EXPLORATORY DATA ANALYSIS OF NETFLIX

Import libraries

In [1]:

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

Read dataset

In [2]:

```
raw_df=pd.read_csv('netflix_dataset.csv')
print(len(raw_df))
raw_df.head()
```

```
8807
```

Out[2]:

	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 [3]:

```
raw_df['show_id'].tail(1)
```

Out[3]:

```
8806    s8807
Name: show_id, dtype: object
```

In [4]:

```
raw_df['director'].unique().tolist() ## one movie can have more than 1 director
```

Out[4]:

```
['Kirsten Johnson',
 nan,
 'Julien Leclercq',
 'Mike Flanagan',
 'Robert Cullen, José Luis Ucha',
 'Haile Gerima',
 'Andy Devonshire',
 'Theodore Melfi',
 'Kongkiat Komesiri',
 'Christian Schwochow',
 'Bruno Garotti',
 'Pedro de Echave García, Pablo Azorín Williams',
 'Adam Salky',
 'Olivier Megaton',
 'K.S. Ravikumar',
 'Alex Woo, Stanley Moore',
 'S. Shankar',
```

In [5]:

```
raw_df['country'].unique().tolist() ## one movie can have more than 1 country maybe publish
```

Out[5]:

```
['United States',
 'South Africa',
 nan,
 'India',
 'United States, Ghana, Burkina Faso, United Kingdom, Germany, Ethiopia',
 'United Kingdom',
 'Germany, Czech Republic',
 'Mexico',
 'Turkey',
 'Australia',
 'United States, India, France',
 'Finland',
 'China, Canada, United States',
 'South Africa, United States, Japan',
 'Nigeria',
 'Japan',
 'Spain, United States',
 'France',
```

In [6]:

```
raw_df[raw_df['country']=='United States, Ghana, Burkina Faso, United Kingdom, Germany, Eth
```

Out[6]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	d
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	1

In [7]:

```
raw_df['duration'].unique().tolist()
```

```
 '119 min',
 '143 min',
 '114 min',
 '118 min',
 '108 min',
 '63 min',
 '121 min',
 '142 min',
 '154 min',
 '120 min',
 '82 min',
 '109 min',
 '101 min',
 '86 min',
 '229 min',
 '76 min',
 '89 min',
 '156 min',
 '112 min',
```

In [8]:

```
raw_df['type'].unique().tolist()
```

Out[8]:

```
['Movie', 'TV Show']
```

In [9]:

```
# nodups=raw_df.drop_duplicates(['title','director'])
# df=nodups.groupby('title',as_index=False)['director'].count()
# df[df['director']>1]
```

In [10]:

```
raw_df.columns
```

Out[10]:

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

Preprocessing of data

Dealing with comma seperated compressed data in cols-'cast','listed_in','director','country'

In [11]:

```
## CAST-SPLIT
cast_b4stack=raw_df[['show_id','title','cast']]
index_lt=[i for i in range(50)]
cast_b4stack[index_lt]=cast_b4stack['cast'].str.split(',',expand=True)
# cast_b4stack[['show_id','title']+index_lt]

cast_afterstack=cast_b4stack[['show_id','title']+index_lt].melt(id_vars=['show_id','title']
        var_name="id",
        value_name="cast_name")
cast_afterstack

cast_afterstack.dropna(axis=0,inplace=True)
cast_afterstack=cast_afterstack[['show_id','title','cast_name']]
cast_afterstack
```

```
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\3179884693.py:4: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  cast_b4stack[index_lt]=cast_b4stack['cast'].str.split(',',expand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\3179884693.py:4: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  cast_b4stack[index_lt]=cast_b4stack['cast'].str.split(',',expand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\3179884693.py:4: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  cast_b4stack[index_lt]=cast_b4stack['cast'].str.split(',',expand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\3179884693.py:4: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  cast_b4stack[index_lt]=cast_b4stack['cast'].str.split(',',expand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\3179884693.py:4: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  cast_b4stack[index_lt]=cast_b4stack['cast'].str.split(',',expand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\3179884693.py:4: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  cast_b4stack[index_lt]=cast_b4stack['cast'].str.split(',',expand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\3179884693.py:4: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  cast_b4stack[index_lt]=cast_b4stack['cast'].str.split(',',expand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\3179884693.py:4: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  cast_b4stack[index_lt]=cast_b4stack['cast'].str.split(',',expand=True)
```

Out[11]:

	show_id	title	cast_name
1	s2	Blood & Water	Ama Qamata
2	s3	Ganglands	Sami Bouajila
4	s5	Kota Factory	Mayur More
5	s6	Midnight Mass	Kate Siegel
6	s7	My Little Pony: A New Generation	Vanessa Hudgens
...	...	...	...
417703	s3775	Black Mirror	Jon Hamm
424590	s1855	Social Distance	Ayize Ma'at
426510	s3775	Black Mirror	Oona Chaplin
433397	s1855	Social Distance	Lovie Simone
435317	s3775	Black Mirror	Rafe Spall

64126 rows × 3 columns

In [12]:

```
## Director-SPLIT
director_b4stack=raw_df[['show_id','title','director']]
index_lt=[i for i in range(len(director_b4stack['director'].str.split(',',expand=True).colu
print(index_lt)
director_b4stack[index_lt]=director_b4stack['director'].str.split(',',expand=True)
director_b4stack[['show_id','title']+index_lt]

director_afterstack=director_b4stack[['show_id','title']+index_lt].melt(id_vars=['show_id',
        var_name="id",
        value_name="director_name")
director_afterstack

director_afterstack.dropna(axis=0,inplace=True)
director_afterstack=director_afterstack[['show_id','title','director_name']]

director_afterstack
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\4094328131.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  director_b4stack[index_lt]=director_b4stack['director'].str.split(',',e
xpand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\4094328131.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  director_b4stack[index_lt]=director_b4stack['director'].str.split(',',e
xpand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\4094328131.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  director_b4stack[index_lt]=director_b4stack['director'].str.split(',',e
xpand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\4094328131.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
```

```
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  director_b4stack[index_lt]=director_b4stack['director'].str.split(',',e
xpand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\4094328131.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  director_b4stack[index_lt]=director_b4stack['director'].str.split(',',e
xpand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\4094328131.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  director_b4stack[index_lt]=director_b4stack['director'].str.split(',',e
xpand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\4094328131.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  director_b4stack[index_lt]=director_b4stack['director'].str.split(',',e
xpand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\4094328131.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  director_b4stack[index_lt]=director_b4stack['director'].str.split(',',e
xpand=True)
```

Out[12]:

	show_id	title	director_name
0	s1	Dick Johnson Is Dead	Kirsten Johnson
2	s3	Ganglands	Julien Leclercq
5	s6	Midnight Mass	Mike Flanagan
6	s7	My Little Pony: A New Generation	Robert Cullen
7	s8	Sankofa	Haile Gerima

	show_id	title	director_name
...	...	...	...
95585	s7516	Movie 43	Rusty Cundieff
102764	s5888	Walt Disney Animation Studios Short Films Coll...	Mike Gabriel
103787	s6911	HALO Legends	Hiroshi Yamazaki
104392	s7516	Movie 43	James Gunn
111571	s5888	Walt Disney Animation Studios Short Films Coll...	Mark Henn

6978 rows × 3 columns

In [13]:

```
## listed_in-SPLIT
listed_in_b4stack=raw_df[['show_id','title','listed_in']]
index_lt=[i for i in range(len(listed_in_b4stack['listed_in'].str.split(',',expand=True).co
print(index_lt)
listed_in_b4stack[index_lt]=listed_in_b4stack['listed_in'].str.split(',',expand=True)
listed_in_b4stack[['show_id','title']+index_lt]

listed_in_afterstack=listed_in_b4stack[['show_id','title']+index_lt].melt(id_vars=['show_id
        var_name="id",
        value_name="listed_in_name")
listed_in_afterstack

listed_in_afterstack.dropna(axis=0,inplace=True)
listed_in_afterstack=listed_in_afterstack[['show_id','title','listed_in_name']]
listed_in_afterstack['listed_in_name']=listed_in_afterstack['listed_in_name'].str.strip()

listed_in_afterstack
```

```
[0, 1, 2]

C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\640419612.py:5: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  listed_in_b4stack[index_lt]=listed_in_b4stack['listed_in'].str.split(',',e
xpand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\640419612.py:5: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  listed_in_b4stack[index_lt]=listed_in_b4stack['listed_in'].str.split(',',e
xpand=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\640419612.py:5: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  listed_in_b4stack[index_lt]=listed_in_b4stack['listed_in'].str.split(',',e
xpand=True)
```

Out[13]:

show_id	title	listed_in_name

	show_id	title	listed_in_name
0	s1	Dick Johnson Is Dead	Documentaries
1	s2	Blood & Water	International TV Shows
2	s3	Ganglands	Crime TV Shows
3	s4	Jailbirds New Orleans	Docuseries
4	s5	Kota Factory	International TV Shows
...	...	...	...
26414	s8801	Zindagi Gulzar Hai	TV Dramas
26415	s8802	Zinzana	Thrillers
26416	s8803	Zodiac	Thrillers
26417	s8804	Zombie Dumb	TV Comedies
26420	s8807	Zubaan	Music & Musicals

In [14]:

```
## country-SPLIT
country_b4stack=raw_df[['show_id','title','country']]
index_lt=[i for i in range(len(country_b4stack['country'].str.split(',',expand=True).column
print(index_lt)
country_b4stack[index_lt]=country_b4stack['country'].str.split(',',expand=True)
country_b4stack[['show_id','title']+index_lt]

country_afterstack=country_b4stack[['show_id','title']+index_lt].melt(id_vars=['show_id','t
        var_name="id",
        value_name="country_name")
country_afterstack

country_afterstack.dropna(axis=0,inplace=True)
country_afterstack=country_afterstack[['show_id','title','country_name']]
country_afterstack['country_name']=country_afterstack['country_name'].str.strip()
country_afterstack
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\2581447538.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  country_b4stack[index_lt]=country_b4stack['country'].str.split(',',expa
nd=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\2581447538.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  country_b4stack[index_lt]=country_b4stack['country'].str.split(',',expa
nd=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\2581447538.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  country_b4stack[index_lt]=country_b4stack['country'].str.split(',',expa
nd=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\2581447538.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
```

```
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  country_b4stack[index_lt]=country_b4stack['country'].str.split(',',expa
nd=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\2581447538.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  country_b4stack[index_lt]=country_b4stack['country'].str.split(',',expa
nd=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\2581447538.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  country_b4stack[index_lt]=country_b4stack['country'].str.split(',',expa
nd=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\2581447538.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  country_b4stack[index_lt]=country_b4stack['country'].str.split(',',expa
nd=True)
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\2581447538.py:5: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  country_b4stack[index_lt]=country_b4stack['country'].str.split(',',expa
nd=True)
```

Out[14]:

	show_id	title	country_name
0	s1	Dick Johnson Is Dead	United States
1	s2	Blood & Water	South Africa
4	s5	Kota Factory	India
7	s8	Sankofa	United States
8	s9	The Great British Baking Show	United Kingdom

	show_id	title	country_name
...	...	...	...
78859	s8404	The Look of Silence	Germany
85496	s6234	Barbecue	Sweden
87666	s8404	The Look of Silence	Netherlands
94303	s6234	Barbecue	United States
103110	s6234	Barbecue	Uruguay

Merge each with main dataset

Main df

In [15]:

```
main_rawdf=raw_df[['show_id', 'type', 'title', 'date_added',
       'release_year', 'rating', 'duration']]
main_rawdf.head()
```

Out[15]:

	show_id	type	title	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons
2	s3	TV Show	Ganglands	September 24, 2021	2021	TV-MA	1 Season
3	s4	TV Show	Jailbirds New Orleans	September 24, 2021	2021	TV-MA	1 Season
4	s5	TV Show	Kota Factory	September 24, 2021	2021	TV-MA	2 Seasons

In [16]:

```
data_frames=[main_rawdf,country_afterstack,listed_in_afterstack,director_afterstack,cast_af
df_merged = reduce(lambda  left,right: pd.merge(left,right,on=['show_id','title'],
                                            how='left'), data_frames)
df_merged
```

Out[16]:

	show_id	type	title	date_added	release_year	rating	duration	country_name	lis
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	United States	[
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	South Africa	Ir
2	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	South Africa	Ir
3	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	South Africa	Ir
4	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	South Africa	Ir
...	...	...	...	...	...	...	...	...	
202060	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	
202061	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	
202062	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	
202063	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	
202064	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	

202065 rows × 11 columns

Dealing with missing data

In [17]:

```
df_merged.isna().sum()
```

Out[17]:

```
show_id              0
type                 0
title                0
date_added         158
release_year         0
rating              67
duration             3
country_name     11897
listed_in_name       0
director_name    50643
cast_name         2149
dtype: int64
```

check rows where rating is null

In [18]:

```
df_merged['rating'].unique()
```

Out[18]:

```
array(['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', nan,
       'TV-Y7-FV', 'UR'], dtype=object)
```

In [19]:

```
df_merged[df_merged['rating'].isna()]['title'].unique()
```

Out[19]:

```
array(['13TH: A Conversation with Oprah Winfrey & Ava DuVernay',
       'Gargantia on the Verdurous Planet', 'Little Lunch',
       'My Honor Was Loyalty'], dtype=object)
```

In [20]:

```
df_merged['rating'].fillna(value='Unknown', inplace=True)
df_merged['cast_name'] = df_merged['cast_name'].fillna('Unknown')

df_merged['rating'].unique()
```

Out[20]:

```
array(['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',
       'Unknown', 'TV-Y7-FV', 'UR'], dtype=object)
```

In [21]:

```
df_merged['country_name'].replace('', np.nan, inplace=True)
df_merged['country_name'] = df_merged['country_name'].fillna(df_merged['country_name'].mode
df_merged['date_added'] = df_merged['date_added'].fillna(df_merged['date_added'].mode()[0])
```

For director missing values, Replace with mode by groupping by country and finding max count

In [22]:

```
temp=df_merged.drop_duplicates(['country_name','director_name','title'])
grpped_cnty2=temp.groupby(['country_name'],as_index=False)['director_name'].value_counts()
grpped_cnty2['max_cntry'] = grpped_cnty2.groupby(['country_name'])['count']\
                        .transform('max')  ##like_windowfunc
grpped_cnty=grpped_cnty2[(grpped_cnty2['count']==grpped_cnty2['max_cntry']) & (grpped_cnty2
grpped_cnty['country_name'].replace('', np.nan, inplace=True)
grpped_cnty.dropna(subset=['country_name'], inplace=True)
grpped_cnty.rename(columns={'director_name':'director_mode'},inplace=True)
grpped_cnty.head()
```

Out[22]:

	country_name	director_mode	count	max_cntry
0	Afghanistan	Pieter-Jan De Pue	1	1
1	Albania	Antonio Morabito	1	1
2	Algeria	Youssef Chahine	1	1
5	Angola	Chris Roland	1	1
7	Argentina	Raúl Campos	5	5

In [23]:

```
df_merged_cp=df_merged.copy()
df_merged=df_merged_cp.merge(grpped_cnty[['country_name','director_mode']],on='country_name
df_merged['director_name'].fillna(df_merged['director_mode'],inplace=True)
df_merged.drop(['director_mode'],axis=1, inplace=True)
df_merged
```

Out[23]:

	show_id	type	title	date_added	release_year	rating	duration	country_name	lis
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	United States	[
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	South Africa	Ir
2	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	South Africa	Ir
3	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	South Africa	Ir
4	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	South Africa	Ir
...	...	...	...	...	...	...	...	...	
202060	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	
202061	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	
202062	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	
202063	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	
202064	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min	India	

202065 rows × 11 columns

Clean 'duration' col

In [24]:

```
df_merged[df_merged['type']=='TV Show']['duration'].unique() #No_of_seasons
```

Out[24]:

```
array(['2 Seasons', '1 Season', '9 Seasons', '4 Seasons', '5 Seasons',
       '3 Seasons', '6 Seasons', '7 Seasons', '10 Seasons', '8 Seasons',
       '17 Seasons', '13 Seasons', '15 Seasons', '12 Seasons',
       '11 Seasons'], dtype=object)
```

In [25]:

```
# raw_df[raw_df['type']=='Movie']['duration'].unique() ##all given in min
```

In [26]:

```
df_merged[df_merged['duration'].isna()]
```

Out[26]:

	show_id	type	title	date_added	release_year	rating	duration	country_name	li
126582	s5542	Movie	Louis C.K. 2017	April 4, 2017	2017	74 min	NaN	United States	
131648	s5795	Movie	Louis C.K.: Hilarious	September 16, 2016	2010	84 min	NaN	United States	
131782	s5814	Movie	Louis C.K.: Live at the Comedy Store	August 15, 2016	2015	66 min	NaN	United States	

In [27]:

```
df_merged['duration'].fillna(df_merged['rating'],inplace=True)
df_merged[df_merged['show_id']=='s5542']
```

Out[27]:

	show_id	type	title	date_added	release_year	rating	duration	country_name	liste
126582	s5542	Movie	Louis C.K. 2017	April 4, 2017	2017	74 min	74 min	United States	

In [28]:

```
## removed seasons or mins written to convert into number
df_merged['duration'] = df_merged['duration'].str.extract('(\d+)', expand=False)
df_merged['duration']=df_merged['duration'].astype(int)
df_merged['duration'].unique()
```

Out[28]:

```
array([ 90,   2,   1,  91, 125,   9, 104, 127,   4,  67,  94,   5, 161,
        61, 166, 147, 103,  97, 106, 111,   3, 110, 105,  96, 124, 116,
        98,  23, 115, 122,  99,  88, 100,   6, 102,  93,  95,  85,  83,
       113,  13, 182,  48, 145,  87,  92,  80, 117, 128, 119, 143, 114,
       118, 108,  63, 121, 142, 154, 120,  82, 109, 101,  86, 229,  76,
        89, 156, 112, 107, 129, 135, 136, 165, 150, 133,  70,  84, 140,
        78,   7,  64,  59, 139,  69, 148, 189, 141, 130, 138,  81, 132,
        10, 123,  65,  68,  66,  62,  74, 131,  39,  46,  38,   8,  17,
       126, 155, 159, 137,  12, 273,  36,  34,  77,  60,  49,  58,  72,
       204, 212,  25,  73,  29,  47,  32,  35,  71, 149,  33,  15,  54,
       224, 162,  37,  75,  79,  55, 158, 164, 173, 181, 185,  21,  24,
        51, 151,  42,  22, 134, 177,  52,  14,  53,  57,  28,  50,  26,
        45, 171,  27,  44, 146,  20, 157, 203,  41,  30, 194, 233, 237,
       230, 195, 253, 152, 190, 160, 208, 180, 144, 174, 170, 192, 209,
       187, 172,  16, 186,  11, 193, 176,  56, 169,  40, 168, 312, 153,
       214,  31, 163,  19, 179,  43, 200, 196, 167, 178, 228,  18, 205,
       201, 191])
```

ANALYSIS

In [29]:

```
movie_raw=df_merged[df_merged['type']=='Movie']
tv_raw=df_merged[df_merged['type']=='TV Show']
```

In [30]:

```
df_merged['date_added']=pd.to_datetime(df_merged['date_added'])
df_merged['date_added'].dtype
```

Out[30]:

```
dtype('<M8[ns]')
```

In [31]:

```
df_merged['year_added']=df_merged['date_added'].dt.year
```

In [32]:

```
df_merged['month_added']=df_merged['date_added'].dt.month
```

In [33]:

```
df_merged_nodups=df_merged.drop_duplicates(['title'])
len(df_merged_nodups)
```

Out[33]:

8807

1. Trend of movies/tv shows being added year on year

In [34]:

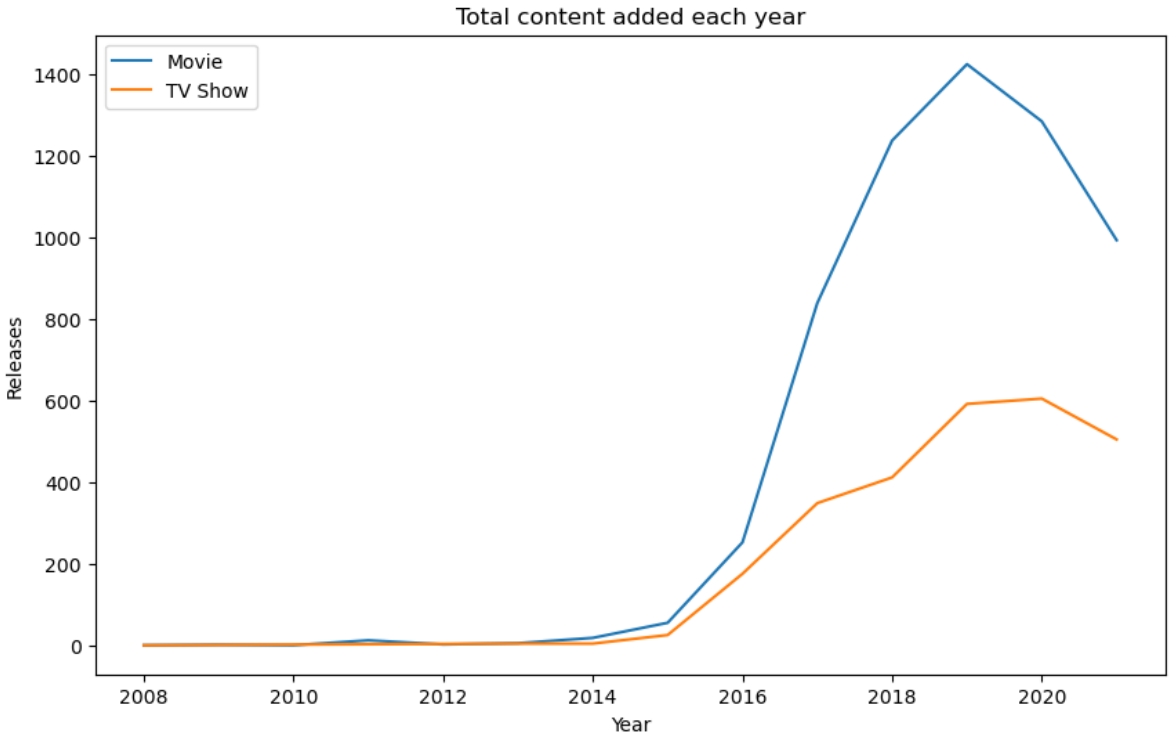
```
movie_grpped=df_merged_nodups[df_merged_nodups['type']=='Movie'].groupby(['year_added'],as_
movie_grpped

tv_grpped=df_merged_nodups[df_merged_nodups['type']=='TV Show'].groupby(['year_added'],as_i
tv_grpped
```

Out[34]:

	year_added	title
0	2008	1
1	2013	5
2	2014	5
3	2015	26
4	2016	176
5	2017	349
6	2018	412
7	2019	592
8	2020	605
9	2021	505

In [35]:

```
fig, ax = plt.subplots(figsize=(10, 6))
sns.lineplot(data=movie_grpped,x='year_added',y='title')
sns.lineplot(data=tv_grpped, x='year_added', y='title')
# ax.set_xticks(np.arange(2008, 2020, 1))
plt.title("Total content added each year")
plt.legend(['Movie','TV Show'])
plt.ylabel("Releases")
plt.xlabel("Year")
plt.show()
```



Observation - There has been consistent growth of movies being added than shows but from 2018 year afterwards the number of movies being added on Netflix start dropping drastically

In [36]:

```
df_merged.head()
```

Out[36]:

	show_id	type	title	date_added	release_year	rating	duration	country_name	listed_i
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90	United States	Docum
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	South Africa	Internat
2	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	South Africa	Internat
3	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	South Africa	Internat
4	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	South Africa	Internat

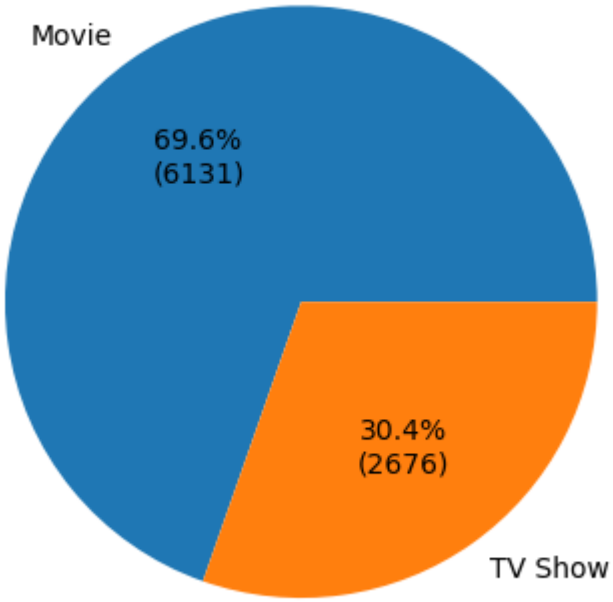
2. Percentage of each content type

In [37]:

```
#create pie chart
def autopct_format(values):
        def my_format(pct):
            total = sum(values)
            val = int(round(pct*total/100.0))
            return '{:.1f}%\n({v:d})'.format(pct, v=val)
        return my_format

s = df_merged_nodups['type'].value_counts()
plt.pie(s,labels = s.index, autopct=autopct_format(s))
```

Out[37]:

```
([<matplotlib.patches.Wedge at 0x264482462e0>,
  <matplotlib.patches.Wedge at 0x2644823f250>],
 [Text(-0.6357552620136555, 0.897672126570692, 'Movie'),
  Text(0.6357552620136554, -0.8976721265706921, 'TV Show')],
 [Text(-0.3467755974619939, 0.4896393417658319, '69.6%\n(6131)'),
  Text(0.3467755974619938, -0.48963934176583196, '30.4%\n(2676)')])
```



Observation - Netflix should try adding more TV Shows

3. % of each genre in each type of content-

a. Top 10 popular overall

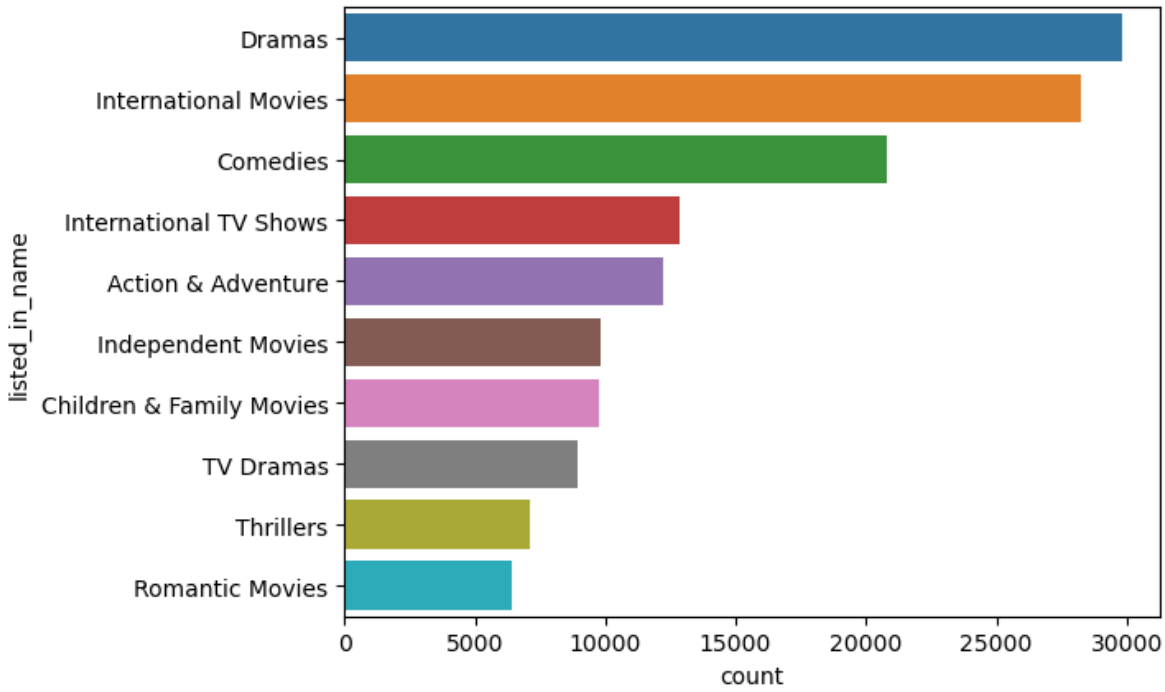
b. Top 5 popular in each type of content

In [38]:

```
sns.countplot(y='listed_in_name',
              data=df_merged,
              order=pd.value_counts(df_merged['listed_in_name']).iloc[:10].index)
```

Out[38]:

```
<AxesSubplot:xlabel='count', ylabel='listed_in_name'>
```



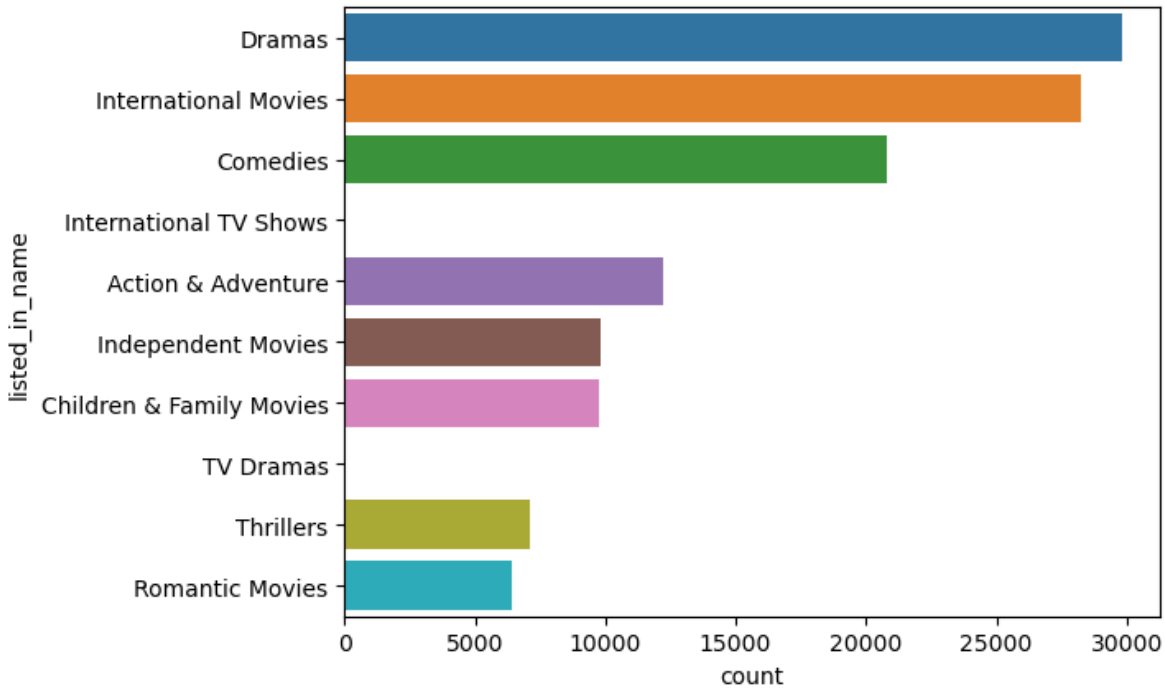
Movie genre popularity

In [39]:

```
sns.countplot(y='listed_in_name',
              data=df_merged[df_merged['type']=='Movie'],
              order=pd.value_counts(df_merged['listed_in_name']).iloc[:10].index)
```

Out[39]:

```
<AxesSubplot:xlabel='count', ylabel='listed_in_name'>
```



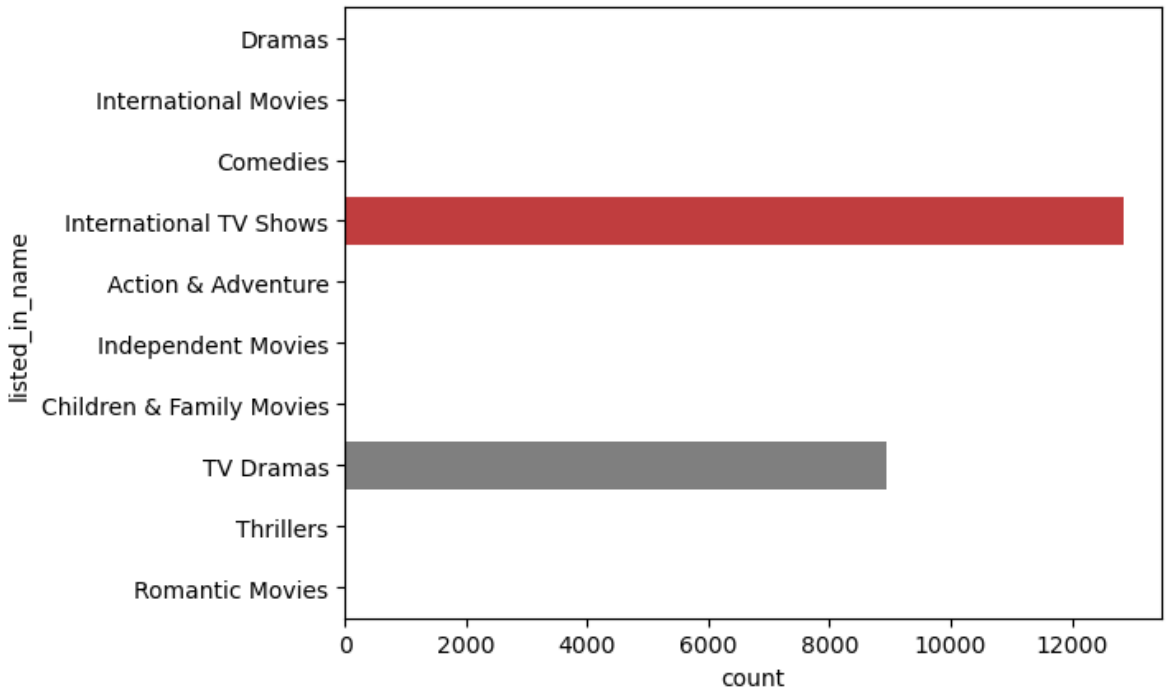
TV Show genre popularity

In [40]:

```
sns.countplot(y='listed_in_name',
              data=df_merged[df_merged['type']=='TV Show'],
              order=pd.value_counts(df_merged['listed_in_name']).iloc[:10].index)
```

Out[40]:

```
<AxesSubplot:xlabel='count', ylabel='listed_in_name'>
```



4. What recommendations will you give to the business with respect to the target audience for these countries -India, USA?

In [41]:

```
two_cnt=df_merged[(df_merged['country_name']=='United States') | (df_merged['country_name']

import plotly.express as px
dfx=two_cnt.groupby(['rating','country_name']).size().reset_index(name='Total Content')
fig4 = px.bar(dfx, x="rating", y="Total Content", color="country_name", title="Most Content
fig4.show()
```

-> Observation - India has more content with TV-14 rating i.e content for age group>14, and US population has TV-MA Mature Audience Only

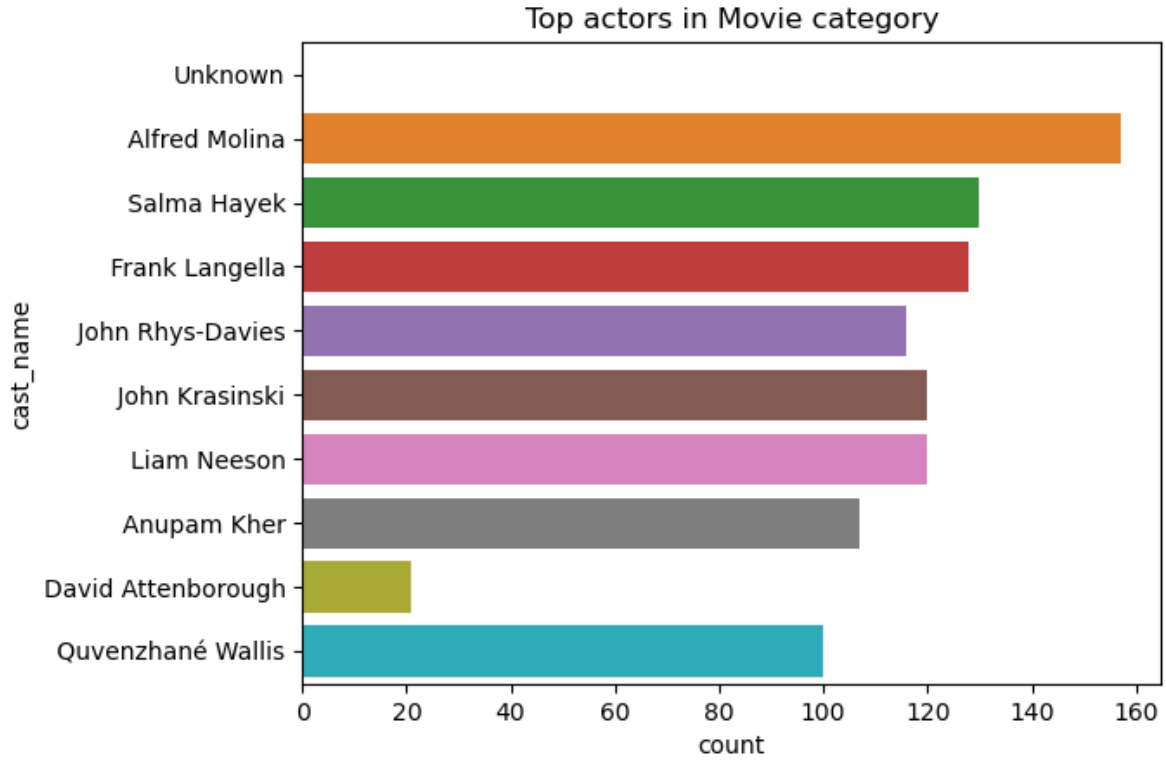
5. Top cast members in each type of content

In [42]:

```
sns.countplot(y='cast_name',
              data=df_merged[(df_merged['type']=='Movie')  & (df_merged['cast_name']!='Unkn
              order=pd.value_counts(df_merged['cast_name']).iloc[:10].index).set(title='Top
```

Out[42]:

```
[Text(0.5, 1.0, 'Top actors in Movie category')]
```

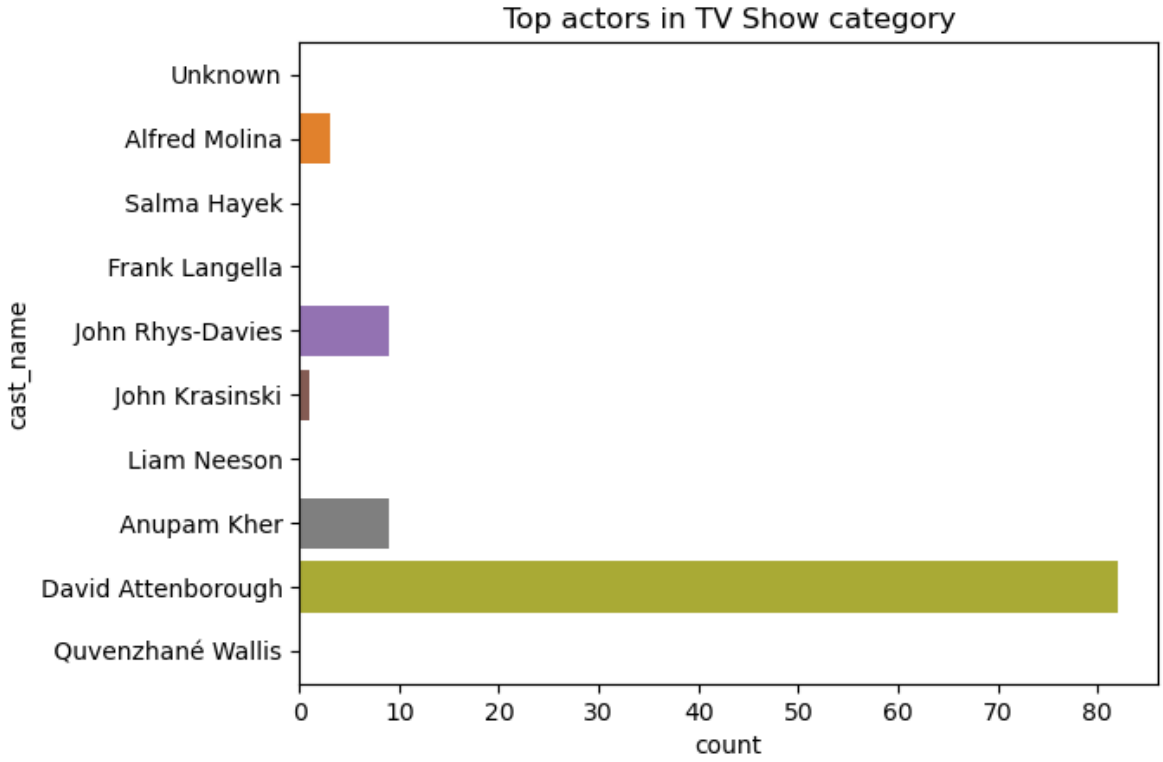


In [43]:

```
sns.countplot(y='cast_name',
              data=df_merged[(df_merged['type']=='TV Show')  & (df_merged['cast_name']!='Un
              order=pd.value_counts(df_merged['cast_name']).iloc[:10].index).set(title='Top
```

Out[43]:

```
[Text(0.5, 1.0, 'Top actors in TV Show category')]
```



In [44]:

```
#### Top 5 popular Actors in India

filtered_cast=df_merged[(df_merged['country_name']=='India') & (df_merged['cast_name']!='Un
actors=filtered_cast.groupby(['cast_name']).size().reset_index(name='Total Content')
actors=actors.sort_values(by=['Total Content'],ascending=False)
actorsTop5=actors.head()
actorsTop5
```

Out[44]:

	cast_name	Total Content
354	Anupam Kher	102
2279	Radhika Apte	80
2077	Paresh Rawal	76
4084	Shah Rukh Khan	73
3658	Akshay Kumar	70

Observation - Anupam Kher's movies have taken precedence over SRK which is surprising to see ;) Netflix should consider adding more women cast movies/shows as well or more South movies so that even south indian audience can cater to their taste of movies

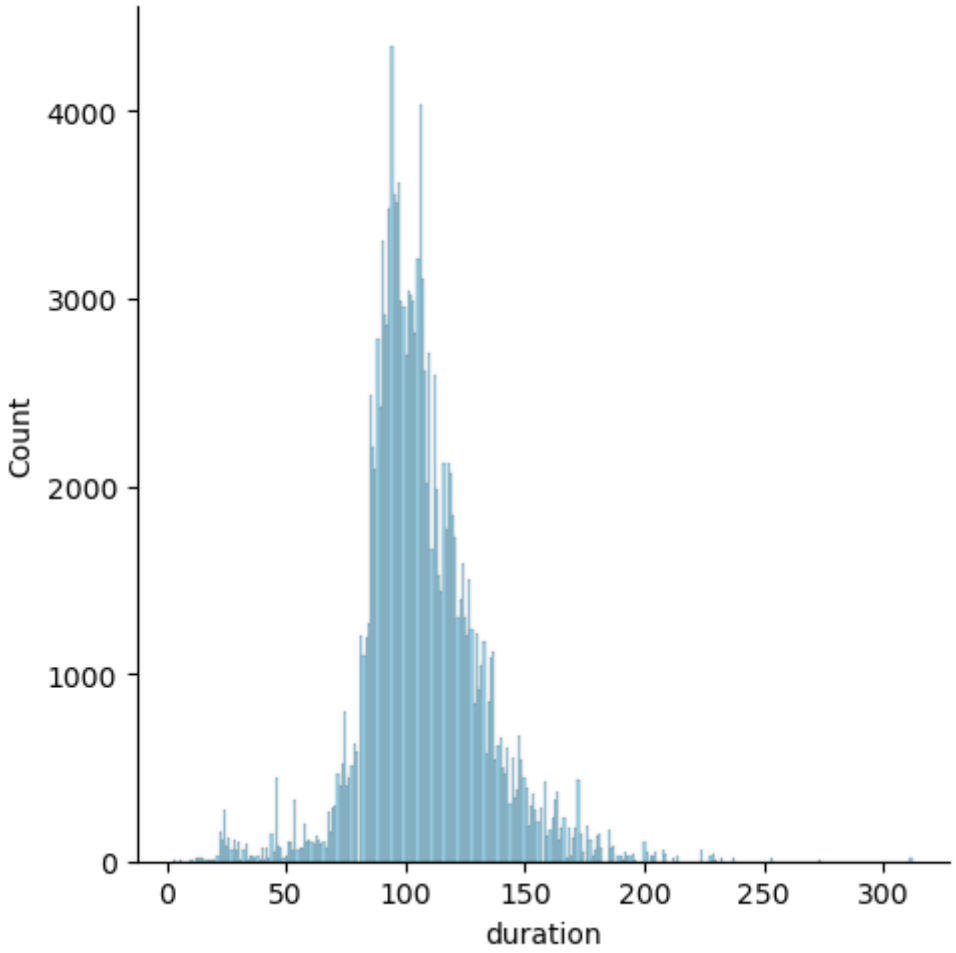
6a. Data Distribution of Time Duration of each content type

In [45]:

```
movie_df=df_merged[df_merged['type']=='Movie']
tv_df=df_merged[df_merged['type']=='TV Show']

# movie
sns.displot(movie_df.duration, color='skyblue')
```

Out[45]:

```
<seaborn.axisgrid.FacetGrid at 0x26452385730>
```

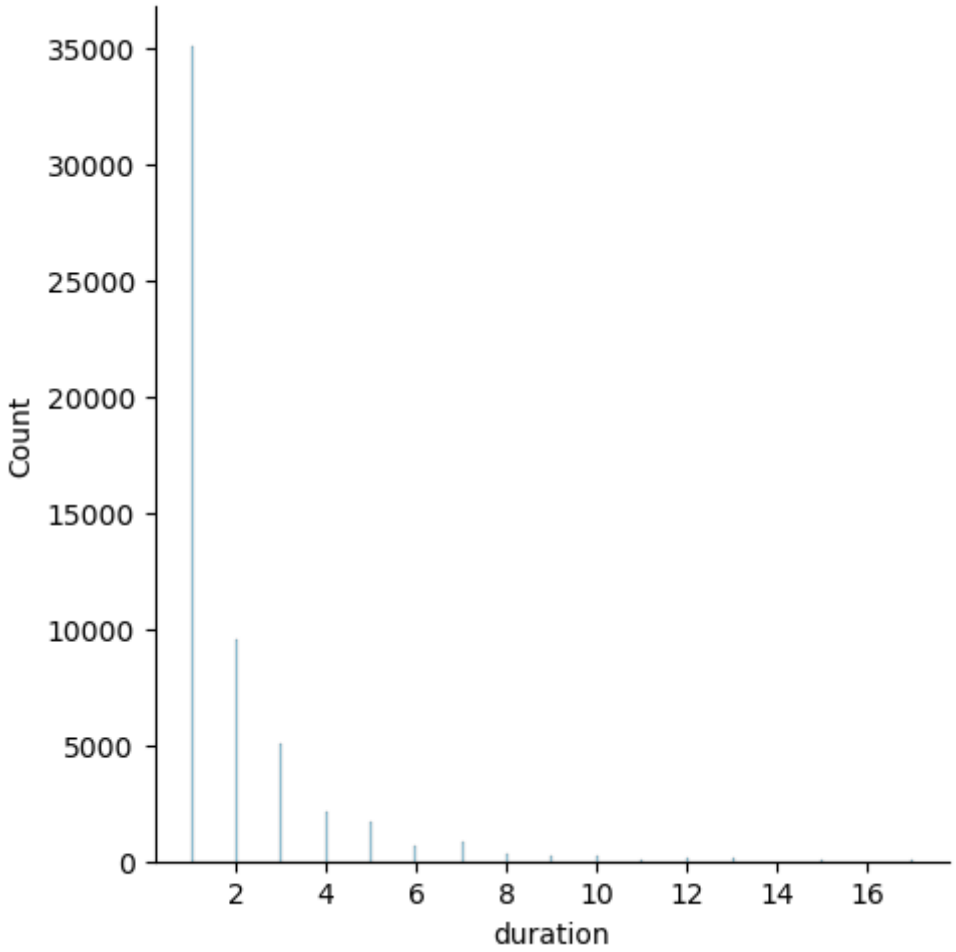


In [46]:

```
sns.displot(tv_df.duration, color='skyblue')
```

Out[46]:

```
<seaborn.axisgrid.FacetGrid at 0x2645237a8b0>
```



Observation - Many TV shows with just one season have been added to Netflix, but if we keep adding/creating more seasons to most liked shows, the viewership and watchtime will increase

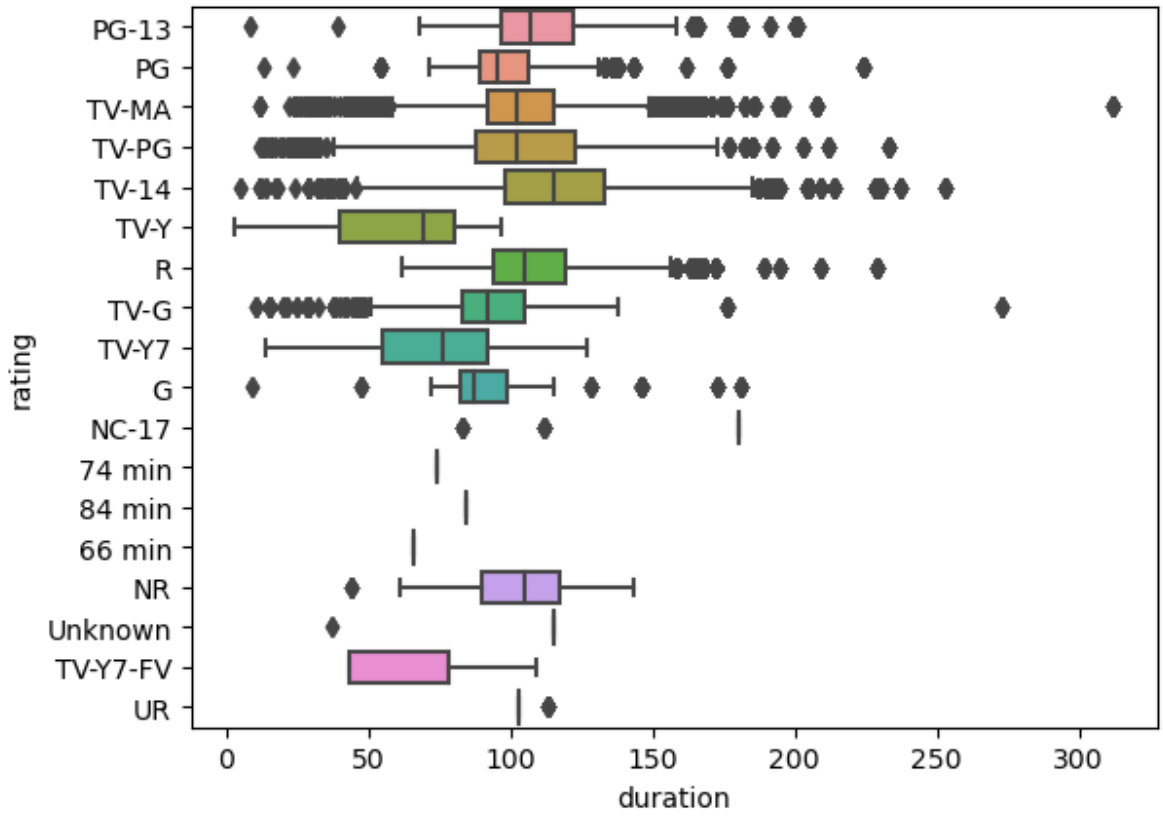
6b. Box plot for distribution of duration across ratings

In [47]:

```
sns.boxplot(data=movie_df, y="rating", x="duration")
```

Out[47]:

```
<AxesSubplot:xlabel='duration', ylabel='rating'>
```

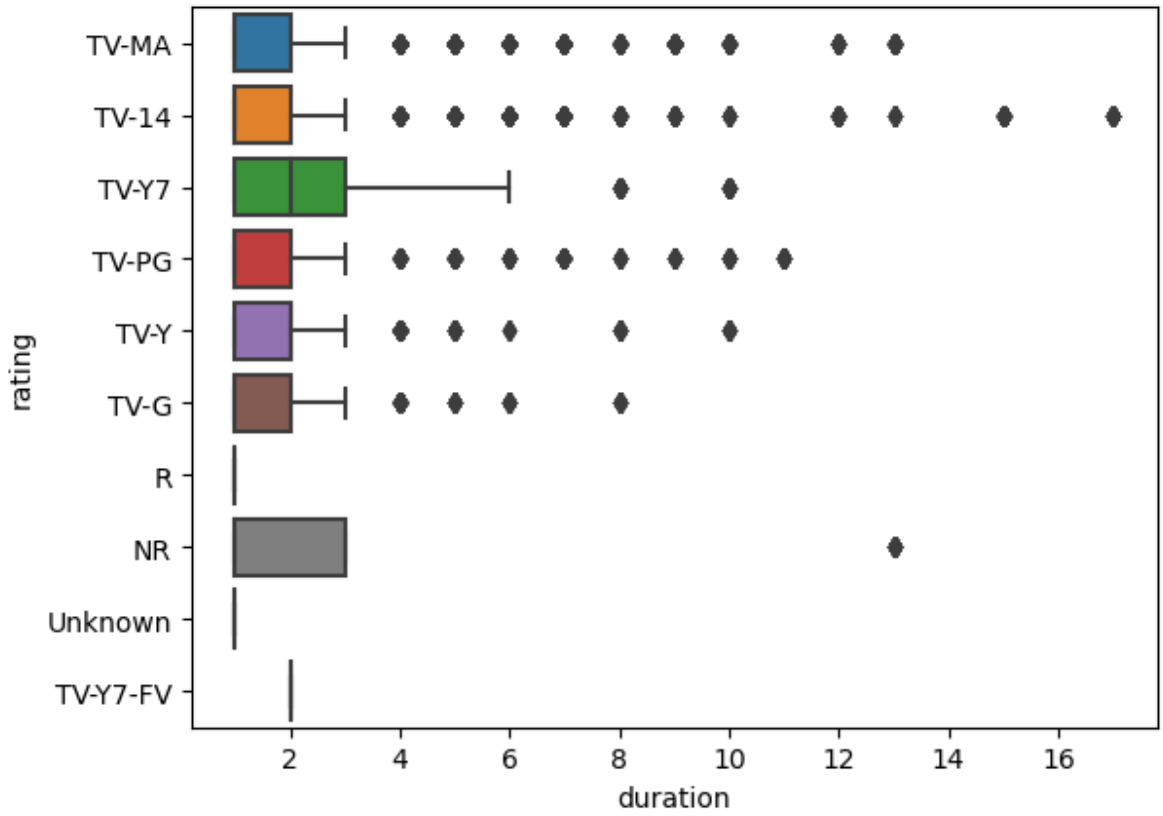


In [48]:

```
sns.boxplot(data=tv_df, y="rating", x="duration")
```

Out[48]:

```
<AxesSubplot:xlabel='duration', ylabel='rating'>
```



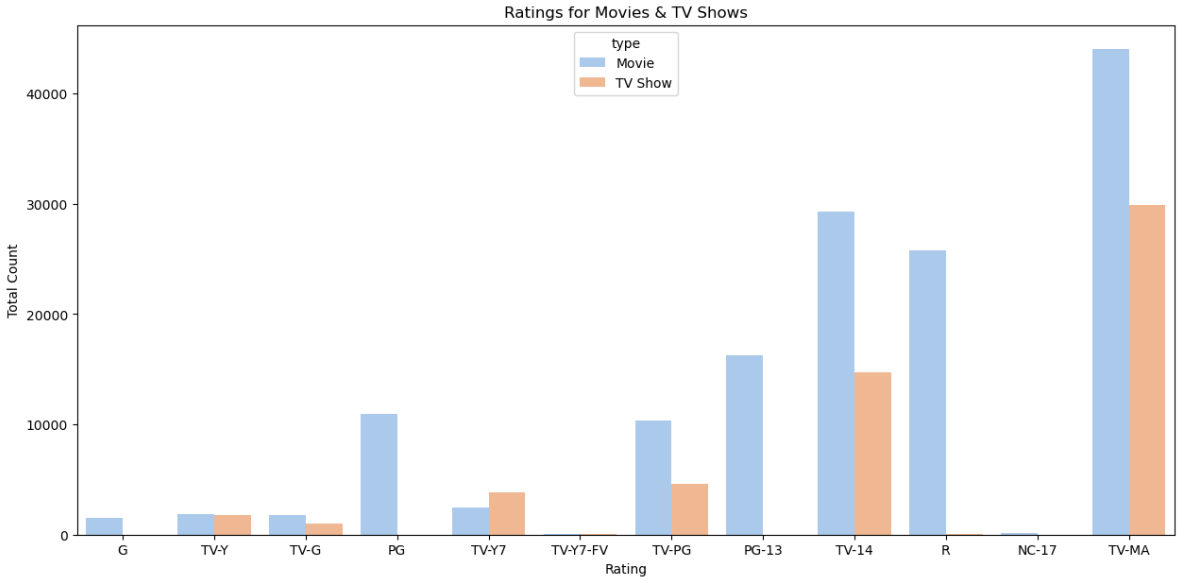
Observation - Mature audiences or 14+ age movies are seen to have higher durations(<100mins) than thats availabe for younger generation, and shows can't be compare as the number to compare is less, need more tv shows added

7. Ratings for Movies & TV Shows

In [49]:

```
order =  ['G', 'TV-Y', 'TV-G', 'PG', 'TV-Y7', 'TV-Y7-FV', 'TV-PG', 'PG-13', 'TV-14', 'R', '
plt.figure(figsize=(15,7))
g = sns.countplot(df_merged.rating, hue=df_merged.type, order=order, palette="pastel");
plt.title("Ratings for Movies & TV Shows")
plt.xlabel("Rating")
plt.ylabel("Total Count")
plt.show()
```

```
C:\Users\nidhi\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning:

Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments withou
t an explicit keyword will result in an error or misinterpretation.
```



Observation - Overall, there is much more content for a more mature audience. For the mature audience, there is much more movie content than there are TV shows. However, for the younger audience (under the age of 17), it is the opposite, there are slightly more TV shows than there are movies.

8.If producer wants to release a movie, which month should he do?

In [50]:

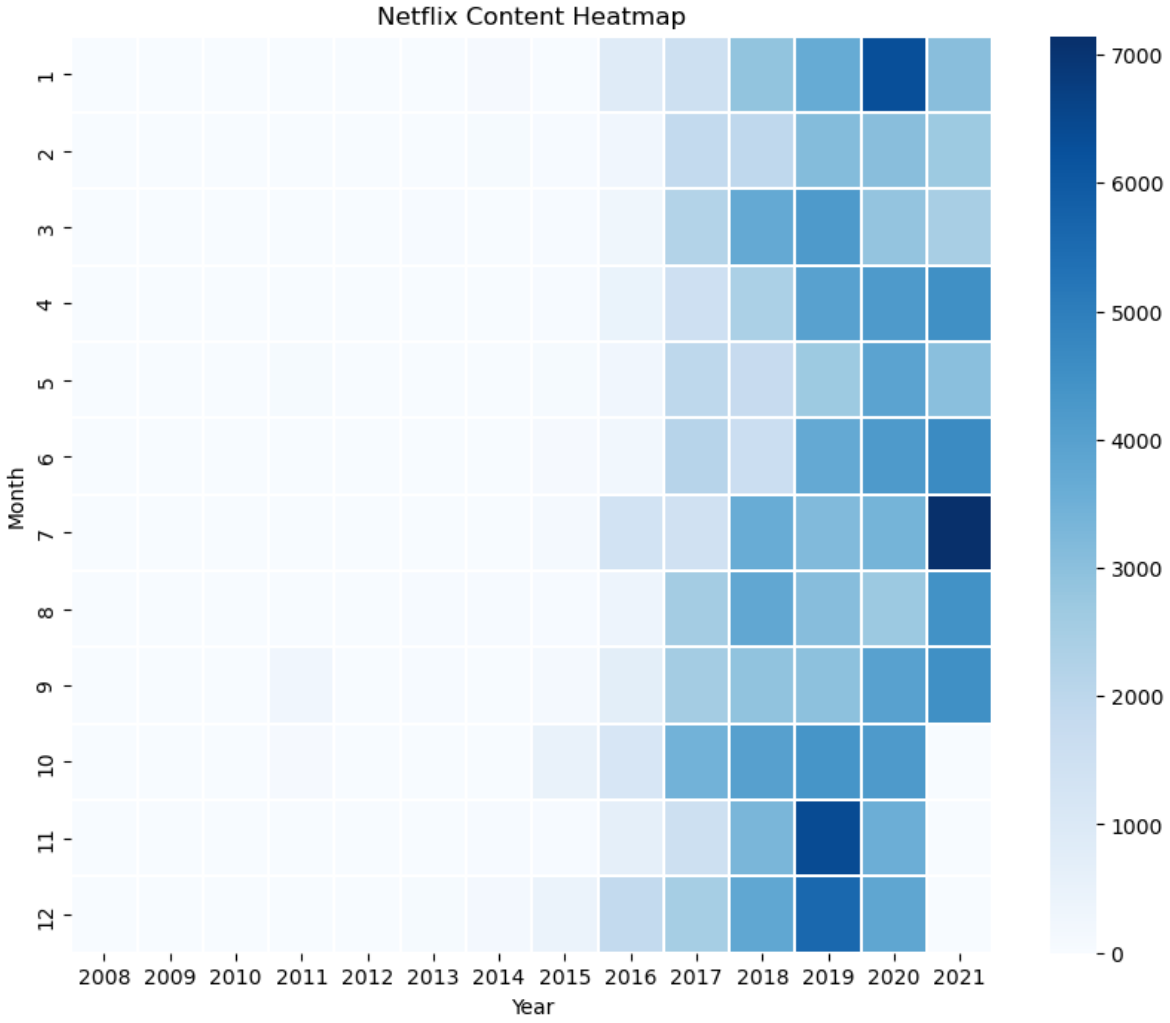
```
month_year_df = df_merged.groupby('year_added',
                                  as_index=False)['month_added'].value_counts().pivot("mont
                                                                                      "year
##this pivot() unstacks()/converts long to wide format
month_year_df
```

Out[50]:

year_added	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	201
month_added												
1	18.0	0.0	0.0	0.0	0.0	0.0	91.0	1.0	884.0	1525.0	2878.0	3672.
2	1.0	0.0	0.0	0.0	2.0	0.0	60.0	44.0	253.0	1847.0	1950.0	3147.
3	0.0	0.0	0.0	0.0	0.0	30.0	0.0	50.0	291.0	2217.0	3732.0	4205.
4	0.0	0.0	0.0	0.0	0.0	0.0	21.0	37.0	469.0	1489.0	2395.0	3972.
5	0.0	24.0	0.0	72.0	0.0	0.0	0.0	75.0	277.0	1956.0	1774.0	2687.
6	0.0	0.0	0.0	0.0	0.0	0.0	19.0	101.0	235.0	2141.0	1591.0	3723.
7	0.0	0.0	0.0	0.0	0.0	0.0	8.0	114.0	1367.0	1450.0	3644.0	3186.
8	0.0	0.0	0.0	0.0	0.0	30.0	36.0	45.0	374.0	2562.0	3777.0	3075.
9	0.0	0.0	0.0	270.0	0.0	49.0	2.0	134.0	745.0	2553.0	2903.0	2976.
10	0.0	0.0	0.0	96.0	0.0	25.0	20.0	504.0	1156.0	3453.0	4002.0	4354.
11	0.0	6.0	20.0	0.0	16.0	10.0	49.0	36.0	652.0	1532.0	3319.0	6394.
12	0.0	0.0	0.0	0.0	18.0	63.0	146.0	419.0	1866.0	2484.0	3820.0	5606.

In [51]:

```
##heatmap
plt.figure(figsize=(10,8))
sns.heatmap(month_year_df, linewidths=0.025, cmap='Blues')
plt.title("Netflix Content Heatmap")
plt.ylabel("Month")
plt.xlabel("Year")
plt.show()
```



Observation - Most movies have been added in Nov, Dec, Jan, so Producer should try selling to Netflix during this time

9. Top countries with highest number of content

In [52]:

```
atthislevel=df_merged.drop_duplicates(['country_name','title'])
top_cntry_highcontent=atthislevel.groupby('country_name',
                                  as_index=False)['title'].count().sort_values(['title'],as


top_cntry_highcontent
```

Out[52]:

	country_name	title
115	United States	4527
45	India	1046
114	United Kingdom	806
20	Canada	445
36	France	393
53	Japan	318
102	Spain	232
100	South Korea	231
38	Germany	226
67	Mexico	169

In [53]:

```
atthislevel=df_merged.drop_duplicates(['country_name','title'])
btm_cntry_highcontent=atthislevel.groupby('country_name',
                                    as_index=False)['title'].count().sort_values(['title'],as


btm_cntry_highcontent
```

Out[53]:

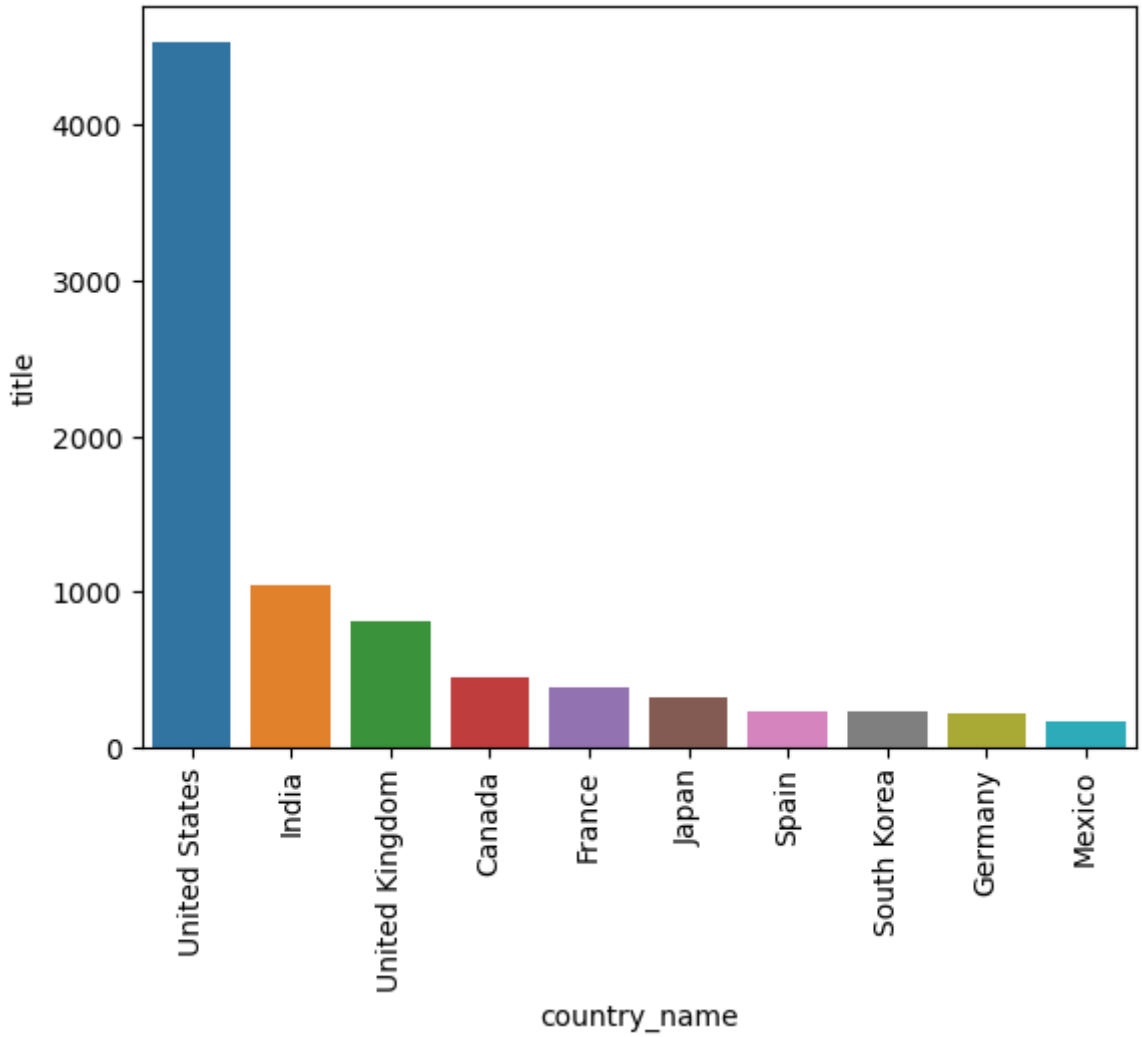
	country_name	title
60	Liechtenstein	1
1	Albania	1
63	Malawi	1
68	Mongolia	1
69	Montenegro	1
71	Mozambique	1
76	Nicaragua	1
80	Palestine	1
81	Panama	1
61	Lithuania	1

In [54]:

```
g=sns.barplot(data=top_cntry_highcontent,x='country_name',y='title')
g.set_xticklabels(g.get_xticklabels(), rotation=90)
```

Out[54]:

```
[Text(0, 0, 'United States'),
 Text(1, 0, 'India'),
 Text(2, 0, 'United Kingdom'),
 Text(3, 0, 'Canada'),
 Text(4, 0, 'France'),
 Text(5, 0, 'Japan'),
 Text(6, 0, 'Spain'),
 Text(7, 0, 'South Korea'),
 Text(8, 0, 'Germany'),
 Text(9, 0, 'Mexico')]
```



10. Top directors based on number of content directed

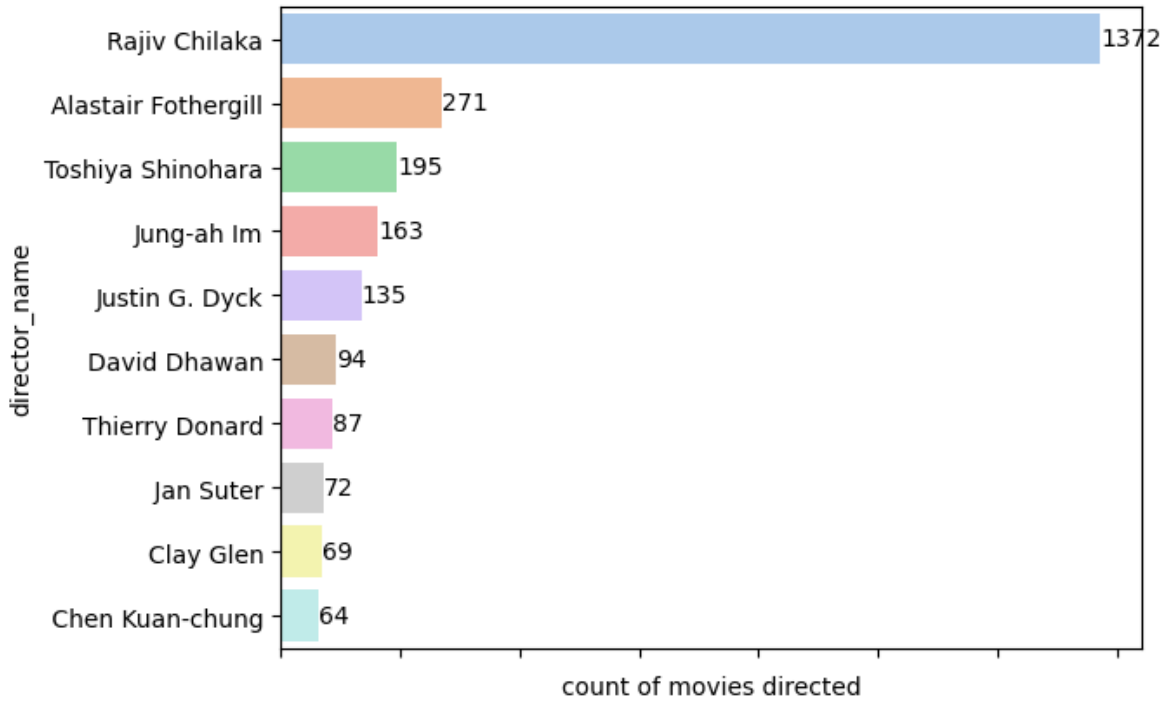
In [55]:

```
nodups_direc=df_merged.drop_duplicates(['director_name','title'])
top_director_highcontent=nodups_direc.groupby('director_name',
                                    as_index=False)['title'].count().sort_values(['title'],as

g = sns.barplot(data=top_director_highcontent,y='director_name',x='title' ,palette="pastel"
g.set_xticklabels(g.get_xticklabels(), rotation=90)
g.set_xlabel('count of movies directed')
g.bar_label(g.containers[0])
```

```
C:\Users\nidhi\AppData\Local\Temp\ipykernel_6432\42676650.py:6: UserWarning:

FixedFormatter should only be used together with FixedLocator
```

Out[55]:

```
[Text(0, 0, '1372'),
 Text(0, 0, '271'),
 Text(0, 0, '195'),
 Text(0, 0, '163'),
 Text(0, 0, '135'),
 Text(0, 0, '94'),
 Text(0, 0, '87'),
 Text(0, 0, '72'),
 Text(0, 0, '69'),
 Text(0, 0, '64')]
```



Recommendations -

1. Netflix should add more TV Shows as the percentage of this content type is very less
2. Since we see drastic decrease in number of movies added after 2018, Netflix should try adding more movies mostly in genres like Thrillers, Romance, Family drama which is less compared to other types on the platform

3. In India, Netflix should consider adding more women cast movies/shows or more South movies so that even south indian audience can cater to their taste of movies
4. Overall, there is much more content for a more mature audience rated content. So Netflix should add more content from other ratings as well so ALL age groups can spend time on Netflix
5. Most movies have been added in Nov, Dec, Jan, so Producer should try selling to Netflix during this time

In [ ]: