In [2]:

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

In [3]:

```
df_new=pd.read_csv(r'C:\Users\suryawaa\OneDrive - TomTom\2022\Scaler\Netflix\netflix.csv')
df_new
```

Out[3]:

	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...
...	...	...	...	...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	2 Seasons	Kids' TV, Korean TV Shows, TV Comedies	While living alone in a spooky town, a young g...
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	Comedies, Horror Movies	Looking to survive in a world taken over by zo...
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

8807 rows × 12 columns

```
In [4]: df_new.shape # number of (rows,columns)
```

```
Out[4]: (8807, 12)
```

```
In [5]: df_new.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
```

- Observation On Data:
- There are total 8807 records(rows) and 12 columns
- 6/12 columns also contains some null values in the records

```
In [6]: df_new.isna().sum() # Number of missing values in each columns
```

```
Out[6]: show_id            0
        type               0
        title              0
        director        2634
        cast             825
        country          831
        date_added        10
        release_year       0
        rating             4
        duration           3
        listed_in          0
        description        0
        dtype: int64
```

```
In [7]: df_new.describe(include=object)
```

Out[7]:

	show_id	type	title	director	cast	country	date_added	rating	duration	listed_in	description
count	8807	8807	8807	6173	7982	7976	8797	8803	8804	8807	8807
unique	8807	2	8807	4528	7692	748	1767	17	220	514	8775
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	TV-MA	1 Season	Dramas, International Movies	Paranormal activity at a lush, abandoned prope...
freq	1	6131	1	19	19	2818	109	3207	1793	362	4

- Observation On Data:
- Top row shows total count of non null in each columns
- Second row shows total uniqure records out of total records in columns
- Third row shows most commonly occuring value (Here as movie title are unique by default top most row)
- Last row show frequency of most common value

- Conversion of data types:
- Conversion of string date into datetime may require

In [8]:

```
df_new['date_added'] = pd.to_datetime(df_new['date_added'])
```

In [9]:

```
df_new.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   datetime64[ns]
 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: datetime64[ns](1), int64(1), object(10)
memory usage: 825.8+ KB
```

- Unnesting of data for directors,cast,genre,country

In [10]:

```
directors_list = df_new['director'].apply(lambda x: str(x).split(',')).tolist()
```

```
In [11]: df_1 = pd.DataFrame(directors_list,index=df_new['title'])
         df_1
```

```
Out[11]:
                                   0     1     2     3     4     5     6     7     8     9    10    11    12
                   title
     Dick Johnson Is Dead  Kirsten Johnson  None  None  None  None  None  None  None  None  None  None  None  None
            Blood & Water              nan  None  None  None  None  None  None  None  None  None  None  None  None
                Ganglands  Julien Leclercq  None  None  None  None  None  None  None  None  None  None  None  None
    Jailbirds New Orleans              nan  None  None  None  None  None  None  None  None  None  None  None  None
             Kota Factory              nan  None  None  None  None  None  None  None  None  None  None  None  None
                      ...              ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
                   Zodiac    David Fincher  None  None  None  None  None  None  None  None  None  None  None  None
              Zombie Dumb              nan  None  None  None  None  None  None  None  None  None  None  None  None
               Zombieland  Ruben Fleischer  None  None  None  None  None  None  None  None  None  None  None  None
                     Zoom     Peter Hewitt  None  None  None  None  None  None  None  None  None  None  None  None
                   Zubaan      Mozez Singh  None  None  None  None  None  None  None  None  None  None  None  None

8807 rows × 13 columns
```

```
In [12]: df_1 = df_1.stack()
         df_1
```

```
Out[12]: title
         Dick Johnson Is Dead   0    Kirsten Johnson
         Blood & Water          0                nan
         Ganglands              0    Julien Leclercq
         Jailbirds New Orleans  0                nan
         Kota Factory           0                nan
                                          ...
         Zodiac                 0      David Fincher
         Zombie Dumb            0                nan
         Zombieland             0    Ruben Fleischer
         Zoom                   0       Peter Hewitt
         Zubaan                 0        Mozez Singh
         Length: 9612, dtype: object
```

In [13]:
```
df_1 = pd.DataFrame(df_1)
df_1
```

Out[13]:

		0
title		
Dick Johnson Is Dead	**0**	Kirsten Johnson
Blood & Water	**0**	nan
Ganglands	**0**	Julien Leclercq
Jailbirds New Orleans	**0**	nan
Kota Factory	**0**	nan
...	**...**	...
Zodiac	**0**	David Fincher
Zombie Dumb	**0**	nan
Zombieland	**0**	Ruben Fleischer
Zoom	**0**	Peter Hewitt
Zubaan	**0**	Mozez Singh

9612 rows × 1 columns

In [14]:

```
df_1.reset_index(inplace=True)
df_1
```

Out[14]:

```
                    title  level_1                0
   0  Dick Johnson Is Dead        0  Kirsten Johnson
   1         Blood & Water        0              nan
   2             Ganglands        0  Julien Leclercq
   3  Jailbirds New Orleans       0              nan
   4          Kota Factory        0              nan
 ...                   ...      ...              ...
9607               Zodiac        0    David Fincher
9608          Zombie Dumb        0              nan
9609           Zombieland        0  Ruben Fleischer
9610                 Zoom        0     Peter Hewitt
9611               Zubaan        0      Mozez Singh

9612 rows × 3 columns
```

In [15]:

```
df_1.drop(["level_1"],axis=1,inplace=True)
df_1.head(10)
```

Out[15]:

	title	0
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan
5	Midnight Mass	Mike Flanagan
6	My Little Pony: A New Generation	Robert Cullen
7	My Little Pony: A New Generation	José Luis Ucha
8	Sankofa	Haile Gerima
9	The Great British Baking Show	Andy Devonshire

In [16]:
```
df_1.columns = ['title','directors']
df_1
```

Out[16]:

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

9612 rows × 2 columns

- Similar for cast,genre & country:

In [17]:

```
cast_list = df_new['cast'].apply(lambda x: str(x).split(',')).tolist()
df_2 = pd.DataFrame(cast_list,index=df_new['title'])
df_2 = df_2.stack()
df_2 = pd.DataFrame(df_2)
df_2.reset_index(inplace=True)
df_2.drop(["level_1"],axis=1,inplace=True)
df_2.columns = ['title','cast']
df_2
```

Out[17]:

	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

In [18]:

```
genre_list = df_new['listed_in'].apply(lambda x: str(x).split(',')).tolist()
df_3 = pd.DataFrame(genre_list,index=df_new['title'])
df_3 = df_3.stack()
df_3 = pd.DataFrame(df_3)
df_3.reset_index(inplace=True)
df_3.drop(["level_1"],axis=1,inplace=True)
df_3.columns = ['title','genre']
df_3
```

Out[18]:

	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 [19]: country_list = df_new['country'].apply(lambda x: str(x).split(',')).tolist()
         df_4 = pd.DataFrame(country_list,index=df_new['title'])
         df_4 = df_4.stack()
         df_4 = pd.DataFrame(df_4)
         df_4.reset_index(inplace=True)
         df_4.drop(["level_1"],axis=1,inplace=True)
         df_4.columns = ['title','country']
         df_4
```

```
Out[19]:
                       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 [20]:
```
df_new.nunique() # Raw data with unique value count
```

Out[20]:
```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1714
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
```

In [21]:
```
# Number of total unique directors

df_1['directors'].nunique()
```

Out[21]:
```
5121
```

In [22]:
```
df_1.describe() # statstically also can verify the same
```

Out[22]:

	title	directors
count	9612	9612
unique	8807	5121
top	Walt Disney Animation Studios Short Films Coll...	nan
freq	13	2634

In [23]:
```
top_directs = df_1['directors'].value_counts()
top10_directors = pd.DataFrame(top_directs)
top10_directors
```

Out[23]:
```
                   directors
              nan       2634
    Rajiv Chilaka         22
        Jan Suter         18
      Raúl Campos         18
    Marcus Raboy          16
              ...        ...
       Eric Bross          1
   Will Eisenberg          1
 Marina Seresesky          1
       Kenny Leon          1
      Mozez Singh          1

5121 rows × 1 columns
```

In [24]:
```
# Number of total unique people as cast

df_2['cast'].nunique()
```

Out[24]:
```
39297
```

In [25]:
```
df_2.describe()
```

Out[25]:

	title	cast
count	64951	64951
unique	8807	39297
top	Social Distance	nan
freq	50	825

In [104]:
```
df_2.cast.value_counts()
```

Out[104]:
```
nan                          825
 Anupam Kher                  39
 Rupa Bhimani                 31
 Takahiro Sakurai             30
 Julie Tejwani                28
                            ... 
 Vedika                        1
 Tedros Teclebrhan             1
 Maryam Zaree                  1
 Melanie Straub                1
 Chittaranjan Tripathy         1
Name: cast, Length: 39297, dtype: int64
```

```
In [105]: df_new.cast.value_counts()
```

```
Out[105]: David Attenborough
          19
          Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mousam, Swapnil
          14
          Samuel West
          10
          Jeff Dunham
          7
          David Spade, London Hughes, Fortune Feimster
          6

          ..
          Michael Peña, Diego Luna, Tenoch Huerta, Joaquin Cosio, José María Yazpik, Matt Letscher, Alyssa Diaz
          1
          Nick Lachey, Vanessa Lachey
          1
          Takeru Sato, Kasumi Arimura, Haru, Kentaro Sakaguchi, Takayuki Yamada, Kendo Kobayashi, Ken Yasuda, Arata Fu
          ruta, Suzuki Matsuo, Koichi Yamadera, Arata Iura, Chikako Kaku, Kotaro Yoshida     1
          Toyin Abraham, Sambasa Nzeribe, Chioma Chukwuka Akpotha, Chioma Omeruah, Chiwetalu Agu, Dele Odule, Femi Ade
          bayo, Bayray McNwizu, Biodun Stephen                                               1
          Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna Malik, Malkeet Rauni, Anita Shabdi
          sh, Chittaranjan Tripathy                                                          1
          Name: cast, Length: 7692, dtype: int64
```

```
In [26]: # Number of total unique people as cast

         df_3['genre'].nunique()
```

```
Out[26]: 73
```

In [161]:

```
df_3.describe()
```

Out[161]:

	title	genre
count	19323	19323
unique	8807	73
top	Zubaan	International Movies
freq	3	2624

In [163]:

```
df_3['genre'] = df_3['genre'].str.strip()
```

In [164]:

```
df_3
```

Out[164]:

	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 [28]:

```
# Top most genre and their count
top_genre = df_3.groupby('genre')['title'].count().sort_values(ascending=False)
top_genre
```

Out[28]:

```
genre
 International Movies        2624
Dramas                       1600
Comedies                     1210
Action & Adventure            859
Documentaries                 829
                             ... 
Romantic Movies                 3
Spanish-Language TV Shows       2
TV Sci-Fi & Fantasy             1
LGBTQ Movies                    1
Sports Movies                   1
Name: title, Length: 73, dtype: int64
```

Top most countries shows are released in by count

In [29]:

```
# Number of total unique countries where movies/shows released


df_4['country'] = df_4['country'].str.strip()
df_4['country'].nunique()
```

Out[29]:

```
124
```

In [30]:

```
df_4.describe() # Actual Unique countries
```

Out[30]:

	title	country
count	10850	10850
unique	8807	124
top	Barbecue	United States
freq	12	3690

In [31]:

```
top_countries = df_4.groupby('country')['title'].nunique().sort_values(ascending=False)
top10_countries = pd.DataFrame(top_countries.head(10))
top10_countries
```

Out[31]:

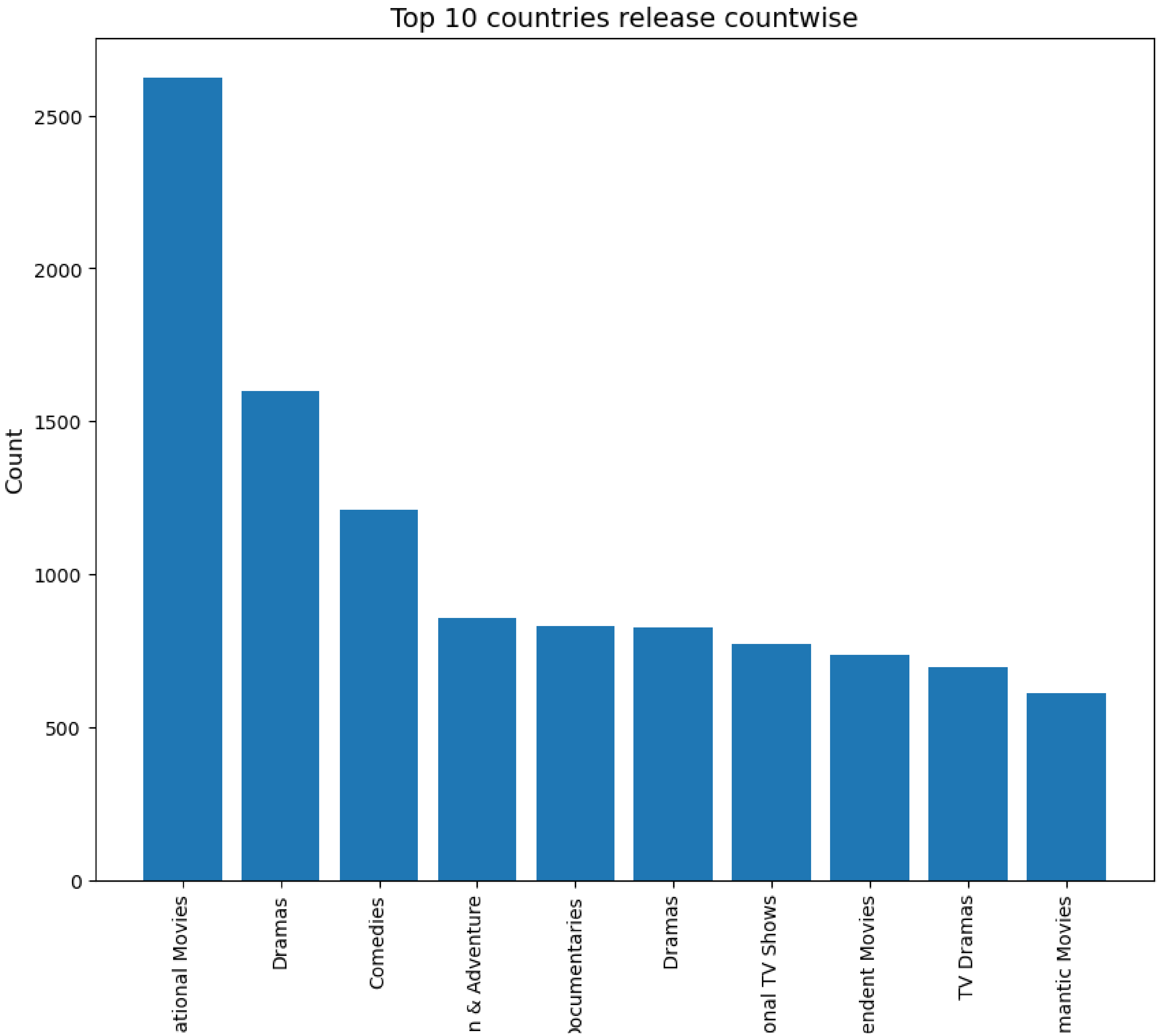
```
                   title
         country
          United States   3690
                  India   1046
                    nan    831
         United Kingdom    806
                 Canada    445
                 France    393
                  Japan    318
                  Spain    232
            South Korea    231
                Germany    226
```

In [32]:

```
x_val = top10_countries.index
y_val = top10_countries['title']
```

In [174]:

```
plt.figure(figsize=(10,8))
plt.bar(x_val,y_val)
plt.xticks(rotation=90)
plt.xlabel('Country',fontsize=12)
plt.ylabel('Count',fontsize=12)
plt.title('Top 10 countries release countwise',fontsize=14)
plt.show()
```


Top 10 countries release countwise
Count
0
500
1000
1500
2000
2500
ational Movies
Dramas
Comedies
n & Adventure
ocumentaries
Dramas
onal TV Shows
endent Movies
TV Dramas
mantic Movies

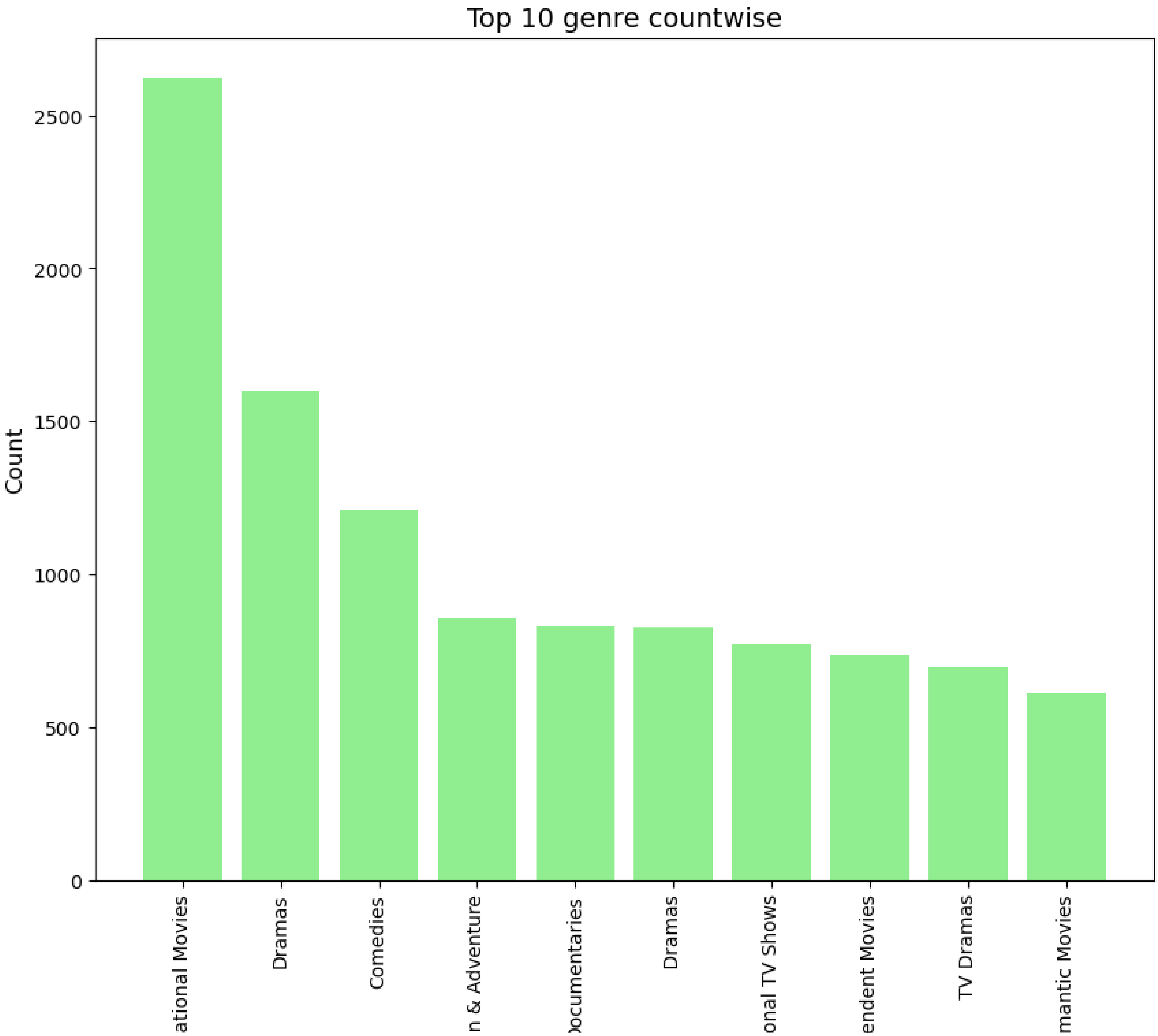
```
In [125]: top10_genre = pd.DataFrame(top_genre.head(10))
          top10_genre
```

```
Out[125]:
                              title
                      genre
        International Movies   2624
                      Dramas   1600
                    Comedies   1210
          Action & Adventure    859
               Documentaries    829
                      Dramas    827
        International TV Shows  774
          Independent Movies    736
                   TV Dramas    696
             Romantic Movies    613
```

```
In [126]: x_val = top10_genre.index
          y_val = top10_genre['title']
```

In [127]:

```
plt.figure(figsize=(10,8))
plt.bar(x_val,y_val,color='lightgreen')
plt.xticks(rotation=90)
plt.xlabel('Genre',fontsize=12)
plt.ylabel('Count',fontsize=12)
plt.title('Top 10 genre countwise',fontsize=14)
plt.show()
```


Top 10 genre countwise
Count
2500
2000
1500
1000
500
0
ational Movies
Dramas
Comedies
n & Adventure
ocumentaries
Dramas
onal TV Shows
endent Movies
TV Dramas
mantic Movies

In [122]:

```python
top10_direct = df_1.groupby('directors')['title'].count().sort_values(ascending=False).head(11)
top10_direct = pd.DataFrame(top10_direct)
top10_direct = top10_direct[1:]
top10_direct
```

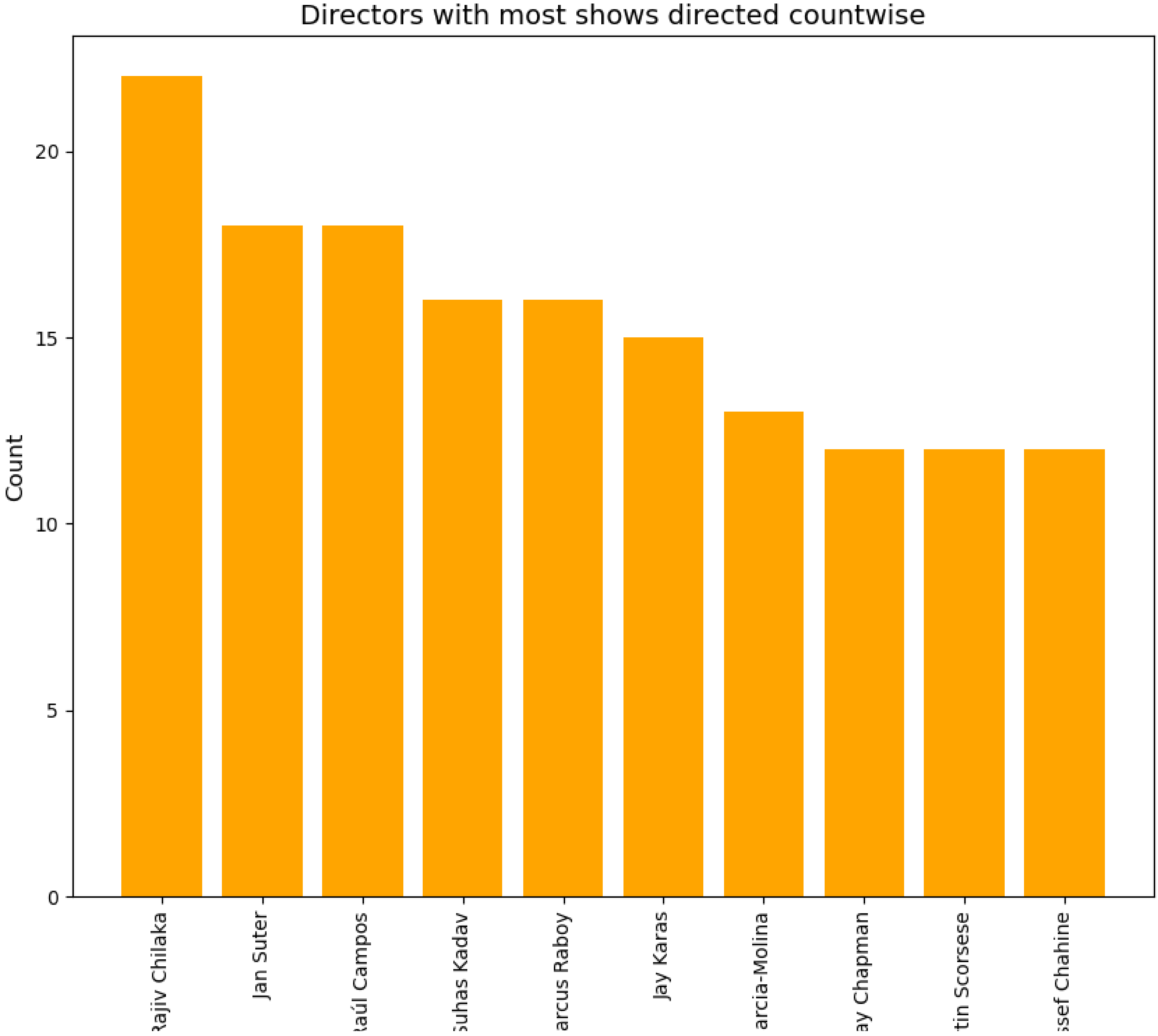
Out[122]:

	title
directors	
Rajiv Chilaka	22
Jan Suter	18
Raúl Campos	18
Suhas Kadav	16
Marcus Raboy	16
Jay Karas	15
Cathy Garcia-Molina	13
Jay Chapman	12
Martin Scorsese	12
Youssef Chahine	12

In [123]:

```python
x_val = top10_direct.index
y_val = top10_direct['title']
```

```
In [124]: plt.figure(figsize=(10,8))
          plt.bar(x_val,y_val,color='orange')
          plt.xticks(rotation=90)
          plt.xlabel('Directors',fontsize=12)
          plt.ylabel('Count',fontsize=12)
          plt.title('Directors with most shows directed countwise',fontsize=14)
          plt.show()
```



Directors with most shows directed countwise

In [40]:
```
df_new['release_year']
```

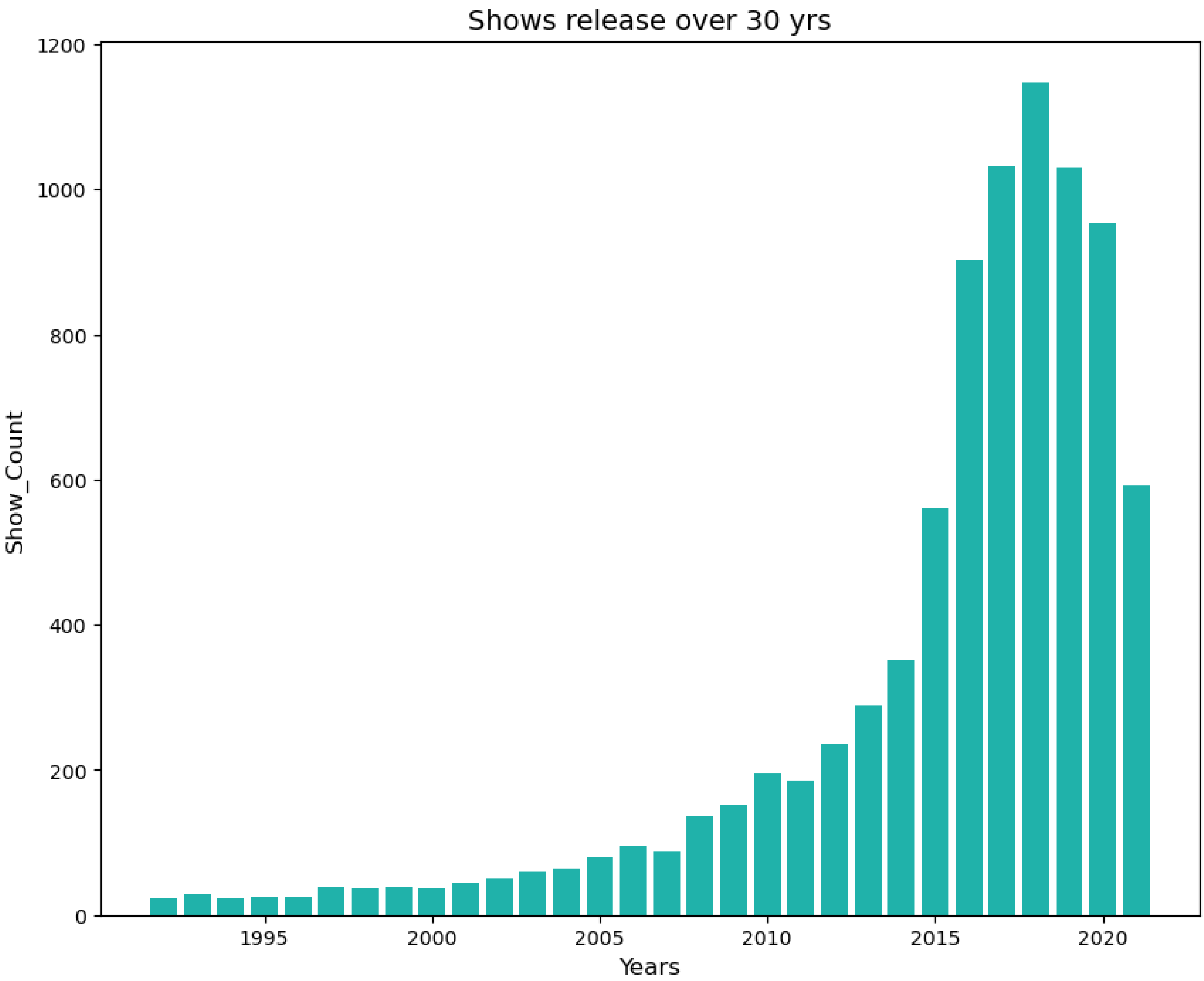
Out[40]:
```
0       2020
1       2021
2       2021
3       2021
4       2021
        ... 
8802    2007
8803    2018
8804    2009
8805    2006
8806    2015
Name: release_year, Length: 8807, dtype: int64
```

In [118]:
```
last_30_yr = df_new.groupby('release_year')['title'].count().reset_index().tail(30)
last_30_yr = pd.DataFrame(last_30_yr)
last_30_yr.columns = ['release_year','show_count']
```

In [119]:
```
x_val = last_30_yr['release_year']
y_val = last_30_yr['show_count']
```

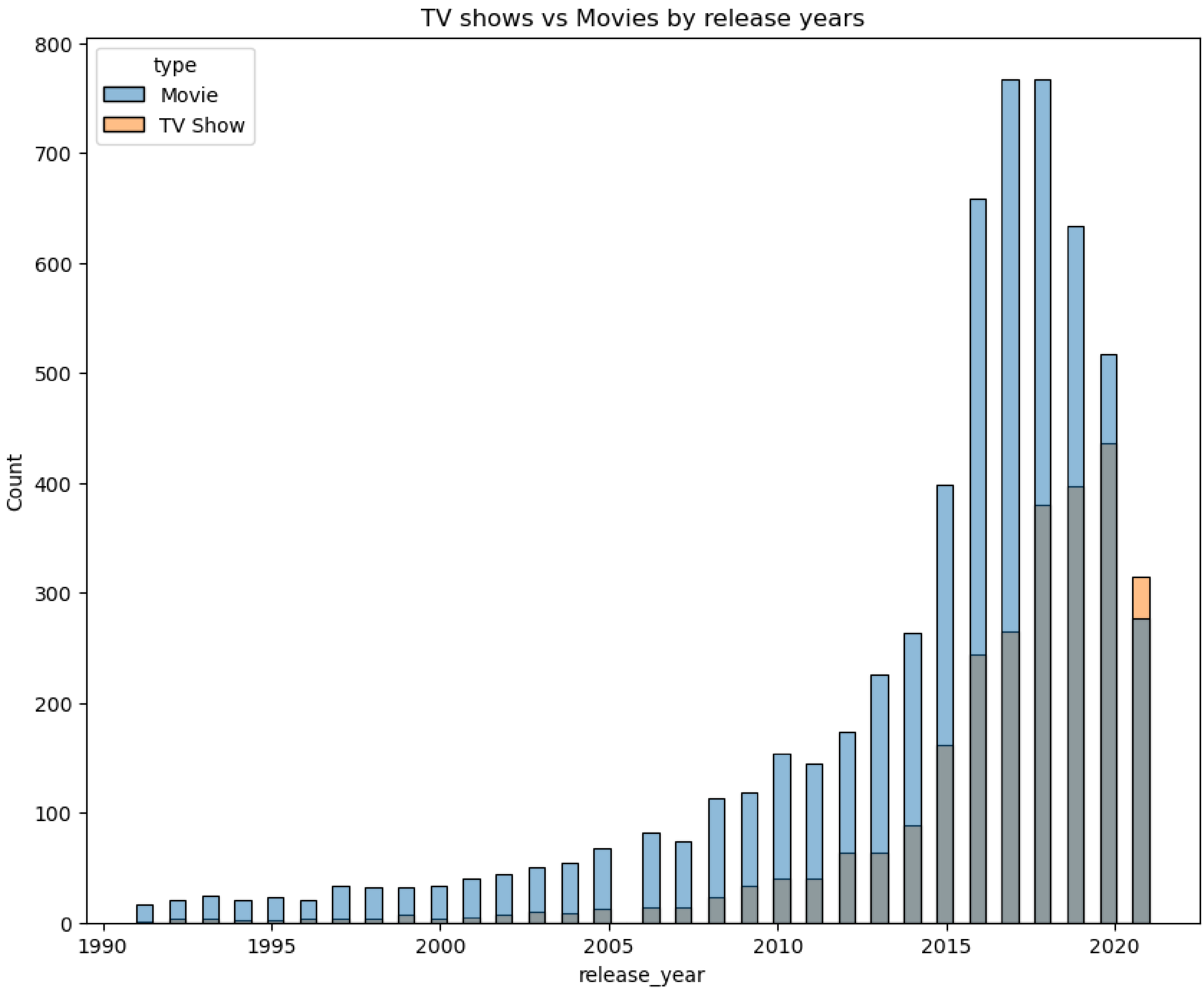
In [120]:

```
plt.figure(figsize=(10,8))
plt.bar(x_val,y_val,color='lightseagreen')
plt.xlabel('Years',fontsize=12)
plt.ylabel('Show_Count',fontsize=12)
plt.title('Shows release over 30 yrs',fontsize=14)
plt.show()
```


Shows release over 30 yrs
Show_Count
0
200
400
600
800
1000
1200
Years
1995
2000
2005
2010
2015
2020

In [178]:

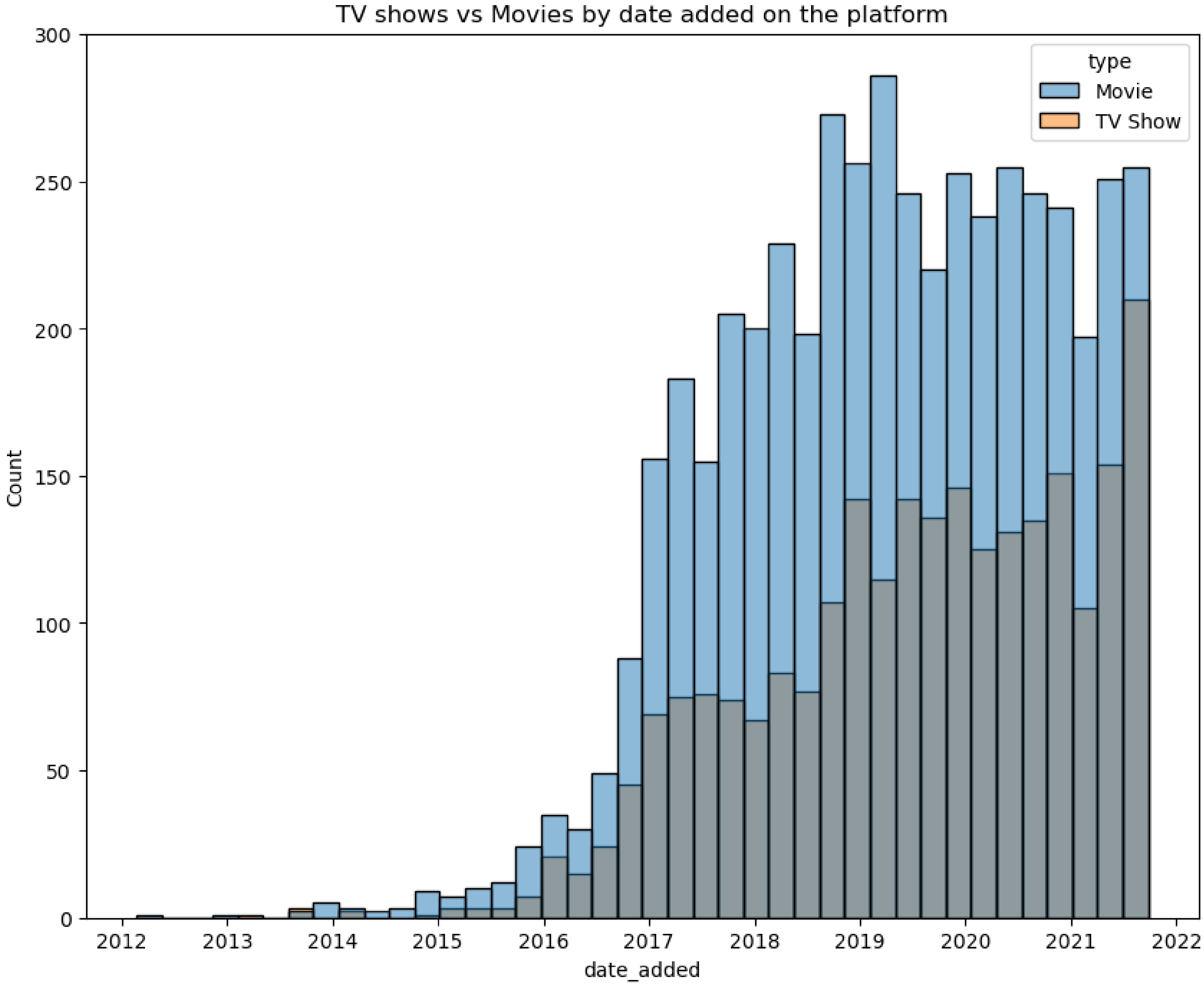
```
plt.figure(figsize=(10,8))
sns.histplot(x='release_year',hue='type',data=df_new.loc[df_new['release_year'] > 1990])
plt.title("TV shows vs Movies by release years")
plt.show()
```


TV shows vs Movies by release years
type
Movie
TV Show
Count
0
100
200
300
400
500
600
700
800
release_year
1990
1995
2000
2005
2010
2015
2020

* Shows on the netflix and their release changed slowly for first 10 to 15 yrs and then it sharply rose from 2014 to 2018 reaches max from where it again dipped for next 3 years
* Overall we can observe there are lot more number of movies realeased than the TV shows till now. Except in the year 2021 where there are more number of TV shows are released than movies.

In [176]:

```
plt.figure(figsize=(10,8))
sns.histplot(x='date_added',hue='type',data=df_new.loc[df_new['release_year'] > 2010])
plt.title("TV shows vs Movies by date added on the platform")
plt.show()
```


TV shows vs Movies by date added on the platform

#### Movies vs TV shows

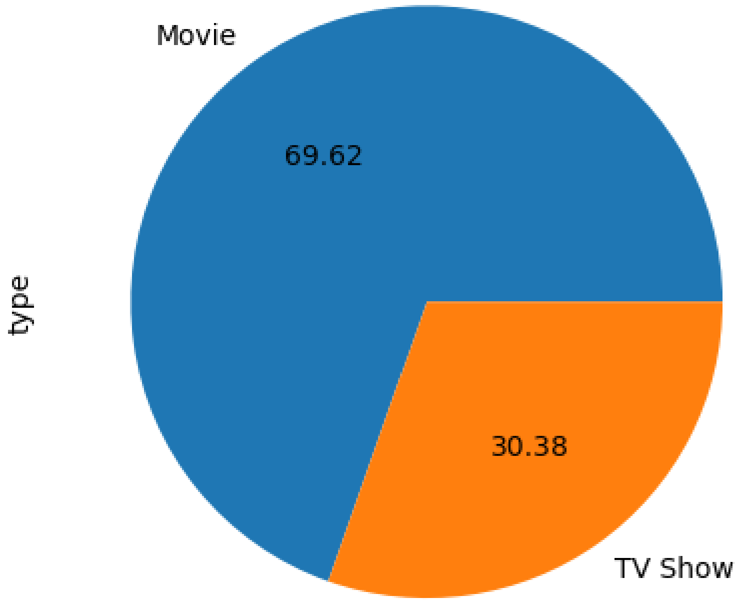
In [44]:
```
tv_movies = df_new.groupby('type')['title'].count()
tv_movies
```

Out[44]:
```
type
Movie      6131
TV Show    2676
Name: title, dtype: int64
```

In [101]:
```
df_new['type'].value_counts(normalize=True)*100
```

Out[101]:
```
Movie      69.615079
TV Show    30.384921
Name: type, dtype: float64
```

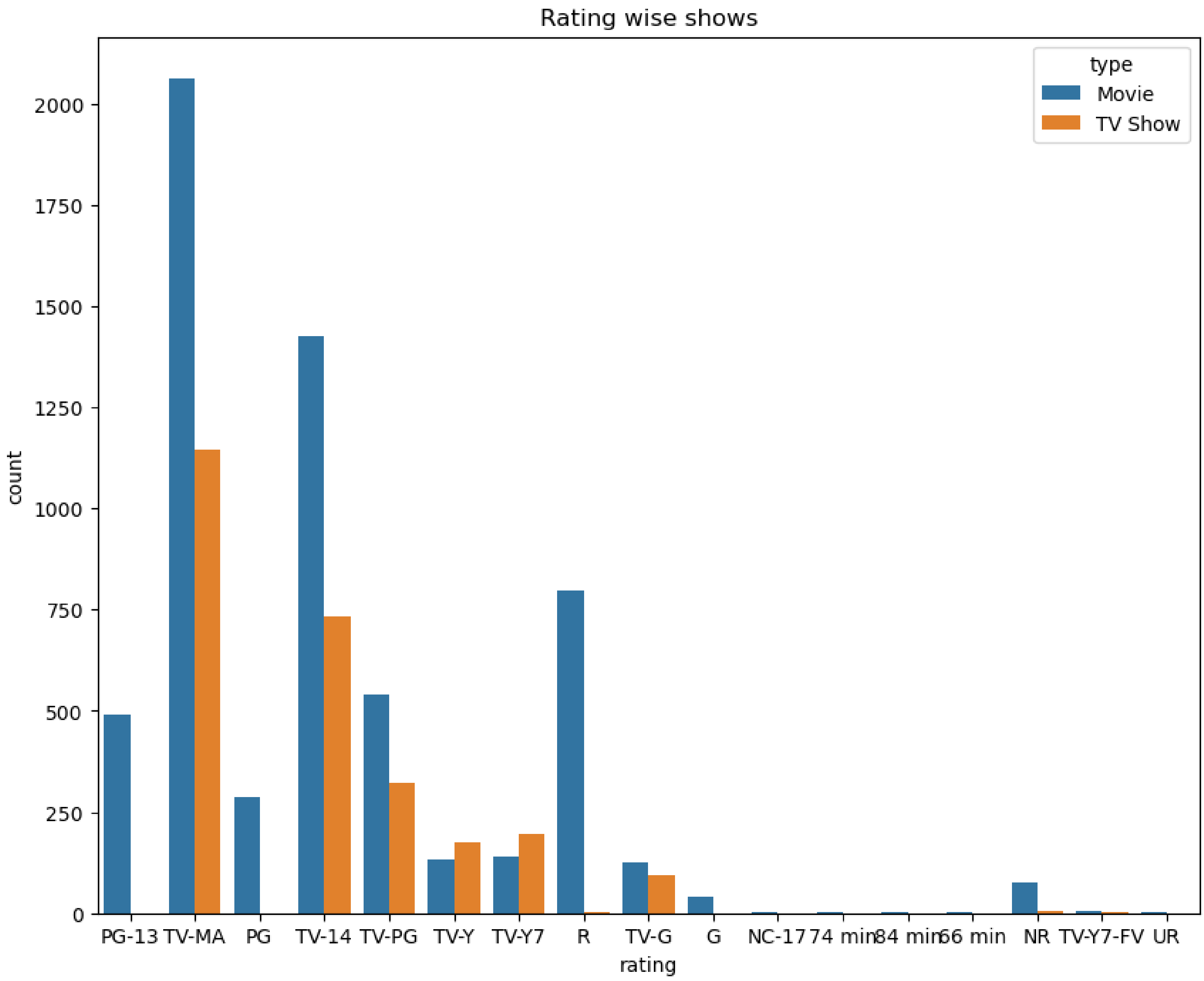
```
In [103]: df_new['type'].value_counts().plot(kind='pie',autopct="%.2f")
          plt.show()
```



- Number of Total 2676 TV shows
- Number of Total 6131 movies
- Overall there are more number of unique movies than TV_shows

In [175]:

```
plt.figure(figsize=(10,8))
sns.countplot(x='rating', hue='type', data=df_new)
plt.title("Rating wise shows")
plt.show()
```


Rating wise shows
type
Movie
TV Show
count
0
250
500
750
1000
1250
1500
1750
2000
PG-13
TV-MA
PG
TV-14
TV-PG
TV-Y
TV-Y7
R
TV-G
G
NC-17
74 min
84 min
66 min
NR
TV-Y7-FV
UR
rating

* Ratingwise with TV-MA is topmost amongst both the type of shows

#### Global distribution of genres

```
In [170]: gen_order = df_3.groupby('genre')['title'].count().sort_values(ascending=False)
```

In [171]:

```
gen_order
```

```
Out[171]: genre
          International Movies            2752
          Dramas                          2427
          Comedies                        1674
          International TV Shows          1351
          Documentaries                    869
          Action & Adventure               859
          TV Dramas                        763
          Independent Movies               756
          Children & Family Movies         641
          Romantic Movies                  616
          TV Comedies                      581
          Thrillers                        577
          Crime TV Shows                   470
          Kids' TV                         451
          Docuseries                       395
          Music & Musicals                 375
          Romantic TV Shows                370
          Horror Movies                    357
          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
Name: title, dtype: int64
```

In [172]:

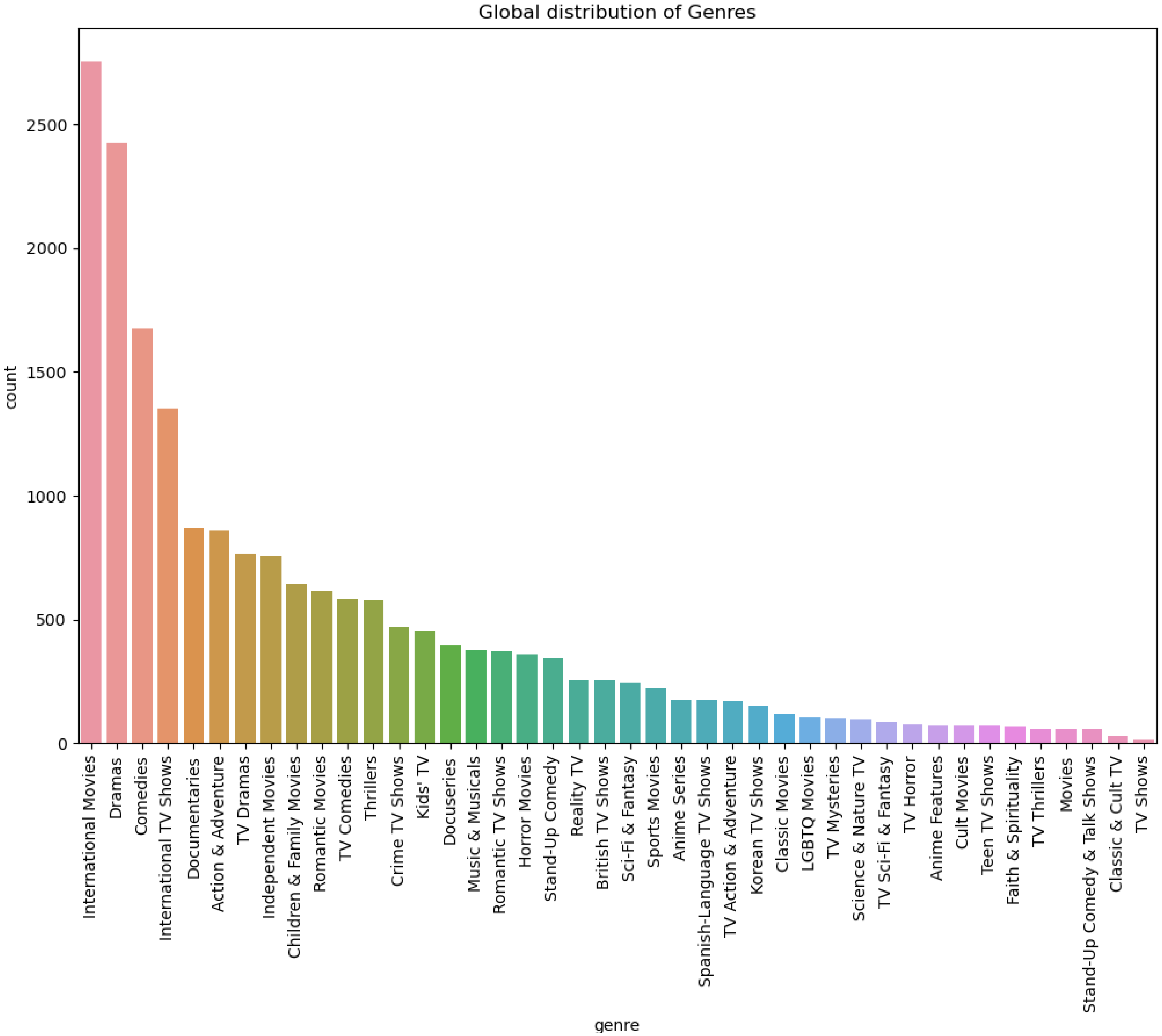
```
gen_order.index
```

Out[172]:

```
Index(['International Movies', 'Dramas', 'Comedies', 'International TV Shows',
       'Documentaries', 'Action & Adventure', 'TV Dramas',
       'Independent Movies', 'Children & Family Movies', 'Romantic Movies',
       'TV Comedies', 'Thrillers', 'Crime TV Shows', 'Kids' TV', 'Docuseries',
       'Music & Musicals', 'Romantic TV Shows', 'Horror Movies',
       'Stand-Up Comedy', 'Reality TV', 'British TV Shows', 'Sci-Fi & Fantasy',
       'Sports Movies', 'Anime Series', 'Spanish-Language TV Shows',
       'TV Action & Adventure', 'Korean TV Shows', 'Classic Movies',
       'LGBTQ Movies', 'TV Mysteries', 'Science & Nature TV',
       'TV Sci-Fi & Fantasy', 'TV Horror', 'Anime Features', 'Cult Movies',
       'Teen TV Shows', 'Faith & Spirituality', 'TV Thrillers', 'Movies',
       'Stand-Up Comedy & Talk Shows', 'Classic & Cult TV', 'TV Shows'],
      dtype='object', name='genre')
```

In [173]:

```
plt.figure(figsize=(12,8))
sns.countplot(data=df_3,x='genre',order=gen_order.index)
plt.xticks(rotation=90)
plt.title("Global distribution of Genres")
plt.show()
```


Global distribution of Genres
count
2500
2000
1500
1000
500
0
genre
International Movies
Dramas
Comedies
International TV Shows
Documentaries
Action & Adventure
TV Dramas
Independent Movies
Children & Family Movies
Romantic Movies
TV Comedies
Thrillers
Crime TV Shows
Kids' TV
Docuseries
Music & Musicals
Romantic TV Shows
Horror Movies
Stand-Up Comedy
Reality TV
British TV Shows
Sci-Fi & Fantasy
Sports Movies
Anime Series
Spanish-Language TV Shows
TV Action & Adventure
Korean TV Shows
Classic Movies
LGBTQ Movies
TV Mysteries
Science & Nature TV
TV Sci-Fi & Fantasy
TV Horror
Anime Features
Cult Movies
Teen TV Shows
Faith & Spirituality
TV Thrillers
Movies
Stand-Up Comedy & Talk Shows
Classic & Cult TV
TV Shows

- International movies,Dramas,comedies, International TV shows, Documentaries, Action & Adventure,TV dramas are amongst the topmost genre worldwide.

```
********************************************************************************************************
*******************
```

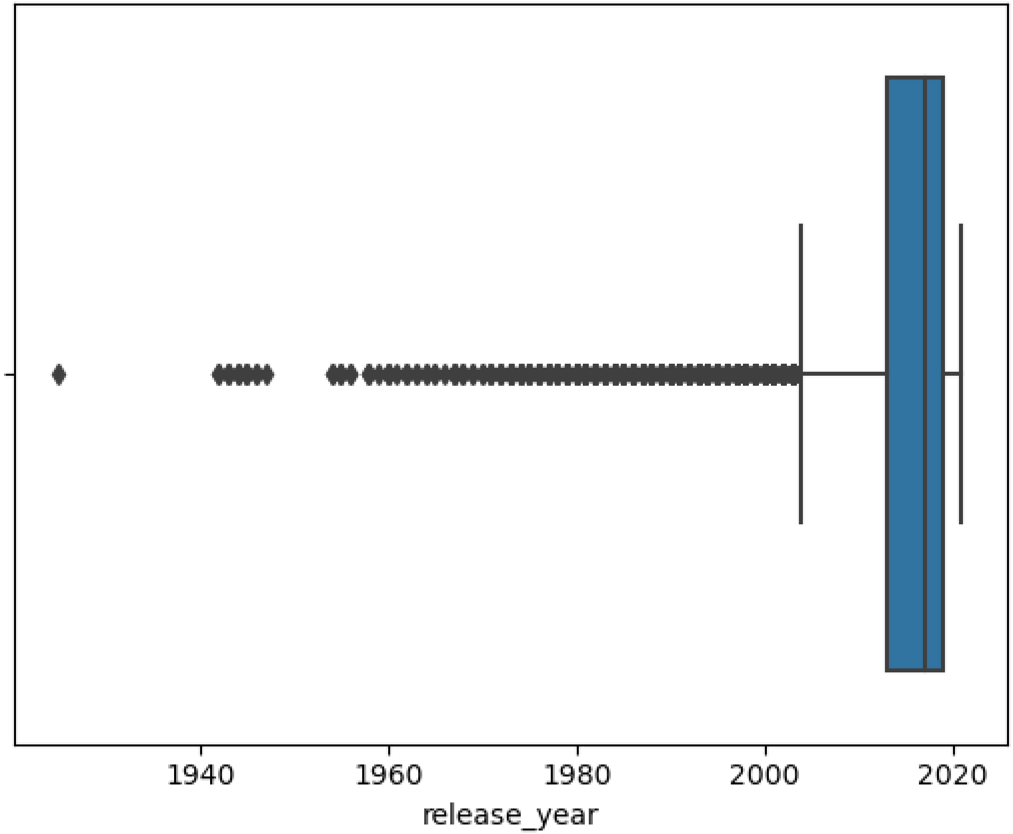
#### Show added and released timeline

In [58]:
```
df_new['date_added']
```

Out[58]:
```
0      2021-09-25
1      2021-09-24
2      2021-09-24
3      2021-09-24
4      2021-09-24
          ...
8802   2019-11-20
8803   2019-07-01
8804   2019-11-01
8805   2020-01-11
8806   2019-03-02
Name: date_added, Length: 8807, dtype: datetime64[ns]
```

In [149]:

```
sns.boxplot(data=df_new,x=df_new['release_year'])
plt.show()
```



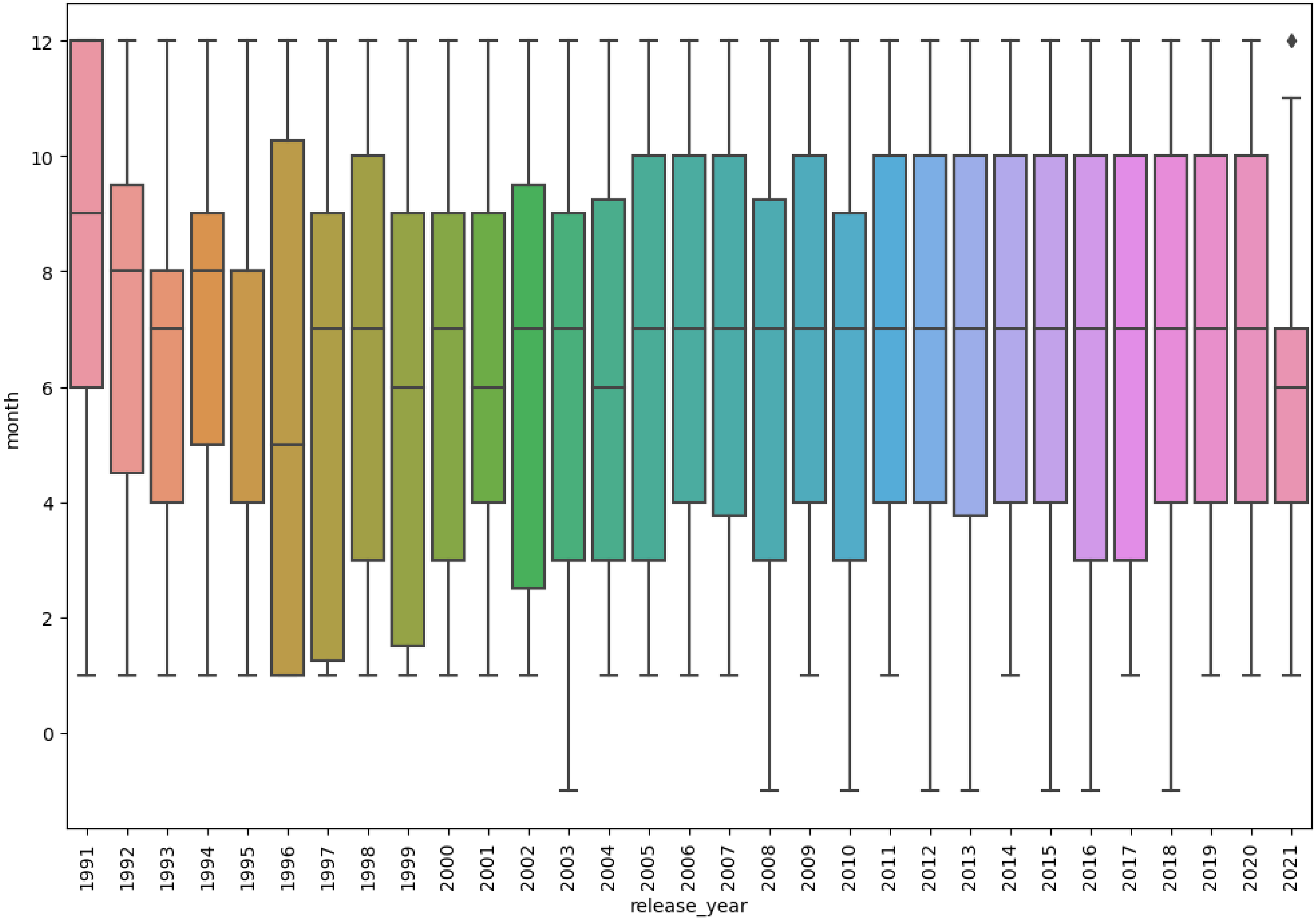
* Significant number of shows added after 2010 to 2022. It signifies there are very less amount of show released before 2012 are available on Netflix.

In [143]:

```
temp_df = df_new.loc[df_new['release_year'] > 1990]
```

In [145]:

```
plt.figure(figsize=(12,8))
sns.boxplot(data=df_new,x=temp_df['release_year'],y=df_new['month'])
plt.xticks(rotation=90)
plt.show()
```



* Also shows added on the netflix are usually after 4th month of year in previous few years. ie. on the second half of the year
* Show release trend shows it is good time to release shows during holidays like thanks giving, Christmas, New year and during Winter , which is Mostly from November to March.

In [61]:

```
grouped = df_new.groupby(pd.Grouper(key='date_added', freq='M')).count().reset_index()
```

In [62]:

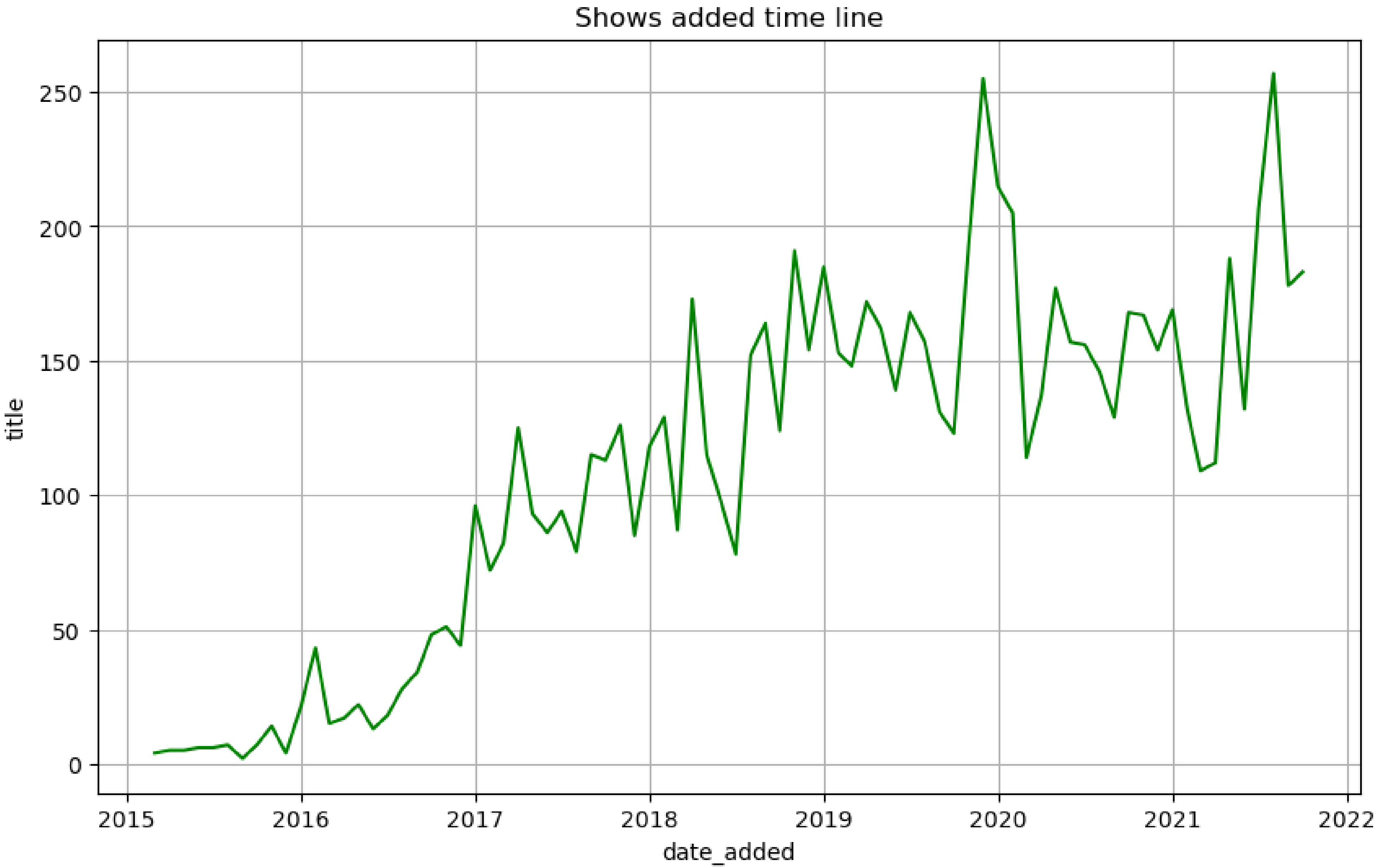
```
grouped.sort_values('date_added')
```

Out[62]:

	date_added	show_id	type	title	director	cast	country	release_year	rating	duration	listed_in	description
0	2008-01-31	1	1	1	1	1	1	1	1	1	1	1
1	2008-02-29	1	1	1	0	0	1	1	1	1	1	1
2	2008-03-31	0	0	0	0	0	0	0	0	0	0	0
3	2008-04-30	0	0	0	0	0	0	0	0	0	0	0
4	2008-05-31	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
160	2021-05-31	132	132	132	95	116	97	132	132	132	132	132
161	2021-06-30	207	207	207	132	190	147	207	207	207	207	207
162	2021-07-31	257	257	257	172	233	162	257	257	257	257	257
163	2021-08-31	178	178	178	126	158	129	178	178	178	178	178
164	2021-09-30	183	183	183	127	164	126	183	183	183	183	183

165 rows × 12 columns

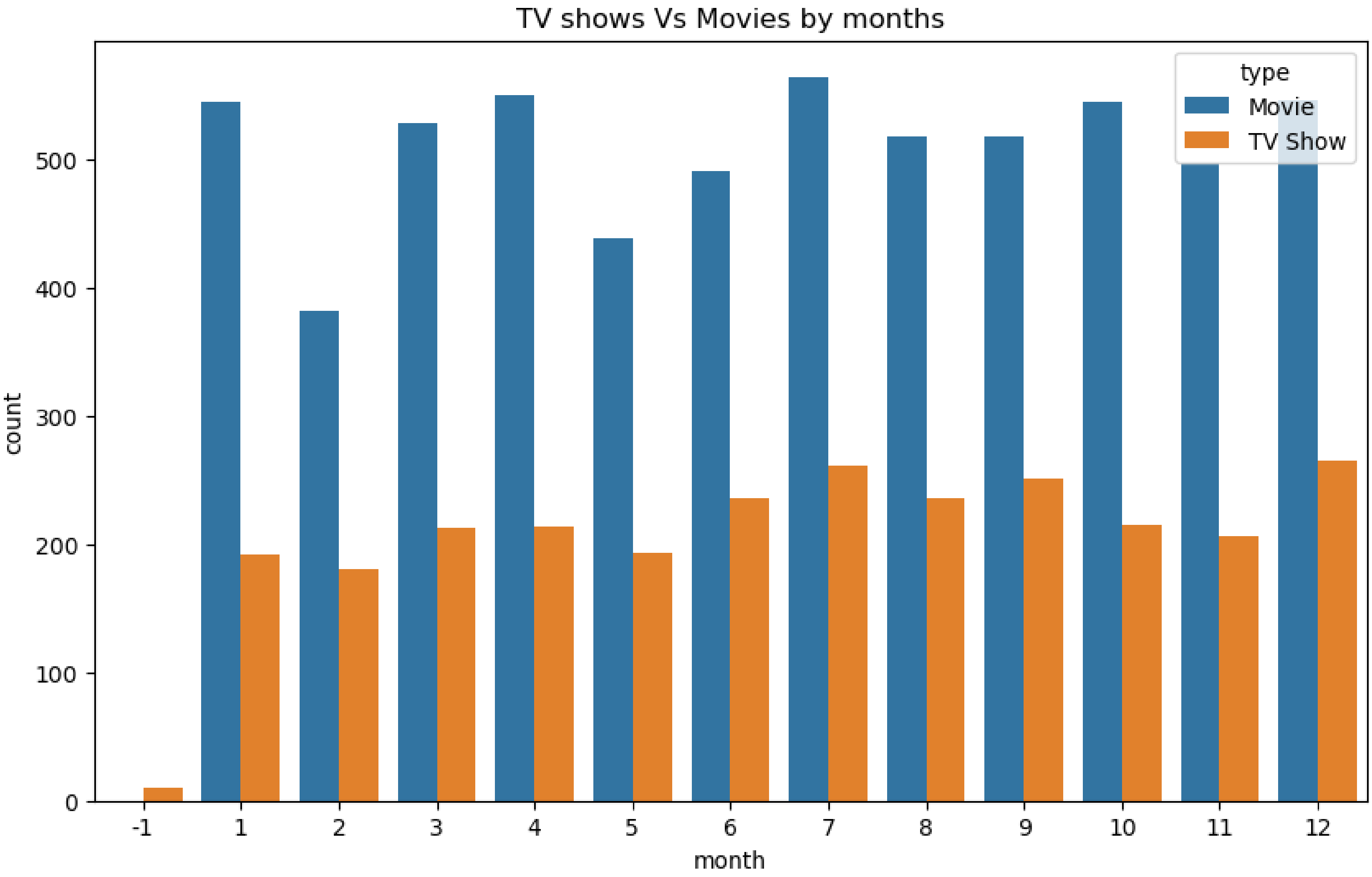
In [63]:
```
plt.figure(figsize=(10,6))
sns.lineplot(data=grouped.tail(80),x='date_added',y="title",color='green')
plt.title("Shows added time line")
plt.grid()
plt.show()
```



```
In [131]: df_new['month']=df_new['date_added'].dt.month.fillna(-1)
```

```
In [133]: df_new['month']=df_new['month'].astype('int64')
```

```
In [157]: plt.figure(figsize=(10,6))
          sns.countplot(x='month',hue='type', data=df_new)
          plt.title("TV shows Vs Movies by months")
          plt.show()
```



```
* TV Shows vs movies added shows that on netflix there are more number of movies added evey month than TV shows and count is greater in second half of the year for both.
```

List of popular genres in particular countries

In [64]:
```
genre_country = df_3.merge(df_4,on='title')
```

In [65]:
```
genre_country.reset_index(drop = True,inplace = True)
```

In [66]:
```
genre_country.groupby('country')['genre'].nunique().sort_values(ascending=False)
```

Out[66]:
```
country
United States     69
nan               61
United Kingdom    60
Canada            59
Australia         50
                  ..
Nicaragua          1
Kazakhstan         1
Botswana           1
Somalia            1
Sudan              1
Name: genre, Length: 124, dtype: int64
```

```
In [67]: genre_country.groupby('country')['genre'].get_group('United States').str.strip().unique()
```

```
Out[67]: array(['Documentaries', 'Dramas', 'Independent Movies',
                'International Movies', 'Comedies', 'TV Comedies', 'TV Dramas',
                'Horror Movies', 'Sci-Fi & Fantasy', 'Thrillers',
                'Action & Adventure', "Kids' TV", 'TV Sci-Fi & Fantasy',
                'Classic Movies', 'Reality TV', 'Crime TV Shows',
                'Children & Family Movies', 'Music & Musicals', 'Sports Movies',
                'TV Action & Adventure', 'Cult Movies', 'Romantic Movies',
                'Faith & Spirituality', 'LGBTQ Movies', 'TV Mysteries',
                'TV Horror', 'Anime Features', 'Movies', 'Docuseries',
                'Stand-Up Comedy', 'Teen TV Shows', 'Classic & Cult TV',
                'International TV Shows', 'Science & Nature TV',
                'Romantic TV Shows', 'Anime Series', 'TV Thrillers',
                'Stand-Up Comedy & Talk Shows', 'TV Shows',
                'Spanish-Language TV Shows', 'British TV Shows', 'Korean TV Shows'],
               dtype=object)
```

```
In [68]: genre_country.loc[genre_country['country']=='United States'].groupby('genre')['title'].count().sort_values(as
```

```
Out[68]: genre
         Dramas                           547
         Documentaries                    488
         Comedies                         406
         Action & Adventure               404
          Independent Movies              375
                                         ...
         Anime Features                     3
          Korean TV Shows                   3
          Stand-Up Comedy & Talk Shows      2
         Romantic Movies                    2
         Spanish-Language TV Shows          2
         Name: title, Length: 69, dtype: int64
```

In [69]:

```
genre_country.loc[genre_country['country']=='India'].groupby('genre')['title'].count().sort_values(ascending=
```

Out[69]:

```
genre
 International Movies            826
Dramas                           415
Comedies                         271
 Dramas                          247
 Independent Movies              166
Action & Adventure               137
 Romantic Movies                 120
 Music & Musicals                 96
 Thrillers                        91
 Comedies                         52
International TV Shows            48
International Movies              38
Horror Movies                     30
 TV Dramas                        28
Documentaries                     27
 TV Comedies                      26
Children & Family Movies          26
 International TV Shows           18
 Sports Movies                    17
 Romantic TV Shows                12
 Sci-Fi & Fantasy                 12
Kids' TV                          11
Crime TV Shows                     9
Classic Movies                     9
Docuseries                         8
 TV Horror                         7
Stand-Up Comedy                    6
 TV Action & Adventure             5
 Horror Movies                     5
 Reality TV                        4
 Faith & Spirituality              4
 Cult Movies                       4
 TV Thrillers                      3
British TV Shows                   3
 TV Sci-Fi & Fantasy               3
 Stand-Up Comedy & Talk Shows      3
TV Shows                           3
Reality TV                         2
 Classic Movies                    2
 TV Mysteries                      2
 LGBTQ Movies                      2
Independent Movies                 1
```

```
Cult Movies                          1
 Teen TV Shows                       1
 Kids' TV                            1
 Docuseries                          1
Thrillers                            1
Name: title, dtype: int64
```

#### Top Actor and director pairs acted in more number of shows

In [88]:
```
act_direct_pair = pd.merge(df_1,df_2)
act_direct_pair
```

Out[88]:

	title	directors	cast
0	Dick Johnson Is Dead	Kirsten Johnson	nan
1	Blood & Water	nan	Ama Qamata
2	Blood & Water	nan	Khosi Ngema
3	Blood & Water	nan	Gail Mabalane
4	Blood & Water	nan	Thabang Molaba
...	...	...	...
70807	Zubaan	Mozez Singh	Manish Chaudhary
70808	Zubaan	Mozez Singh	Meghna Malik
70809	Zubaan	Mozez Singh	Malkeet Rauni
70810	Zubaan	Mozez Singh	Anita Shabdish
70811	Zubaan	Mozez Singh	Chittaranjan Tripathy

70812 rows × 3 columns

```
In [71]: act_direct_pair = pd.DataFrame(act_direct_pair.groupby(['cast','directors'])["title"].nunique().sort_values(as
```

```
In [72]: act_direct_pair.head(100)
```

```
Out[72]:
                                             title
                    cast          directors
                     nan                nan    352
        Takahiro Sakurai                nan     23
           Julie Tejwani      Rajiv Chilaka     19
             Rajesh Kava      Rajiv Chilaka     19
            Rupa Bhimani      Rajiv Chilaka     18
                     ...                ...    ...
          Satsuki Yukino  Toshiya Shinohara      7
               Sathyaraj     S.S. Rajamouli      7
        Tamannaah Bhatia     S.S. Rajamouli      7
         Fumiko Orikasa                nan      7
        Chinatsu Akasaki                nan      7

100 rows × 1 columns
```

```
In [73]: act_direct_pair.reset_index(inplace=True)
```

In [74]:

```
act_direct_pair
```

Out[74]:

	cast	directors	title
0	nan	nan	352
1	Takahiro Sakurai	nan	23
2	Julie Tejwani	Rajiv Chilaka	19
3	Rajesh Kava	Rajiv Chilaka	19
4	Rupa Bhimani	Rajiv Chilaka	18
...	...	...	...
63805	Jason Sudeikis	Will Graham	1
63806	Jason Sudeikis	Andy Tennant	1
63807	Jason Sudeikis	Garry Marshall	1
63808	Jason Sudeikis	Kevin R. Adams	1
63809	Ṣọpẹ́ Dìrísù	Remi Weekes	1

63810 rows × 3 columns

In [75]:

```
act_direct_pair.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63810 entries, 0 to 63809
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   cast       63810 non-null  object
 1   directors  63810 non-null  object
 2   title      63810 non-null  int64 
dtypes: int64(1), object(2)
memory usage: 1.5+ MB
```

In [76]:

```
act_direct_pair.replace('nan', np.nan, inplace=True)
```

In [77]:
```
act_direct_pair.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63810 entries, 0 to 63809
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   cast       63256 non-null  object
 1   directors  49212 non-null  object
 2   title      63810 non-null  int64 
dtypes: int64(1), object(2)
memory usage: 1.5+ MB
```

In [78]:
```
act_direct_pair = act_direct_pair.set_index('title').dropna()
```

In [79]:
```
temp = pd.merge(df_3,df_4)
```

In [87]:
```
temp
```

Out[87]:

	title	genre	country
0	Dick Johnson Is Dead	Documentaries	United States
1	Blood & Water	International TV Shows	South Africa
2	Blood & Water	TV Dramas	South Africa
3	Blood & Water	TV Mysteries	South Africa
4	Ganglands	Crime TV Shows	nan
...	...	...	...
23759	Zoom	Children & Family Movies	United States
23760	Zoom	Comedies	United States
23761	Zubaan	Dramas	India
23762	Zubaan	International Movies	India
23763	Zubaan	Music & Musicals	India

23764 rows × 3 columns

In [89]:
```
merged_df = pd.merge(act_direct_pair,temp)
merged_df.replace('nan', np.nan, inplace=True)
merged_df.dropna(inplace=True)
```

In [91]:
```
merged_df.rename({'index':'index'},axis=1)
```

Out[91]:

	title	directors	cast	genre	country
179	Sankofa	Haile Gerima	Kofi Ghanaba	Dramas	United States
180	Sankofa	Haile Gerima	Kofi Ghanaba	Dramas	Ghana
181	Sankofa	Haile Gerima	Kofi Ghanaba	Dramas	Burkina Faso
182	Sankofa	Haile Gerima	Kofi Ghanaba	Dramas	United Kingdom
183	Sankofa	Haile Gerima	Kofi Ghanaba	Dramas	Germany
...	...	...	...	...	...
202060	Zubaan	Mozez Singh	Anita Shabdish	International Movies	India
202061	Zubaan	Mozez Singh	Anita Shabdish	Music & Musicals	India
202062	Zubaan	Mozez Singh	Chittaranjan Tripathy	Dramas	India
202063	Zubaan	Mozez Singh	Chittaranjan Tripathy	International Movies	India
202064	Zubaan	Mozez Singh	Chittaranjan Tripathy	Music & Musicals	India

143173 rows × 5 columns

#### Most famous actor-director pair overall

```
In [92]: merged_df.groupby(['cast','directors'])["title"].nunique().sort_values(ascending=False).head(50)
```

```
Out[92]: cast                 directors
         Kappei Yamaguchi     Toshiya Shinohara      7
          Kumiko Watanabe     Toshiya Shinohara      7
          Satsuki Yukino      Toshiya Shinohara      7
          Houko Kuwashima     Toshiya Shinohara      7
          Koji Tsujitani      Toshiya Shinohara      7
          Anupam Kher         David Dhawan           6
          Joross Gamboa       Cathy Garcia-Molina    6
         Yılmaz Erdoğan       Yılmaz Erdoğan         6
          John Paul Tremblay   Robb Wells            5
          Wille Lindberg      Thierry Donard         5
          John Paul Tremblay   John Paul Tremblay    5
          Pat Roach            John Paul Tremblay    5
          Alok Nath           Sooraj R. Barjatya     5
          Ken Narita          Toshiya Shinohara      5
         Ahmed Helmy          Khaled Marei           5
         John Lloyd Cruz      Cathy Garcia-Molina    5
         Mike Smith            John Paul Tremblay    5
                               Robb Wells            5
         Donnie Yen           Wilson Yip             5
          Noriko Hidaka       Toshiya Shinohara      5
          John Dunsworth      Mike Clattenburg       5
          Pat Roach            Robb Wells            5
          Robb Wells           Robb Wells            5
          Alexa PenaVega      Robert Rodriguez       5
         Kevin Hart           Leslie Small           5
          Robb Wells           John Paul Tremblay    5
          Salih Kalyon        Hakan Algül            5
          Hiroomi Tosaka      Shigeaki Kubo          4
          Mahmoud El Meleigy  Youssef Chahine        4
          Sho Aoyagi          Shigeaki Kubo          4
         Robert De Niro       Martin Scorsese        4
         Junko Takeuchi       Masahiko Murata        4
         Jeff Dunham          Michael Simon          4
         David Attenborough   Alastair Fothergill    4
          Akira               Shigeaki Kubo          4
          Barrie Dunn         Mike Clattenburg       4
          Harvey Keitel       Martin Scorsese        4
          Keiji Kuroki        Shigeaki Kubo          4
          Patrick Roach       Mike Clattenburg       4
          John Dunsworth       John Paul Tremblay    4
         Clint Eastwood       Clint Eastwood         4
          Lucy Decoutere      Mike Clattenburg       4
```

```
 Kate Higgins        William Lau          4
 Cezmi Baskın        Yılmaz Erdoğan       4
Salman Khan          Sooraj R. Barjatya   4
 John Dunsworth       Robb Wells          4
Adam Sandler         Steve Brill          4
 Mohnish Bahl        Sooraj R. Barjatya   4
Omoni Oboli          Omoni Oboli          4
Harrison Ford        Steven Spielberg     4
Name: title, dtype: int64
```

#### Actors Acted in most different Movies

In [93]:

```
merged_df.groupby('cast')["title"].nunique().sort_values(ascending=False).head(50)
```

```
Out[93]: cast
          Anupam Kher             38
          Om Puri                 27
          Boman Irani             25
         Shah Rukh Khan           25
          Paresh Rawal            25
         Akshay Kumar             23
          Kareena Kapoor          20
          Naseeruddin Shah        20
         Adam Sandler             20
         Amitabh Bachchan         20
          Yashpal Sharma          17
          Asrani                  17
          Amrish Puri             16
          Tinnu Anand             16
          Gulshan Grover          16
         Ajay Devgn               16
          Vijay Raaz              16
          Rajesh Sharma           16
          Manoj Joshi             16
         Nicolas Cage             15
          John Cleese             15
          Rajpal Yadav            15
          Shakti Kapoor           14
          Alfred Molina           14
          Kay Kay Menon           14
          Kulbhushan Kharbanda    14
          Sharat Saxena           14
          Molly Shannon           14
         Aamir Khan               14
          Priyanka Chopra         13
          Mohnish Bahl            13
          Manoj Bajpayee          13
         Ahmed Helmy              13
          Michael Peña            13
          Fred Tatasciore         13
          Maya Rudolph            13
          Willem Dafoe            13
          Radhika Apte            13
          Samuel L. Jackson       13
          Erin Fitzgerald         13
          Hassan Hosny            13
          Danny Trejo             13
```

```
 Ben Kingsley           13
 Sanjay Mishra          13
 Vipin Sharma           13
Anil Kapoor             13
 Fred Armisen           13
 Laurence Fishburne     13
 Brendan Gleeson        13
 Katrina Kaif           13
Name: title, dtype: int64
```

#### Most appeared Actor-Director Pairs in different countries

In [94]:
```
merged_df.groupby(['country','cast','directors'])['title'].nunique().sort_values(ascending=False).head(30)
```

Out[94]:
```
country        cast                 directors
Japan          Kappei Yamaguchi     Toshiya Shinohara      7
                Houko Kuwashima     Toshiya Shinohara      7
                Satsuki Yukino      Toshiya Shinohara      7
                Koji Tsujitani      Toshiya Shinohara      7
                Kumiko Watanabe     Toshiya Shinohara      7
Philippines     Joross Gamboa       Cathy Garcia-Molina    6
India           Anupam Kher         David Dhawan           6
Turkey         Yılmaz Erdoğan       Yılmaz Erdoğan         6
France          Wille Lindberg      Thierry Donard         5
Canada         Mike Smith            Robb Wells            5
                                     John Paul Tremblay    5
India           Alok Nath           Sooraj R. Barjatya     5
United States  Kevin Hart           Leslie Small           5
Japan           Noriko Hidaka       Toshiya Shinohara      5
Canada          John Paul Tremblay   Robb Wells            5
                John Dunsworth      Mike Clattenburg       5
                John Paul Tremblay   John Paul Tremblay    5
United States   Alexa PenaVega      Robert Rodriguez       5
Japan           Ken Narita          Toshiya Shinohara      5
Philippines    John Lloyd Cruz      Cathy Garcia-Molina    5
Egypt          Ahmed Helmy          Khaled Marei           5
Canada          Robb Wells           Robb Wells            5
Hong Kong      Donnie Yen           Wilson Yip             5
Canada          Robb Wells           John Paul Tremblay    5
China          Donnie Yen           Wilson Yip             5
Turkey          Salih Kalyon        Hakan Algül            5
Canada          Pat Roach            John Paul Tremblay    5
                                     Robb Wells            5
Turkey          Cezmi Baskın        Yılmaz Erdoğan         4
France          Jesse Richman       Thierry Donard         4
Name: title, dtype: int64
```

* TWo Most appeared actor-director pair in USA is Kevin Hart-Leslie Small & Alexa PenaVega-Robert Rodriguez, in India it is Anupam Kher-David Dhawan & Alok Nath Sooraj Barjatya

## Business Insights

* There are total 8807 number of shows available with 5121 directors worldwide involved in filming and 39297 actors involved as per the data available on the platform.
* International movies, dramas, comedies, International TV shows, Documentaries, Action & Adventure, TV dramas are amongst the topmost genre available worldwide.
* There is wide variety of Movies already available like International, Independent, children & Family, Romantic, Horror & Thriller and less movies in anime, cult and sci-fy compared to them.
* There are a greater number of movies than TV shows available on the Netflix.
* Also, there is huge number of viewers from USA already where most of the shows are released. India & UK are next two countries where most shows are released.
* Shows on the Netflix and their release changed slowly for first 10 to 15 years and then it sharply rose from 2014 to 2018 reaches max from where it again dipped for next 3 years.
* Total there are a greater number of unique movies (6131) than the TV shows (2676)
* Trend for Show release shows, it is good time to release shows during holidays like thanksgiving, Christmas, New year and during Winter, which is Mostly from November to March.
* Most genre present in the USA are Dramas, comedies, documentaries, Action & adventure.
* Anime, British TV shows, classic cult, TV shows, Korean TV shows are least in count.
* Significant number of shows added after 2010 to 2022. It signifies there are very less amount of show released before 2012 are available on Netflix.
* Also shows added on the Netflix are usually after 4th month of year in previous few years. i.e., on the second half of the year
* Show release trend shows it is good time to release shows during holidays like thanksgiving, Christmas, New year and during Winter, which is Mostly from November to March.
* Two most appeared actor-director pair in USA is Kevin Hart-Leslie Small & Alexa PenaVega-Robert Rodriguez, in India it is Anupam Kher-David Dhawan & Alok Nath Sooraj Barjatya
* Rating wise with TV-MA is topmost amongst both the type of shows

## Business Recommendations

* Increase focus on international movies and TV shows: Since there is already a large number of movies available on the platform, Netflix could consider increasing its investment in acquiring and producing international TV shows to attract a wider audience.
* Invest in genres with high demand: Based on the data provided, Netflix could focus on producing and acquiring more content in the most popular genres, such as dramas, comedies, documentaries, and action & adventure.
* Release shows during holidays and winter months: Given the trend that shows released during holidays and winter months tend to perform well, Netflix could consider strategically timing the release of its original content to coincide with these periods.

```
* Strengthen presence in India and UK: Since India and UK are the next two countries with the most shows
released on Netflix after the USA, Netflix could consider increasing its investment in these markets to
strengthen its presence there and attract more viewers.
* Expand content before 2012: Since there are very few shows released before 2012 available on Netflix, the
company could consider acquiring more content from earlier years to expand its library and appeal to viewers
interested in older shows.
* Focus on TV-MA shows: Given that TV-MA shows are the top-rated among both types of shows, Netflix could
invest in producing and acquiring more mature content to appeal to its audience.
* Leverage popular actor-director pairs: Since Kevin Hart-Leslie Small and Alexa PenaVega-Robert Rodriguez
are the most popular actor-director pairs in the USA, and Anupam Kher-David Dhawan and Alok Nath Sooraj
Barjatya are the most popular in India, Netflix could consider partnering with these pairs to produce
exclusive content for the platform.
```