

In [1]:

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

In [2]:

```
df=pd.read_csv("netflix.csv")
```

In [3]:

```
df.head()
```

Out[3]:

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

In [4]:

```
df.tail()
```

Out[4]:

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

In [5]:

```
df.shape
```

Out[5]:

(8807, 12)

In [6]:

```
df.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
```

In [7]:

```
df.describe(include='all')
```

Out[7]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
count	8807	8807	8807	6173	7982	7976	8797	8807.000000	8803
unique	8807	2	8807	4528	7692	748	1767	NaN	8804
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	NaN	8.819312
freq	1	6131	1	19	19	2818	109	NaN	1
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014.180198	8.819312
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	8.819312	1
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1925.000000	1
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2013.000000	1
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2017.000000	1
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2019.000000	1
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2021.000000	1



In [8]:

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

Out[8]:

```
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
```

Stacking

Stacking Cast Column

In [9]:

```
cast_s=df['cast'].apply(lambda x: str(x).split(",")).tolist()
```

In [10]:

```
cast_df=pd.DataFrame(cast_s,index=df['title'])
```

In [11]:

```
cast_df=cast_df.stack()
```

In [12]:

```
cast_df_new=pd.DataFrame(cast_df)
```

In [13]:

```
cast_df_new
```

Out[13]:

0

title		
Dick Johnson Is Dead	0	nan
Blood & Water	0	Ama Qamata
	1	Khosi Ngema
	2	Gail Mabalane
	3	Thabang Molaba
...
Zubaan	3	Manish Chaudhary
	4	Meghna Malik
	5	Malkeet Rauni
	6	Anita Shabdish
	7	Chittaranjan Tripathy

64951 rows × 1 columns

In [14]:

```
cast_df_new.reset_index(inplace=True)
```

In [15]:

```
cast_df_new
```

Out[15]:

	title	level_1	0
0	Dick Johnson Is Dead	0	nan
1	Blood & Water	0	Ama Qamata
2	Blood & Water	1	Khosi Ngema
3	Blood & Water	2	Gail Mabalane
4	Blood & Water	3	Thabang Molaba
...
64946	Zubaan	3	Manish Chaudhary
64947	Zubaan	4	Meghna Malik
64948	Zubaan	5	Malkeet Rauni
64949	Zubaan	6	Anita Shabdish
64950	Zubaan	7	Chittaranjan Tripathy

64951 rows × 3 columns

In [16]:

```
cast_df_new=cast_df_new[['title',0]]
```

In [17]:

```
cast_df_new.columns=['title','cast']
```

In [18]:

```
cast_df_new
```

Out[18]:

	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
...
64946	Zubaan	Manish Chaudhary
64947	Zubaan	Meghna Malik
64948	Zubaan	Malkeet Rauni
64949	Zubaan	Anita Shabdish
64950	Zubaan	Chittaranjan Tripathy

64951 rows × 2 columns

Merging

In [19]:

```
df_cast=df.reset_index().merge(cast_df_new,on='title',how='inner')
```

In [20]:

df_cast

Out[20]:

	index	show_id	type	title	director	cast_x	country	date_added	release
0	0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	
1	1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	
2	1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	
3	1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	
4	1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	
...	
64946	8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	
64947	8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	
64948	8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	

	index	show_id	type	title	director	cast_x	country	date_added	release
64949	8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	
64950	8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	

64951 rows × 14 columns

In [21]:

```
df_cast.drop(['index', 'cast_x'], axis=1, inplace=True)
```

In [22]:

```
df_cast.rename(columns={"cast_y": "cast"}, inplace=True)
```

In [23]:

```
df_cast
```

Out[23]:

	show_id	type	title	director	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	September 25, 2021	2020	PG-13	90m
1	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Season 2
2	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Season 2
3	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Season 2
4	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Season 2
...
64946	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	111m
64947	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	111m
64948	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	111m
64949	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	111m

	show_id	type	title	director	country	date_added	release_year	rating	duration
64950	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	111

64951 rows × 12 columns

Stacking Listed In Column

In [24]:

```
constraint=df['listed_in'].apply(lambda x: str(x).split(",")).tolist()
```

In [25]:

```
genre_df=pd.DataFrame(constraint,index=df['title'])
```

In [26]:

```
genre_df=genre_df.stack()
```

In [27]:

```
genre_df_new=pd.DataFrame(genre_df)
```

In [28]:

```
genre_df_new.reset_index(inplace=True)
```

In [29]:

```
genre_df_new=genre_df_new[['title',0]]
```

In [30]:

```
genre_df_new.columns=['title','genre']
```

In [31]:

```
genre_df_new
```

Out[31]:

	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
...
19318	Zoom	Children & Family Movies
19319	Zoom	Comedies
19320	Zubaan	Dramas
19321	Zubaan	International Movies
19322	Zubaan	Music & Musicals

19323 rows × 2 columns

In [32]:

```
df_genre=df_cast.reset_index().merge(genre_df_new,on='title',how='inner')
```

In [33]:

df_genre

Out[33]:

	index	show_id	type	title	director	country	date_added	release_year	rating
0	0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	September 25, 2021	2020	F
1	1	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	
2	1	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	
3	1	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	
4	2	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	
...
149507	64949	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV
149508	64949	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV
149509	64950	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV
149510	64950	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV

	index	show_id	type	title	director	country	date_added	release_year	rat	
	149511	64950	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV ▼

◀

▶

In [34]:

```
df_genre.drop(['index', 'listed_in'],axis=1,inplace=True)
```

In [35]:

```
df_genre
```

Out[35]:

	show_id	type	title	director	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	September 25, 2021	2020	PG-13	9
1	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Se
2	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Se
3	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Se
4	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Se
...
149507	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	1h
149508	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	1h
149509	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	1h
149510	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	1h

show_id	type	title	director	country	date_added	release_year	rating	du	
149511	s8807	Movie	Zubaan	Mozez Singh	India	March 2, 2019	2015	TV-14	1'

149512 rows × 12 columns

Stacking Director Column

In [36]:

```
constraint=df['director'].apply(lambda x: str(x).split(",")).tolist()
```

In [37]:

```
director_df=pd.DataFrame(constraint,index=df['title'])
```

In [38]:

```
director_df=director_df.stack()
```

In [39]:

```
director_df_new=pd.DataFrame(director_df)
```

In [40]:

```
director_df_new.reset_index(inplace=True)
```

In [41]:

```
director_df_new=director_df_new[['title',0]]
```

In [42]:

```
director_df_new.columns=['title','director']
```


In [43]:

```
director_df_new
```

Out[43]:

	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
...
9607	Zodiac	David Fincher
9608	Zombie Dumb	nan
9609	Zombieland	Ruben Fleischer
9610	Zoom	Peter Hewitt
9611	Zubaan	Mozez Singh

9612 rows × 2 columns

In [44]:

```
df_director=df_genre.reset_index().merge(director_df_new,on='title',how='inner')
```

In [45]:

```
df_director.drop(['director_x'],axis=1,inplace=True)
```

In [46]:

```
df_director.rename(columns={"director_y":"director"},inplace=True)
```

In [47]:

```
df_director.drop(['index'],axis=1,inplace=True)
```

In [48]:

```
df_director
```

Out[48]:

	show_id	type	title	country	date_added	release_year	rating	duration	de
0	s1	Movie	Dick Johnson Is Dead	United States	September 25, 2021	2020	PG-13	90 min	fat t
1	s2	TV Show	Blood & Water	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	I C
2	s2	TV Show	Blood & Water	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	I C
3	s2	TV Show	Blood & Water	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	I C
4	s2	TV Show	Blood & Water	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	I C
...	
161211	s8807	Movie	Zubaan	India	March 2, 2019	2015	TV-14	111 min	b his
161212	s8807	Movie	Zubaan	India	March 2, 2019	2015	TV-14	111 min	b his
161213	s8807	Movie	Zubaan	India	March 2, 2019	2015	TV-14	111 min	b his
161214	s8807	Movie	Zubaan	India	March 2, 2019	2015	TV-14	111 min	b his

show_id	type	title	country	date_added	release_year	rating	duration	de	
161215	s8807	Movie	Zubaan	India	March 2, 2019	2015	TV-14	111 min	b his

161216 rows × 12 columns

Stacking Country Column

In [49]:

```
constraint=df['country'].apply(lambda x: str(x).split(",")).tolist()
```

In [50]:

```
country_df=pd.DataFrame(constraint,index=df['title'])
```

In [51]:

```
country_df=country_df.stack()
```

In [52]:

```
country_df_new=pd.DataFrame(country_df)
```

In [53]:

```
country_df_new.reset_index(inplace=True)
```

In [54]:

```
country_df_new=country_df_new[['title',0]]
```

In [55]:

```
country_df_new.columns=['title','country']
```

In [56]:

```
country_df_new
```

Out[56]:

	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
...
10845	Zodiac	United States
10846	Zombie Dumb	nan
10847	Zombieland	United States
10848	Zoom	United States
10849	Zubaan	India

10850 rows × 2 columns

In [57]:

```
df_final=df_director.reset_index().merge(country_df_new,on='title',how='inner')
```

In [58]:

```
df_final.drop(['index','country_x'],axis=1,inplace=True)
```

In [59]:

```
df_final.rename(columns={'country_y':'country'},inplace=True)
```

In [60]:

```
df_final
```

Out[60]:

	show_id	type	title	date_added	release_year	rating	duration	description
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
2	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
3	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
4	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
...
202060	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	A scrappy but poor boy worms his way into a ty...
202061	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	A scrappy but poor boy worms his way into a ty...
202062	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	A scrappy but poor boy worms his way into a ty...
202063	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	A scrappy but poor boy worms his way into a ty...

show_id	type	title	date_added	release_year	rating	duration	description	
202064	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	A scrappy but poor boy worms his way into a ty...

202065 rows × 12 columns

Unique Attributes and Value Counts

Director

In [61]:

```
k=df['director'].value_counts().keys().tolist()
v=df['director'].value_counts().tolist()
```

In [62]:

```
print("Director and their Movies")
for i in range(len(k)):
    print(f"{k[i]:{30}}{v[i]}")
```

```
Director and their Movies
Rajiv Chilaka                19
Raúl Campos, Jan Suter       18
Marcus Raboy                 16
Suhas Kadav                  16
Jay Karas                    14
Cathy Garcia-Molina          13
Martin Scorsese               12
Youssef Chahine               12
Jay Chapman                  12
Steven Spielberg             11
Don Michael Paul              10
David Dhawan                  9
Yılmaz Erdoğan              8
Lance Bangs                   8
Kunle Afolayan                8
Quentin Tarantino             8
Ryan Polito                   8
Troy Miller                   8
Shannon Hartman               8
```

Genre

In [63]:

```
k=df_final['genre'].value_counts().keys().tolist()
v=df_final['genre'].value_counts().tolist()
```

In [64]:

```
for i in range(len(k)):
    print(f"{k[i].strip():{30}}{v[i]}")
```

International Movies	27141
Dramas	19657
Comedies	13894
Action & Adventure	12216
Dramas	10149
Independent Movies	9564
Children & Family Movies	9294
TV Dramas	7956
International TV Shows	7065
Comedies	6935
Romantic Movies	6392
Thrillers	6283
International TV Shows	5780
Crime TV Shows	4020
Sci-Fi & Fantasy	3875
Kids' TV	3809
TV Comedies	3768
Horror Movies	3259
Music & Musicals	3012
Romantic TV Shows	2729
Anime Series	2313
Documentaries	2285
Spanish-Language TV Shows	2088
British TV Shows	1808
TV Action & Adventure	1724
Sports Movies	1528
Horror Movies	1312
TV Mysteries	1281
TV Comedies	1195
Korean TV Shows	1122
International Movies	1102
TV Sci-Fi & Fantasy	1038
TV Dramas	986
Classic Movies	946
Cult Movies	908
LGBTQ Movies	833
Thrillers	824
TV Thrillers	768
Anime Features	765
Kids' TV	759
TV Horror	750
Teen TV Shows	742
Faith & Spirituality	719
Crime TV Shows	713
TV Action & Adventure	564
Docuseries	546
Stand-Up Comedy	516
Classic Movies	497
Children & Family Movies	477
Reality TV	413
Movies	412
TV Shows	337
Reality TV	322
Romantic TV Shows	320
Docuseries	299

Anime Features	280
Independent Movies	270
Classic & Cult TV	236
TV Horror	191
Stand-Up Comedy & Talk Shows	188
Cult Movies	169
Sci-Fi & Fantasy	162
Science & Nature TV	157
Documentaries	124
Stand-Up Comedy & Talk Shows	80
Music & Musicals	65
Spanish-Language TV Shows	38
Classic & Cult TV	36
Stand-Up Comedy	24
Romantic Movies	20
TV Sci-Fi & Fantasy	7
LGBTQ Movies	5
Sports Movies	3

In [65]:

```
k=df['country'].value_counts().keys().tolist()
v=df['country'].value_counts().tolist()
for i in range(len(k)):
    print(f"{k[i]:{30}}{v[i]}")
```

United States	2818
India	972
United Kingdom	419
Japan	245
South Korea	199
Canada	181
Spain	145
France	124
Mexico	110
Egypt	106
Turkey	105
Nigeria	95
Australia	87
Taiwan	81
Indonesia	79
Brazil	77
Philippines	75
United Kingdom, United States	75
United States, Canada	73
~	--

In [66]:

```
k=df_final['cast'].value_counts().keys().tolist()
v=df_final['cast'].value_counts().tolist()
for i in range(len(k)):
    print(f"{k[i].strip():{30}}{v[i]}")
```

nan	2149
Alfred Molina	160
Salma Hayek	130
Frank Langella	128
John Rhys-Davies	125
John Krasinski	121
Liam Neeson	120
Anupam Kher	116
David Attenborough	103
Quvenzhané Wallis	100
James Faulkner	93
Radhika Apte	92
Jim Broadbent	92
Ben Whishaw	86
Luci Christian	86
Pareesh Rawal	82
Om Puri	81
Boman Irani	80
Takahiro Sakurai	79
Chloe Pirrie	77

Getting Insights

In [67]:

```
df_final.shape
```

Out[67]:

```
(202065, 12)
```

In [68]:

```
df.head()
```

Out[68]:

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



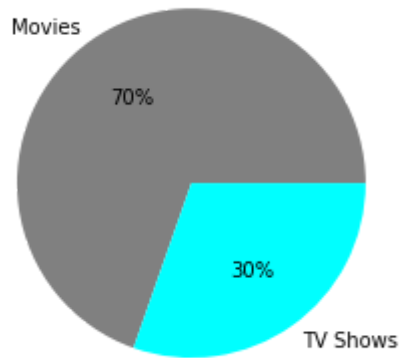
Movies and TV Shows

In [69]:

```
type_ = df['type'].value_counts().keys().tolist()
count_ = df['type'].value_counts().tolist()
```

In [70]:

```
plt.pie(count_,
        labels=['Movies', 'TV Shows'],
        colors=['gray', 'cyan'], autopct='%1.0f%%')
plt.show()
```



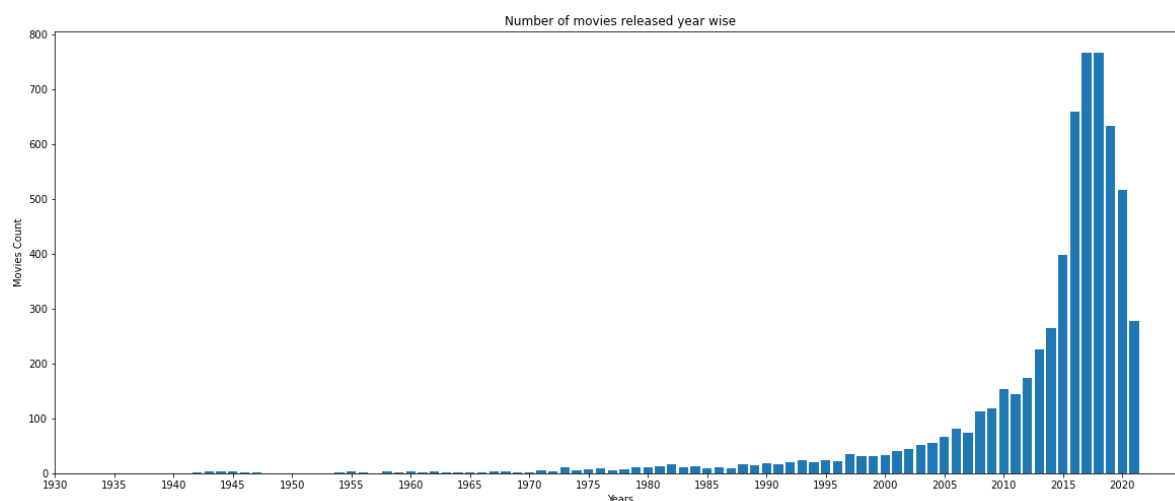
Number of movies released per year changed over last 20-30 years

In [71]:

```
df_n_movies=df[df['type']=='Movie']
year_movies = df_n_movies['release_year'].value_counts().keys().tolist()
count_movies = df_n_movies['release_year'].value_counts().tolist()
```

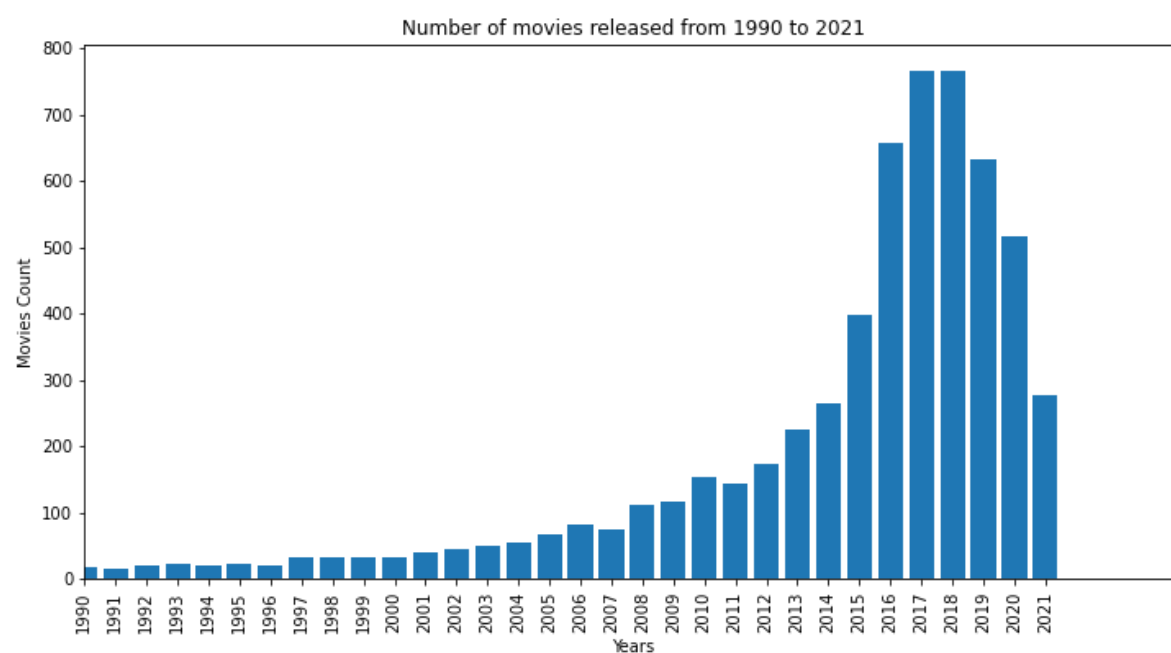
In [72]:

```
plt.figure(figsize=(20,8))
plt.title("Number of movies released year wise")
plt.xlabel("Years")
plt.ylabel("Movies Count")
plt.bar(x=year_movies,height=count_movies)
plt.xticks(np.arange(1930,2025,5))
plt.show()
```



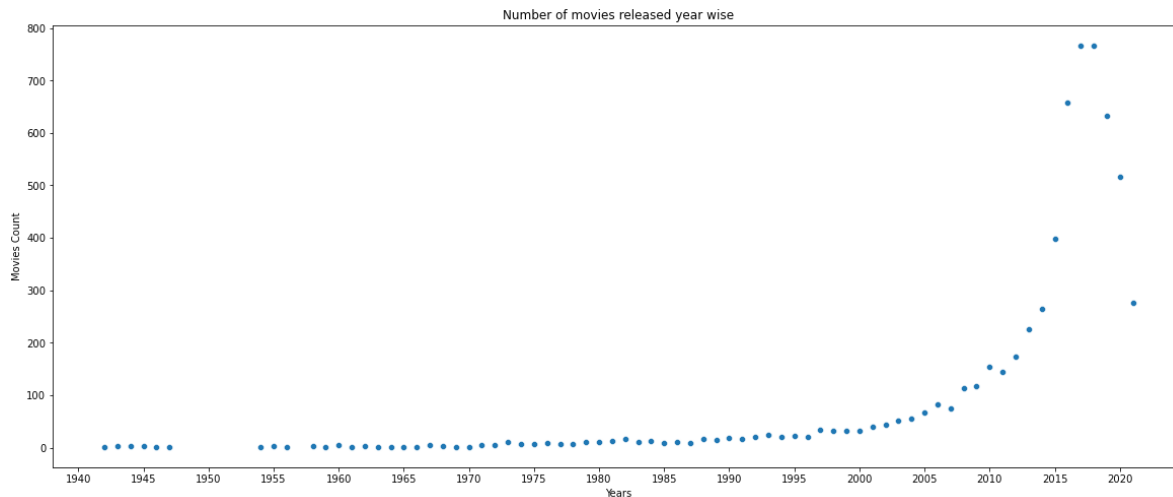
In [73]:

```
plt.figure(figsize=(12,6))
plt.title("Number of movies released from 1990 to 2021")
plt.xlabel("Years")
plt.ylabel("Movies Count")
plt.bar(x=year_movies,height=count_movies)
plt.xlim(left=1990)
plt.xticks(np.arange(1990,2022),rotation=90)
plt.show()
```



In [74]:

```
plt.figure(figsize=(20,8))
plt.title("Number of movies released year wise")
plt.xlabel("Years")
plt.ylabel("Movies Count")
plt.xticks(np.arange(1930,2025,5))
sns.scatterplot(x= year_movies, y = count_movies)
plt.show()
```



Number of movies released in Netflix over years

In [75]:

```
date_added=df_n_movies['date_added'].apply(lambda x: str(x).split(',')[1])
```

In [76]:

```
date_added=pd.DataFrame(date_added)
```

In [77]:

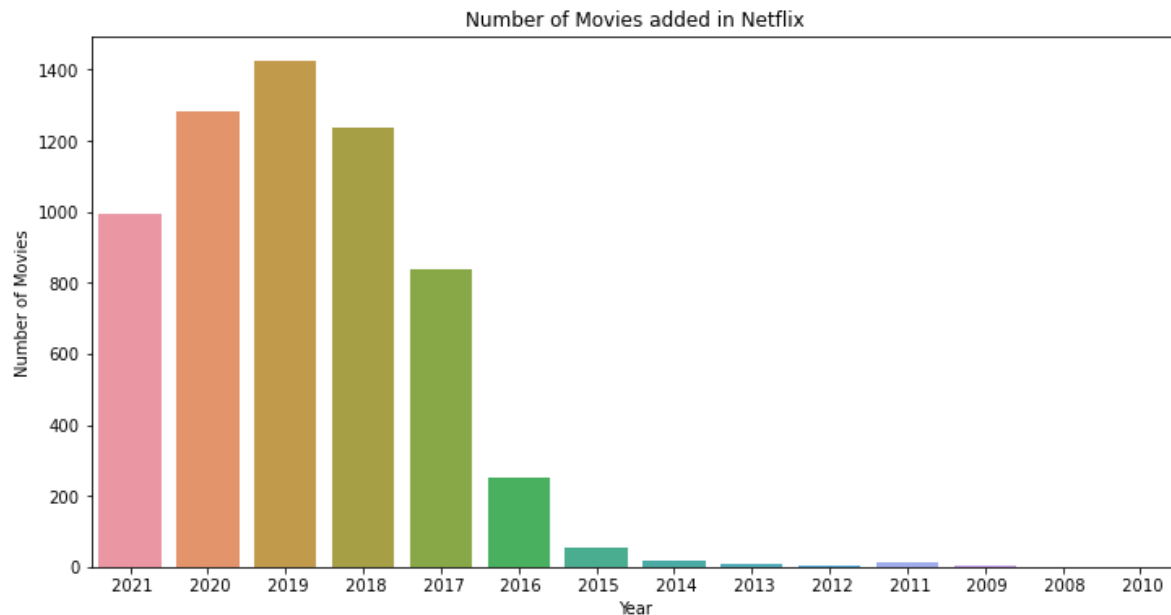
```
date_added.dropna(inplace=True)
date_added.columns=['date_added_netflix']
```

In [78]:

```
date_added=date_added[date_added['date_added_netflix']!="nan"]
```

In [79]:

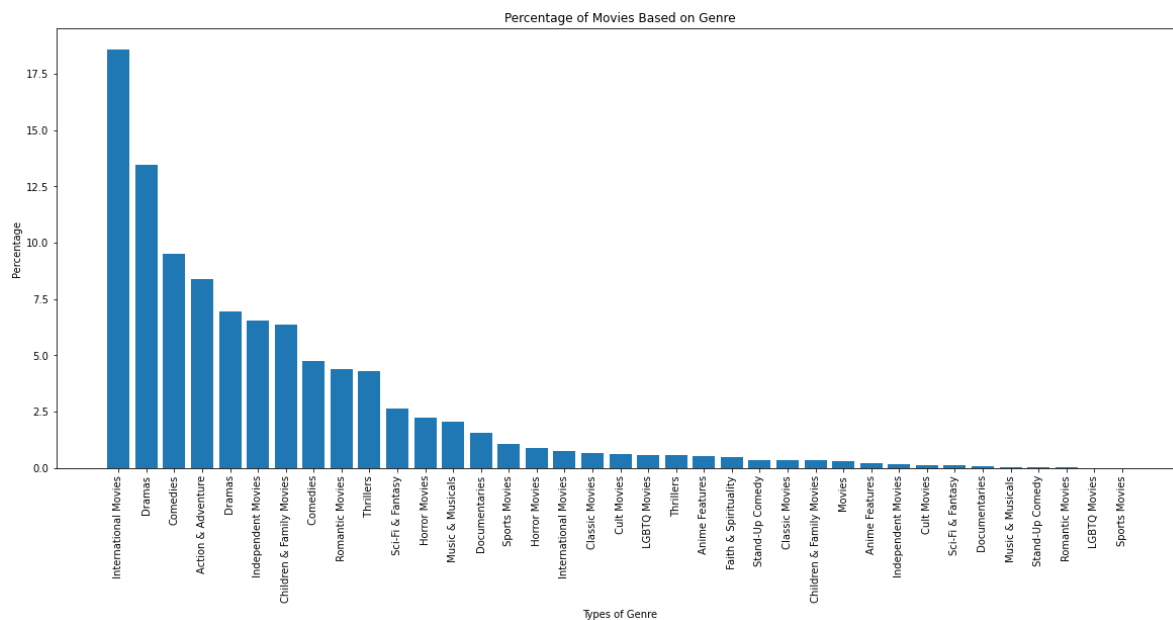
```
plt.figure(figsize=(12,6))
sns.countplot(x = 'date_added_netflix', data = date_added)
plt.title("Number of Movies added in Netflix")
plt.xlabel("Year")
plt.ylabel("Number of Movies")
plt.show()
```



Percentage of movies based on Genre

In [80]:

```
df_g_movies=df_final[df_final['type']=="Movie"]
genre=df_g_movies['genre'].value_counts().keys().tolist()
count=df_g_movies['genre'].value_counts().tolist()
count=np.array(count)
s=np.sum(count)
count=(count/s)*100
count=np.round(count,3)
count=list(count)
plt.figure(figsize=(20,8))
plt.title("Percentage of Movies Based on Genre")
plt.xlabel("Types of Genre")
plt.ylabel("Percentage")
plt.bar(x=genre, height=count)
plt.xticks(rotation=90)
plt.show()
```



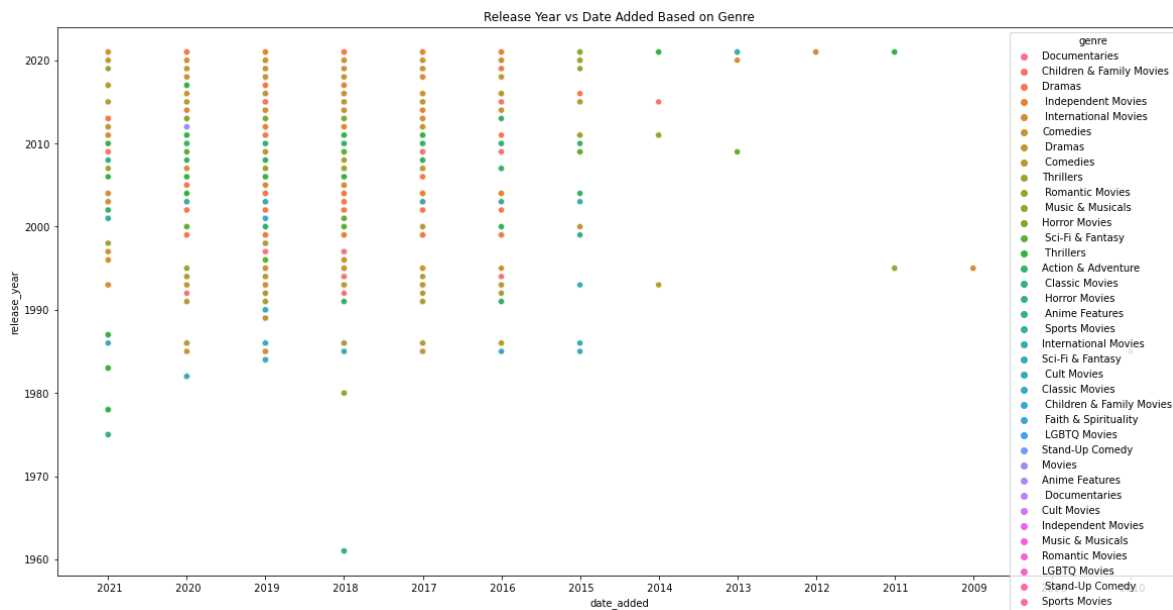
Release Year and Date Added Based on Genre

In [81]:

```
date_added=df_n_movies['date_added'].apply(lambda x: str(x).split(',')[ -1])
df_g_movies['date_added']=date_added
```

In [82]:

```
plt.figure(figsize=(20,10))
plt.title("Release Year vs Date Added Based on Genre")
sns.scatterplot(x='date_added', y='release_year', data=df_g_movies, hue='genre')
plt.show()
```



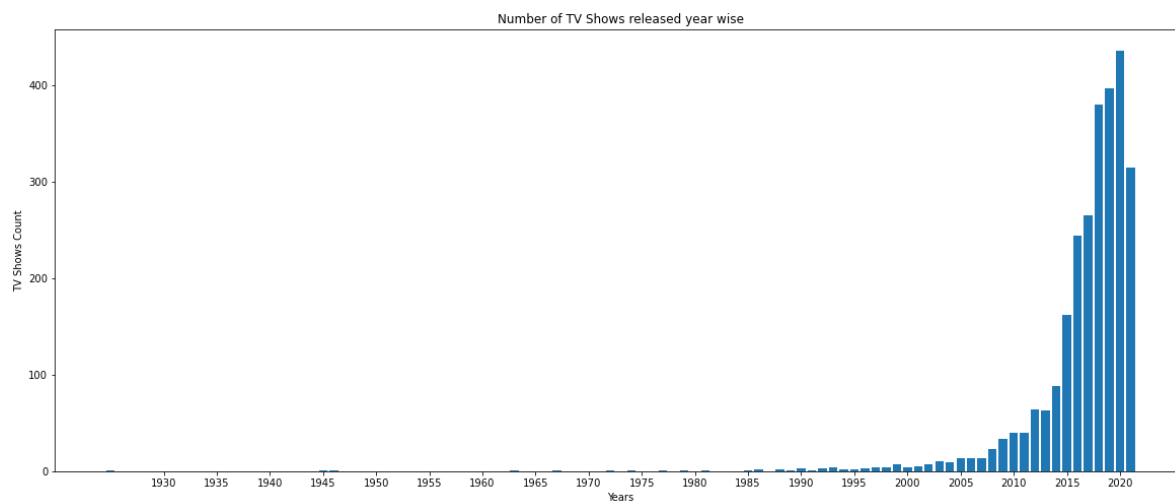
Number of TV Shows released per year changed over last 20-30 years

In [84]:

```
df_n_shows=df[df['type']=='TV Show']
year_shows = df_n_shows['release_year'].value_counts().keys().tolist()
count_shows = df_n_shows['release_year'].value_counts().tolist()
```

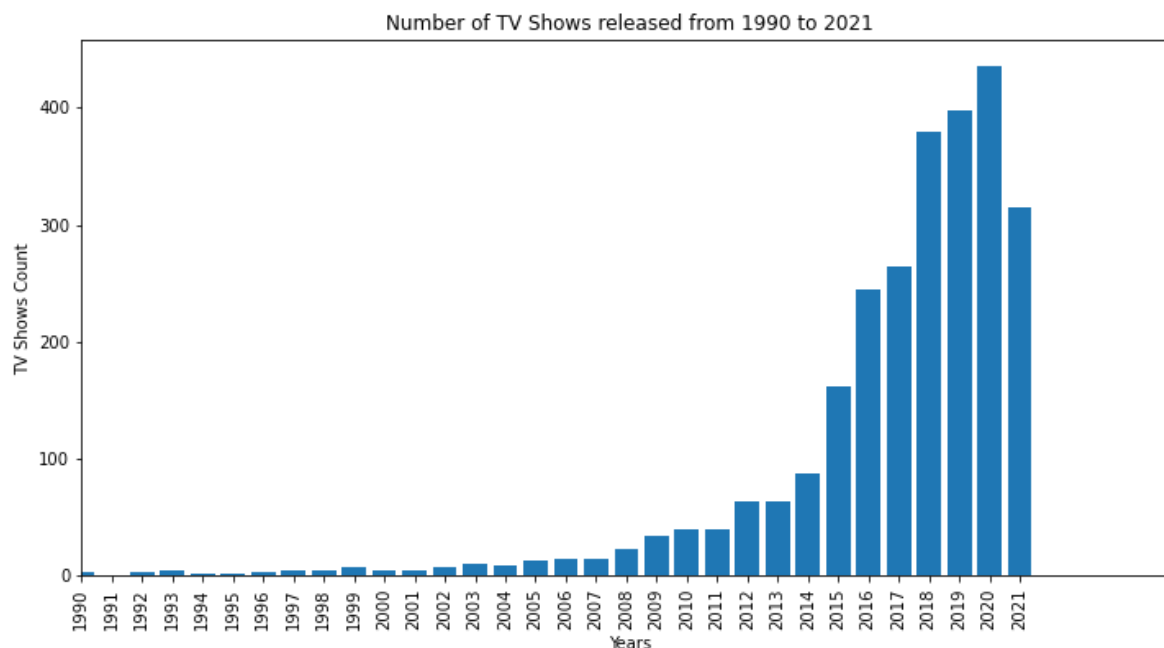

In [85]:

```
plt.figure(figsize=(20,8))
plt.title("Number of TV Shows released year wise")
plt.xlabel("Years")
plt.ylabel("TV Shows Count")
plt.bar(x=year_shows,height=count_shows)
plt.xticks(np.arange(1930,2025,5))
plt.show()
```



In [86]:

```
plt.figure(figsize=(12,6))
plt.title("Number of TV Shows released from 1990 to 2021")
plt.xlabel("Years")
plt.ylabel("TV Shows Count")
plt.bar(x=year_shows,height=count_shows)
plt.xlim(left=1990)
plt.xticks(np.arange(1990,2022),rotation=90)
plt.show()
```



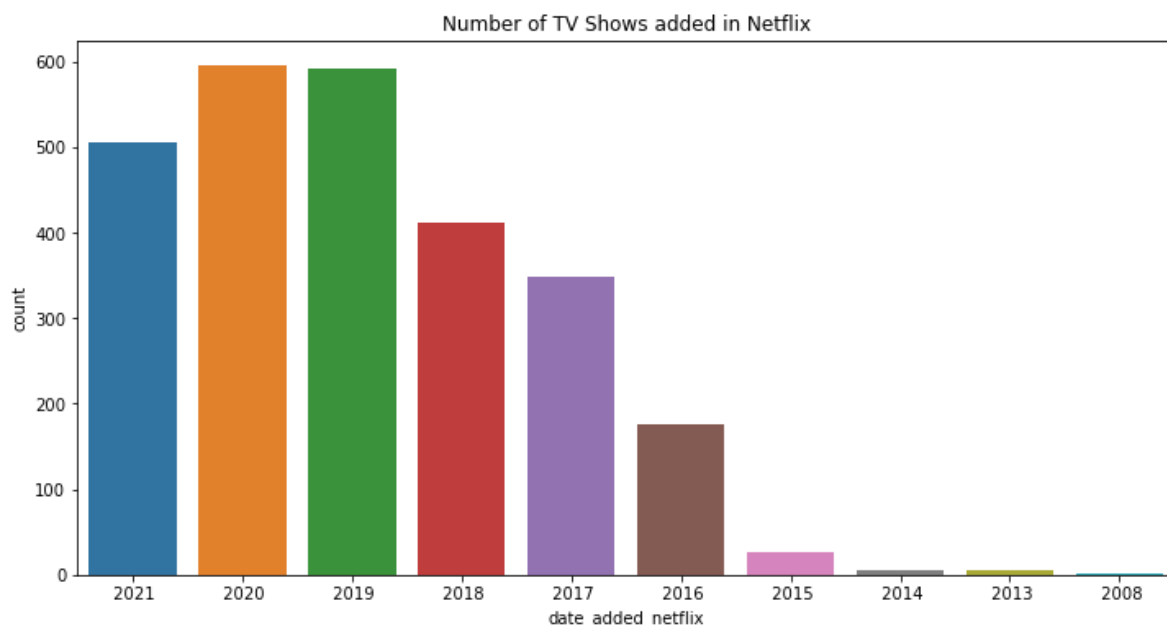
Number of TV Shows released in Netflix over years

In [87]:

```
date_added=df_n_shows['date_added'].apply(lambda x: str(x).split(',')[1])
```

In [88]:

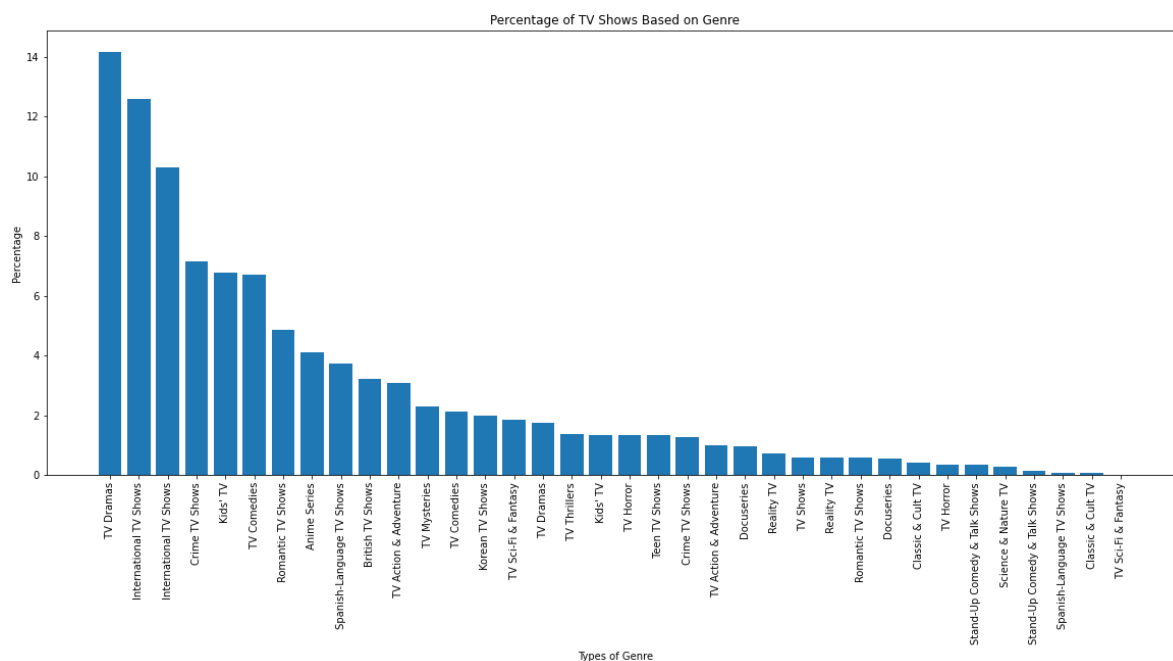
```
date_added=pd.DataFrame(date_added)
date_added.dropna(inplace=True)
date_added.columns=['date_added_netflix']
date_added=date_added[date_added['date_added_netflix']!="nan"]
plt.figure(figsize=(12,6))
plt.title("Number of TV Shows added in Netflix")
plt.xlabel("Year")
plt.ylabel("Number of TV Shows")
sns.countplot(x = 'date_added_netflix', data = date_added)
plt.show()
```



Percentage of TV Shows based on Genre

In [89]:

```
df_g_shows=df_final[df_final['type']=="TV Show"]
genre=df_g_shows['genre'].value_counts().keys().tolist()
count=df_g_shows['genre'].value_counts().tolist()
count=np.array(count)
s=np.sum(count)
count=(count/s)*100
count=np.round(count,3)
count=list(count)
plt.figure(figsize=(20,8))
plt.title("Percentage of TV Shows Based on Genre")
plt.xlabel("Types of Genre")
plt.ylabel("Percentage")
plt.bar(x=genre, height=count)
plt.xticks(rotation=90)
plt.show()
```



Insights on Duration

In [90]:

```
duration=df['duration'].apply(lambda x: str(x).split(" ")).tolist()
```

In [91]:

```
def movie_show(x):
    if len(x)>1:
        if x[1].lower()=="min":
            return "movie"
        elif x[1].lower()=="seasons":
            return "seasons"
```

In [92]:

```

duration_movie=[]
duration_show=[]
for i in duration:
    if movie_show(i)=="movie":
        duration_movie.append(int(i[0]))
    elif movie_show(i)=="seasons":
        duration_show.append(int(i[0]))

```

In [93]:

```

print("Maximum Duration Movie : {}".format(max(duration_movie)))
print("Minumum Duration Movie : {}".format(min(duration_movie)))
print("Maximum Seasons Show : {}".format(max(duration_show)))
print("Minumum Seasons Show : {}".format(min(duration_show)))

```

Maximum Duration Movie : 312 Minutes
 Minumum Duration Movie : 3 Minutes
 Maximum Seasons Show : 17 Seasons
 Minumum Seasons Show : 2 Seasons

In [94]:

```

print("Longthiest Movie in Netflix")
df[df['duration']==str(max(duration_movie))+ " min"]

```

Longthiest Movie in Netflix

Out[94]:

	show_id	type	title	director	cast	country	date_added	release_year	r
4253	s4254	Movie	Black Mirror: Bandersnatch	NaN	Fionn Whitehead, Will Poulter, Craig Parkinson...	United States	December 28, 2018	2018	



In [95]:

```
print("Shortest Movie in Netflix")
df[df['duration']==str(min(duration_movie))+ " min"]
```

Shortest Movie in Netflix

Out[95]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	durat
3777	s3778	Movie	Silent	Limbert Fabian, Brandon Oldenburg	NaN	United States	June 4, 2019	2014	TV-Y	3 i



In [96]:

```
print("Longthiest TV Show in Netflix")
df[df['duration']==str(max(duration_show))+ " Seasons"]
```

Longthiest TV Show in Netflix

Out[96]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	d
548	s549	TV Show	Grey's Anatomy	NaN	Ellen Pompeo, Sandra Oh, Katherine Heigl, Just...	United States	July 3, 2021	2020	TV-14	ε



In [97]:

```
print("Shortest TV Show in Netflix")
df[df['duration']==str(min(duration_show))+ " Seasons"].head()
```

Shortest TV Show in Netflix

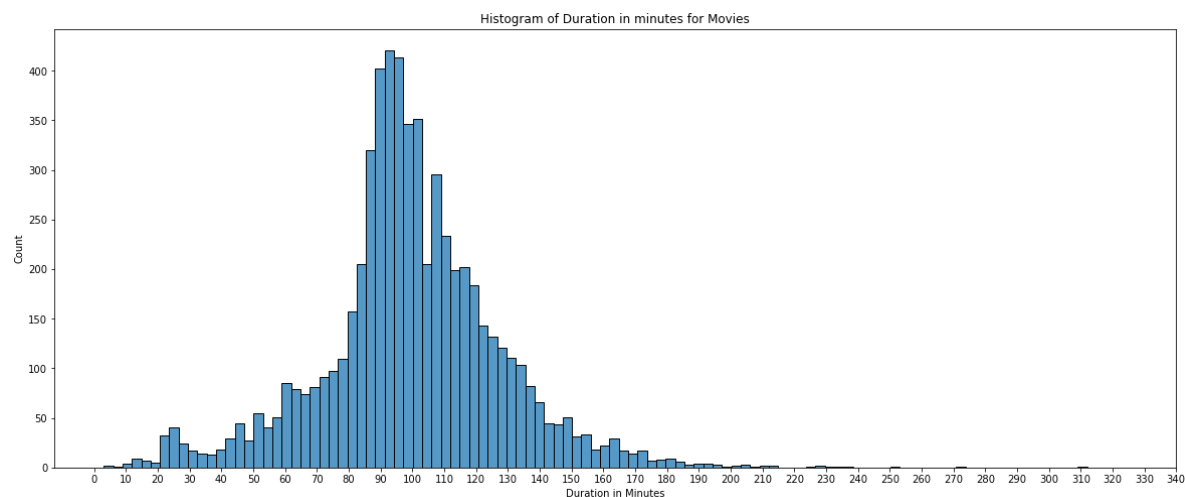
Out[97]:

show_id	type	title	director	cast	country	date_added	release_year	rating	
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA
17	s18	TV Show	Falsa identidad	NaN	Luis Ernesto Franco, Camila Sodi, Sergio Goyri...	Mexico	September 22, 2021	2020	TV-MA
25	s26	TV Show	Love on the Spectrum	NaN	Brooke Satchwell	Australia	September 21, 2021	2021	TV-14
49	s50	TV Show	Castle and Castle	NaN	Richard Mofe-Damijo, Dakore Akande, Bimbo Manu...	Nigeria	September 15, 2021	2021	TV-MA



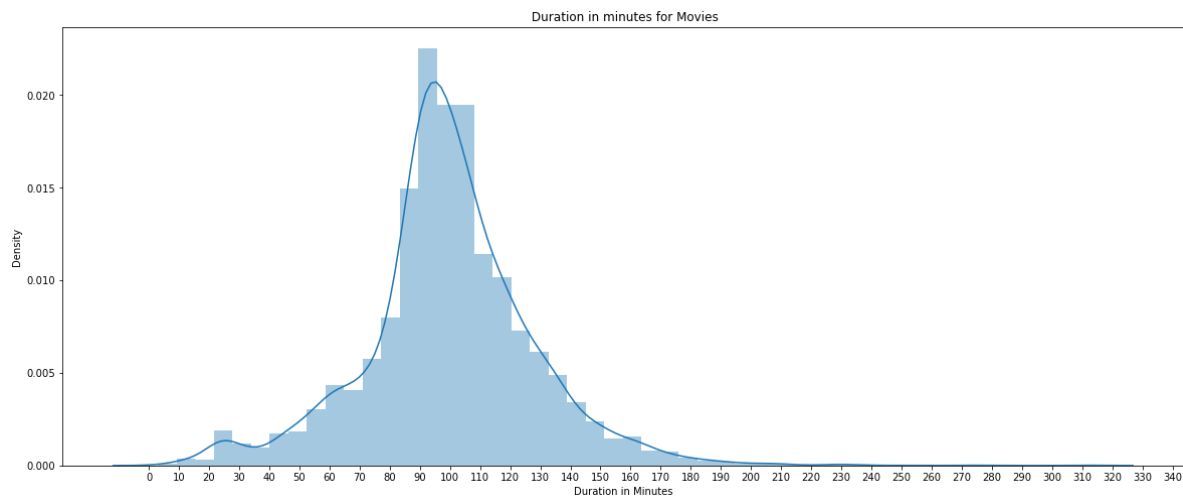
In [98]:

```
plt.figure(figsize=(20,8))
sns.histplot(duration_movie)
plt.title("Histogram of Duration in minutes for Movies")
plt.xlabel("Duration in Minutes")
plt.xticks(np.arange(0,350,10))
plt.show()
```



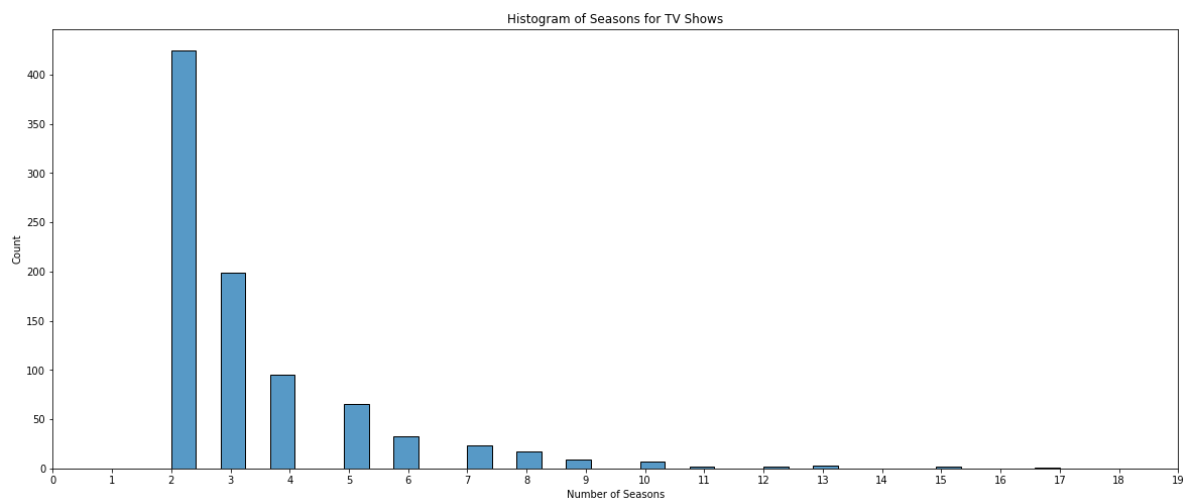
In [99]:

```
plt.figure(figsize=(20,8))
sns.distplot(duration_movie)
plt.title(" Duration in minutes for Movies")
plt.xlabel("Duration in Minutes")
plt.xticks(np.arange(0,350,10))
plt.show()
```



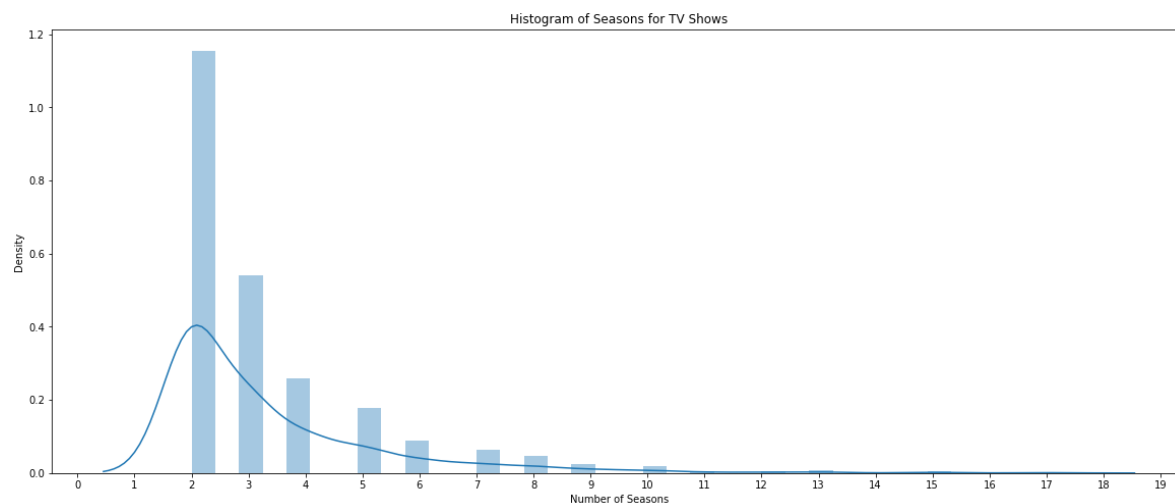
In [100]:

```
plt.figure(figsize=(20,8))
sns.histplot(duration_show)
plt.title("Histogram of Seasons for TV Shows")
plt.xlabel("Number of Seasons")
plt.xticks(np.arange(0,20))
plt.show()
```



In [101]:

```
plt.figure(figsize=(20,8))
sns.distplot(duration_show)
plt.title("Histogram of Seasons for TV Shows")
plt.xlabel("Number of Seasons")
plt.xticks(np.arange(0,20))
plt.show()
```



Correlation

In [302]:

```
df_corr=df_final.copy()
duration=df_corr['duration'].apply(lambda x: str(x).split(" ")[0])
duration=duration.replace('nan',0)
```

In [303]:

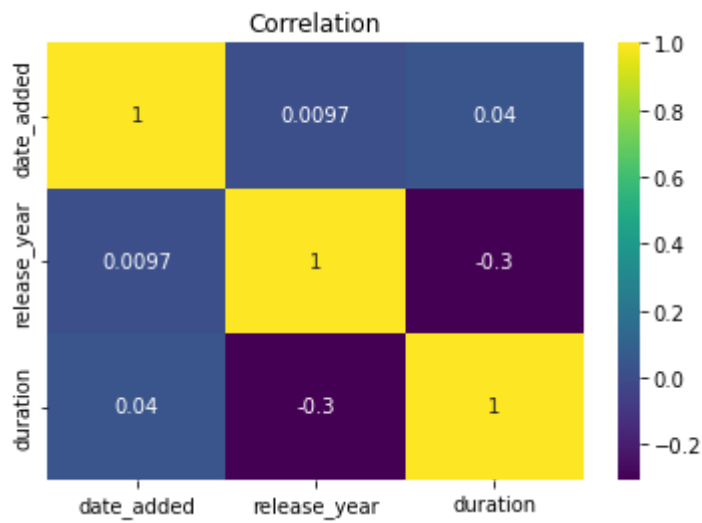
```
date_added=df_corr['date_added'].apply(lambda x: str(x).split(',')[0])
date_added=date_added.replace('nan',0)
```

In [304]:

```
df_corr['date_added']=np.array(date_added).astype("int32")
df_corr['duration']=np.array(duration).astype("int32")
```

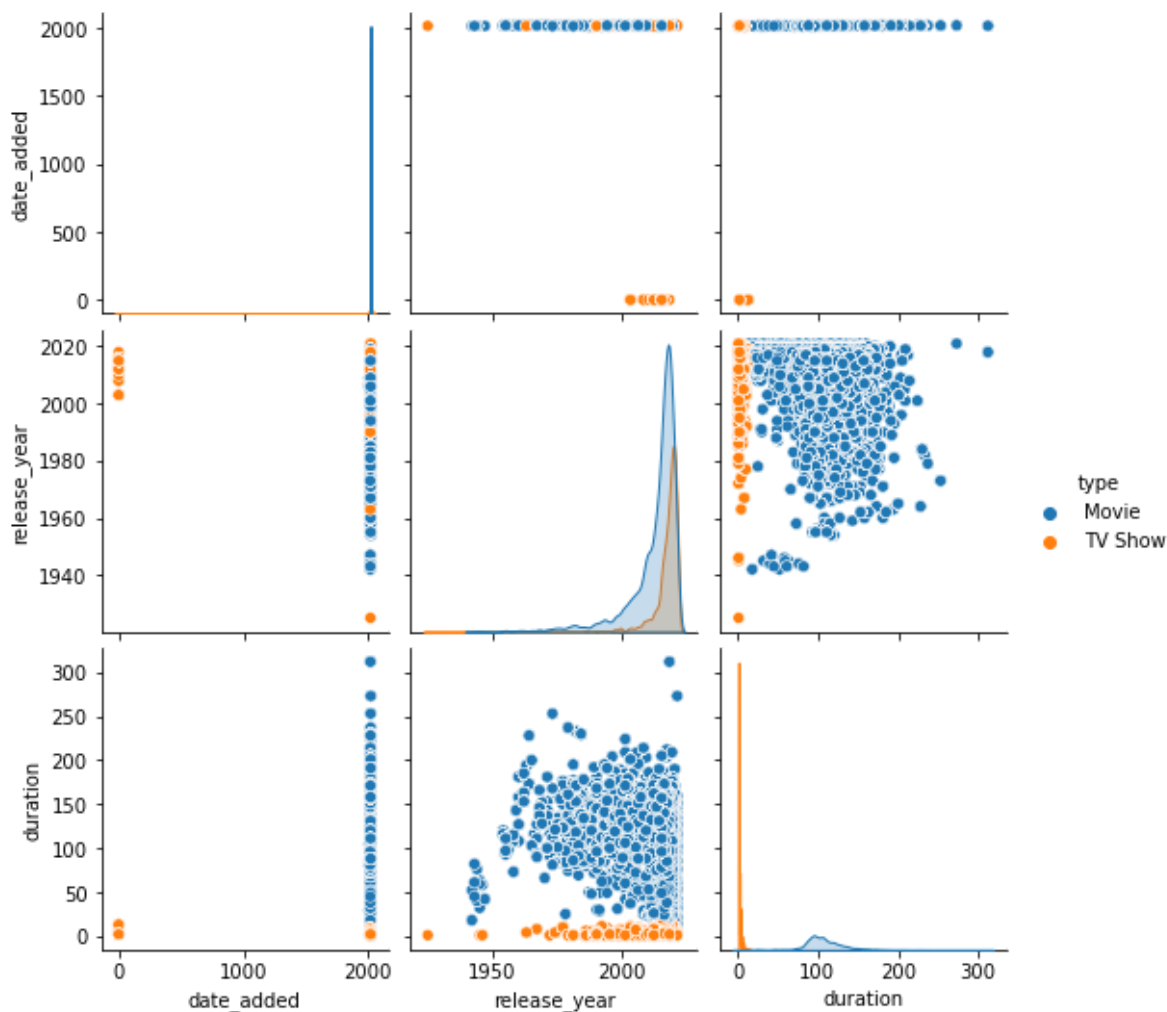
In [308]:

```
plt.title("Correlation")
sns.heatmap(df_corr.corr(), cmap= "viridis", annot=True)
plt.show()
```



In [312]:

```
sns.pairplot(df_corr,hue="type")
plt.show()
```



Insights on Cast

Getting Popular Actor/Actress based on Genre

In [102]:

```
unique_genre=list(df_final['genre'].unique())
```

In [103]:

```
group_cast=df_final.groupby('genre')
```

In [104]:

```

print("Popular Actor/Actress Based on Genre combined both Movies and TV Shows")
for i in unique_genre:
    t=group_cast.get_group(i)['cast'].value_counts(ascending=False).index.tolist()
    if t[0]!="nan":
        t=t[0]
    else:
        t=t[1]
    print(f"{i.strip():{30}}{t.strip()}")

```

Popular Actor/Actress Based on Genre combined both Movies and TV Shows

Documentaries	Samuel West
International TV Shows	Randy Kerber
TV Dramas	Joanna Kulig
TV Mysteries	Manoj Bajpayee
Crime TV Shows	Raúl Méndez
International TV Shows	Takahiro Sakurai
TV Action & Adventure	José María Yazpik
Docuseries	Bob Brisbane
Reality TV	Reina Triendl
Romantic TV Shows	Amanda Chou
TV Comedies	Richard Webber
TV Dramas	Fiona Fullerton
TV Horror	Jon Jon Briones
Children & Family Movies	Alfred Molina
Dramas	Gael García Bernal
Independent Movies	Radhika Apte
International Movies	Anupam Kher
British TV Shows	David Attenborough
Comedies	Seth Green
Dramas	Liam Neeson
Docuseries	David Attenborough
Comedies	Carlos Alazraqui
Crime TV Shows	Lee Ingleby
TV Comedies	Jeanna Harrison
Spanish-Language TV Shows	Fabián Ríos
Thrillers	Tom Hanks
Romantic Movies	Ben Whishaw
Music & Musicals	Amrish Puri
Horror Movies	Lindsay Burdge
Sci-Fi & Fantasy	Luci Christian
TV Thrillers	Janel Tsai
Kids' TV	Vincent Tong
Thrillers	Tom Wilkinson
Action & Adventure	Luci Christian
TV Sci-Fi & Fantasy	Lena Headey
Classic Movies	James Robertson Justice
Horror Movies	Lorenza Izzo
Anime Features	John Swasey
Reality TV	Francesco Facchinetti
Sports Movies	Jay Baruchel
Anime Series	Takahiro Sakurai
Kids' TV	Justin Fletcher
International Movies	Carice van Houten
Korean TV Shows	Bae Doona
Sci-Fi & Fantasy	Sophie Thatcher
Science & Nature TV	David Attenborough
Teen TV Shows	Ai Kayano
Cult Movies	Doona Bae

Classic Movies	Burgess Meredith
TV Shows	Prayaga Martin
Children & Family Movies	Koichi Yamadera
Faith & Spirituality	Ait Youssef Youssef
LGBTQ Movies	Allan Paule
Stand-Up Comedy	Jeff Dunham
TV Action & Adventure	Lena Headey
Movies	David Attenborough
Stand-Up Comedy & Talk Shows	Fortune Feimster
Classic & Cult TV	John Dunsworth
Stand-Up Comedy & Talk Shows	Se-yoon Yoo
Anime Features	Minako Kotobuki
Documentaries	Burgess Meredith
Romantic TV Shows	Lisa Vidal
Cult Movies	Charlie Ruedpokanon
Independent Movies	Natalie Martinez
TV Horror	Catherine Lemieux
Spanish-Language TV Shows	Daniela Bascopé
Classic & Cult TV	Ford Kiernan
Music & Musicals	Michael Brons
Romantic Movies	Sarah Troyer
LGBTQ Movies	Hannah Emily Anderson
Stand-Up Comedy	Tig Notaro
TV Sci-Fi & Fantasy	Marc Bendavid
Sports Movies	Lee Dixon

Insights on Average Run Time

Movies

In [105]:

```
df_n_movies=df_final[df_final['type']=='Movie']
unique_genre_movies=list(df_n_movies['genre'].unique())
```

In [106]:

```
df_n_movies.dropna(inplace=True)
```

In [107]:

```
duration=df_n_movies['duration'].apply(lambda x: str(x).split(" ")[0]).tolist()
```

In [108]:

```
duration=np.array(duration).astype("int32")
```

In [109]:

```
df_n_movies["Duration"]=duration
```

In [110]:

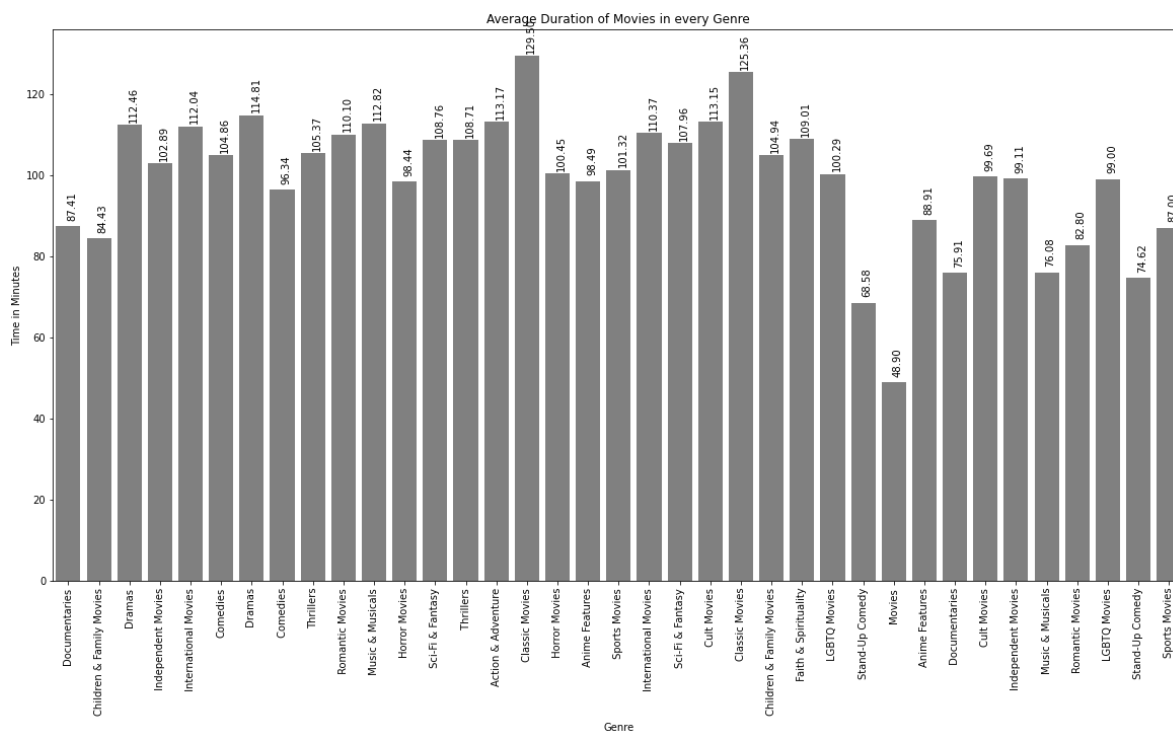
```
group_duration=df_n_movies.groupby('genre')
duration_min=[]
for i in unique_genre_movies:
    t=group_duration.get_group(i)['Duration'].mean()
    t=round(t,3)

    duration_min.append(t)
```

In [111]:

```
plt.figure(figsize=(20,10))
plt.title("Average Duration of Movies in every Genre")
plt.xlabel("Genre")
plt.ylabel("Time in Minutes")
plots = sns.barplot(x = unique_genre_movies, y=duration_min, color="gray")
for bar in plots.patches:

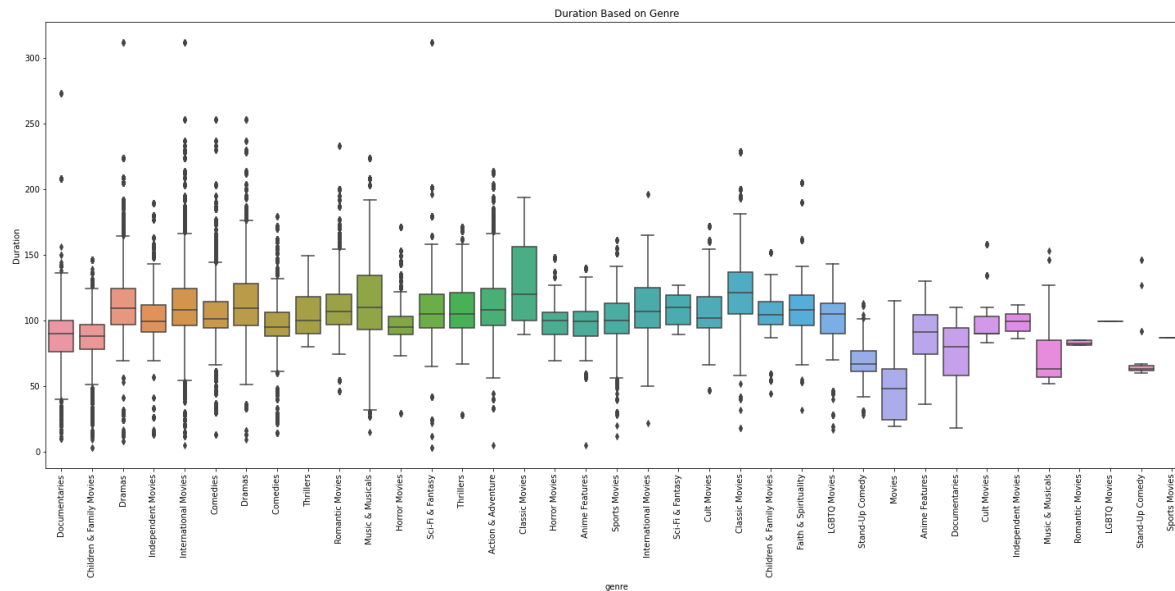
    plots.annotate(format(bar.get_height(), '.2f'),
                    (bar.get_x() + bar.get_width() / 1,
                     bar.get_height()), ha='right', va='center',
                    size=10, xytext=(-3, 20),rotation=90,
                    textcoords='offset points')
plt.xticks(rotation=90)
plt.show()
```



Movie Duration Based on Genre

In [112]:

```
plt.figure(figsize=(25,10))
sns.boxplot(x = 'genre', y = 'Duration', data = df_n_movies)
plt.title("Duration Based on Genre")
plt.xticks(rotation=90)
plt.show()
```



Shows

In [113]:

```
df_n_shows=df_final[df_final['type']=='TV Show']
unique_genre_shows=list(df_n_shows['genre'].unique())
df_n_shows.dropna(inplace=True)
```

In [114]:

```
duration=df_n_shows['duration'].apply(lambda x: str(x).split(" ")[0]).tolist()
duration=np.array(duration).astype("int32")
```

In [115]:

```
df_n_shows["Duration"]=duration
```

In [116]:

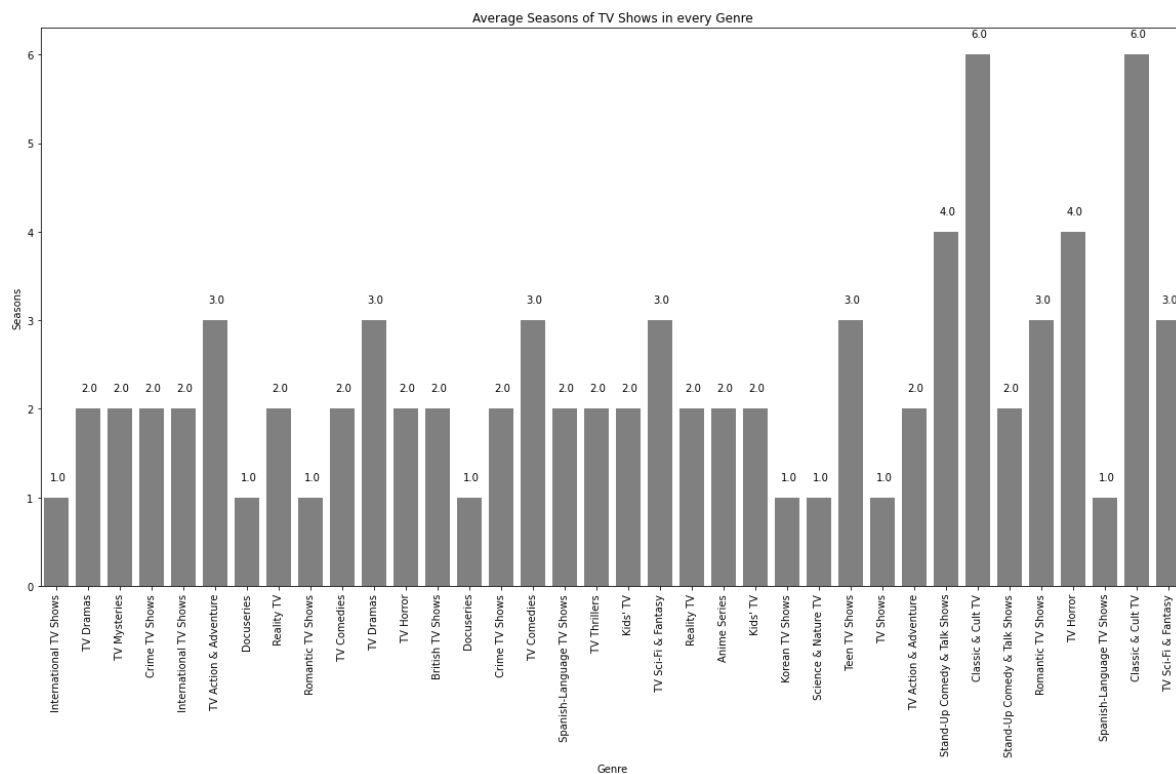
```
group_duration=df_n_shows.groupby('genre')
duration_min=[]
for i in unique_genre_shows:
    t=group_duration.get_group(i)['Duration'].mean()
    t=round(t)

    duration_min.append(int(t))
```


In [117]:

```
plt.figure(figsize=(20,10))
plt.title("Average Seasons of TV Shows in every Genre")
plt.xlabel("Genre")
plt.ylabel("Seasons")
plots = sns.barplot(x = unique_genre_shows, y=duration_min, color="gray")
for bar in plots.patches:

    plots.annotate(format(bar.get_height()),
                    (bar.get_x() + bar.get_width() / 1,
                     bar.get_height()), ha='right', va='center',
                    size=10, xytext=(-3, 20),
                    textcoords='offset points')
plt.xticks(rotation=90)
plt.show()
```



Popular Actor/ Actress

Movies

In [118]:

```
actor_more_movies=df_final[df_final['type']=="Movie"]
```

In [119]:

```
actor_more_movies=actor_more_movies['cast'].value_counts(ascending=False).index.tolist()
```

In [120]:

```

if actor_more_movies[0]=='nan':
    actor_more_movies1=actor_more_movies[1]
else:
    actor_more_movies1=actor_more_movies[0]

print("Actor/Actress who are acted in more number of movies: ",actor_more_movies1)

```

Actor/Actress who are acted in more number of movies: Alfred Molina

In [121]:

```

print("Genres that actor/actress acted")
print(df_final[df_final['cast']==actor_more_movies1]['genre'].unique())

```

Genres that actor/actress acted

```

['Children & Family Movies' ' Dramas' 'Dramas' ' Independent Movies'
 ' Comedies' 'Thrillers' "Kids' TV" ' TV Action & Adventure'
 ' TV Sci-Fi & Fantasy' 'Comedies' 'Independent Movies' ' Thrillers'
 'Action & Adventure' ' Children & Family Movies' ' Classic Movies']

```

Top 20 Popular Actors in Movies

In [122]:

```

for i in range(22):
    if actor_more_movies[i]=='nan':
        pass
    else:
        actor_more_movies1=actor_more_movies[i]
    print(actor_more_movies1.strip())

```

Alfred Molina
 Alfred Molina
 Salma Hayek
 Frank Langella
 John Krasinski
 Liam Neeson
 John Rhys-Davies
 Anupam Kher
 Quvenzhané Wallis
 Ben Whishaw
 Jim Broadbent
 Paresh Rawal
 Om Puri
 James Faulkner
 Boman Irani
 Luci Christian
 Radhika Apte
 Shah Rukh Khan
 Akshay Kumar
 Andy McAvin
 John Swasey
 Naseeruddin Shah

TV Shows

In [123]:

```
actor_more_shows=df_final[df_final['type']=="TV Show"]  
actor_more_shows=actor_more_shows['cast'].value_counts(ascending=False).index.tolist()
```

In [124]:

```
if actor_more_shows[0]!='nan':  
    actor_more_shows1=actor_more_shows[1]  
else:  
    actor_more_shows1=actor_more_shows[0]  
  
print("Actor/Actress who are acted in more number of TV Shows: ",actor_more_shows1)
```

Actor/Actress who are acted in more number of TV Shows: David Attenborough

In [125]:

```
print("Genres that actor/actress acted")  
print(df_final[df_final['cast']==actor_more_shows1]['genre'].unique())
```

Genres that actor/actress acted

```
['Documentaries' 'British TV Shows' ' Docuseries'  
 ' International TV Shows' 'Docuseries' ' Science & Nature TV' 'Movies']
```

In [126]:

```
for i in range(22):  
    if actor_more_shows[i]!='nan':  
        pass  
    else:  
        actor_more_shows1=actor_more_movies[i]  
    print(actor_more_shows1.strip())
```

David Attenborough
Alfred Molina
Salma Hayek
Frank Langella
John Krasinski
Liam Neeson
John Rhys-Davies
Anupam Kher
Quvenzhané Wallis
Ben Whishaw
Jim Broadbent
Paresh Rawal
Om Puri
James Faulkner
Boman Irani
Luci Christian
Radhika Apte
Shah Rukh Khan
Akshay Kumar
Andy McAvin
John Swasey
Naseeruddin Shah

Insights on Date Added

Number of Movies added in Netflix Month Wise

In [127]:

```
month_movies=df[df['type']=="Movie"]
```

In [128]:

```
month_added=month_movies['date_added'].apply(lambda x: str(x).split(" ")[0])
```

In [129]:

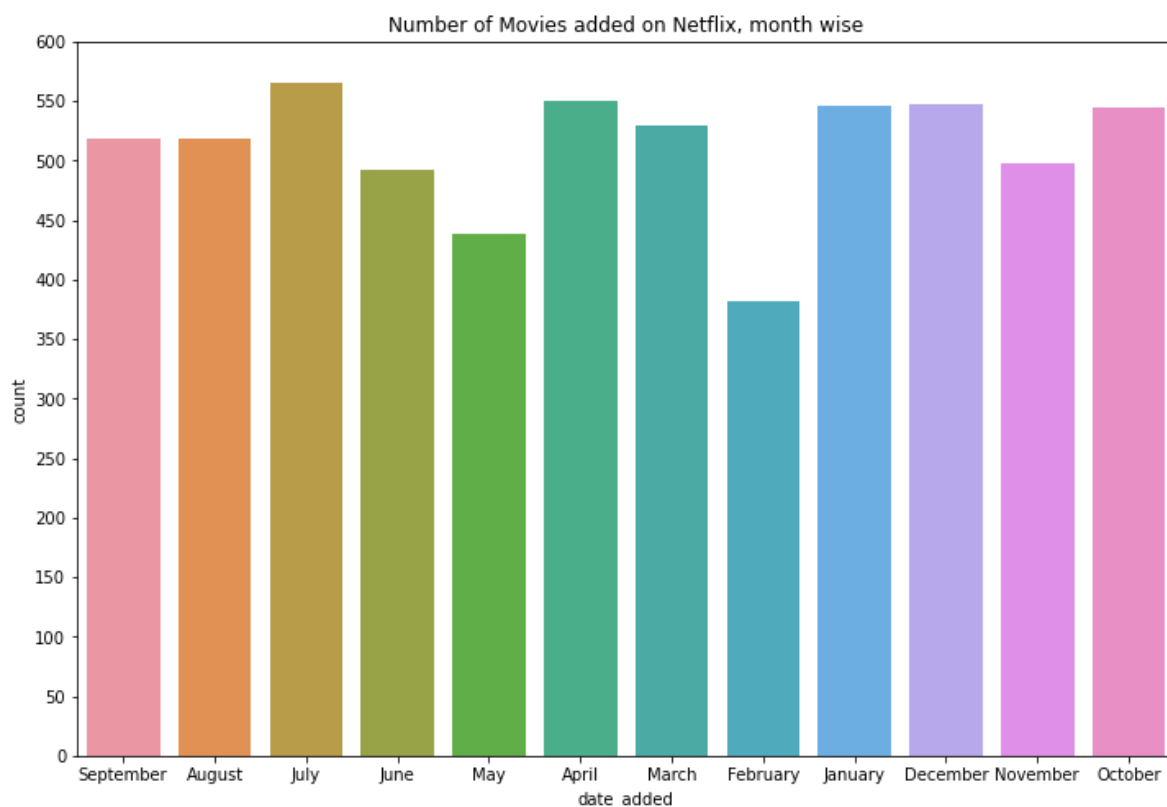
```
month_added=pd.DataFrame(month_added)
```

In [130]:

```
month_added.dropna(inplace=True)
```

In [131]:

```
plt.figure(figsize=(12,8))
sns.countplot(x='date_added',data=month_added)
plt.title("Number of Movies added on Netflix, month wise")
plt.yticks(np.arange(0,650,50))
plt.show()
```



Number of Shows added in Netflix Month Wise

In [139]:

```
month_shows=df[df['type']=="TV Show"]
```

In [133]:

```
month_shows.dropna(inplace=True)
```

In [140]:

```
month_shows=month_shows['date_added'].apply(lambda x: str(x).split(" ")[0])
```

In [141]:

```
month_shows=pd.DataFrame(month_shows)
```

In [142]:

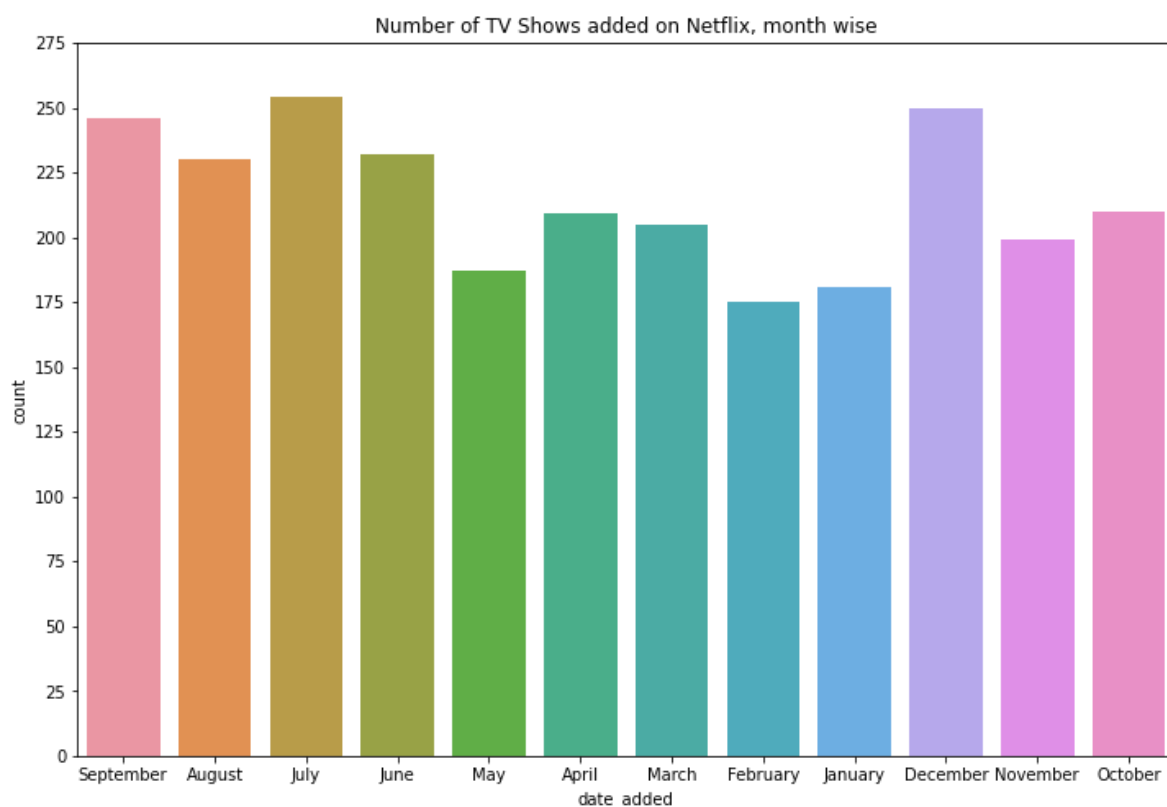
```
drop_=month_shows.index[month_shows['date_added']=='nan'].tolist()  
drop_+=month_shows.index[month_shows['date_added']==''].tolist()
```

In [143]:

```
month_shows.drop(drop_,inplace=True)
```

In [144]:

```
plt.figure(figsize=(12,8))  
sns.countplot(x='date_added',data=month_shows)  
plt.title("Number of TV Shows added on Netflix, month wise")  
plt.yticks(np.arange(0,276,25))  
plt.show()
```



Insights on Country

Top 10 Highest Movies Producing Countries

In [145]:

```
country_movies=df[df['type']=="Movie"]
```

In [146]:

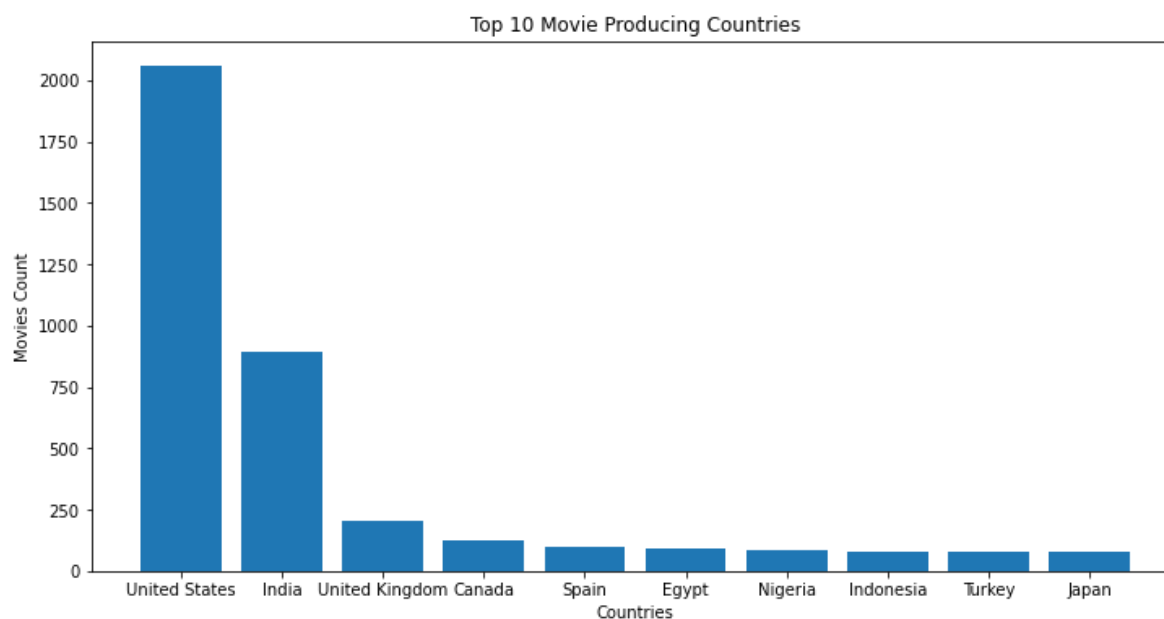
```
countries=country_movies['country'].value_counts().keys().tolist()  
count=country_movies['country'].value_counts().tolist()
```

In [147]:

```
countries=countries[:10]  
count=count[:10]
```

In [148]:

```
plt.figure(figsize=(12,6))  
plt.title("Top 10 Movie Producing Countries")  
plt.bar(x=countries,height=count)  
plt.xlabel("Countries")  
plt.ylabel("Movies Count")  
plt.show()
```



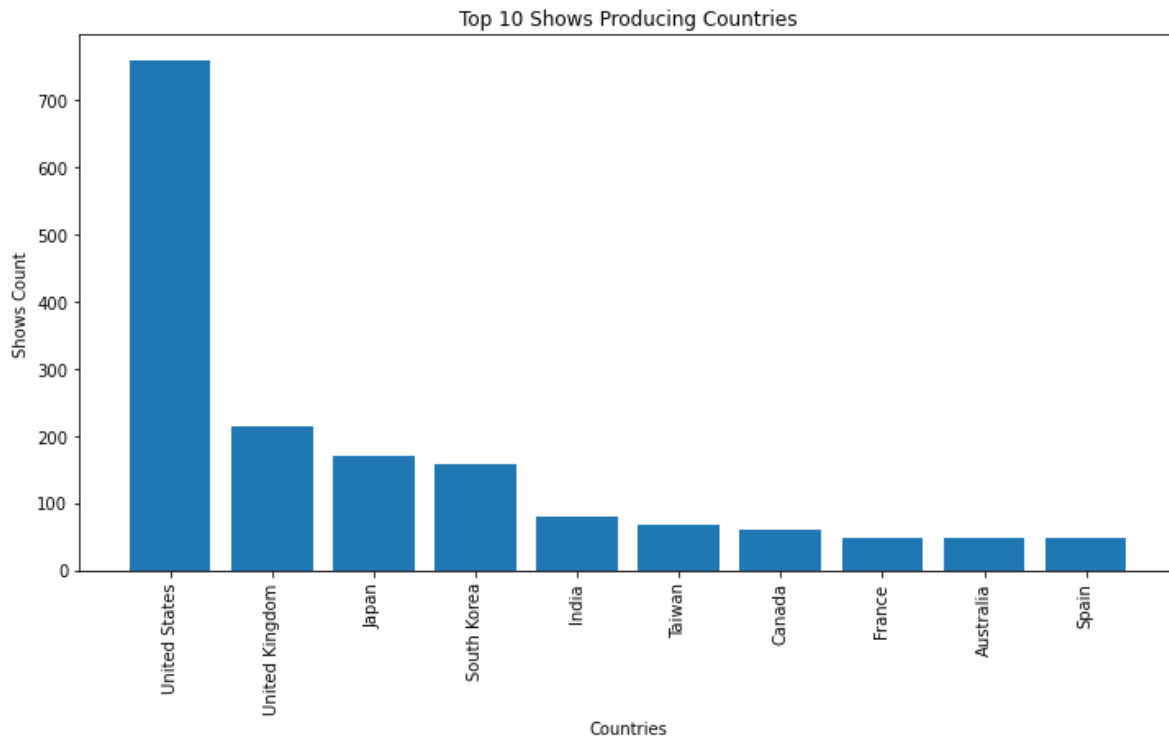
Top 10 Highest Shows Producing Countries

In [149]:

```
country_shows=df[df['type']=="TV Show"]  
countries=country_shows['country'].value_counts().keys().tolist()  
count=country_shows['country'].value_counts().tolist()  
countries=countries[:10]  
count=count[:10]
```

In [150]:

```
plt.figure(figsize=(12,6))
plt.title("Top 10 Shows Producing Countries")
plt.bar(x=countries,height=count)
plt.xlabel("Countries")
plt.ylabel("Shows Count")
plt.xticks(rotation=90)
plt.show()
```



Insights on Director

Top 20 popular Directors

In [151]:

```
directors=df_final['director'].value_counts(ascending=False).index.tolist()
```


In [152]:

```
for i in range(22):  
    if directors[i]!='nan':  
        pass  
    else:  
        popular_director=directors[i]  
        print(popular_director.strip())
```

Martin Scorsese
Youssef Chahine
Cathy Garcia-Molina
Steven Spielberg
Lars von Trier
Raja Gosnell
Tom Hooper
McG
David Dhawan
Wilson Yip
Don Michael Paul
Martin Campbell
Noah Baumbach
Olivier Assayas
Yorgos Lanthimos
Yılmaz Erdoğan
Ron Howard
Kunle Afolayan
Robert Vince
Terry Gilliam
Noam Murro