
data_release_year=data_usa_movies.groupby(['release_year']).agg({"title":"nunique"}).rese
sns.lineplot(data=data_release_year, x='release_year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("release_year")
plt.show()
```




```
plt.figure(figsize=(6,8))
plt.subplot(2,1,1)
data_release_year=data_usa_shows[data_usa_shows['release_year']>=1980].groupby(['release_
sns.lineplot(data=data_release_year, x='release_year', y='title')
plt.ylabel("Shows Released in the Year")
```

```
plt.xlabel("release_year")
plt.show()

plt.figure(figsize=(6,8))
plt.subplot(2,1,2)
data_release_year=data_usa_movies[data_usa_movies['release_year']>=1980].groupby(['releas
sns.lineplot(data=data_release_year, x='release_year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("release_year")
plt.show()
```



In USA, both recently released TV shows and Movies are declining which means less. Not adding recently released movies to the platform may also be a reason for this.

```
#Analysing Popular combination of actors and directors
data_usa_shows['Actor_Director_Combination'] = data_usa_shows.cast.str.cat(data_usa_shows
```

```
data_usa_shows_subset=data_usa_shows[data_usa_shows['cast']!='Unknown Cast']
data_usa_shows_subset=data_usa_shows_subset[data_usa_shows_subset['director']!='Unknown D
data_usa_shows_subset.head()
```

```
<ipython-input-133-c2893dab49b0>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  data_usa_shows['Actor_Director_Combination'] = data_usa_shows.cast.str.cat(data_usa
```

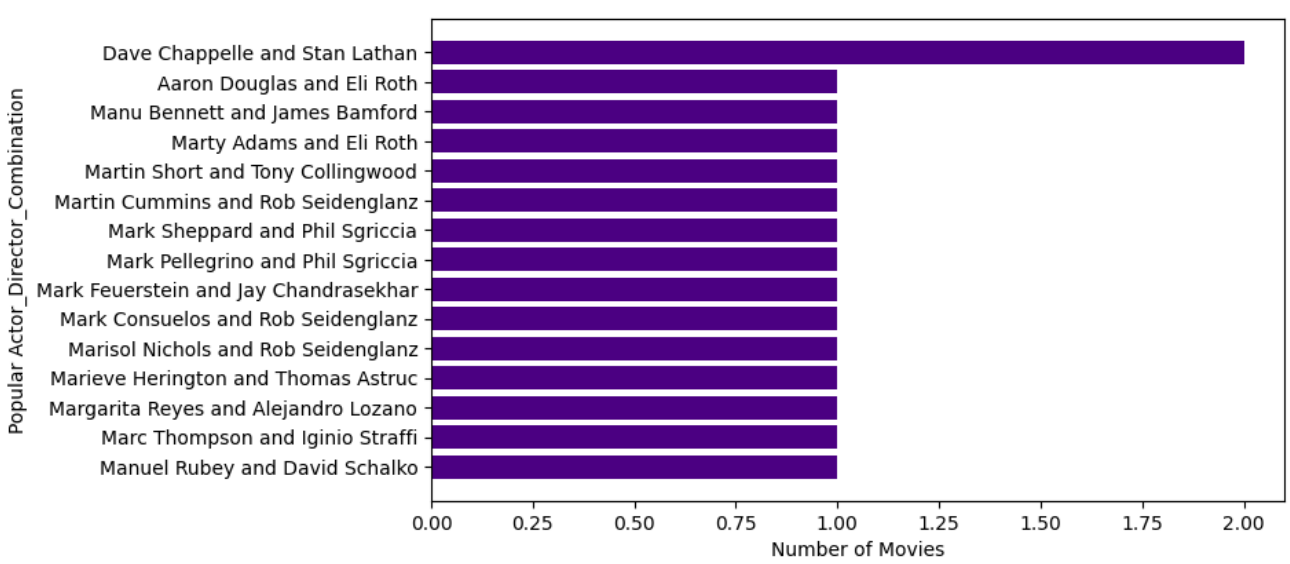
	title	cast	director	country	listed_in	show_id	type	date_added	releas
111	Midnight Mass	Kate Siegel	Mike Flanagan	United States	TV Dramas	s6	TV Show	September 24, 2021	
112	Midnight Mass	Kate Siegel	Mike Flanagan	United States	TV Horror	s6	TV Show	September 24, 2021	
113	Midnight Mass	Kate Siegel	Mike Flanagan	United States	TV Mysteries	s6	TV Show	September 24, 2021	
114	Midnight Mass	Zach Gilford	Mike Flanagan	United States	TV Dramas	s6	TV Show	September 24, 2021	
115	Midnight Mass	Zach Gilford	Mike Flanagan	United States	TV Horror	s6	TV Show	September 24, 2021	

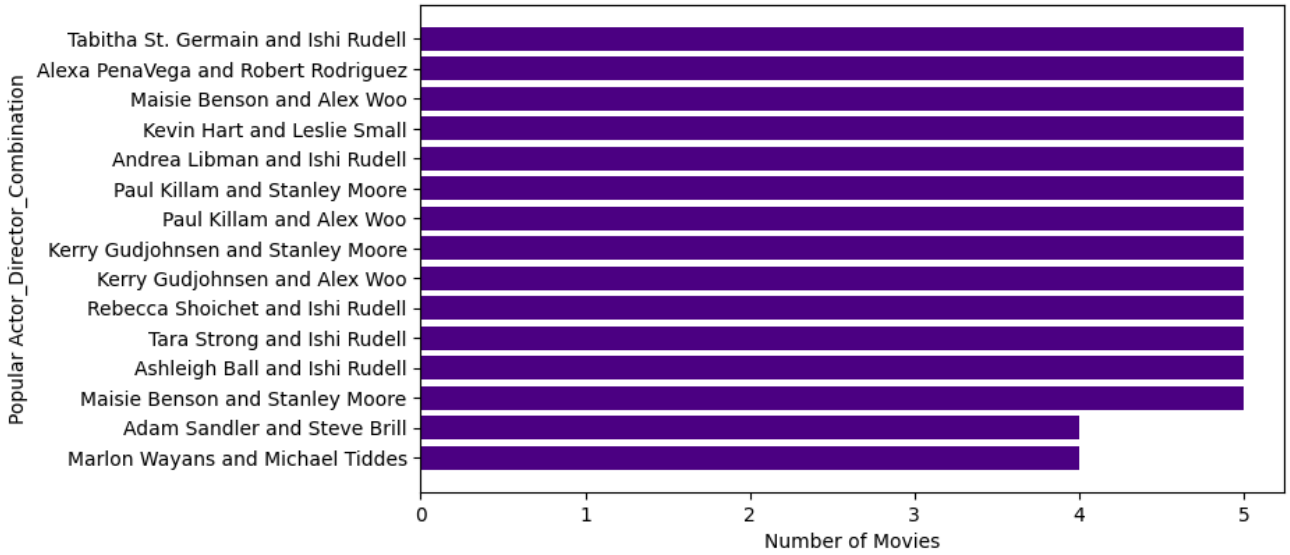
```
#Analysing Popular combination of actors and directors
data_usa_movies['Actor_Director_Combination'] = data_usa_movies.cast.str.cat(data_usa_mov
data_usa_movies_subset=data_usa_movies[data_usa_movies['cast']!='Unknown Cast']
data_usa_movies_subset=data_usa_movies_subset[data_usa_movies_subset['director']!='Unknow
data_usa_movies_subset.head()
```

```
<ipython-input-134-2582c98f3c78>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  data_usa_movies['Actor_Director_Combination'] = data_usa_movies.cast.str.cat(data_u
```

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

```
plt.figure(figsize=(8,10))

plt.subplot(2,1,1)
data_cast=data_usa_shows_subset.groupby(['Actor_Director_Combination']).agg({"title":"nun
data_cast=data_cast[data_cast['Actor_Director_Combination']!='Unknown Director']
plt.barh(data_cast[::-1]['Actor_Director_Combination'], data_cast[::-1]['title'],color=['
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actor_Director_Combination')
plt.show()

plt.figure(figsize=(8,10))
plt.subplot(2,1,2)
data_cast=data_usa_movies_subset.groupby(['Actor_Director_Combination']).agg({"title":"nu
data_cast=data_cast[data_cast['Actor_Director_Combination']!='Unknown Director']
plt.barh(data_cast[::-1]['Actor_Director_Combination'], data_cast[::-1]['title'],color=['
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actor_Director_Combination')
plt.show()
```






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

```
array(['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)
```

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

```
array(['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 across Movies on Netflix in USA are:-

1. Tabitha St. Germain and Ishi Rudell
2. Alexa PenaVega and Robert Rodriguez
3. Maisie Benson and Alex Woo
4. Kevin Hart and Leslie Small
5. Andrea Libman and Ishi Rudell
6. Paul Killam and Stanley Moore
7. Paul Killam and Alex Woo
8. Kerry Gudjohnsen and Stanley Moore
9. Kerry Gudjohnsen and Alex Woo
10. Rebecca Shoichet and Ishi Rudell
11. Tara Strong and Ishi Rudell
12. Ashleigh Ball and Ishi Rudell
13. Maisie Benson and Stanley Moore

The most popular Actor Director Combination across TV Shows on Netflix in USA are:-

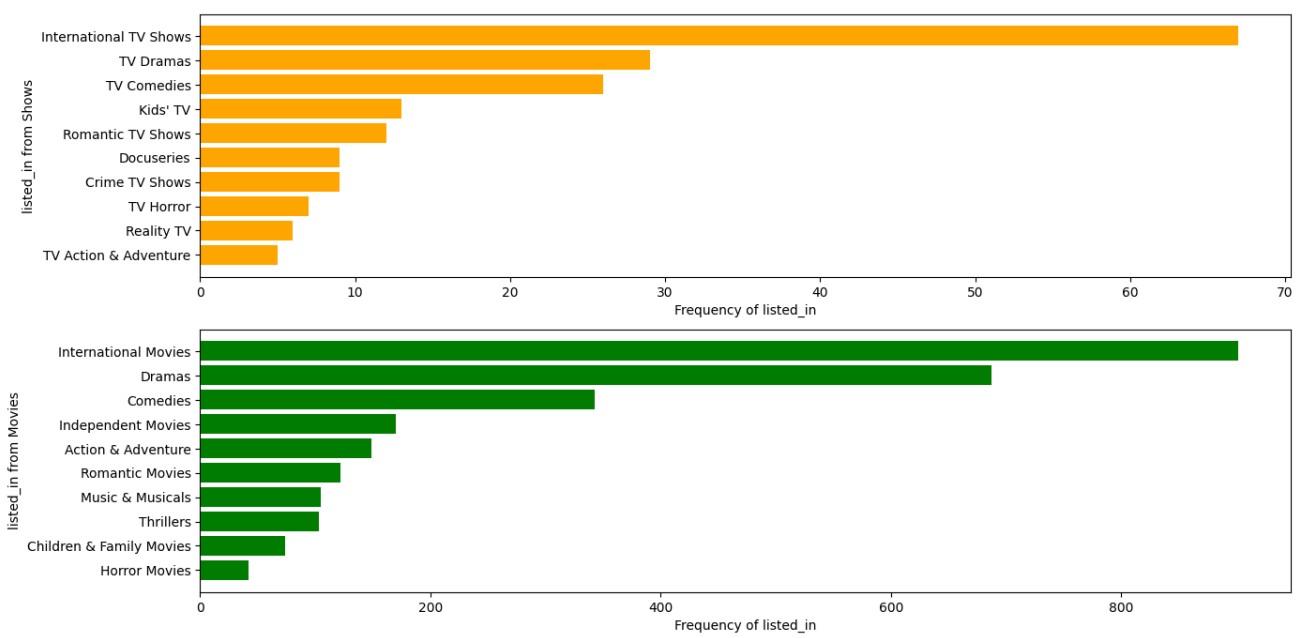
1. Dave Chappelle and Stan Lathan
2. Manuel Rubey and David Schalko
3. Marc Thompson and Iginio Straffi
4. Margarita Reyes and Alejandro Lozano
5. Marieve Herington and Thomas Astruc
6. Marisol Nichols and Rob Seidenglanz
7. Mark Consuelos and Rob Seidenglanz
8. Mark Feuerstein and Jay Chandrasekhar
9. Mark Pellegrino and Phil Sgriccia
10. Martin Cummins and Rob Seidenglanz
11. Martin Short and Tony Collingwood
12. Marty Adams and Eli Roth

13. Manu Bennett and James Bamford
14. Aaron Douglas and Eli Roth

India

```
data_india_movies=data_final[(data_final['country']=='India') & (data_final['type']=='Mov
data_india_shows=data_final[(data_final['country']=='India') & (data_final['type']=='TV S
```

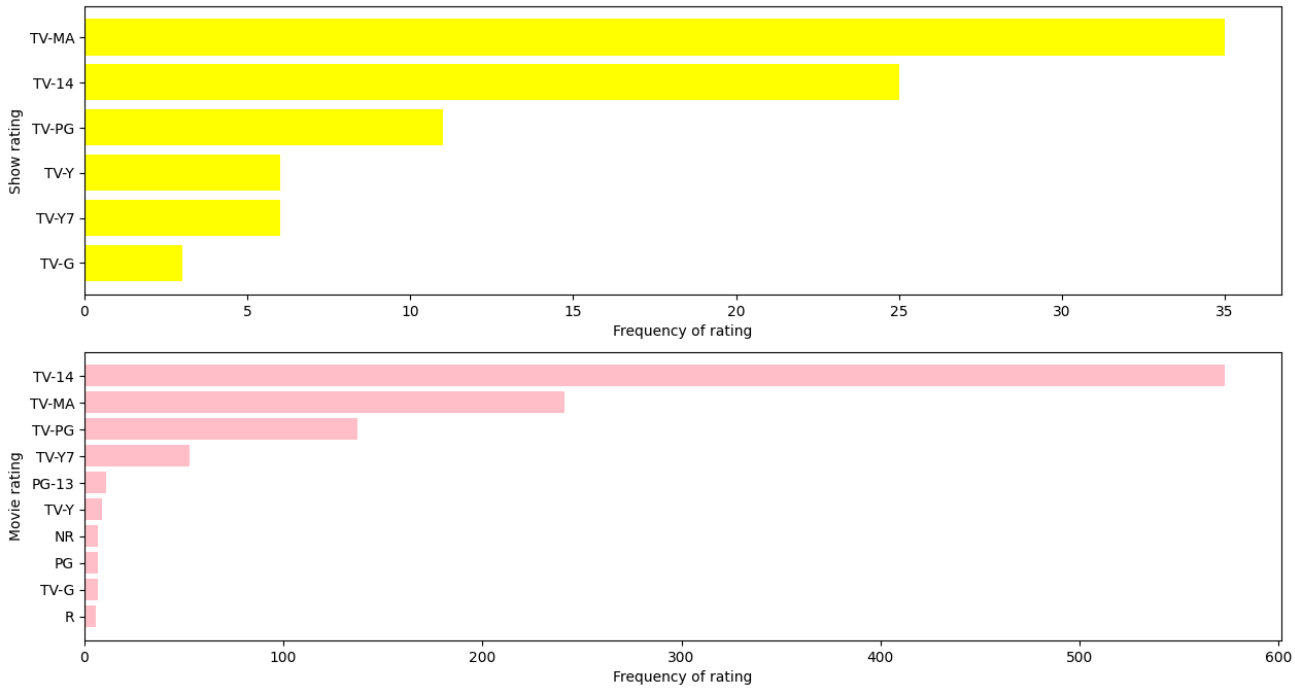
```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_listed_in=data_india_shows.groupby(['listed_in']).agg({"title":"nunique"}).reset_ind
plt.barh(data_listed_in[::-1]['listed_in'], data_listed_in[::-1]['title'],color=['orange'
plt.xlabel('Frequency of listed_in')
plt.ylabel('listed_in from Shows')


plt.subplot(2,1,2)
data_listed_in=data_india_movies.groupby(['listed_in']).agg({"title":"nunique"}).reset_in
plt.barh(data_listed_in[::-1]['listed_in'], data_listed_in[::-1]['title'],color=['Green']
plt.xlabel('Frequency of listed_in')
plt.ylabel('listed_in from Movies')
plt.show()
```



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

International Movies, Dramas, Comedies, Independent Movies and Action & Adventure Genres in Movies are popular in India

```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_rating=data_india_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().s
plt.barh(data_rating[::-1]['rating'], data_rating[::-1]['title'],color=['yellow'])
plt.xlabel('Frequency of rating')
plt.ylabel('Show rating')


plt.subplot(2,1,2)
data_rating=data_india_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().
plt.barh(data_rating[::-1]['rating'], data_rating[::-1]['title'],color=['pink'])
plt.xlabel('Frequency of rating')
plt.ylabel('Movie rating')
plt.show()
```



In India, TV Show ratings were TV-MA, TV-14, TV-PG and TV-Y. Movies were TV-14, TV-MA, TV-PG and TV-Y7.

```
data_min_india_movies=data_min[(data_final['country']=='India')]
data_min_india_movies['country'] = data_min_india_movies['country'].str.replace(',', '')
data_min_india_movies['country'].value_counts()
```

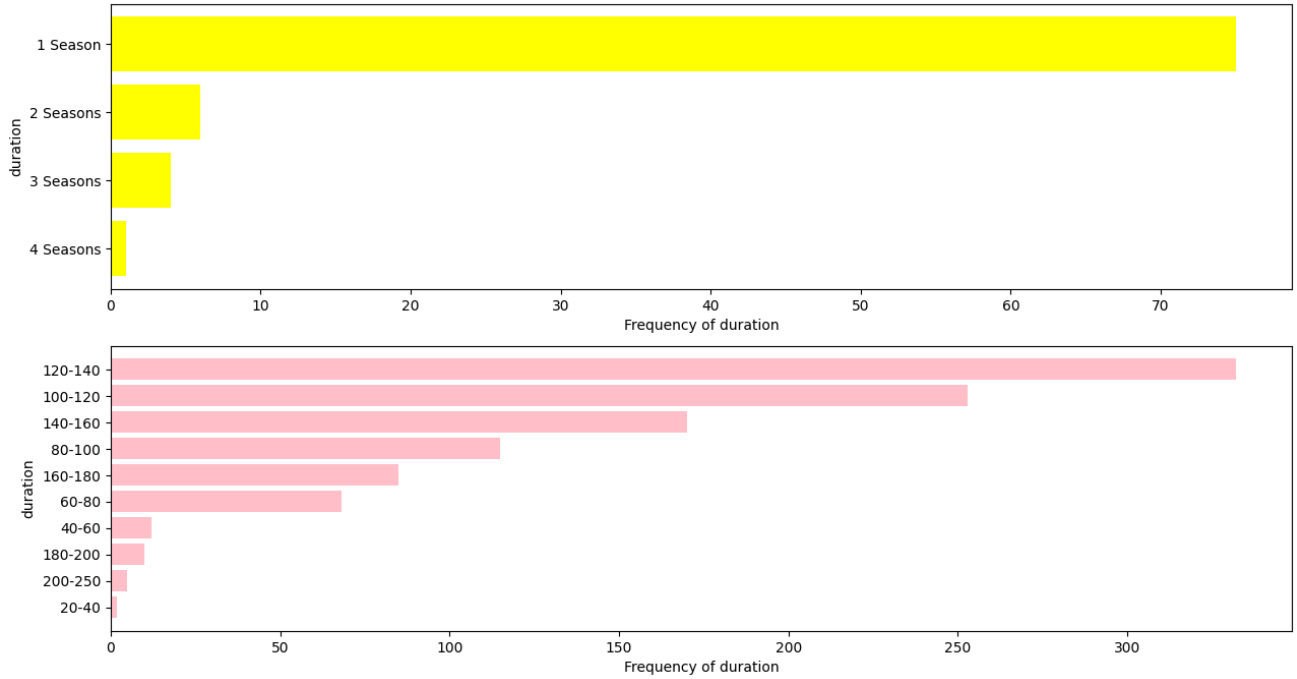
```
<ipython-input-141-cb6e012532c8>:1: UserWarning: Boolean Series key will be reindexed
  data_min_india_movies=data_min[(data_final['country']=='India')]
<ipython-input-141-cb6e012532c8>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
  data_min_india_movies['country'] = data_min_india_movies['country'].str.replace(','
```

country	count
India	22546

dtype: int64

```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_rating=data_india_shows.groupby(['duration']).agg({"title":"nunique"}).reset_index()
plt.barh(data_rating[::-1]['duration'], data_rating[::-1]['title'],color=['yellow'])
plt.xlabel('Frequency of duration')
plt.ylabel('duration')


plt.subplot(2,1,2)
data_rating=data_min_india_movies.groupby(['duration']).agg({"title":"nunique"}).reset_in
plt.barh(data_rating[::-1]['duration'], data_rating[::-1]['title'],color=['pink'])
plt.xlabel('Frequency of duration')
plt.ylabel('duration')
plt.show()
```

```
<ipython-input-142-b81d2557169d>:11: FutureWarning: The default of observed=False is
  data_rating=data_min_india_movies.groupby(['duration']).agg({"title":"nunique"}).re
```

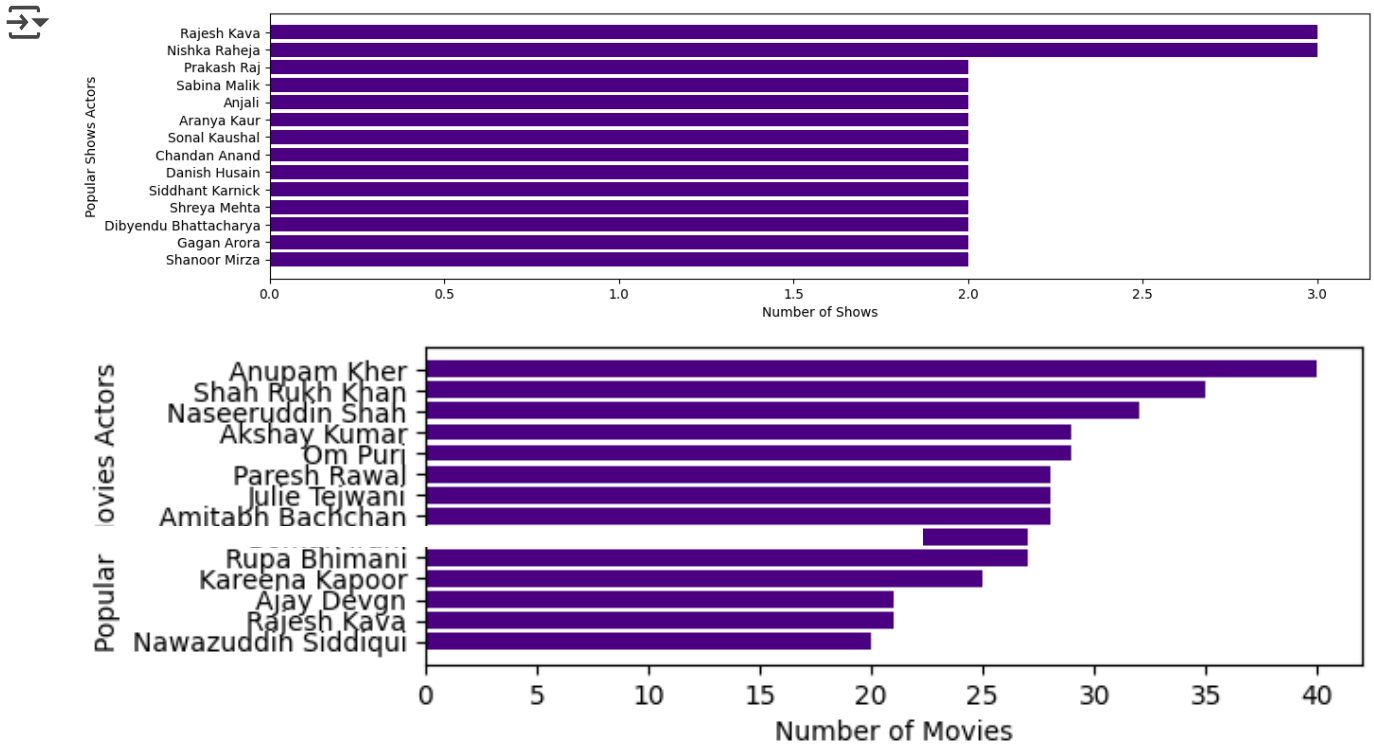


Across movies 120-140, 100-120 is the ranges of minutes for which most movies lie.

Across Tv Shows 1 Season shows were dominated followed by 2 season shows.

```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_cast=data_india_shows.groupby(['cast']).agg({"title":"nunique"}).reset_index().sort_
data_cast=data_cast[data_cast['cast']!='Unknown Cast']
plt.barh(data_cast[::-1]['cast'], data_cast[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Shows')
plt.ylabel('Popular Shows Actors')
plt.show()


plt.subplot(2,1,2)
data_cast=data_india_movies.groupby(['cast']).agg({"title":"nunique"}).reset_index().sort
data_cast=data_cast[data_cast['cast']!='Unknown Cast']
```

```
plt.barh(data_cast[::-1]['cast'], data_cast[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Movies')
plt.ylabel('Popular Movies Actors')
plt.show()
```

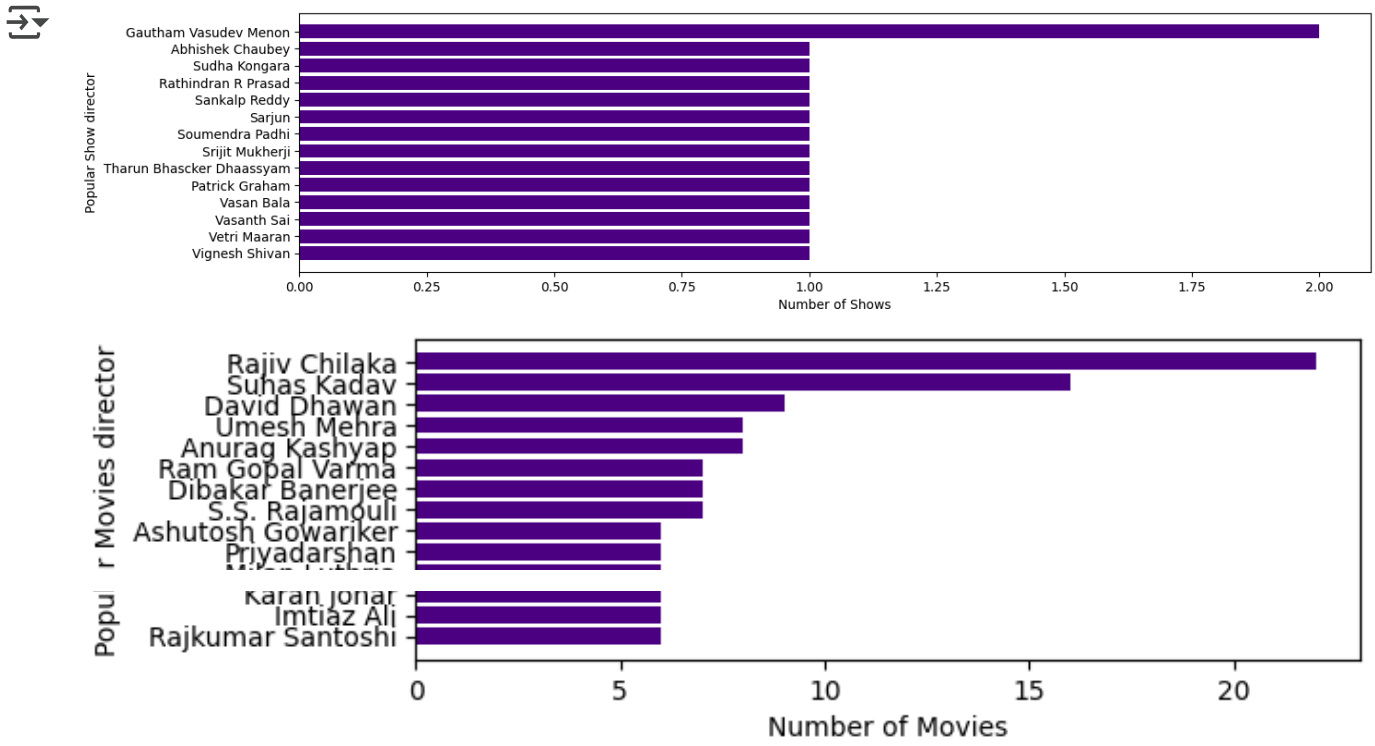


Rajesh Kava, Nishka Raheja and Prakash Raj are the popular actors across TV Shows in India.

Anupam Kher, Shah Rukh Khan and Naseeruddin Shah are the popular actors across movies on Netflix in India.

```
plt.figure(figsize=(15,8))

plt.subplot(2,1,1)
data_cast=data_india_shows.groupby(['director']).agg({"title":"nunique"}).reset_index().s
data_cast=data_cast[data_cast['director']!='Unknown Director']
plt.barh(data_cast[::-1]['director'], data_cast[::-1]['title'],color=['indigo'])
plt.xlabel('Number of Shows')
plt.ylabel('Popular Show director')
plt.show()


plt.subplot(2,1,2)
data_cast=data_india_movies.groupby(['director']).agg({"title":"nunique"}).reset_index().
data_cast=data_cast[data_cast['director']!='Unknown Director']
plt.barh(data_cast[::-1]['director'], data_cast[::-1]['title'],color=['indigo'])
```

```
plt.xlabel('Number of Movies')
plt.ylabel('Popular Movies director')
plt.show()
```

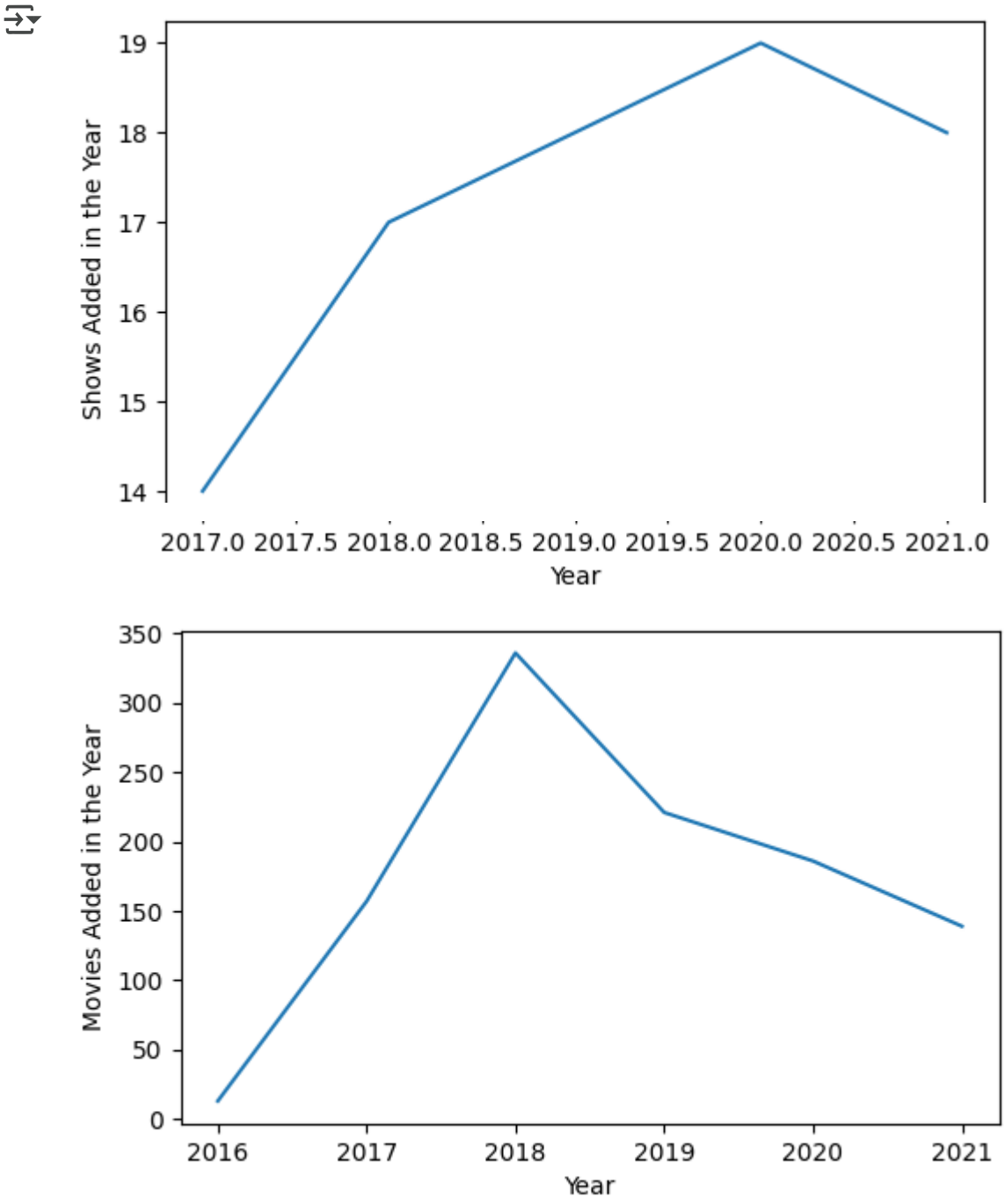


Gautham Vasudev Menon, Abhishek Chaubey, Sudha Kongara, Rathindran R Prasada and some more are the popular directors across TV Shows on Netflix in India.

Rajiv Chilaka, Suhas Kadav, David Dhawan , Umesh Mehra and Anurag KashyapAnurag Kashyap are the popular directors across movies on Netflix in India

```
plt.figure(figsize=(6,8))

plt.subplot(2,1,1)
data_year=data_india_shows.groupby(['Content_Added_Year']).agg({"title":"nunique"}).reset
sns.lineplot(data=data_year, x='Content_Added_Year', y='title')
plt.ylabel("Shows Added in the Year")
plt.xlabel("Year")
plt.show()

plt.figure(figsize=(6,8))
plt.subplot(2,1,2)
data_year=data_india_movies.groupby(['Content_Added_Year']).agg({"title":"nunique"}).rese
sns.lineplot(data=data_year, x='Content_Added_Year', y='title')
plt.ylabel("Movies Added in the Year")
```

```
plt.xlabel("Year")
plt.show()
```

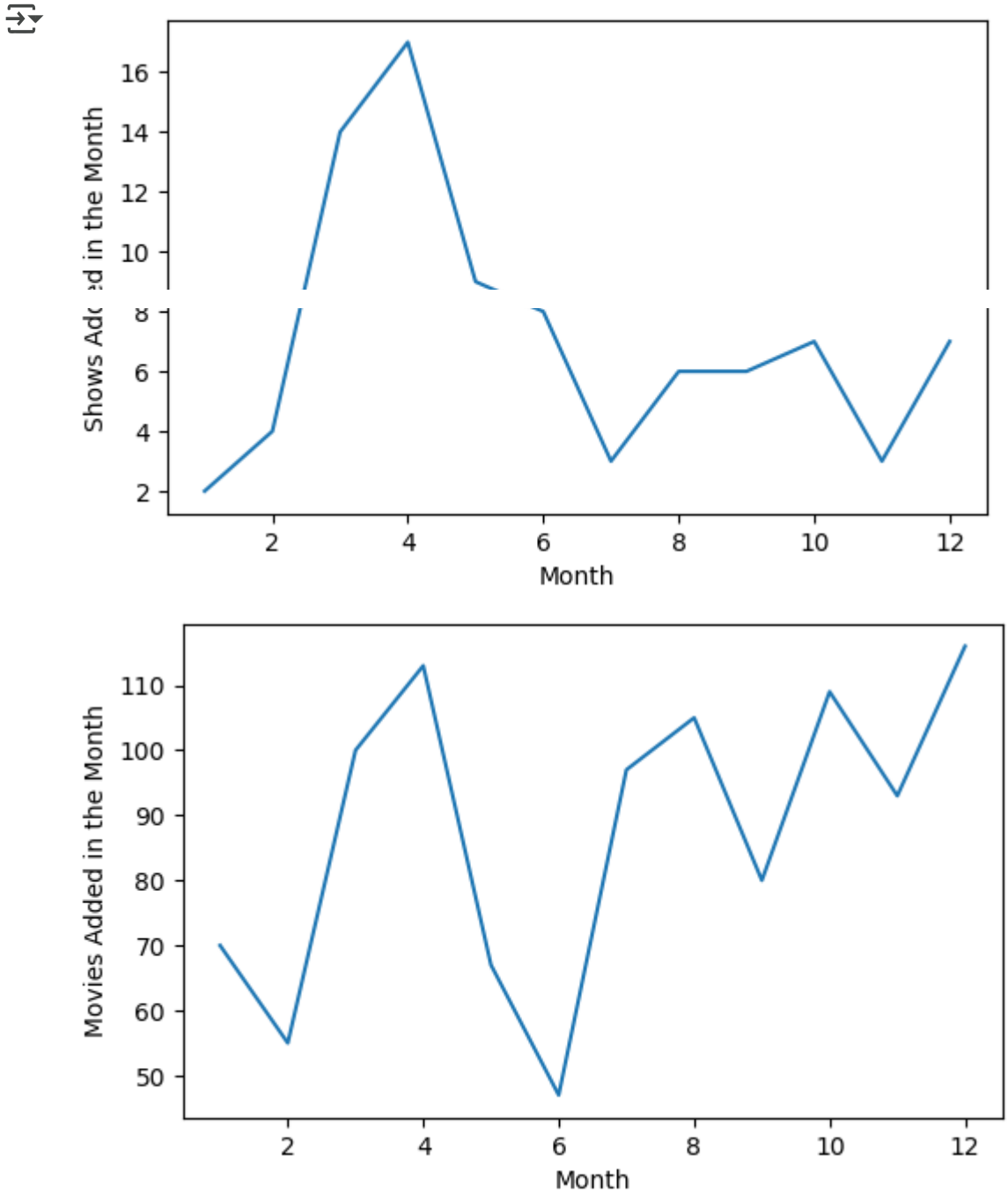


In India, Tv shows addition remains declining for the recent years.

In India, Movies addition in India started to decline after 2018.

```
plt.figure(figsize=(6,8))
plt.subplot(2,1,1)
data_month=data_india_shows.groupby(['Content_Added_Month']).agg({"title":"nunique"}).res
sns.lineplot(data=data_month, x='Content_Added_Month', y='title')
plt.ylabel("Shows Added in the Month")
plt.xlabel("Month")
plt.show()
```

```
plt.figure(figsize=(6,8))
plt.subplot(2,1,2)
data_month=data_india_movies.groupby(['Content_Added_Month']).agg({"title":"nunique"}).re
sns.lineplot(data=data_month, x='Content_Added_Month', y='title')
plt.ylabel("Movies Added in the Month")
plt.xlabel("Month")
plt.show()
```

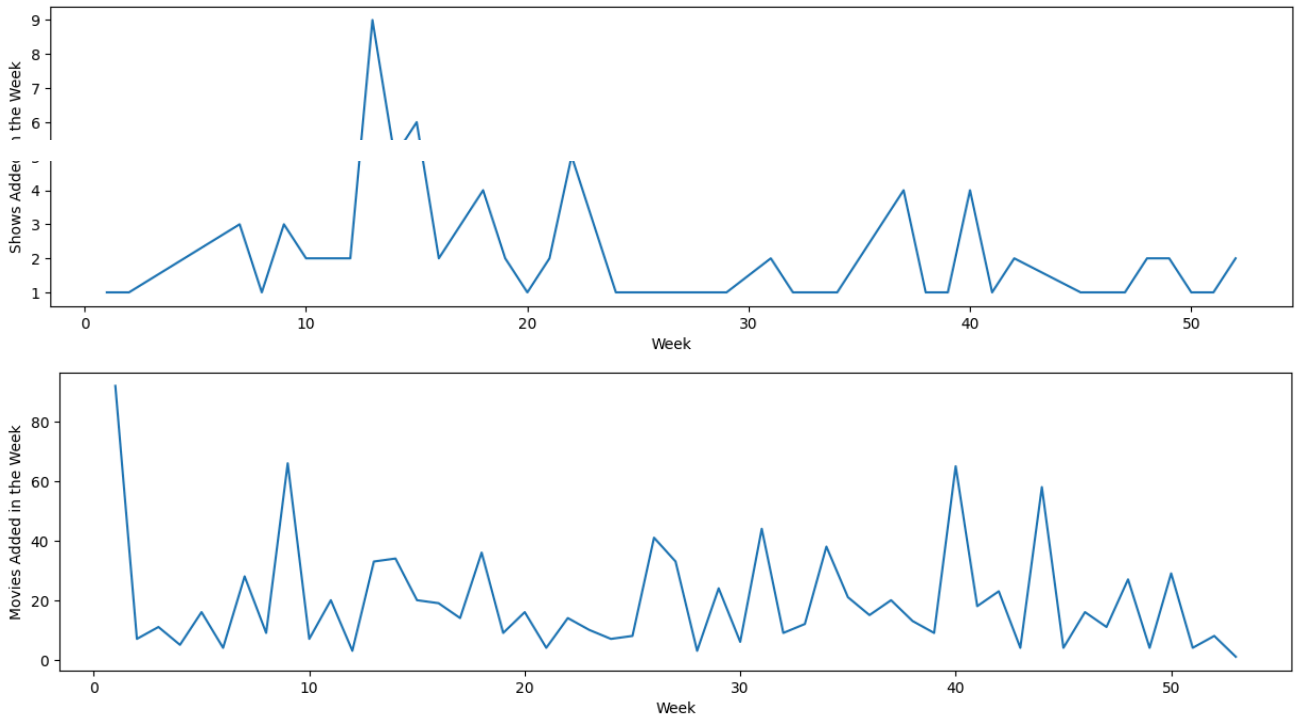


In India, Tv shows were added mostly on April month of the year

In India, Movies were added mostly on April and Last month of the year.

```
plt.figure(figsize=(15,8))
plt.subplot(2,1,1)
data_week=data_india_shows.groupby(['Content_Added_Week']).agg({"title":"nunique"}).reset
sns.lineplot(data=data_week, x='Content_Added_Week', y='title')
```

```
plt.ylabel("Shows Added in the Week")
plt.xlabel("Week")
plt.show()

plt.figure(figsize=(15,8))
plt.subplot(2,1,2)
data_week=data_india_movies.groupby(['Content_Added_Week']).agg({"title":"nunique"}).rese
sns.lineplot(data=data_week, x='Content_Added_Week', y='title')
plt.ylabel("Movies Added in the Week")
plt.xlabel("Week")
plt.show()
```



In India, Tv shows were added between these ranges 12-14 & 21-24 of week in the year

In India, Movies were added between these ranges 0-2, 39-41 & 8-10 of week in the year

```
plt.figure(figsize=(6,8))
plt.subplot(2,1,1)
data_release_year=data_india_shows.groupby(['release_year']).agg({"title":"nunique"}).res
sns.lineplot(data=data_release_year, x='release_year', y='title')
plt.ylabel("Shows Released in the Year")
plt.xlabel("Year")
plt.show()

plt.figure(figsize=(6,8))
plt.subplot(2,1,2)
data_release_year=data_india_movies.groupby(['release_year']).agg({"title":"nunique"}).re
sns.lineplot(data=data_release_year, x='release_year', y='title')
plt.ylabel("Movies Released in the Year")
```