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

```
df = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv")
df
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...
...	...	...	...	...	...	...	...	...	...	...	...	...
					Mark Buffalo							A political

Exploration of the dataset

```
df.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

```
df.isna().sum()
```

```
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
```

Bifercation and Unnesting of the dataset

Cast

```
bifer = df['cast'].apply(lambda x: str(x).split(', ')).tolist()
df_1 = pd.DataFrame(bifer, index=df['title'])
df_1 = df_1.stack()
df_1 = pd.DataFrame(df_1.reset_index())
df_act = df_1.rename(columns = {0:'Actor'})
df_act.drop(['level_1'], axis = 1, inplace = True)
df_act.head()
```

	title	Actor
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

Director

```
bifer = df['director'].apply(lambda x: str(x).split(', ')).tolist()
df_1 = pd.DataFrame(bifer, index=df['title'])
df_1 = df_1.stack()
df_1 = pd.DataFrame(df_1.reset_index())
df_dir = df_1.rename(columns = {0:'Director'})
df_dir.drop(['level_1'], axis = 1, inplace = True)
df_dir.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

Genre

```
bifer = df['listed_in'].apply(lambda x: str(x).split(', ')).tolist()
df_1 = pd.DataFrame(bifer, index=df['title'])
df_1 = df_1.stack()
df_1 = pd.DataFrame(df_1.reset_index())
df_gen = df_1.rename(columns = {0:'Genre'})
df_gen.drop(['level_1'], axis = 1, inplace = True)
df_gen.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

Country

```
bifer = df['country'].apply(lambda x: str(x).split(', ')).tolist()
df_1 = pd.DataFrame(bifer, index=df['title'])
df_1 = df_1.stack()
df_1 = pd.DataFrame(df_1.reset_index())
df_cou = df_1.rename(columns = {0:'Country'})
df_cou.drop(['level_1'], axis = 1, inplace = True)
df_cou.head()
```

	title	Country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	nan
3	Jailbirds New Orleans	nan

Merging the dataset

```
df_new1 = df_act.merge(df_dir, on=['title'], how = 'inner')
df_new2 = df_new1.merge(df_gen, on=['title'], how = 'inner')
df_new = df_new2.merge(df_cou, on=['title'], how = 'inner')
df_new.head()
```

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

Null Values Treatment

```
df_new['Actor'].replace(['nan'],['Unknown Actors'], inplace = True)
df_new['Director'].replace(['nan'],['Unknown Directors'], inplace = True)
df_new['Country'].replace(['nan'],[np.nan], inplace = True)
df_new.head()
```

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

```
df2 = df_new.merge(df[['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description']], on=['title'], how='left')

df2.head()
```

title	Actor	Director	Genre	Country	show_id	type	director	cast	country	date_added	release_year	rating	dur
Is Dead	Actors	Johnson		States			Johnson		States	25, 2021			

```
df2.drop(['show_id', 'director', 'cast', 'country', 'listed_in'], axis = 1, inplace=True)
```

```
df2.head()
```

	title	Actor	Director	Genre	Country	type	date_added	release_year	rating	duration	description
0	Dick Johnson Is Dead	Unknown Actors	Kirsten Johnson	Documentaries	United States	Movie	September 25, 2021	2020	PG-13	90 min	As her father nears the end of his life, filmm...
1	Blood & Water	Ama Qamata	Unknown Directors	International TV Shows	South Africa	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
2	Blood & Water	Ama Qamata	Unknown Directors	TV Dramas	South Africa	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
3	Blood & Water	Ama Qamata	Unknown Directors	TV Mysteries	South Africa	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
4	Blood &	Khosi	Unknown	International TV	South	TV	September	2021	TV-MA	2	After crossing paths at a party, a Cape Town

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201991 entries, 0 to 201990
Data columns (total 11 columns):
 #   Column        Non-Null Count   Dtype 
---  ------        --------------   ----- 
 0   title         201991 non-null  object
 1   Actor         201991 non-null  object
 2   Director      201991 non-null  object
 3   Genre         201991 non-null  object
 4   Country       190094 non-null  object
 5   type          201991 non-null  object
 6   date_added    201833 non-null  object
 7   release_year  201991 non-null  int64 
 8   rating        201924 non-null  object
 9   duration      201988 non-null  object
 10  description   201991 non-null  object
dtypes: int64(1), object(10)
memory usage: 18.5+ MB
```

```
df2.isna().sum()
```

```
title              0
Actor              0
Director           0
Genre              0
Country        11897
type               0
date_added       158
release_year       0
rating            67
duration           3
description        0
dtype: int64
```

```
df2.columns
```

```
Index(['title', 'Actor', 'Director', 'Genre', 'Country', 'type', 'date_added',
       'release_year', 'rating', 'duration', 'description'],
      dtype='object')
```

Making uniform data as rating has values in min, which is not possible

```
df2.loc[df2['rating'].str.contains('min', na=False),'rating'] = 'NR'
df2['rating'].fillna('NR',inplace=True)
pd.set_option('display.max_rows',None)
```

```
df.duration.value_counts()
```

```
1 Season      1793
2 Seasons      425
3 Seasons      199
90 min         152
94 min         146
97 min         146
93 min         146
91 min         144
95 min         137
96 min         130
92 min         129
102 min        122
98 min         120
99 min         118
101 min        116
88 min         116
103 min        114
106 min        111
100 min        108
89 min         106
104 min        104
86 min         103
105 min        101
87 min         101
107 min         98
110 min         97
4 Seasons       95
108 min         87
116 min         80
112 min         74
85 min          73
113 min         69
109 min         69
111 min         68
84 min          67
118 min         65
83 min          65
5 Seasons       65
119 min         63
81 min          62
115 min         61
117 min         61
120 min         56
114 min         56
121 min         54
82 min          52
124 min         52
127 min         48
122 min         45
78 min          45
123 min         44
126 min         44
80 min          43
133 min         42
128 min         41
130 min         40
135 min         39
137 min         38
```

```
df2['month_added']=pd.to_datetime(df2['date_added']).dt.strftime('%B-%Y')
```

Non Graphical Analysis

```
df2.shape
```

```
(201991, 12)
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201991 entries, 0 to 201990
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype 
---  ------        --------------   ----- 
 0   title         201991 non-null  object
 1   Actor         201991 non-null  object
 2   Director      201991 non-null  object
 3   Genre         201991 non-null  object
 4   Country       190094 non-null  object
 5   type          201991 non-null  object
```

```
 6   date_added    201833 non-null  object
 7   release_year  201991 non-null  int64
 8   rating        201991 non-null  object
 9   duration      201988 non-null  object
 10  description   201991 non-null  object
 11  month_added   201833 non-null  object
dtypes: int64(1), object(11)
memory usage: 20.0+ MB
```

```
df2.describe()
```

	release_year
count	201991.000000
mean	2013.452891
std	9.003933
min	1925.000000
25%	2012.000000
50%	2016.000000
75%	2019.000000
max	2021.000000

```
df2['Genre'].value_counts()
```

```
Dramas                          29775
International Movies            28211
Comedies                        20829
International TV Shows          12845
Action & Adventure              12216
Independent Movies               9834
Children & Family Movies         9771
TV Dramas                        8942
Thrillers                        7107
Romantic Movies                  6412
TV Comedies                      4963
Crime TV Shows                   4733
Horror Movies                    4571
Kids' TV                         4568
Sci-Fi & Fantasy                 4037
Music & Musicals                 3077
Romantic TV Shows                3049
Documentaries                    2407
Anime Series                     2313
TV Action & Adventure            2288
Spanish-Language TV Shows        2126
British TV Shows                 1808
Sports Movies                    1531
Classic Movies                   1434
TV Mysteries                     1281
Korean TV Shows                  1122
Cult Movies                      1077
TV Sci-Fi & Fantasy              1045
Anime Features                   1045
TV Horror                         941
Docuseries                        845
LGBTQ Movies                      838
TV Thrillers                      768
Teen TV Shows                     742
Reality TV                        735
Faith & Spirituality              719
Stand-Up Comedy                   540
Movies                            412
TV Shows                          337
Classic & Cult TV                 272
Stand-Up Comedy & Talk Shows      268
Science & Nature TV               157
Name: Genre, dtype: int64
```

```
df2['Genre'].nunique()
```

```
42
```

```
df2['Country'].nunique()
```

```
127
```

```
df2['Actor'].nunique()
```

```
36440
```

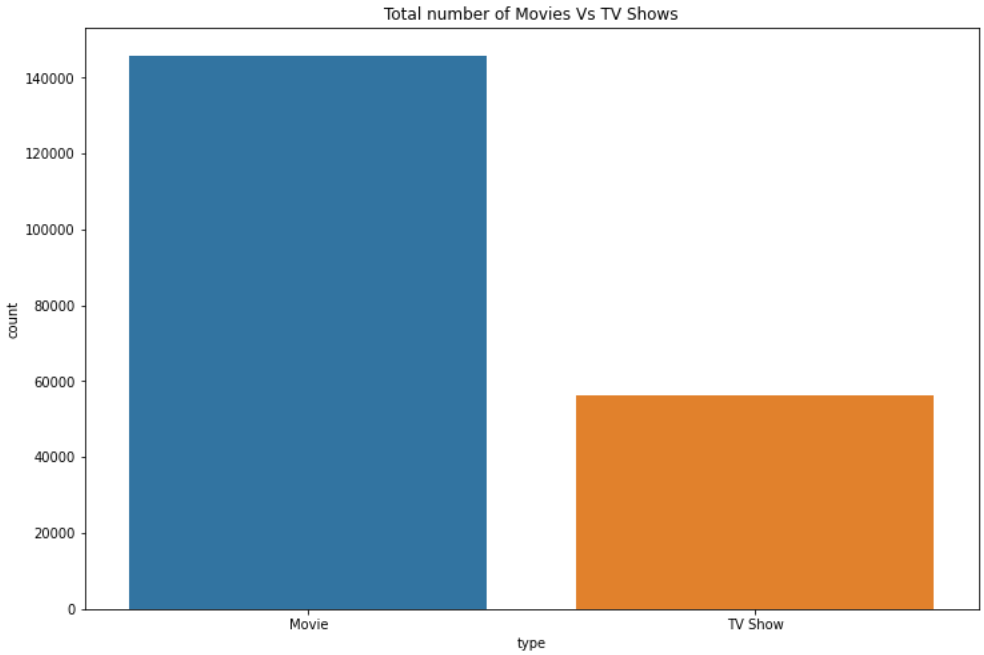
```
df2['Director'].nunique()
```

```
4994
```

```
df2['duration'].nunique()
```

```
220
```

```
plt.figure(figsize=(12,8))
plt.title("Total number of Movies Vs TV Shows")
sns.countplot(data = df2,
              x = "type")
plt.show()
```



Top 10 Country wise analysis on Movies and TV Shows

```
top_10_Country = df2["Country"].value_counts().index[:10]
top_10_Country
```

```
Index(['United States', 'India', 'United Kingdom', 'Japan', 'France', 'Canada',
       'Spain', 'South Korea', 'Germany', 'Mexico'],
      dtype='object')
```

```
top_10_data = df2.loc[(df2["Country"].isin(top_10_Country))]
top_10_data.head()
```

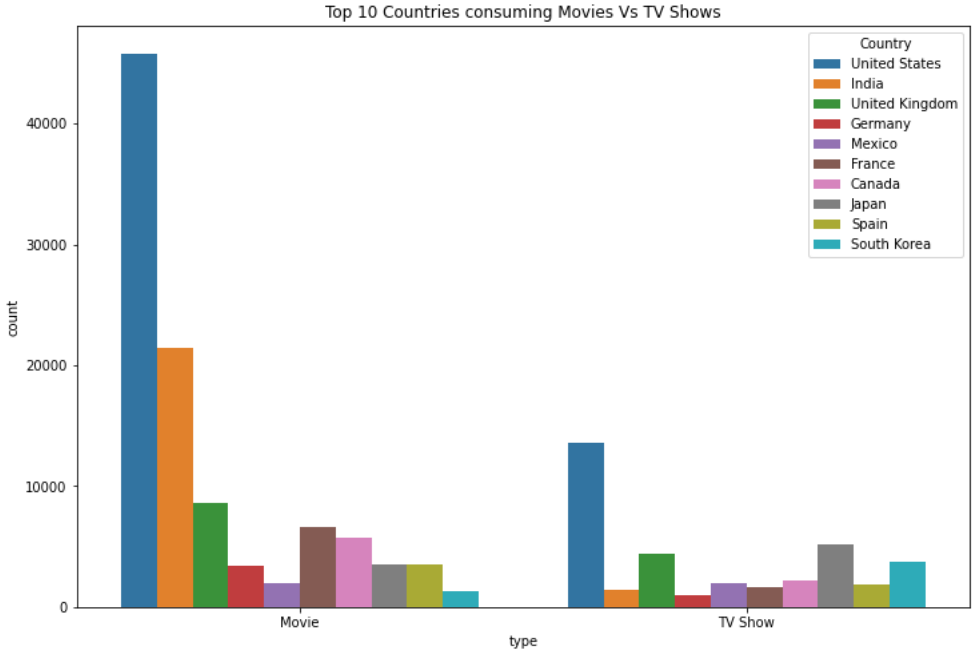
	title	Actor	Director	Genre	Country	type	date_added	release_year	rating	duration	description	month_added
0	Dick Johnson	Unknown	Kirsten	Documentaries	United	Movie	September	2020	PG-13	90 min	As her father nears the end	September-

```
top_10_data["Country"].value_counts()
```

```
United States     59349
India             22814
United Kingdom    12945
Japan              8679
France             8254
Canada             7915
Spain              5315
South Korea        5043
Germany            4383
Mexico             3941
Name: Country, dtype: int64
```

89	Factory	Mons	Directors	TV Comedies	India	Show	24-2021	2021	TV-MA	Seasons	content known	2021

```
plt.figure(figsize=(12,8))
plt.title("Top 10 Countries consuming Movies Vs TV Shows")
sns.countplot(data = top_10_data,
              x = "type",
              hue = "Country")
plt.show()
```



Division of the datasets as per Movies and TV Show matrics

```
df_movies = df2[df2['type']=="Movie"]
df_web = df2[df2['type']=="TV Show"]
```

```
a=df_movies.groupby('release_year').size().sort_values(ascending=False).reset_index()
a.rename({0:'Count_of_movies_produced_per_year'},axis=1,inplace=True)
```

```
b=df_web.groupby('release_year').size().sort_values(ascending=False).reset_index()
b.rename({0:'Count_of_tvshows_produced_per_year'},axis=1,inplace=True)
```

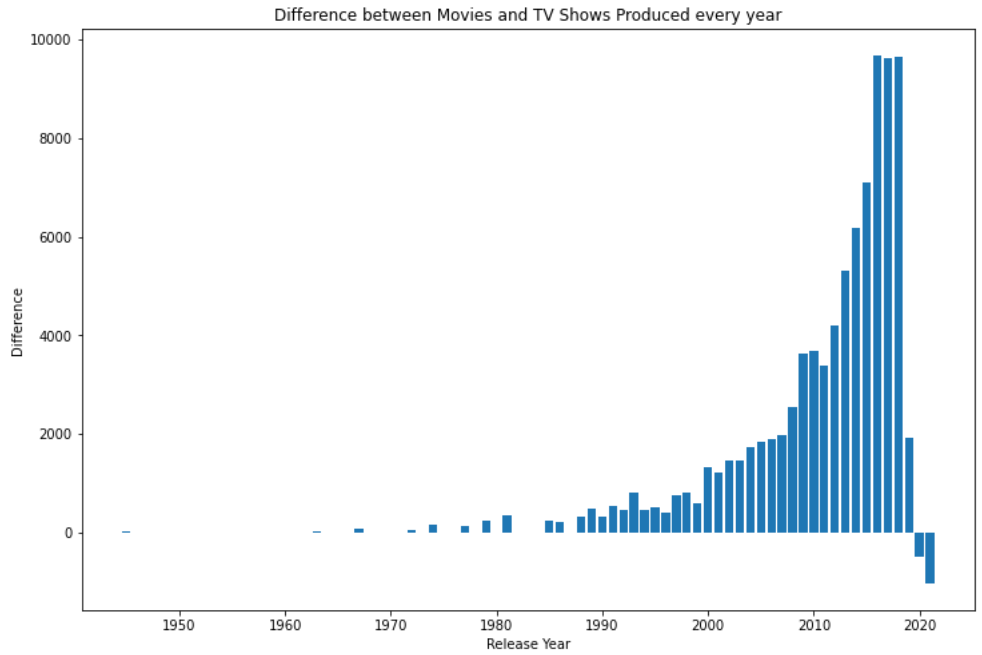
```
c=pd.merge(a,b,on='release_year',how='inner')
```

```
c['difference']=c['Count_of_movies_produced_per_year']-c['Count_of_tvshows_produced_per_year']
```

```
c.set_index('release_year',inplace=True)
```

```
plt.figure(figsize=(12,8))
plt.bar(c['difference'].index, c['difference'])
```

```
plt.title('Difference between Movies and TV Shows Produced each year')
plt.xlabel('Release Year')
plt.ylabel('Difference')
plt.show()
```



Data Analysis on Movies

```
df_movies.head()
```

	title	Actor	Director	Genre	Country	type	date_added	release_year	rating	duration	description	month_added
0	Dick Johnson Is Dead	Unknown Actors	Kirsten Johnson	Documentaries	United States	Movie	September 25, 2021	2020	PG-13	90 min	As her father nears the end of his life, filmm...	September-2021
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	Children & Family Movies	NaN	Movie	September 24, 2021	2021	PG	91 min	Equestria's divided. But a bright-eyed hero be...	September-2021
160	My Little Pony: A New Generation	Vanessa Hudgens	José Luis Ucha	Children & Family Movies	NaN	Movie	September 24, 2021	2021	PG	91 min	Equestria's divided. But a bright-eyed hero be...	September-2021
161	My Little Pony: A	Kimiko	Robert	Children &	NaN	Movie	September	2021	PG	91 min	Equestria's divided. But a	September-

```
df_movies["duration"]=df_movies["duration"].str.replace(" min","")
```

```
<ipython-input-42-f724a670af50>: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/user_guide/indexing.html#returning-a-view-versus-a-co
  df_movies["duration"]=df_movies["duration"].str.replace(" min","")
```

```
df_movies.head()
```

	title	Actor	Director	Genre	Country	type	date_added	release_year	rating	duration	description	month_added
0	Dick Johnson Is Dead	Unknown Actors	Kirsten Johnson	Documentaries	United States	Movie	September 25, 2021	2020	PG-13	90	As her father nears the end of his life, filmm...	September-2021
	My Little Pony: A	Vanessa	Robert	Children &			September				Equestria's divided. But a	September-

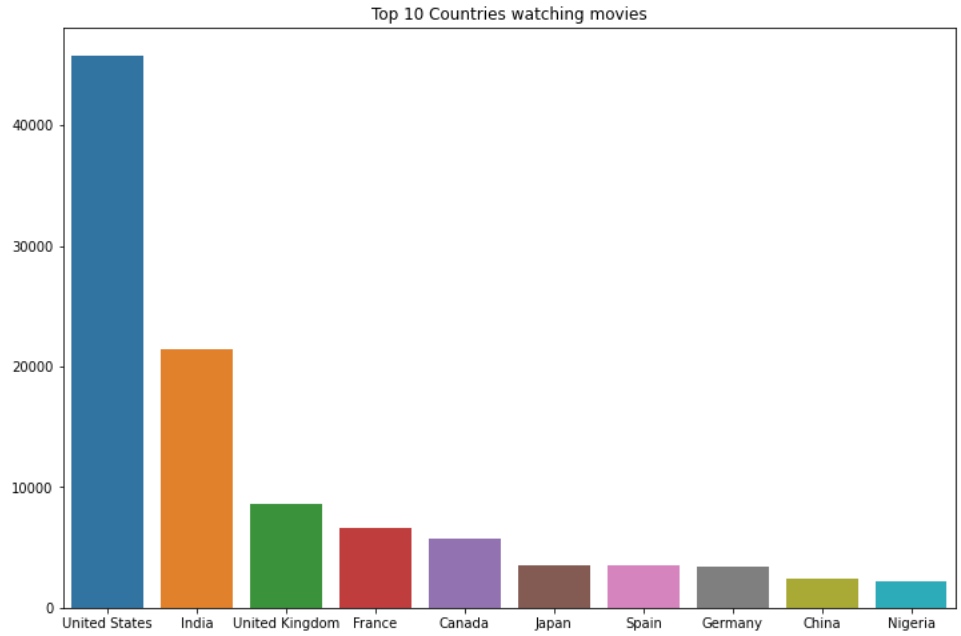
```
df_movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 145843 entries, 0 to 201990
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype 
---  ------        --------------   ----- 
 0   title         145843 non-null  object
 1   Actor         145843 non-null  object
 2   Director      145843 non-null  object
 3   Genre         145843 non-null  object
 4   Country       139644 non-null  object
 5   type          145843 non-null  object
 6   date_added    145843 non-null  object
 7   release_year  145843 non-null  int64 
 8   rating        145843 non-null  object
 9   duration      145840 non-null  object
 10  description   145843 non-null  object
 11  month_added   145843 non-null  object
dtypes: int64(1), object(11)
memory usage: 14.5+ MB
```

```
Country_max = df_movies.Country.value_counts().head(10)
Country_max
```

```
United States     45816
India             21411
United Kingdom     8560
France             6607
Canada             5738
Japan              3525
Spain              3469
Germany            3427
China              2377
Nigeria            2236
Name: Country, dtype: int64
```

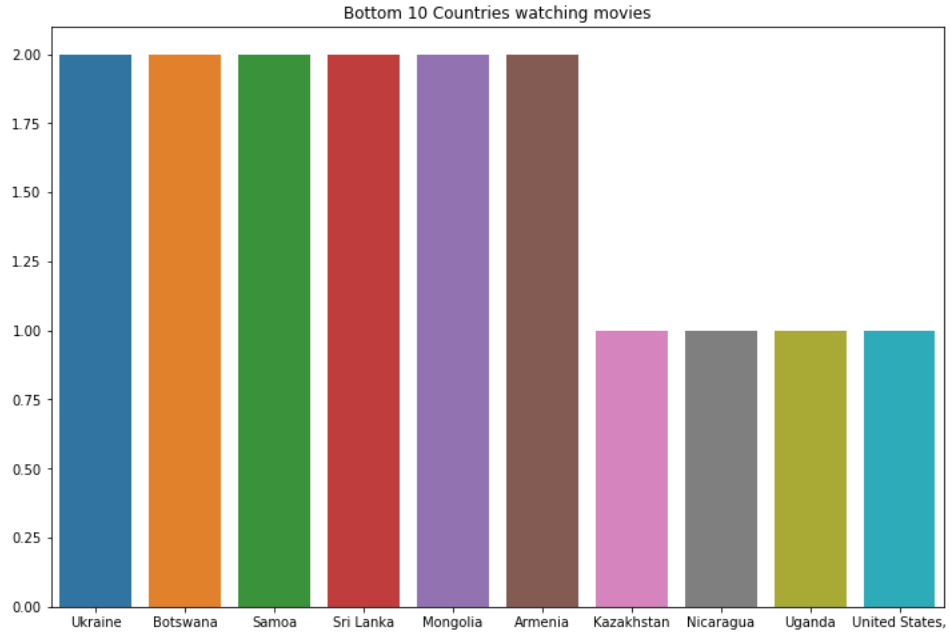
```
plt.figure(figsize=(12,8))
plt.title("Top 10 Countries watching movies")
sns.barplot(x = Country_max.index, y = Country_max.values)
plt.show()
```



```
Country_min = df_movies.Country.value_counts().tail(10)
Country_min
```

```
Ukraine           2
Botswana          2
Samoa             2
Sri Lanka         2
Mongolia          2
Armenia           2
Kazakhstan        1
Nicaragua         1
Uganda            1
United States,    1
Name: Country, dtype: int64
```

```
plt.figure(figsize=(12,8))
plt.title("Bottom 10 Countries watching movies")
sns.barplot(x = Country_min.index, y = Country_min.values)
plt.show()
```

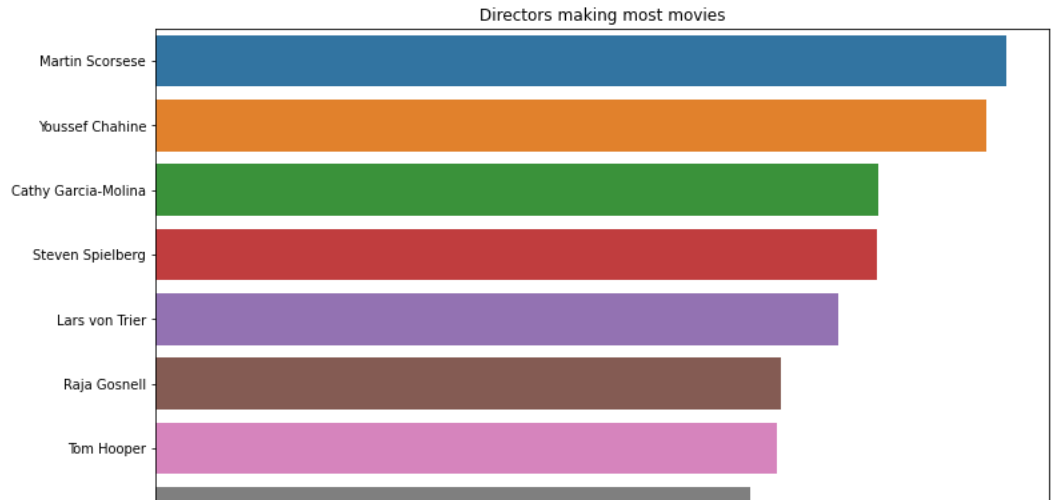



```
Director_max = df_movies.Director.value_counts().head(10)
Director_max
```

```
Unknown Directors      1285
Martin Scorsese         419
Youssef Chahine         409
Cathy Garcia-Molina     356
Steven Spielberg        355
Lars von Trier          336
Raja Gosnell            308
Tom Hooper              306
McG                     293
David Dhawan            270
Name: Director, dtype: int64
```

```
Director_max.drop(['Unknown Directors'], axis = 0, inplace=True )
```

```
plt.figure(figsize=(12,8))
plt.title("Directors making most movies")
sns.barplot(y = Director_max.index, x = Director_max.values)
plt.show()
```

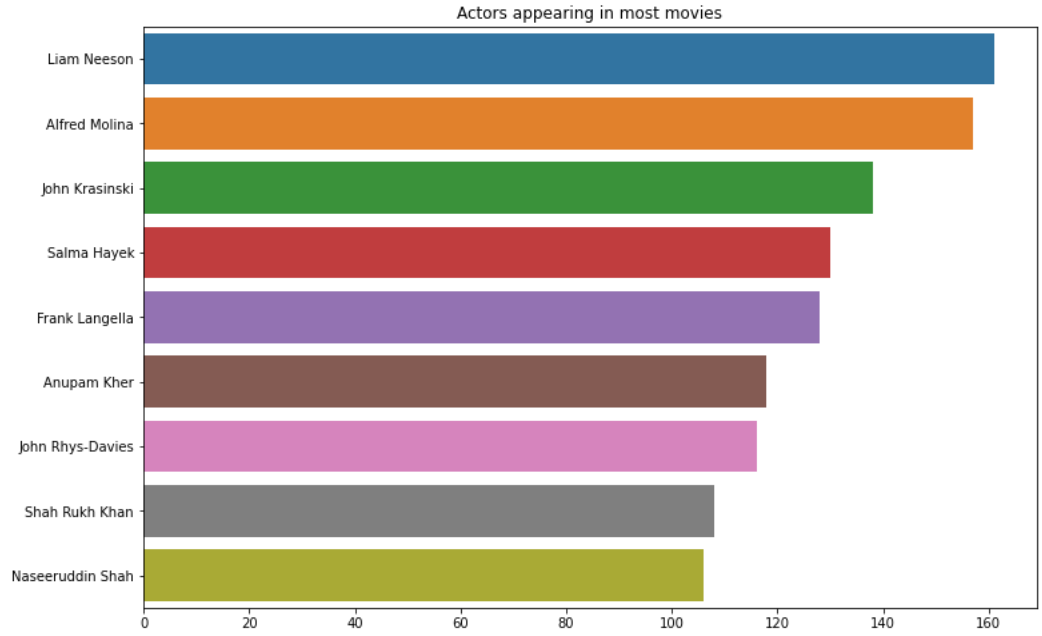



```
Actors_max = df_movies.Actor.value_counts().head(10)
Actors_max
```

```
Unknown Actors      1328
Liam Neeson          161
Alfred Molina        157
John Krasinski       138
Salma Hayek          130
Frank Langella       128
Anupam Kher          118
John Rhys-Davies     116
Shah Rukh Khan       108
Naseeruddin Shah     106
Name: Actor, dtype: int64
```

```
Actors_max.drop(['Unknown Actors'], axis = 0, inplace=True )
```

```
plt.figure(figsize=(12,8))
plt.title("Actors appearing in most movies")
sns.barplot(y = Actors_max.index, x = Actors_max.values)
plt.show()
```

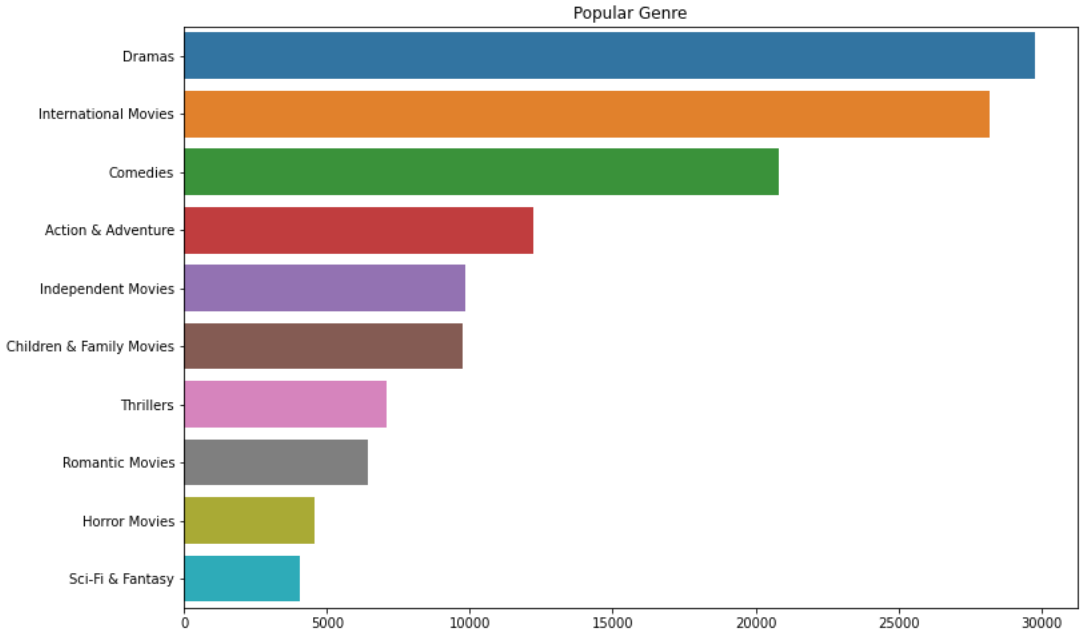



```
pop_genre = df_movies.Genre.value_counts().head(10)
pop_genre
```

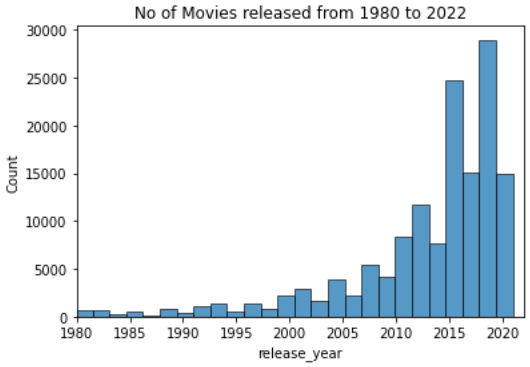
```
Dramas                    29775
International Movies      28211
Comedies                  20829
Action & Adventure        12216
```

```
Independent Movies           9834
Children & Family Movies     9771
Thrillers                    7107
Romantic Movies              6412
Horror Movies                4571
Sci-Fi & Fantasy             4037
Name: Genre, dtype: int64
```

```
plt.figure(figsize=(12,8))
plt.title("Popular Genre")
sns.barplot(y = pop_genre.index, x = pop_genre.values)
plt.show()
```




```
a=sns.histplot(data=df_movies,x='release_year',bins=50)
a.set_xlim(1980,2022)
plt.title('No of Movies released from 1980 to 2022')
plt.show()
```




```
df_movies['duration'] = df_movies['duration'].astype(str).astype("float")
```

```
Duration_max = df_movies.duration.value_counts().head(10)
Duration_max
```

```
94.0     4343
106.0    4040
97.0     3624
95.0     3560
96.0     3484
93.0     3480
90.0     3305
105.0    3209
107.0    3103
101.0    3048
Name: duration, dtype: int64
```

```
max_duration = df_movies["duration"].value_counts().index[:10]
max_duration
```

```
Float64Index([94.0, 106.0, 97.0, 95.0, 96.0, 93.0, 90.0, 105.0, 107.0, 101.0], dtype='float64')
```

```
top_10_Genre = df_movies["Genre"].value_counts().index[:10]
top_10_Genre
```

```
Index(['Dramas', 'International Movies', 'Comedies', 'Action & Adventure',
       'Independent Movies', 'Children & Family Movies', 'Thrillers',
       'Romantic Movies', 'Horror Movies', 'Sci-Fi & Fantasy'],
      dtype='object')
```

```
max_data = df_movies.loc[(df_movies["duration"].isin(max_duration))&(df_movies["Genre"].isin(top_10_Genre))]
max_data.head()
```

	title	Actor	Director	Genre	Country	type	date_added	release_year	rating	duration	description	month_added
554	Intrusion	Freida Pinto	Adam Salky	Thrillers	NaN	Movie	September 22, 2021	2021	TV-14	94.0	After a deadly home invasion at a couple's new...	September-2021
555	Intrusion	Logan Marshall-Green	Adam Salky	Thrillers	NaN	Movie	September 22, 2021	2021	TV-14	94.0	After a deadly home invasion at a couple's new...	September-2021
556	Intrusion	Robert	Adam	Thrillers	NaN	Movie	September	2021	TV-14	94.0	After a deadly home invasion at a	September-

```
df_movies['duration'].max()
```

```
312.0
```

```
df_movies['duration'].min()
```

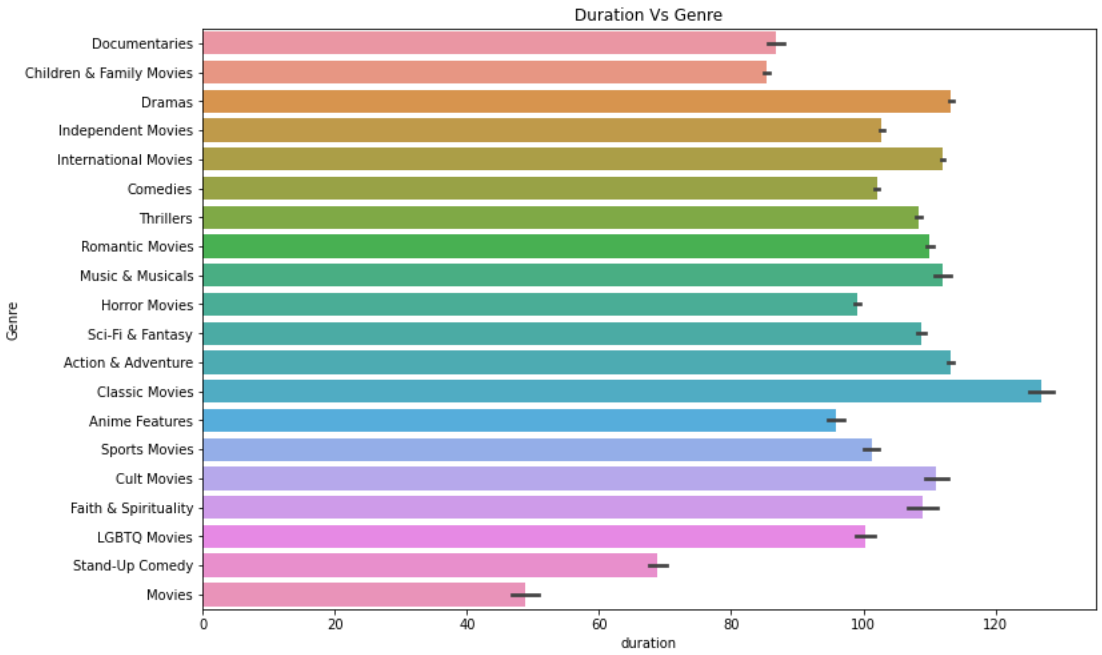
```
3.0
```

The maximum duration of a movies is 312 minutes and the minimum duration of a movie is 3 minutes

```
duration_max = df_movies.loc[df_movies['duration']==312.0]
duration_max
```

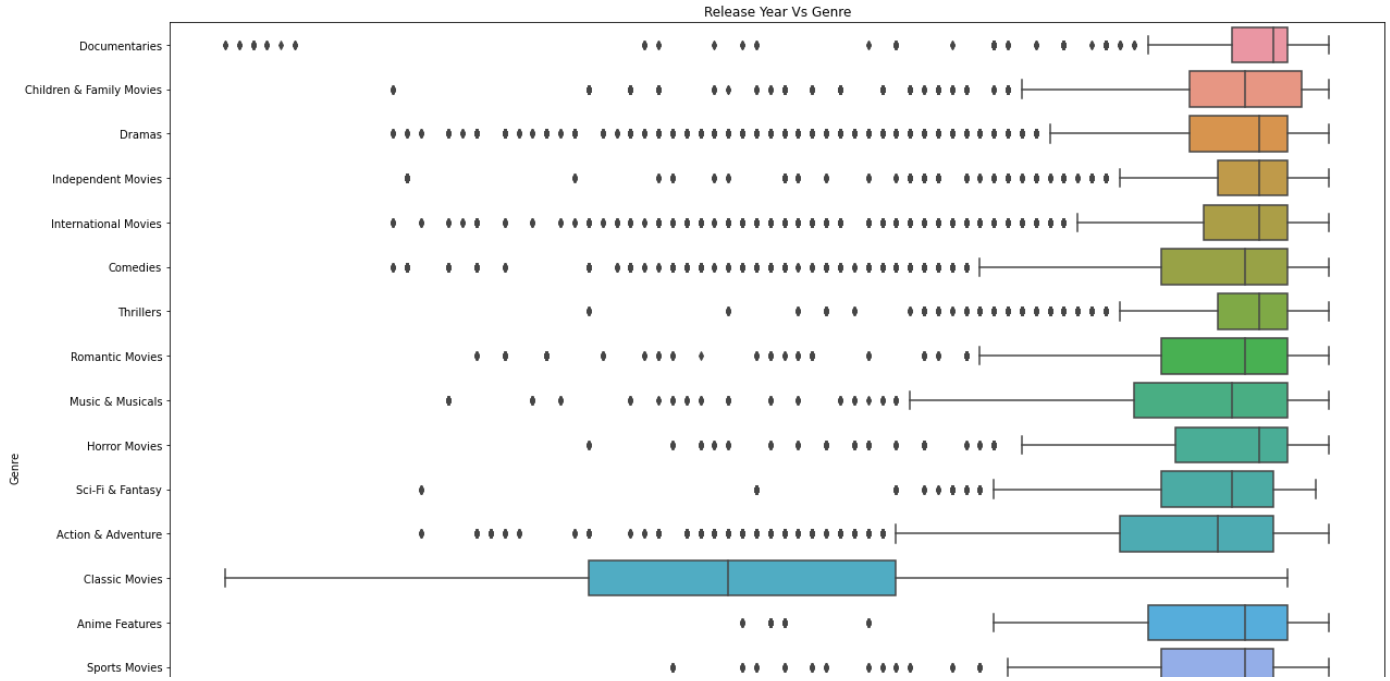
	title	Actor	Director	Genre	Country	type	date_added	release_year	rating	duration	description	month_added
99337	Black Mirror: Bandersnatch	Fionn Whitehead	Unknown Directors	Dramas	United States	Movie	December 28, 2018	2018	TV-MA	312.0	In 1984, a young programmer begins to question...	December-2018
99338	Black Mirror: Bandersnatch	Fionn Whitehead	Unknown Directors	International Movies	United States	Movie	December 28, 2018	2018	TV-MA	312.0	In 1984, a young programmer begins to	December-2018

```
plt.figure(figsize=(12,8))
plt.title("Duration Vs Genre")
sns.barplot(data = df_movies, x = "duration", y = "Genre")
plt.show()
```




```
# plt.figure(figsize=(15,10))
# plt.scatter(y = df_movies.Genre, x = df_movies.duration)
# plt.title("Genre V/s Duration")
# plt.xlabel("Genre")
# plt.ylable("Duration")
# plt.show()
```

```
plt.figure(figsize=(20,15))
plt.title("Release Year Vs Genre")
sns.boxplot(x = 'release_year', y = 'Genre', data = df_movies)
plt.show()
```



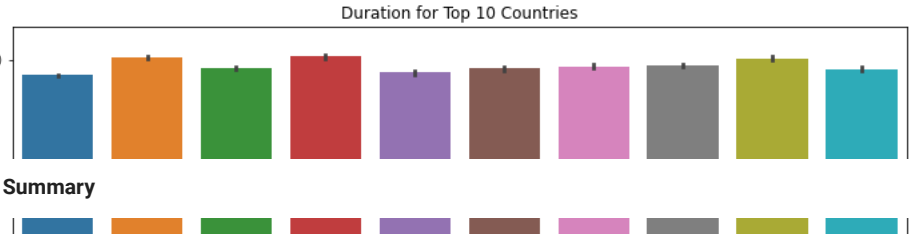

```
top_10_Country = df_movies["Country"].value_counts().index[:10]
top_10_Country
```

```
Index(['United States', 'India', 'United Kingdom', 'France', 'Canada', 'Japan',
       'Spain', 'Germany', 'China', 'Nigeria'],
      dtype='object')
```

```
max_data = df_movies.loc[(df_movies["duration"].isin(max_duration))&(df_movies["Genre"].isin(top_10_Genre))&(df_movies["Country"].isin(top_10
max_data.head()
```

	title	Actor	Director	Genre	Country	type	date_added	release_year	rating	duration	description	month_added
685	Dark Skies	Keri Russell	Scott Stewart	Horror Movies	United States	Movie	September 19, 2021	2013	PG-13	97.0	A family's idyllic suburban life shatters when...	September-2021
686	Dark Skies	Keri Russell	Scott Stewart	Sci-Fi & Fantasy	United States	Movie	September 19, 2021	2013	PG-13	97.0	A family's idyllic suburban life shatters when...	September-2021
687	Dark Skies	Josh Hamilton	Scott Stewart	Horror Movies	United States	Movie	September 19, 2021	2013	PG-13	97.0	A family's idyllic suburban life shatters when...	September-2021

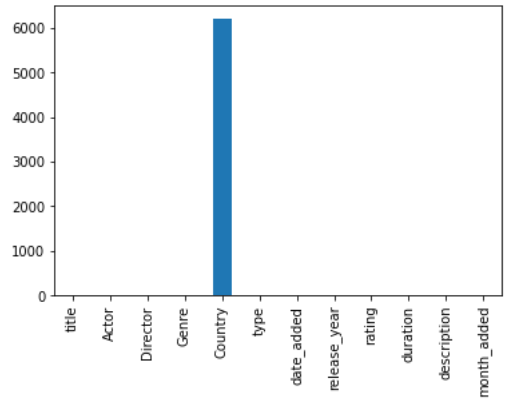
```
plt.figure(figsize=(12,8))
plt.title("Duration for Top 10 Countries")
sns.barplot(data = max_data, y = "duration", x = "Country")
plt.show()
```



Statistical Summary

```
df_movies.isna().sum().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f334b68eaf0>
```




```
df_movies.isna().sum()
```

```
title              0
Actor              0
Director           0
Genre              0
Country         6199
type               0
date_added         0
release_year       0
rating             0
duration           3
description        0
month_added        0
dtype: int64
```

```
df_movies.shape
```

```
(145843, 12)
```

```
df_movies.describe()
```

	release_year	duration
count	145843.000000	145840.000000
mean	2012.135454	106.856452
std	9.805300	24.696519
min	1942.000000	3.000000
25%	2010.000000	93.000000
50%	2016.000000	104.000000
75%	2018.000000	119.000000
max	2021.000000	312.000000

```
df_movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 145843 entries, 0 to 201990
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   title         145843 non-null  object
```

```
 1   Actor         145843 non-null  object
 2   Director      145843 non-null  object
 3   Genre         145843 non-null  object
 4   Country       139644 non-null  object
 5   type          145843 non-null  object
 6   date_added    145843 non-null  object
 7   release_year  145843 non-null  int64
 8   rating        145843 non-null  object
 9   duration      145840 non-null  float64
 10  description   145843 non-null  object
 11  month_added   145843 non-null  object
dtypes: float64(1), int64(1), object(10)
memory usage: 18.5+ MB
```

```
df_movies['Genre'].value_counts()
```

```
Dramas                      29775
International Movies        28211
Comedies                    20829
Action & Adventure          12216
Independent Movies           9834
Children & Family Movies     9771
Thrillers                    7107
Romantic Movies              6412
Horror Movies                4571
Sci-Fi & Fantasy             4037
Music & Musicals             3077
Documentaries                2407
Sports Movies                1531
Classic Movies               1434
Cult Movies                  1077
Anime Features               1045
LGBTQ Movies                  838
Faith & Spirituality          719
Stand-Up Comedy               540
Movies                        412
Name: Genre, dtype: int64
```

```
df_movies['Genre'].unique()
```

```
array(['Documentaries', 'Children & Family Movies', 'Dramas',
       'Independent Movies', 'International Movies', 'Comedies',
       'Thrillers', 'Romantic Movies', 'Music & Musicals',
       'Horror Movies', 'Sci-Fi & Fantasy', 'Action & Adventure',
       'Classic Movies', 'Anime Features', 'Sports Movies', 'Cult Movies',
       'Faith & Spirituality', 'LGBTQ Movies', 'Stand-Up Comedy',
       'Movies'], dtype=object)
```

```
df_movies['Genre'].nunique()
```

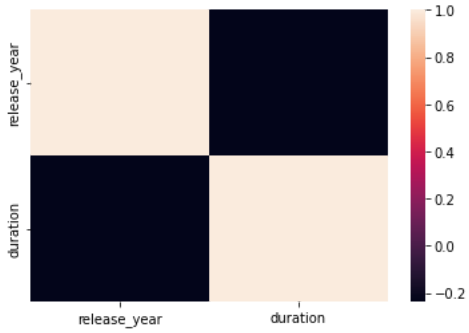
```
20
```

```
df_movies.corr()
```

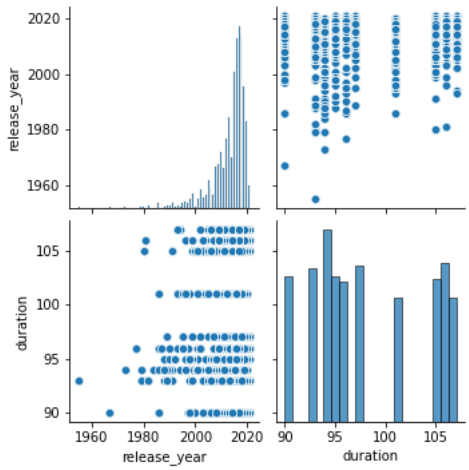
	release_year	duration
release_year	1.000000	-0.235663
duration	-0.235663	1.000000

```
sns.heatmap(data = df_movies.corr())
```

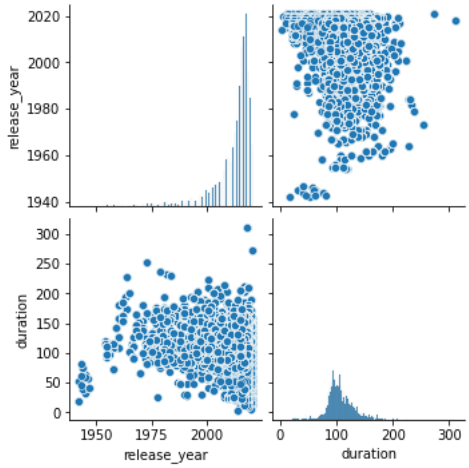
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f334af85a00>
```



```
sns.pairplot(data = max_data)
plt.show()
```




```
sns.pairplot(data = df_movies)
plt.show()
```



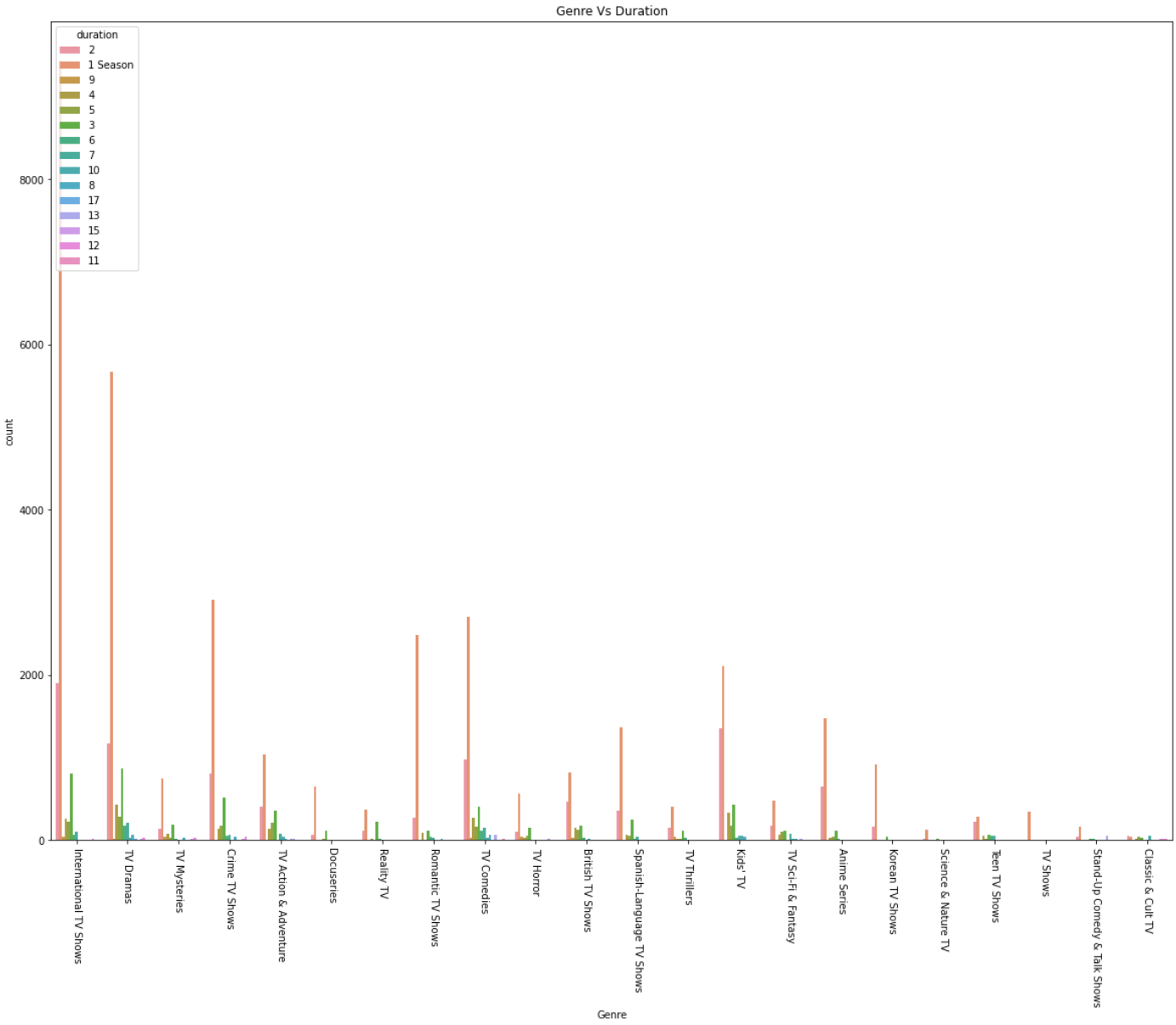
Data analysis on web series

```
df_web["duration"]=df_web["duration"].str.replace(" Seasons","")
```

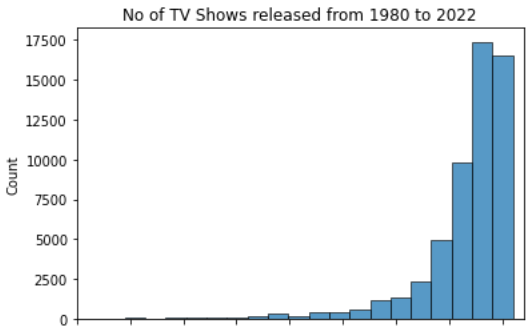
```
df_web.head(10)
```

	title	Actor	Director	Genre	Country	type	date_added	release_year	rating	duration	description	month_added
1	Blood & Water	Ama Qamata	Unknown Directors	International TV Shows	South Africa	TV Show	September-2021	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	September-2021
2	Blood &	Ama Qamata	Unknown Directors	TV Dramas	South Africa	TV Show	September-2021	2021	TV-MA	2	After crossing paths at a party, a	September-2021

```
plt.figure(figsize=(20,15))
plt.title("Genre Vs Duration")
sns.countplot(data = df_web, x = 'Genre', hue = 'duration')
plt.xticks(rotation=-90)
plt.show()
```



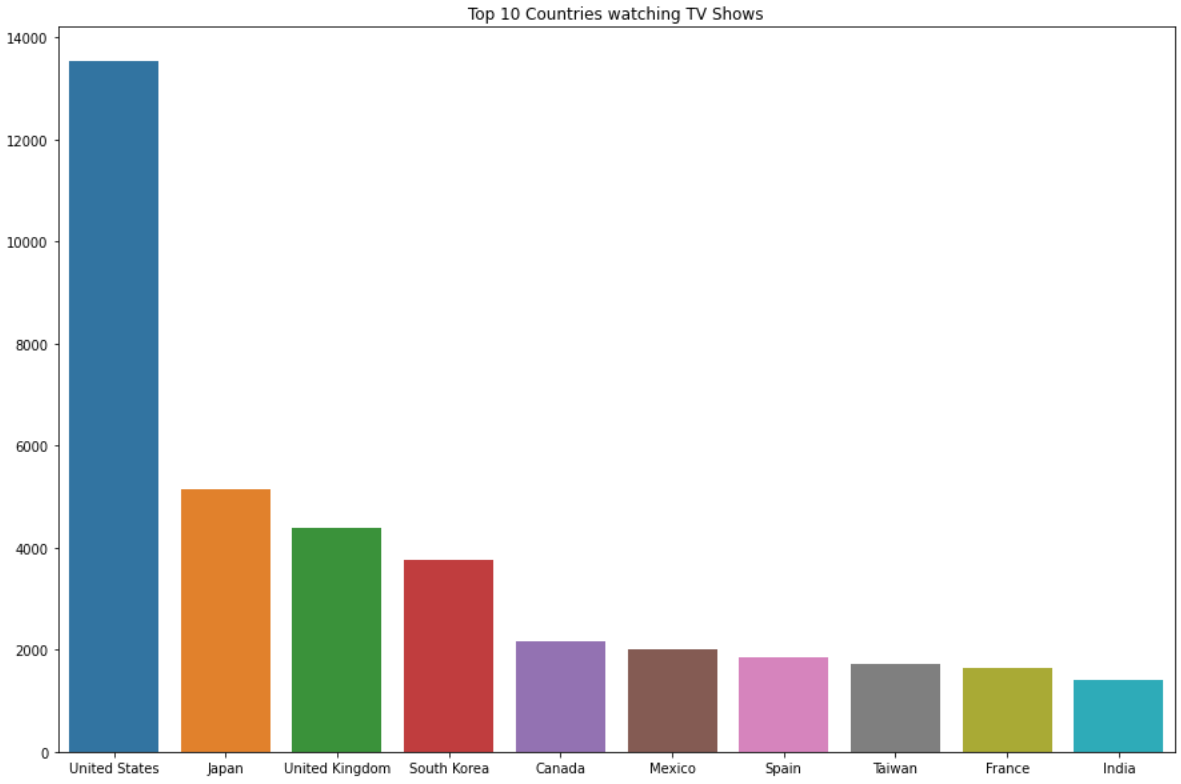

```
a=sns.histplot(data=df_web,x='release_year',bins=50)
a.set_xlim(1980,2022)
plt.title('No of TV Shows released from 1980 to 2022')
plt.show()
```




```
Country_max_web = df_web.Country.value_counts().head(10)
Country_max_web
```

```
United States     13533
Japan              5154
United Kingdom     4385
South Korea        3754
Canada             2177
Mexico             2018
Spain              1846
Taiwan             1719
France             1647
India              1403
Name: Country, dtype: int64
```

```
plt.figure(figsize=(15,10))
plt.title("Top 10 Countries watching TV Shows")
sns.barplot(x = Country_max_web.index, y = Country_max_web.values)
plt.show()
```



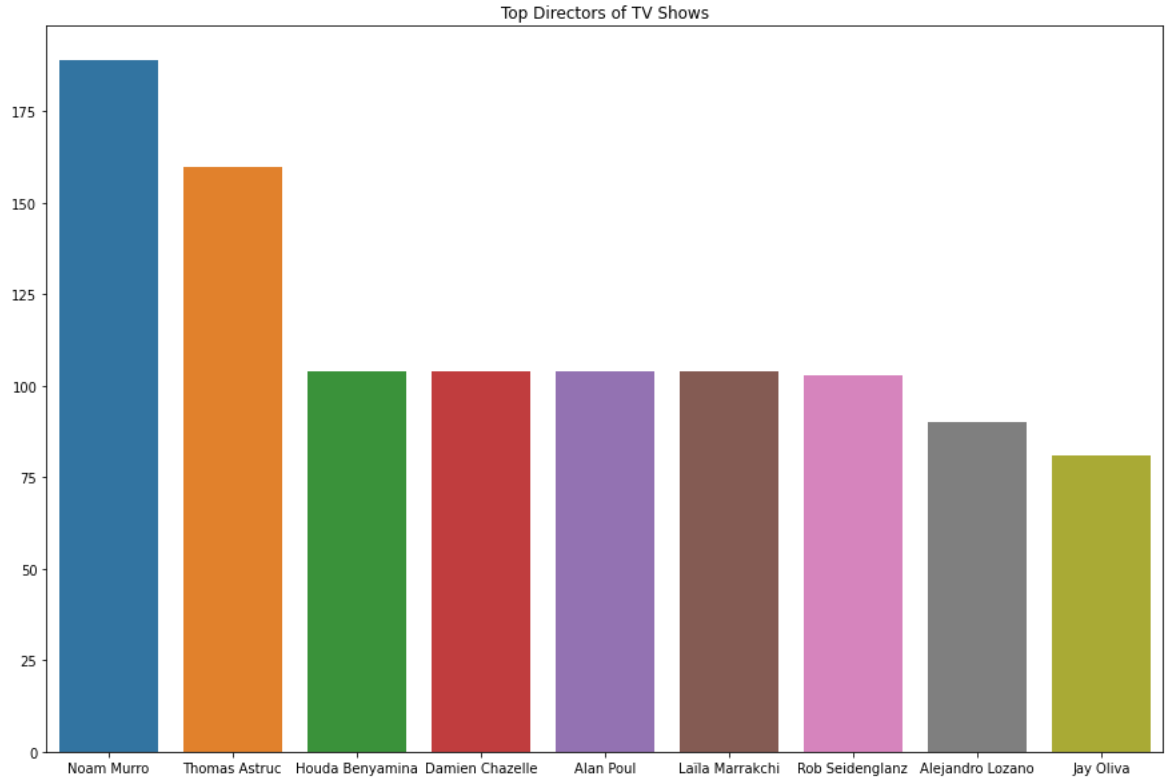

```
Director_max_web = df_web.Director.value_counts().head(10)
Director_max_web
```

```
Unknown Directors    49358
Noam Murro             189
Thomas Astruc          160
Houda Benyamina        104
Damien Chazelle        104
Alan Poul              104
Laïla Marrakchi        104
Rob Seidenglanz        103
```

```
Alejandro Lozano        90
Jay Oliva               81
Name: Director, dtype: int64
```

```
Director_max_web.drop(['Unknown Directors'], axis = 0, inplace=True )
```

```
plt.figure(figsize=(15,10))
plt.title("Top Directors of TV Shows")
sns.barplot(x = Director_max_web.index, y = Director_max_web.values)
plt.show()
```

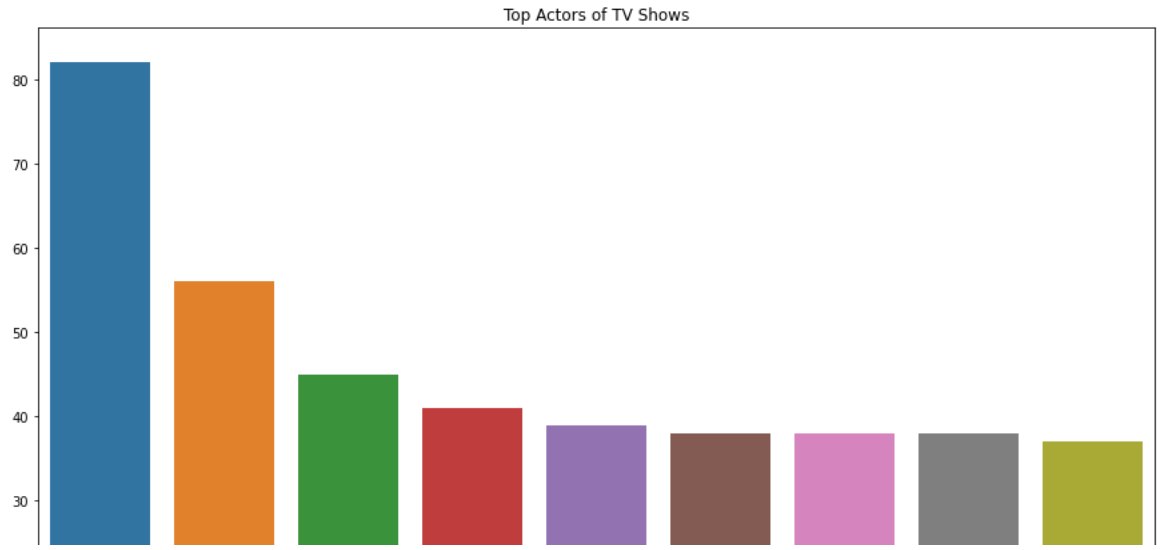



```
Actors_max_web = df_web.Actor.value_counts().head(10)
Actors_max_web
```

```
Unknown Actors        818
David Attenborough     82
Takahiro Sakurai       56
Yuki Kaji              45
Ai Kayano              41
Junichi Suwabe         39
Jun Fukuyama           38
Daisuke Ono            38
Yuichi Nakamura        38
Kate Harbour           37
Name: Actor, dtype: int64
```

```
Actors_max_web.drop(['Unknown Actors'], axis = 0, inplace=True )
```

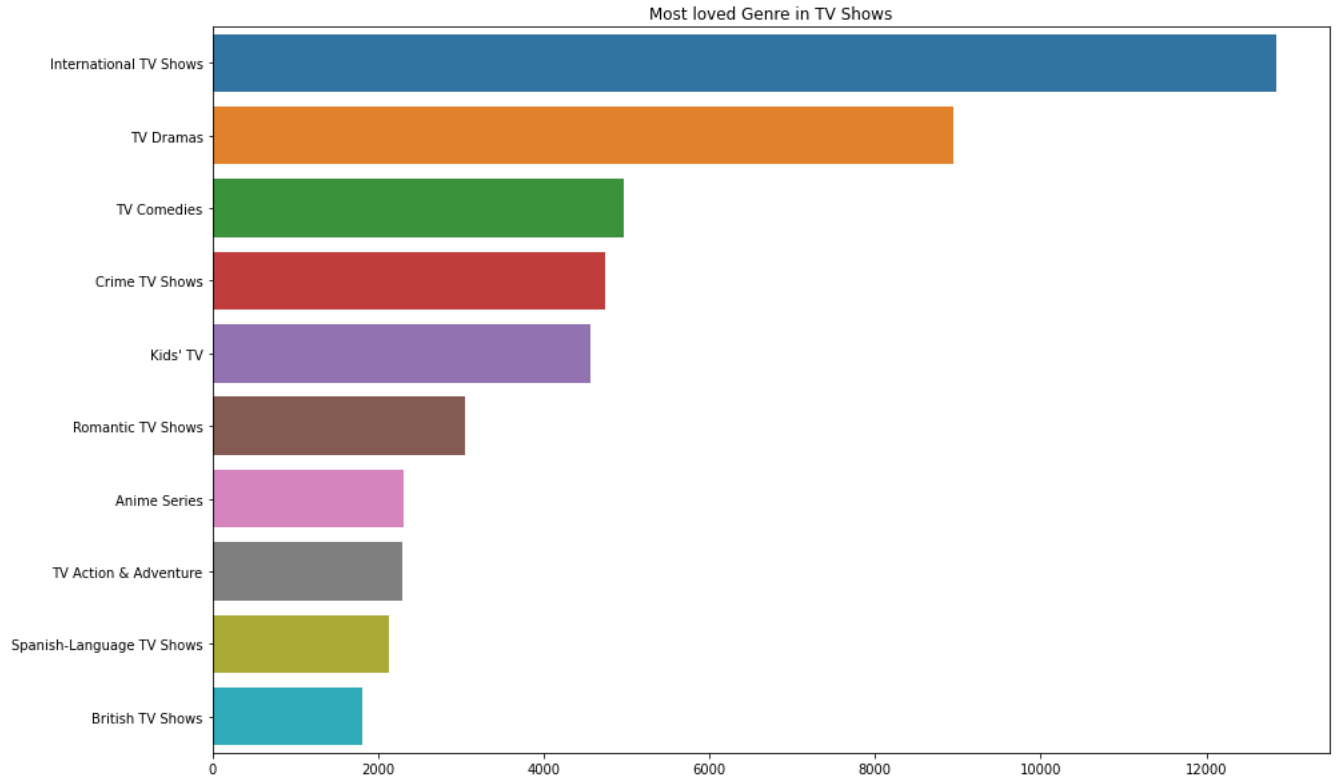
```
plt.figure(figsize=(15,10))
plt.title("Top Actors of TV Shows")
sns.barplot(x = Actors_max_web.index, y = Actors_max_web.values)
plt.show()
```



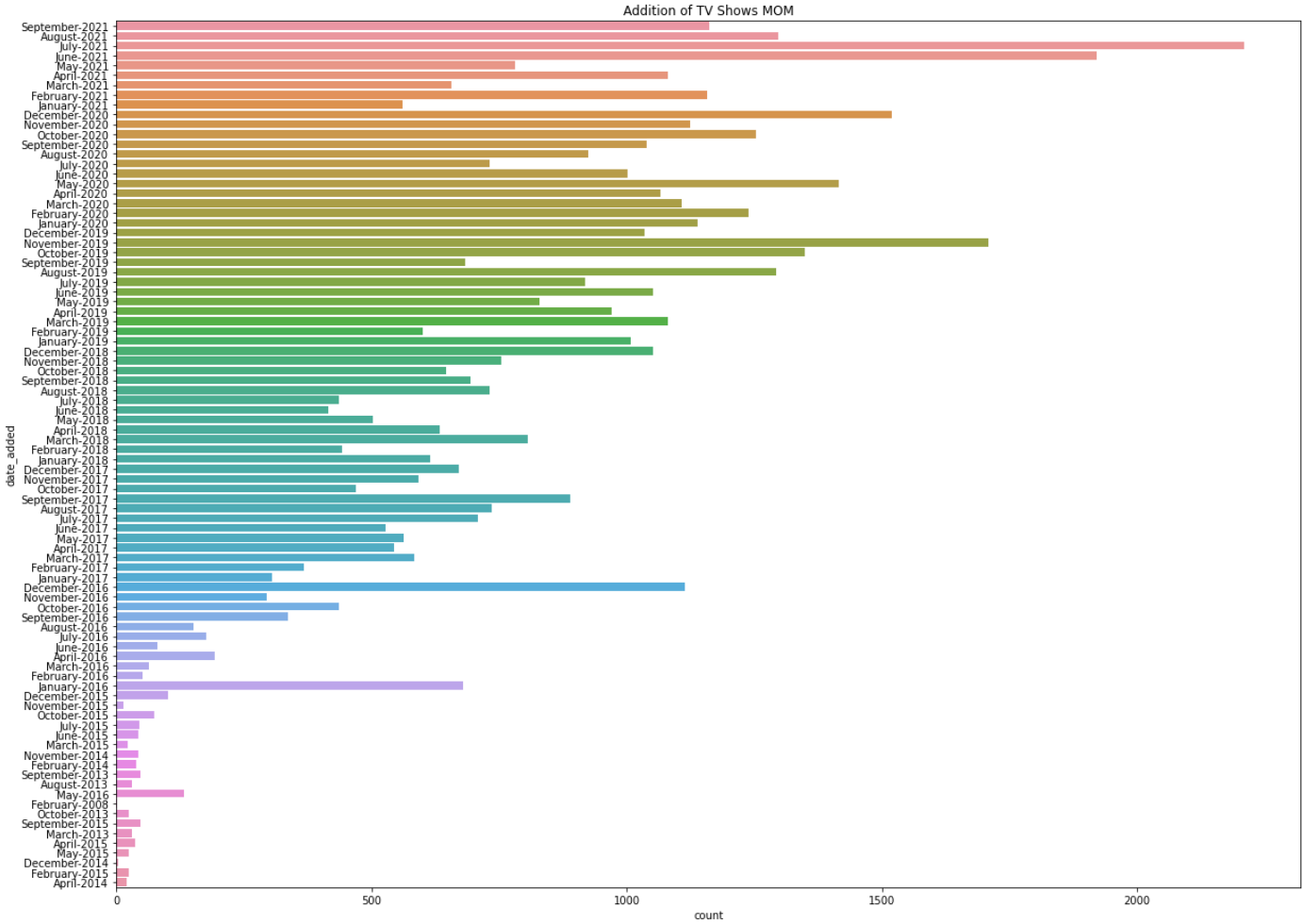

```
pop_genre_web = df_web.Genre.value_counts().head(10)
pop_genre_web
```

```
International TV Shows       12845
TV Dramas                     8942
TV Comedies                   4963
Crime TV Shows                4733
Kids' TV                      4568
Romantic TV Shows             3049
Anime Series                  2313
TV Action & Adventure         2288
Spanish-Language TV Shows     2126
British TV Shows              1808
Name: Genre, dtype: int64
```

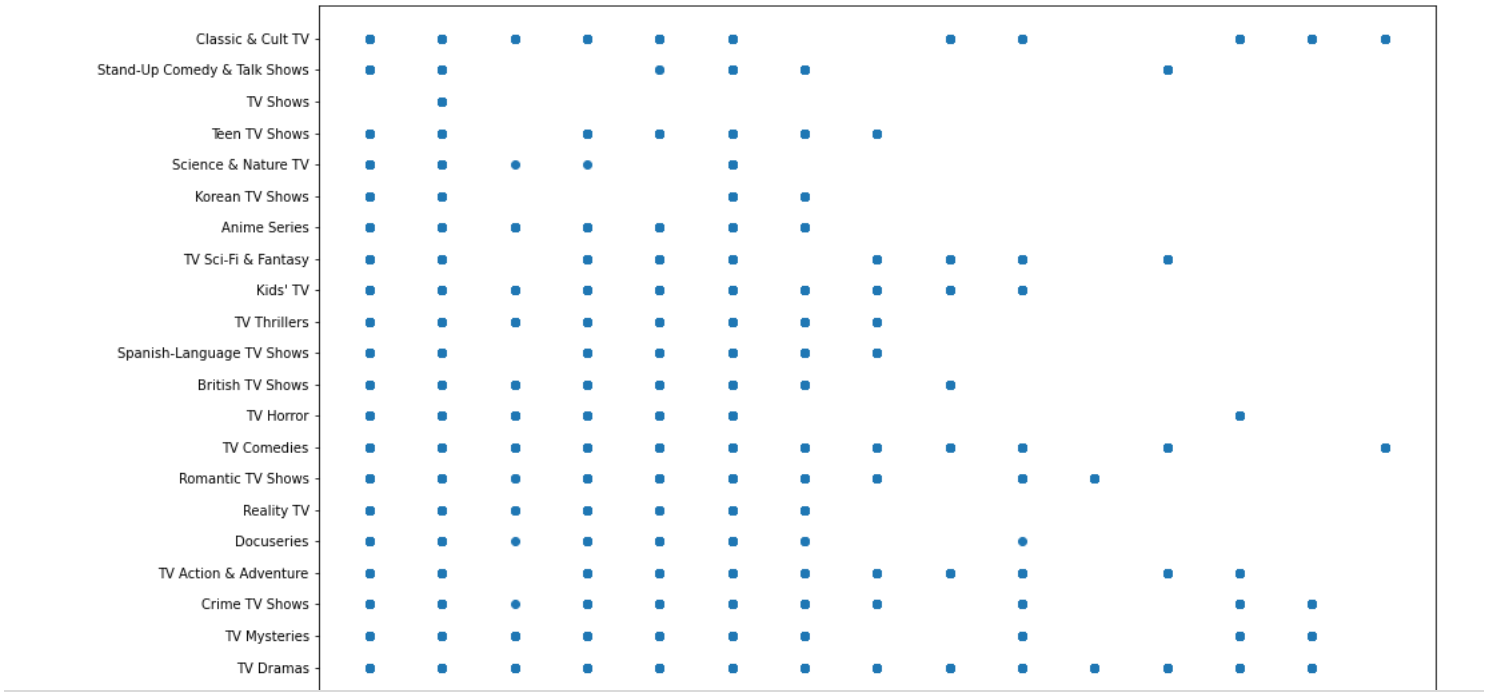
```
plt.figure(figsize=(15,10))
plt.title("Most loved Genre in TV Shows")
sns.barplot(y = pop_genre_web.index, x = pop_genre_web.values)
plt.show()
```



```
plt.figure(figsize=(20,15))
plt.title("Addition of TV Shows MOM")
sns.countplot(data = df_web, y = "date_added")
plt.show()
```




```
plt.figure(figsize=(15,10))
plt.scatter(y = df_web.Genre, x = df_web.duration)
plt.show()
```


Classic & Cult TV
Stand-Up Comedy & Talk Shows
TV Shows
Teen TV Shows
Science & Nature TV
Korean TV Shows
Anime Series
TV Sci-Fi & Fantasy
Kids' TV
TV Thrillers
Spanish-Language TV Shows
British TV Shows
TV Horror
TV Comedies
Romantic TV Shows
Reality TV
Docuseries
TV Action & Adventure
Crime TV Shows
TV Mysteries
TV Dramas
0s completed at 9:23 AM