AIM :- Analyze the data and generate insights that could help Netflix in deciding which type of shows/movies to produce and how they can grow the business in different countries.

Double-click (or enter) to edit

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df=pd.read_csv('/content/drive/MyDrive/Copy of d2beiqkhq929f0.cloudfront.net_public_assets_assets_000_000_940_original_netfl
df.head()
```

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

```
#df=pd.read_csv('/content/drive/MyDrive/bq-results-20230623-060109-1687502380344/d2beiqkhq929f0.cloudfront.net_public_assets_
```

```
#df.head()
```

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

```
8807
```

```
#checking data types
df.dtypes
```

```
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
```

```
#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
```

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

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

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

```
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64
```

```
#unnesting the directors column, i.e creating separate lines for each director
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()
```

	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

```
#unnesting the cast column, i.e creating separate lines for each cast member
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()
```

title	Actors

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

	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

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

	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	nan
3	Jailbirds New Orleans	nan
4	Kota Factory	India

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

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

```
df_new.head()
```

	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
[illegible]	[illegible]	[illegible]	Unknown	[illegible]	[illegible]

```
#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()
```

	title	Actors	Directors	Genre	country	show_id	type	date_add
0	Dick Johnson	Unknown	Kirsten	Documentaries	United	s1	Movie	Septem

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

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

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

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

```
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
```

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

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

	title	Actors	Directors	Genre	country	show_id	type	date_ad
136893	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	
	A Young Doctor's							

```
#date added column is imputed on the basis of release year,
#when release year was 2013.So below piece of code just checks the mode of date added in group
# and imputes in place of nulls the corresponding mode
for i in df_final[df_final['date_added'].isnull()]['release_year'].unique():
 imp=df_final[df_final['release_year']==i]['date_added'].mode().values[0]
 df_final.loc[df_final['release_year']==i,'date_added']=df_final.loc[df_final['release_year']==i,'date_added'].fillna(imp)
```

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

title	Actors	Directors	Genre	country	show_id	type	date_added	releas

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

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 occurence 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

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

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

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

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

```
df_final.head()
```

	title	Actors	Directors	Genre	country	show_id	type	date_add
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	Septem 25, 2(
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	Septem 24, 2(
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	Septem 24, 2(

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

```
253 min          21
15 min           20
167 min          20
233 min          18
237 min          18
49 min           16
37 min           16
43 min           16
312 min          15
12 min           14
31 min           13
191 min          13
230 min          12
41 min           11
19 min            8
273 min           7
34 min            6
17 min            5
39 min            5
10 min            4
16 min            4
196 min           4
20 min            4
18 min            4
3 min             4
5 min             3
11 min            2
8 min             2
9 min             2
Name: duration, dtype: int64
```

```
#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_add
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	Septem 25, 2(
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	Septem 24, 2(
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	Septem 24, 2(

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

```
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)
```

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

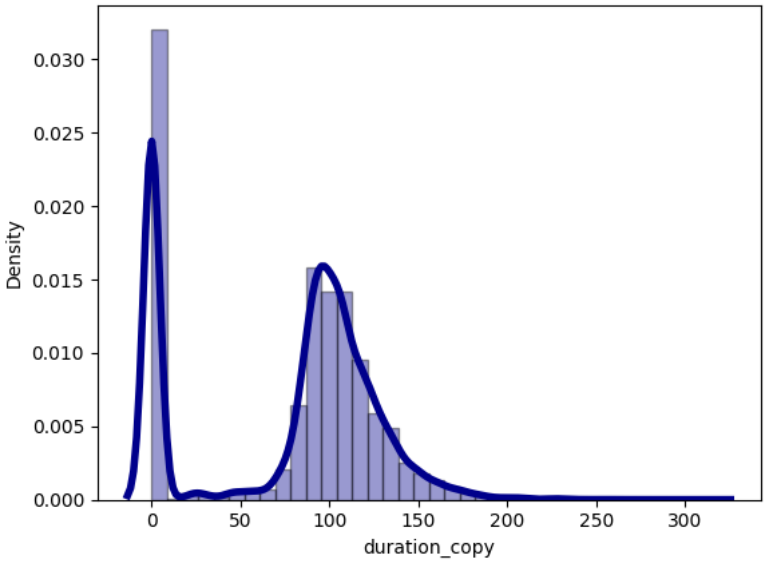
```
df_final1.loc[df_final1['duration_copy'].str.contains('Season'),'duration_copy']=0
df_final1['duration_copy']=df_final1['duration_copy'].astype('int')
df_final1.head()
```

	title	Actors	Directors	Genre	country	show_id	type	date_add
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	Septem 25, 2(

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

```
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
```

```
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()
```




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

```
49                16
37                16
43                16
312               15
12                14
31                13
191               13
230               12
41                11
19                 8
273                7
34                 6
17                 5
39                 5
10                 4
16                 4
196                4
20                 4
18                 4
3                  4
5                  3
11                 2
8                  2
9                  2
Name: duration, dtype: int64
```

```
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()
```

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

```
#presence of brackets and content between brackets is removed.
df_final1['title']=df_final1['title'].str.replace(r"\(.*\)","")
df_final1.head()
```

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

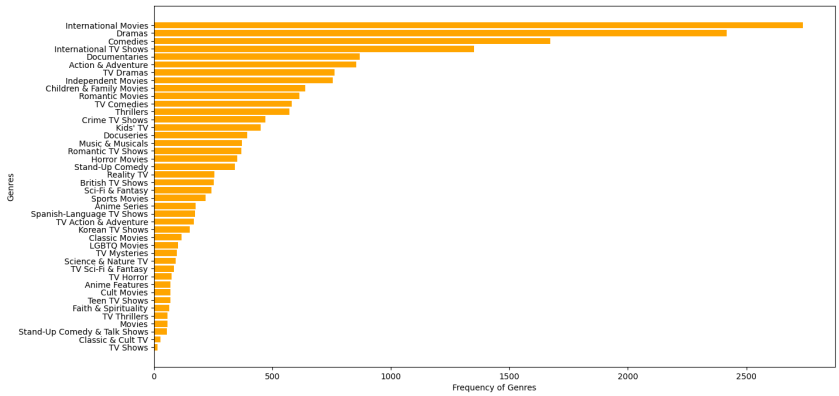
Univariate Analysis in terms of counts of each column

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

Genre	title
International Movies	2738
Dramas	2418
Comedies	1673
International TV Shows	1351
Documentaries	869
Action & Adventure	854
TV Dramas	763
Independent Movies	756
Children & Family Movies	639
Romantic Movies	615
TV Comedies	581
Thrillers	573
Crime TV Shows	470
Kids' TV	451
Docuseries	395
Music & Musicals	372
Romantic TV Shows	370
Horror Movies	353
Stand-Up Comedy	343
Reality TV	255
British TV Shows	253
Sci-Fi & Fantasy	243
Sports Movies	219
Anime Series	176
Spanish-Language TV Shows	174
TV Action & Adventure	168
Korean TV Shows	151
Classic Movies	116
LGBTQ Movies	102
TV Mysteries	98
Science & Nature TV	92
TV Sci-Fi & Fantasy	84
TV Horror	75
Anime Features	71
Cult Movies	71
Teen TV Shows	69
Faith & Spirituality	65
TV Thrillers	57
Movies	57
Stand-Up Comedy & Talk Shows	56
Classic & Cult TV	28
TV Shows	16

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

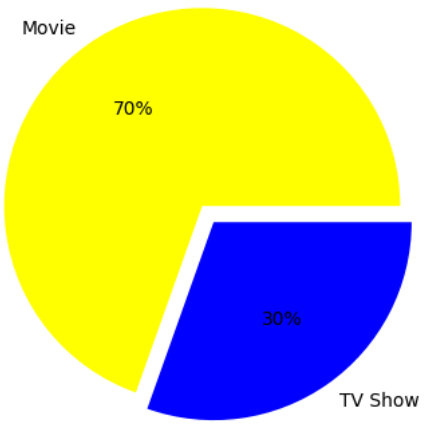
```
plt.ylabel('Genres')
plt.show()
```




```
df_final1.groupby(['type']).agg({"title":"nunique"})
```

type	title
Movie	6115
TV Show	2676

```
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=['yellow','blue'],autopct='%.lf%%')
plt.show()
```




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

country	title
	3
Afghanistan	1
Albania	1
Algeria	3
Angola	2
Argentina	94
Armenia	1
Australia	162
Austria	12
Azerbaijan	1
Bahamas	1
Bangladesh	4
Belarus	1
Belgium	94
Bermuda	1
Botswana	1
Brazil	103
Bulgaria	10
Burkina Faso	1
Cambodia	5
Cambodia,	1
Cameroon	2
Canada	460
Cayman Islands	2
Chile	30
China	166
Colombia	54
Croatia	4
Cuba	2
Cyprus	1
Czech Republic	23
Denmark	50
Dominican Republic	1
East Germany	1
Ecuador	1
Egypt	134
Ethiopia	1
Finland	12
France	409
Georgia	2
Germany	231
Ghana	8
Greece	11
Guatemala	2
Hong Kong	110
Hungary	11
Iceland	11
India	1126

Indonesia	97
Iran	4
Iraq	2
Ireland	46
Israel	30
Italy	102
Jamaica	1
Japan	338
Jordan	10
Kazakhstan	1
Kenya	6
Kuwait	9
Latvia	1
Lebanon	33
Liechtenstein	1
Lithuania	1
Luxembourg	12
Malawi	1
Malaysia	26
Malta	3
Mauritius	3
Mexico	175
Mongolia	1
Montenegro	1
Morocco	6
Mozambique	1
Namibia	2
Nepal	2
Netherlands	50
New Zealand	33
Nicaragua	1
Nigeria	140
Norway	30
Pakistan	24
Palestine	1
Panama	1
Paraguay	1
Peru	11
Philippines	90
Poland	41
Poland,	1
Portugal	6
Puerto Rico	1
Qatar	10
Romania	14
Russia	27
Samoa	1
Saudi Arabia	14
Senegal	3
Serbia	7

Singapore	41
Slovakia	1
Slovenia	3
Somalia	1
South Africa	65
South Korea	235
Soviet Union	3
Spain	239
Sri Lanka	1
Sudan	1
Sweden	44
Switzerland	19
Syria	3
Taiwan	94
Thailand	74
Turkey	115
Uganda	1
Ukraine	3
United Arab Emirates	38
United Kingdom	829
United Kingdom,	2
United States	4245
United States,	1
Unknown Country	175
Uruguay	14
Vatican City	1
Venezuela	4
Vietnam	7
West Germany	5

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

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

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

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

country	title
	3
Afghanistan	1
Albania	1
Algeria	3
Angola	2
Argentina	94
Armenia	1
Australia	162
Austria	12
Azerbaijan	1
Bahamas	1
Bangladesh	4
Belarus	1
Belgium	94
Bermuda	1
Botswana	1
Brazil	103
Bulgaria	10
Burkina Faso	1
Cambodia	6
Cameroon	2
Canada	460
Cayman Islands	2
Chile	30
China	166
Colombia	54
Croatia	4
Cuba	2
Cyprus	1
Czech Republic	23
Denmark	50
Dominican Republic	1
East Germany	1
Ecuador	1
Egypt	134
Ethiopia	1
Finland	12
France	409
Georgia	2
Germany	231
Ghana	8
Greece	11
Guatemala	2
Hong Kong	110
Hungary	11
Iceland	11
India	1126
Indonesia	97

Iran	4
Iraq	2
Ireland	46
Israel	30
Italy	102
Jamaica	1
Japan	338
Jordan	10
Kazakhstan	1
Kenya	6
Kuwait	9
Latvia	1
Lebanon	33
Liechtenstein	1
Lithuania	1
Luxembourg	12
Malawi	1
Malaysia	26
Malta	3
Mauritius	3
Mexico	175
Mongolia	1
Montenegro	1
Morocco	6
Mozambique	1
Namibia	2
Nepal	2
Netherlands	50
New Zealand	33
Nicaragua	1
Nigeria	140
Norway	30
Pakistan	24
Palestine	1
Panama	1
Paraguay	1
Peru	11
Philippines	90
Poland	42
Portugal	6
Puerto Rico	1
Qatar	10
Romania	14
Russia	27
Samoa	1
Saudi Arabia	14
Senegal	3
Serbia	7
Singapore	41
Slovakia	1

Slovenia	3
Somalia	1
South Africa	65
South Korea	235
Soviet Union	3
Spain	239
Sri Lanka	1
Sudan	1
Sweden	44
Switzerland	19
Syria	3
Taiwan	94
Thailand	74
Turkey	115
Uganda	1
Ukraine	3
United Arab Emirates	38
United Kingdom	831
United States	4246
Unknown Country	175
Uruguay	14
Vatican City	1
Venezuela	4
Vietnam	7

```
df_country=df_final1.groupby(['country']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)
plt.figure(figsize=(15,10))
plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=['blue'])
plt.xlabel('Titles by Countries')
plt.ylabel('Countries')
plt.show()
```


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

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

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

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

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

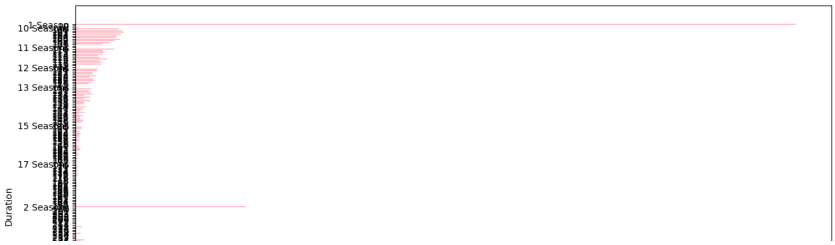
duration	title
1 Season	1793
10	1
10 Seasons	7
100	108
101	116
102	122
103	114
104	104
105	101
106	111
107	98
108	87
109	69
11	2
11 Seasons	2
110	97
111	68
112	74
113	69
114	56
115	61
116	80
117	61
118	65
119	63
12	3
12 Seasons	2
120	56
121	54
122	43
123	44
124	52
125	36
126	44
127	48
128	41
129	32
13	3
13 Seasons	3
130	40
131	32
132	37
133	42
134	22
135	38
136	23
137	38
138	21

139	22
14	3
140	25
141	19
142	13
143	23
144	9
145	18
146	12
147	12
148	19
149	15
15	3
15 Seasons	2
150	17
151	15
152	5
153	11
154	13
155	10
156	10
157	6
158	12
159	5
16	1
160	6
161	10
162	14
163	11
164	4
165	8
166	7
167	1
168	7
169	2
17	3
17 Seasons	1
170	5
171	7
172	4
173	6
174	2
176	5
177	5
178	1
179	2
18	1
180	2
181	4
182	3

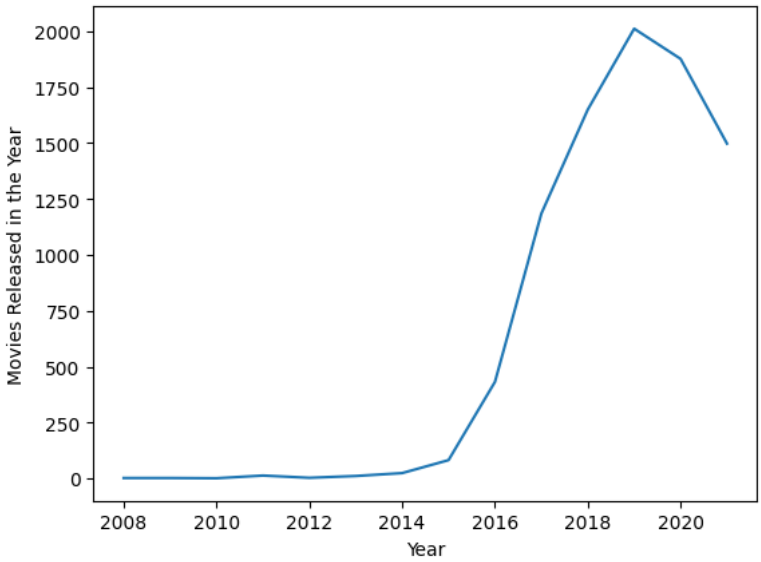
185	6
186	1
187	2
189	1
19	2
190	2
191	1
192	2
193	1
194	1
195	2
196	1
2 Seasons	425
20	2
200	1
201	1
203	1
204	2
205	1
208	1
209	2
21	3
212	1
214	1
22	16
224	1
228	1
229	1
23	13
230	1
233	1
237	1
24	23
25	11
253	1
26	6
27	3
273	1
28	10
29	11
3	1
3 Seasons	199
30	6
31	2
312	1
32	9
33	6
34	3
35	5
36	5

37	3
38	5
39	2
4 Seasons	95
40	13
41	3
42	9
43	1
44	19
45	10
46	24
47	11
48	8
49	9
5	1
5 Seasons	65
50	10
51	11
52	20
53	24
54	24
55	16
56	12
57	14
58	25
59	25
6 Seasons	33
60	29
61	31

```
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()
```




```
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()
```

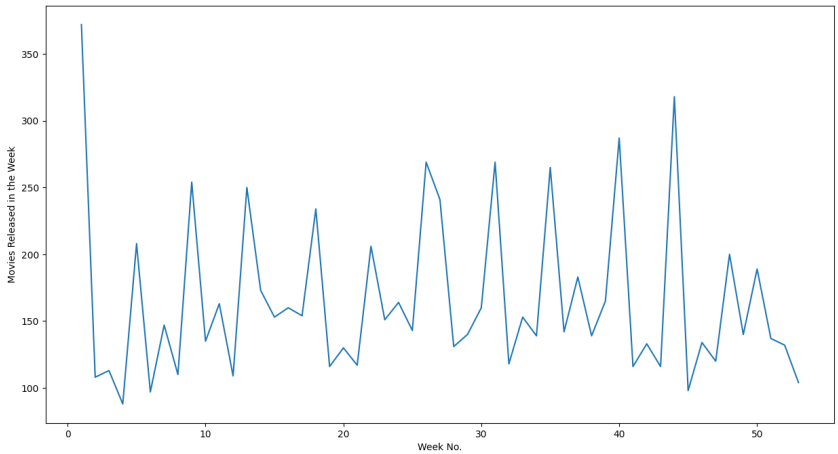


The Amount of Content across Netflix has increased from 2008 continuously till 2019. Then started decreasing from here(probably due to Covid)

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

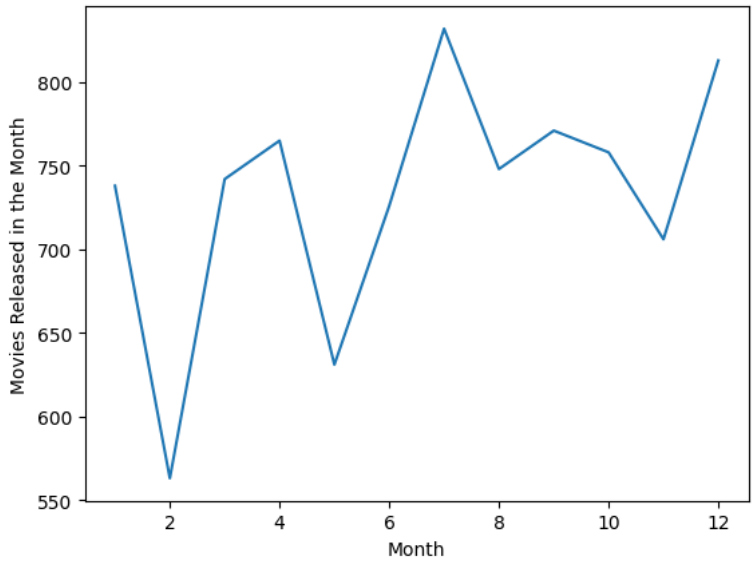
week_Added	title
1	372
2	108
3	113
4	88
5	208
6	97
7	147
8	110
9	254
10	135
11	163
12	109
13	250
14	173
15	153
16	160
17	154
18	234
19	116
20	130
21	117
22	206
23	151
24	164
25	143
26	269
27	241
28	131
29	140
30	160
31	269
32	118
33	153

```
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

```
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()
```

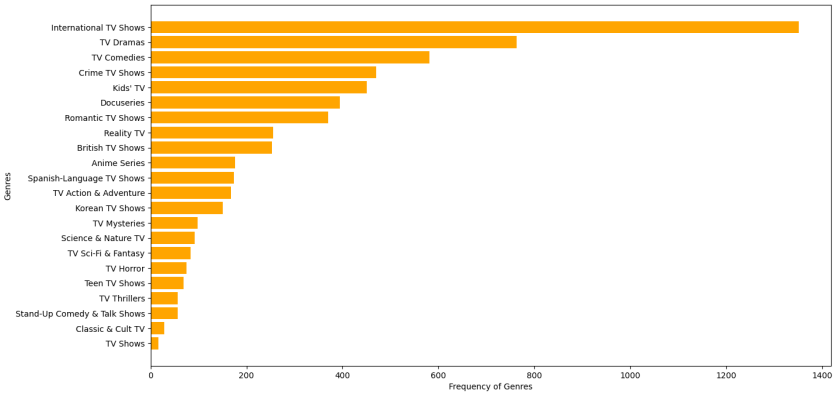


Most of the content is added in the first and last months across Netflix(reinstating what we observed for first week in baove plot)

Univariate Analysis separately for shows and movies

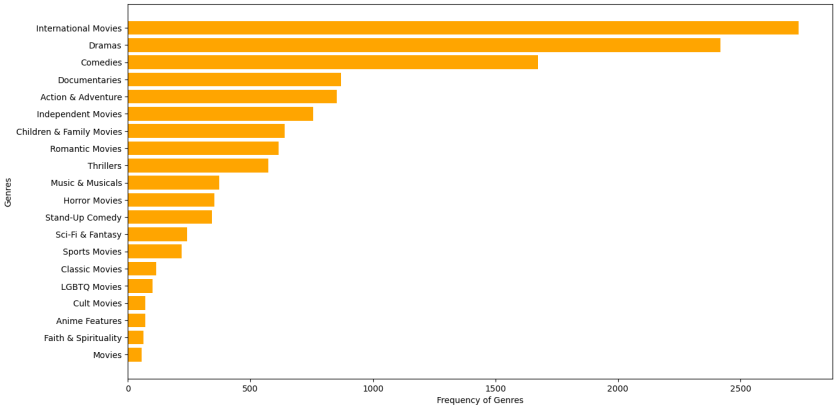
```
df_shows=df_final1[df_final1['type']=='TV Show']
df_movies=df_final1[df_final1['type']=='Movie']
```

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



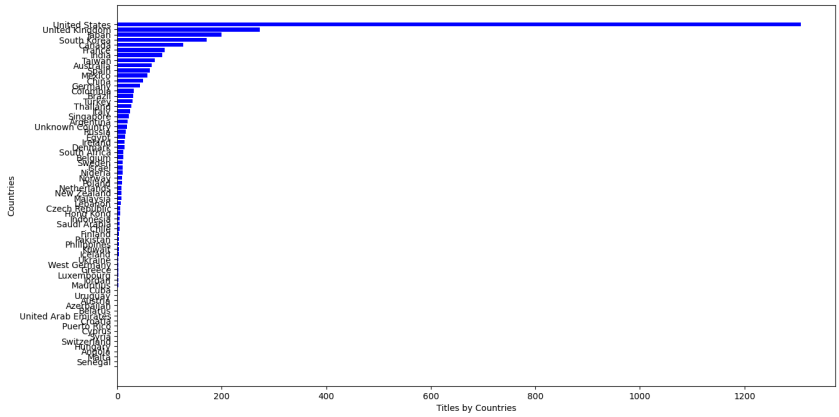
International TV Shows, Dramas and Comedy Genres are popular across TV Shows in Netflix

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

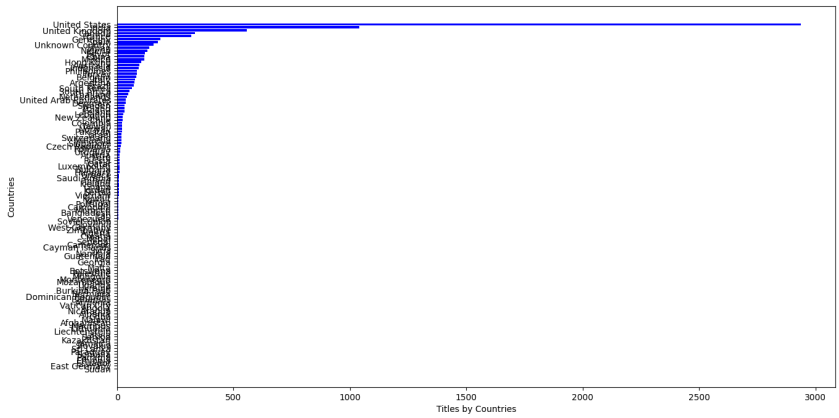


International Movies, Dramas and Comedy Genres are popular followed by Documentaries across Movies on Netflix

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



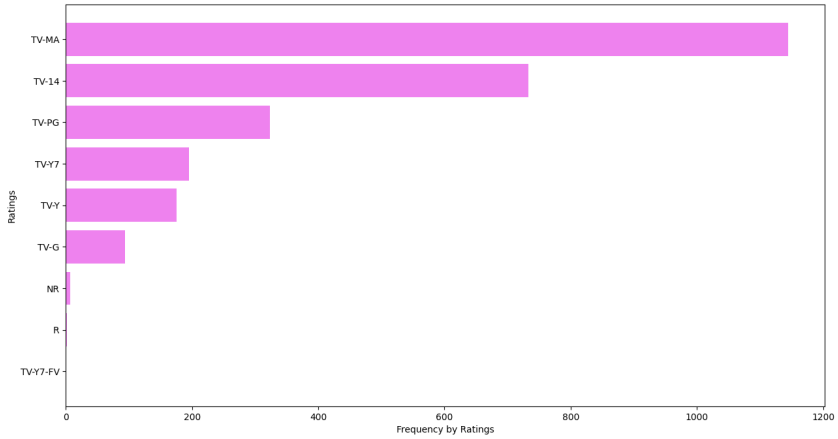

```
df_country=df_movies.groupby(['country']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=['blue'])
plt.xlabel('Titles by Countries')
plt.ylabel('Countries')
plt.show()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



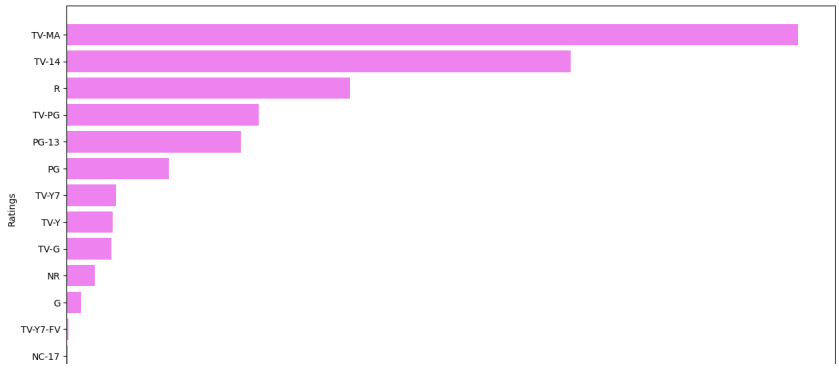
United States is leading across both TV Shows and Movies, UK also provides great content across TV Shows and Movies. Surprisingly India is much more prevalent in Movies as compared TV Shows.

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

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

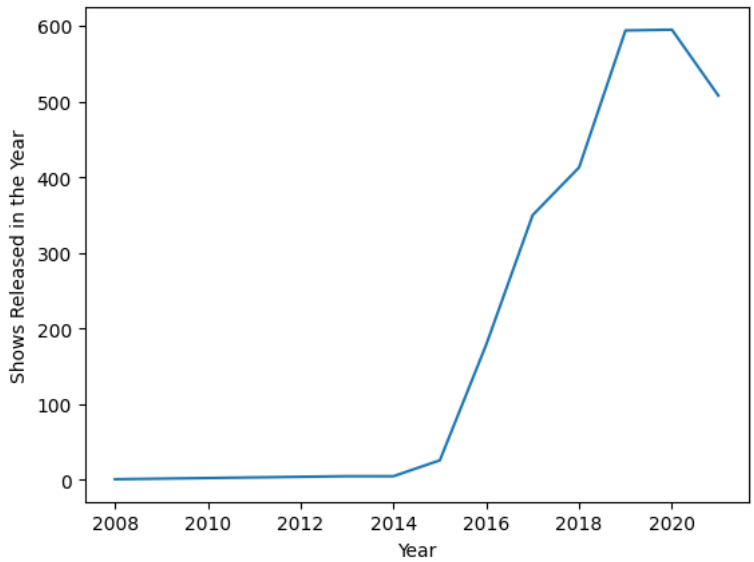



```
df_rating=df_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```

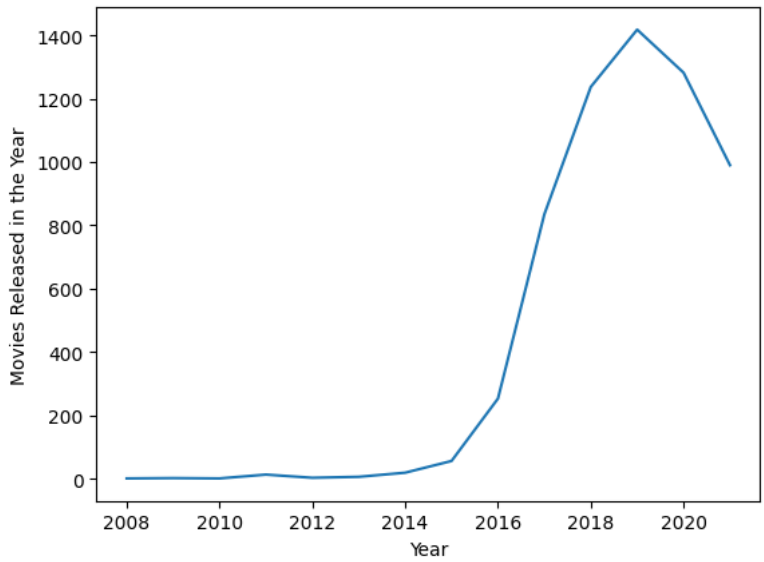


So it seems plaussible to conclude that the popular ratings across Netflix includes Mature Audiences and those appropriate for over 14/over 17 ages. Moreover there are no TV Shows having a rating of R

```
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()
```

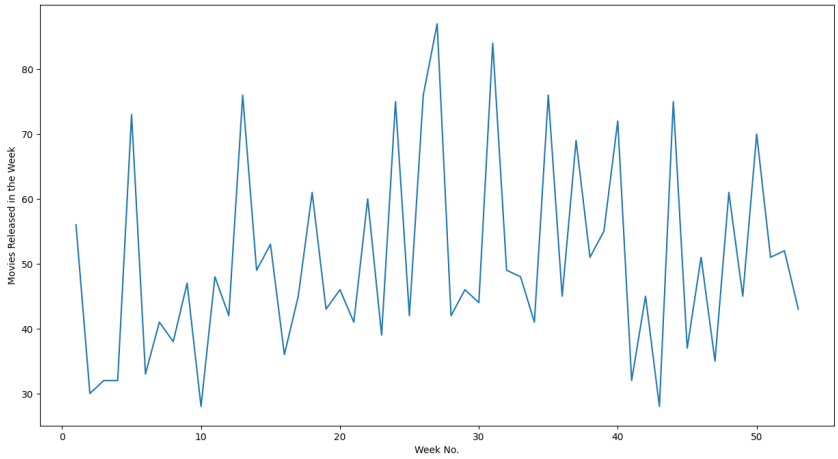



```
df_year=df_movies.groupby(['year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```

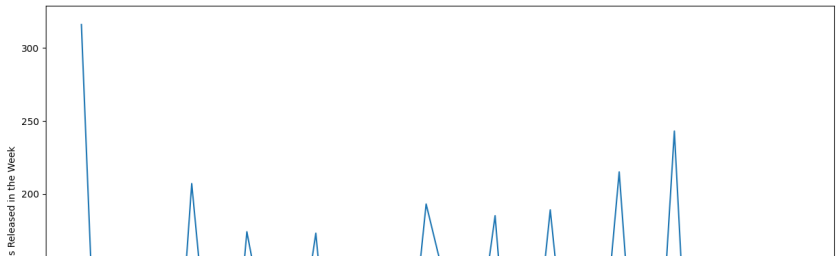


Till 2019, overall content across Netflix was increasing but due to Covid in 2020, though TV Shows didn't take a hit then Movies did take a hit. Well later in 2021, content across both was reduced significantly

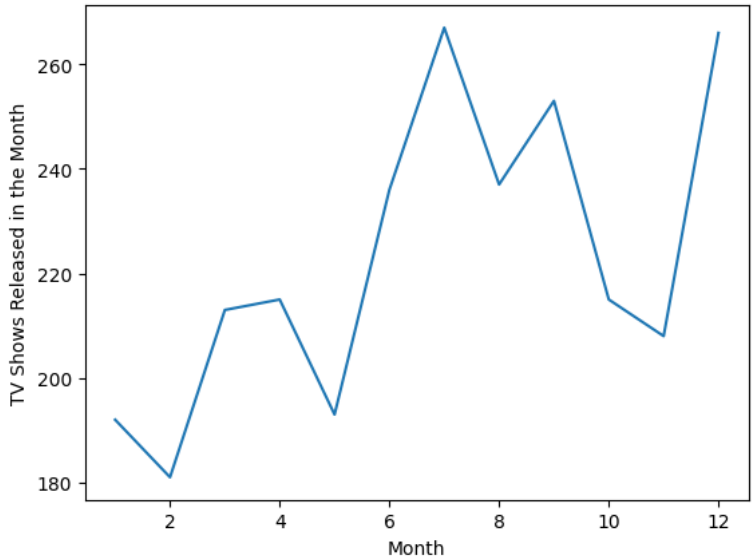
```
df_week=df_shows.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```



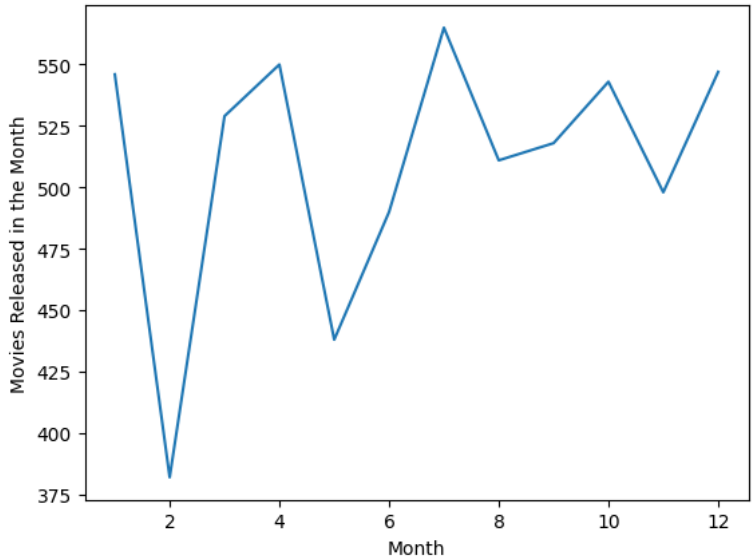

```
df_week=df_movies.groupby(['week_Added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='week_Added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```




```
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()
```




```
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()
```

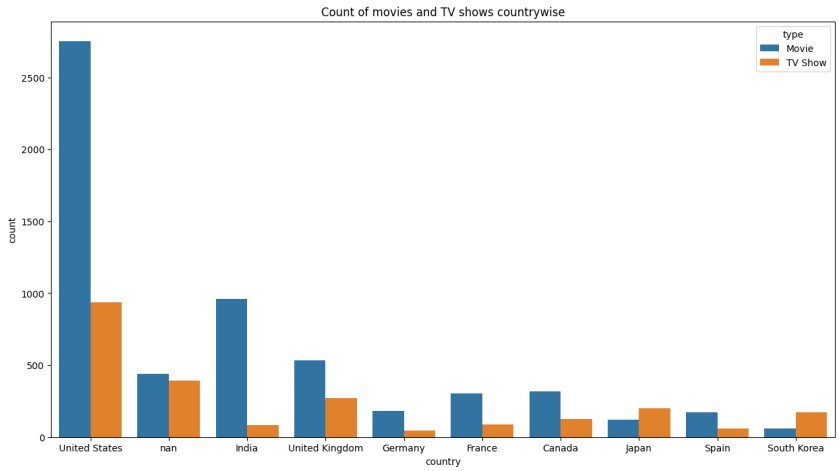


TV Shows are added in Netflix by a tremendous amount in mid weeks/months of the year, i.e- July Movies are added in Netflix by a tremendous amount in first week/last month of current year and first month of next year

```
#exploding country column
country = df["country"].apply(lambda x: str(x).split(", ")).tolist() #exploding the country column
df_country = pd.DataFrame(country, index = df["title"])
df_country = df_country.stack()
df_country = df_country.reset_index()
df_country.drop(columns = "level_1" , inplace = True)
```

```
df_country.columns = ["title" , "country"]
```

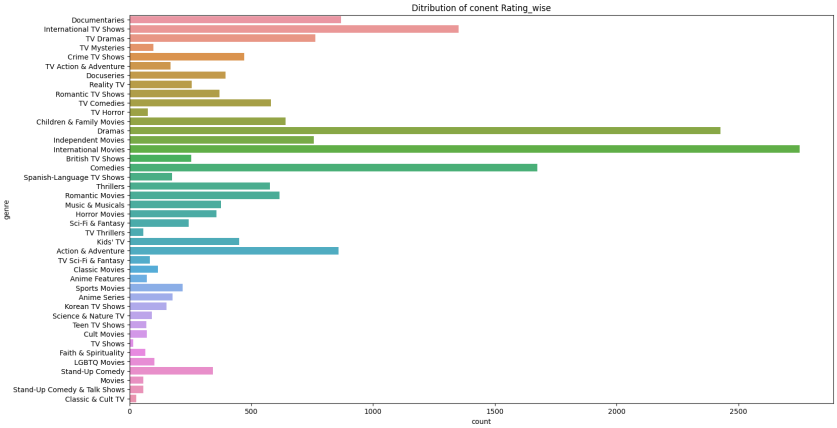
```
Country_wise_trend = df.merge(df_country , on = "title") #making new dataframe by merfing df_country and original dataframe.
Country_wise_trend.drop(columns = "country_x" , inplace = True)
Country_wise_trend.rename(columns = {"country_y" : "country"}, inplace = True)
Country_wise_trend = Country_wise_trend.loc[Country_wise_trend["country"] != "Unknown"]
top10_country = Country_wise_trend["country"].value_counts().head(10).reset_index()
top10_country.rename(columns = {"index" :"country" , "country" : "count"}, inplace = True)
Country_wise_trend = Country_wise_trend.merge(top10_country, how = "inner" , on = "country")
plt.figure(figsize = (15,8))
sns.countplot(x ="country" , data =Country_wise_trend , hue = "type" )
plt.title("Count of movies and TV shows countrywise")
plt.show()
```




```
#exploding listed_in column
listed_in = df["listed_in"].apply(lambda x: str(x).split(", ")).tolist()
df_genre = pd.DataFrame(listed_in, index = df["title"])
df_genre = df_genre.stack()
df_genre = df_genre.reset_index()
df_genre.drop(columns = "level_1" , inplace = True)
df_genre.columns = ["title" , "genre"]
df_genre.head()
```

	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

```
plt.figure(figsize = (18,10))
sns.countplot(y = "genre" , data =df_genre )
plt.title("Ditribution of conent Rating_wise")
plt.show()
```




```
df_trend_country = df.merge(df_country , on = "title")
df_trend_country.drop(columns = "country_x" , inplace = True)
df_trend_country.rename(columns = {"country_y":"country"}, inplace = True)
```

```
temp = df_trend_country['country'].value_counts()[:11].reset_index()
temp.rename(columns = {'index':'country', 'country':'count'}, inplace=True)
country_list = temp['country'].tolist()
df_top10country = df_trend_country.loc[df_trend_country['country'].isin(country_list)]
df_top10country = df_top10country.loc[df_top10country["country"]!="Unknown"]
```

```
genre_country_df= df_trend_country.merge(df_genre , on= "title")
genre_country_df.head(5)
```

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

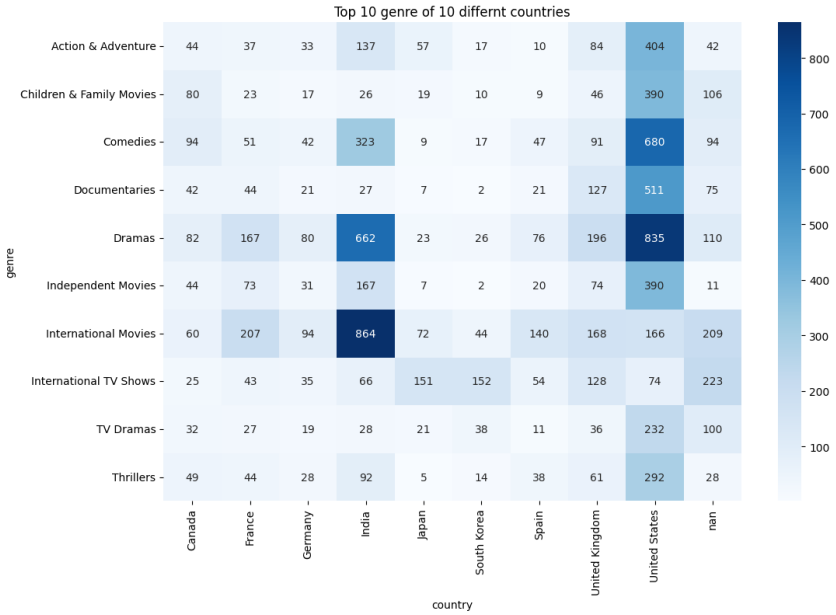
```
temp_genre = genre_country_df['genre'].value_counts()[:10].reset_index()
temp_genre.rename(columns = {'index':'genre', 'genre':'count'}, inplace=True)
genre_list = temp_genre['genre'].tolist()
df_top10_genre = genre_country_df.loc[genre_country_df['genre'].isin(genre_list)]
df_top10_genre.head()
```

	show_id	type	title	director	cast	date_added	release_year	ra
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	September 25, 2021	2020	P
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	September 24, 2021	2021	T\
2	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	September 24, 2021	2021	T\

```
df_top10_genre = df_top10_genre.loc[df_top10_genre["country"] != "Unknown"]
df_top10_genre["country"].value_counts()[:10]

temp_c = df_top10_genre["country"].value_counts()[:10].reset_index()
temp_c.rename(columns = {'index':'country', 'country':'count'}, inplace=True)
country_list = temp_c["country"].tolist()
df_top10_genre_countrywise = df_top10_genre.loc[df_top10_genre['country'].isin(country_list)]
df_top10_genre_countrywise.head()

heat_genre= pd.DataFrame(df_top10_genre_countrywise.groupby("genre")["country"].value_counts())
heat_genre.rename(columns = {"country" : "count"}, inplace = True)
heat_genre.reset_index(inplace = True)
heat_genre_final = heat_genre.pivot("genre" , "country" , "count")
plt.figure(figsize = (12,8))
sns.heatmap(heat_genre_final , annot = True,  cmap="Blues", fmt = "d")
plt.title("Top 10 genre of 10 differnt countries")
plt.show()
```



Conclusion :-

For India, netflix should add more content of genre International movies , Comedies and Dramas. For United States , Netflix should add more content of genre Dramas and Comedy. For Canada, Netflix should add more content of genre Dramas & Children and family movies.

Summary :-

Netflix added more movies as compare to TV shows Content for United States on netflix is maximum as compare to other countries. Netflix content is mostly availabe for adults only Most popular genres in recent years are International movies, Dramas, Comedies, International TV Shows and Action & Adventure. In 2021 , there is significant amount of drop in content added due to COVID pandemic. *Most of viewers of Netflix is from United States followed by India & United Kingdom

Movies:-

In United States , India and United kingdom movies are more popular as comapre to other countires Almost same no. of movies are added on netflix every month. Mostly movies are of "100 min" duration. Top people casted in Movies are from India. "Rajiv Chilakaa" is the most famous director among all.

TV Shows :-

TV Shows mostly are having season 1 and season 2 respectively. For Japan and South Korea, netflix should focus more on TV showes as compare to movies

✓ 2s completed at 22:26