In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import re
import plotly.express as px
import plotly.graph_objs as go
import plotly. figure_factory as ff
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [2]:
```
df=pd.read_csv('netflix.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	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...

In [4]:
```
df.tail()
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
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...
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...

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64 
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

In [6]:
```
df.shape
```

Out[6]:
```
(8807, 12)
```

In [7]:
```
df.describe()
```

Out[7]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

In [8]:
```
df.columns
```

Out[8]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

Inspecting Missing Values in the Dataset

In [9]:
```
df.isnull().sum().sort_values(ascending = False)
```

Out[9]:
```
director        2634
country          831
cast             825
date_added        10
rating             4
duration           3
show_id            0
type               0
title              0
release_year       0
listed_in          0
description        0
dtype: int64
```

In [10]:
```
round(df.isnull().sum() / df.shape[0] * 100,2).sort_values(ascending = False)
```

Out[10]:
```
director        29.91
country          9.44
cast             9.37
date_added       0.11
rating           0.05
duration         0.03
show_id          0.00
type             0.00
title            0.00
release_year     0.00
listed_in        0.00
description      0.00
dtype: float64
```

In [11]:
```
df["director"].value_counts()
```

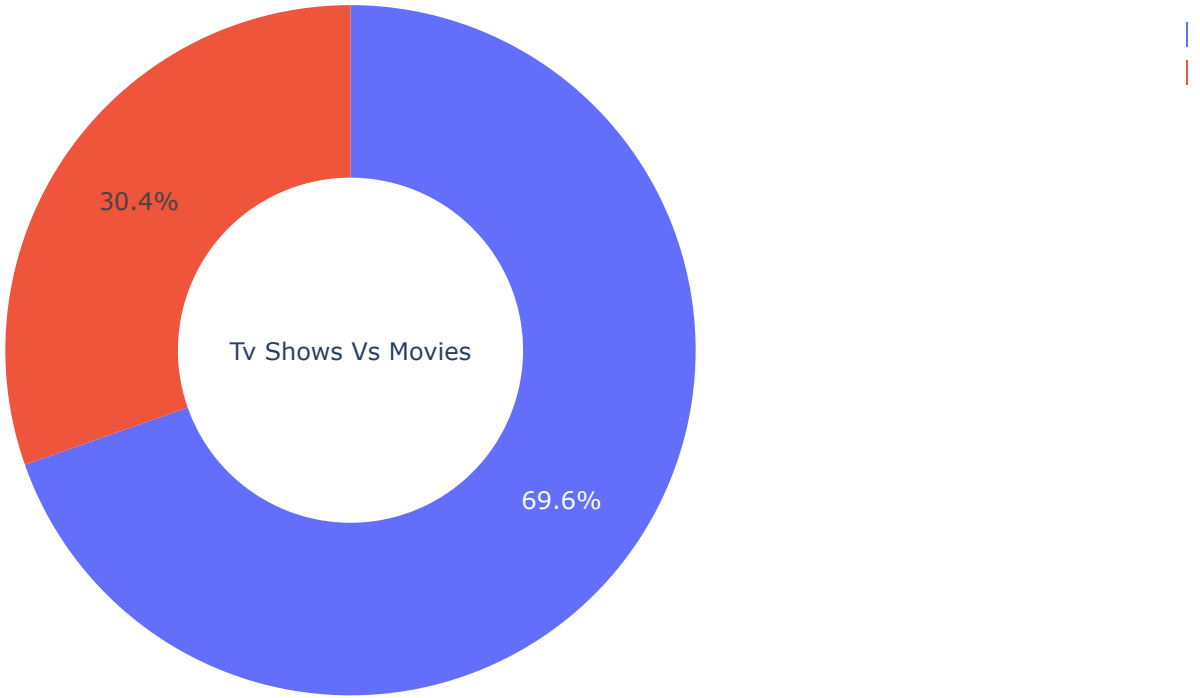
Out[11]:
```
Rajiv Chilaka                     19
Raúl Campos, Jan Suter            18
Marcus Raboy                      16
Suhas Kadav                       16
Jay Karas                         14
                                  ..
Raymie Muzquiz, Stu Livingston     1
Joe Menendez                       1
Eric Bross                         1
Will Eisenberg                     1
Mozez Singh                        1
Name: director, Length: 4528, dtype: int64
```

In [12]:
```
df["director"].value_counts().head(10) #Top 10 directors based on count.
```

Out[12]:
```
Rajiv Chilaka             19
Raúl Campos, Jan Suter    18
Marcus Raboy              16
Suhas Kadav               16
Jay Karas                 14
Cathy Garcia-Molina       13
Martin Scorsese           12
Youssef Chahine           12
Jay Chapman               12
Steven Spielberg          11
Name: director, dtype: int64
```

2. Comparison of tv shows vs. movies.

In [13]:
```
go.Figure(data = [go.Pie(labels = df.type.value_counts(normalize = True).index,
                         values = df.type.value_counts(normalize = True).values, hole = 0.5,
                         title = "Tv Shows Vs Movies")])
```



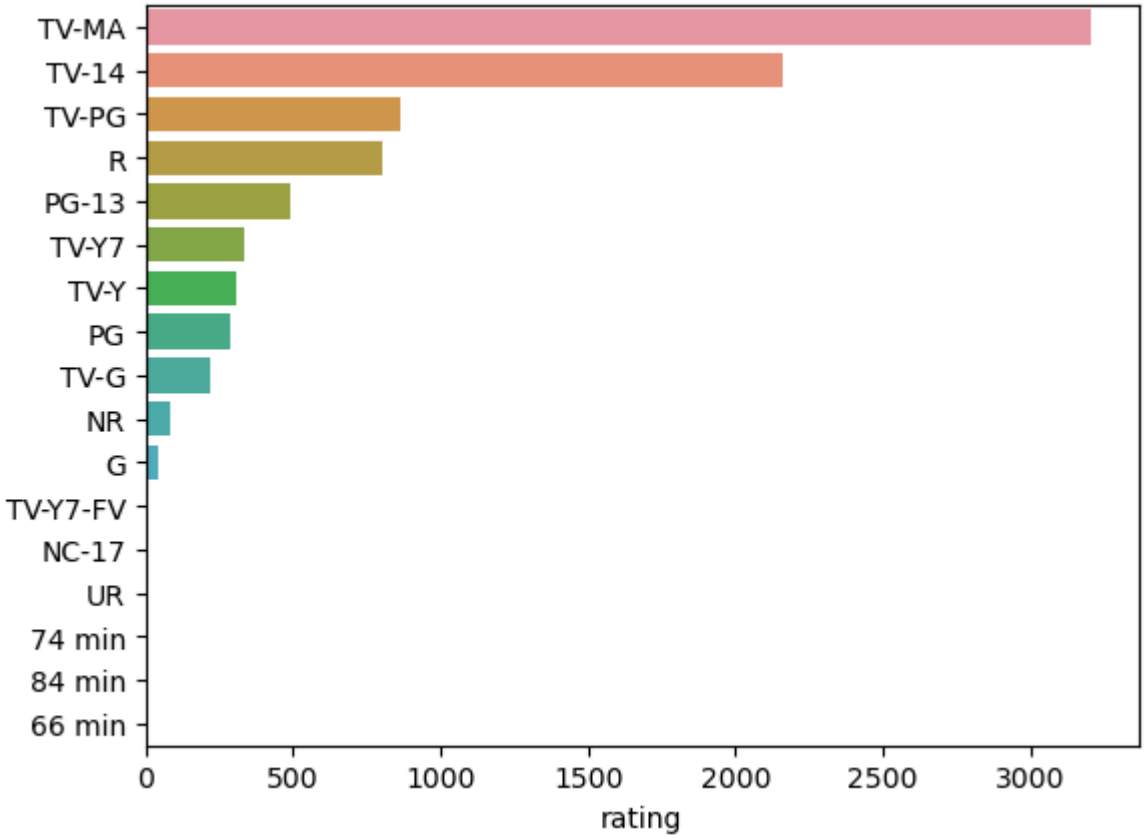
In [14]:
```
df.type.value_counts()
```

Out[14]:
```
Movie      6131
TV Show    2676
Name: type, dtype: int64
```

In [15]:
```
df.rating.value_counts()
```

Out[15]:
```
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64
```

In [16]:
```
sns.barplot(y=df.rating.value_counts().index, x=df.rating.value_counts(), data=df, orient="h")
plt.show()
```



The highest count - TV-MA is the rating that shows that a program is intended for adults. 'MA stands for 'mature audiences. Children aged 17 and younger should not view these programs.

Second largest is the 'TV-14'. A TV-14 program is meant for children over 14 years of age. It is generally not recommended to let children watch the program without parental attendance, or at least without them vetting it first. It can contain crude humor, the use of harmful substances, strong language, violence, and complex or upsetting themes.

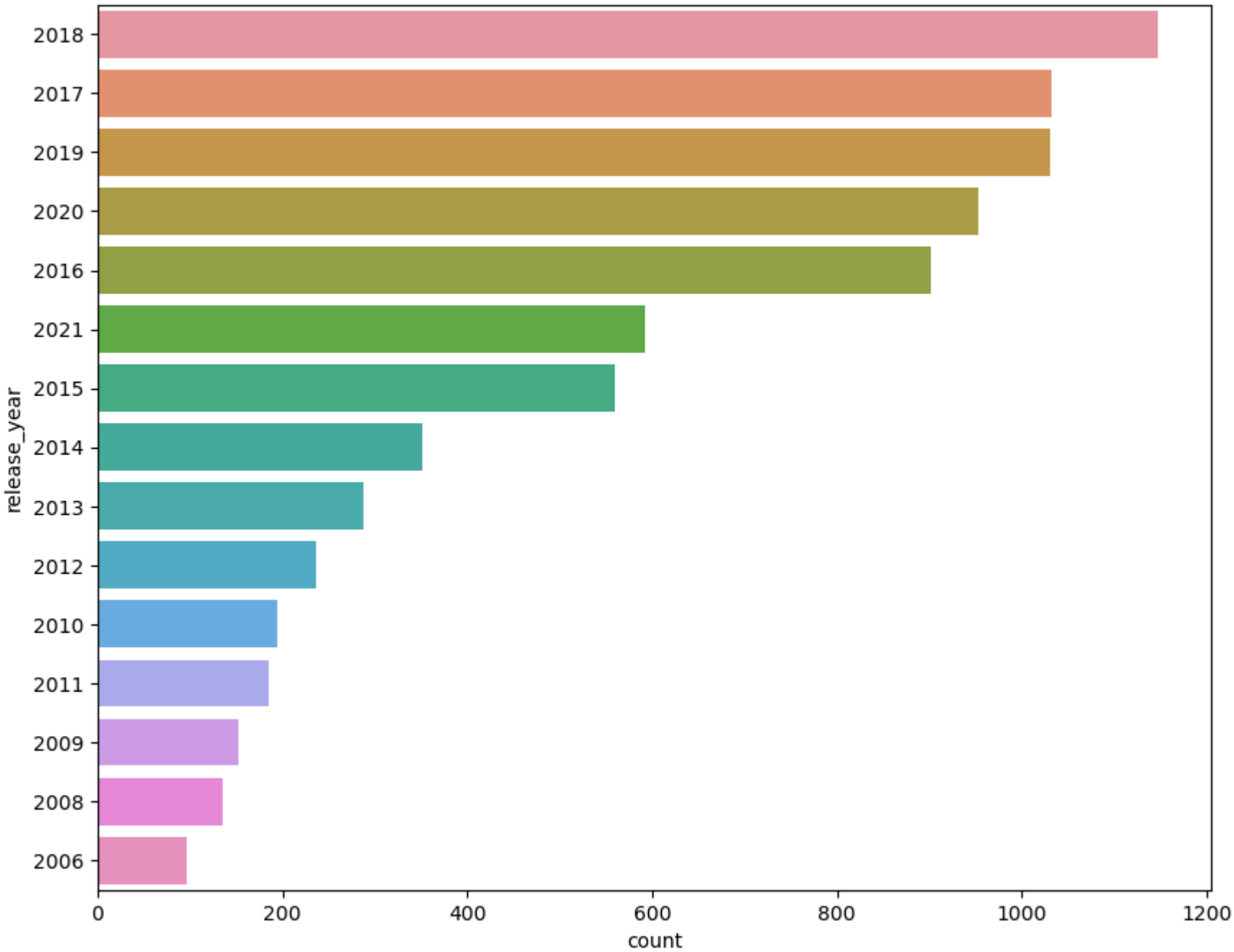
Third largest is the very popular 'R' rating. R is the short for retricted, so any young person under 17 should not watch.

In [17]:
```
df.country.value_counts().head(10)
```

Out[17]:
```
United States     2818
India              972
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Egypt              106
Name: country, dtype: int64
```

1. How has the number of movies released per year changed over the last 20-30 years?

In [18]:
```
plt.figure(figsize=(10,8))
ax= sns.countplot(y="release_year", data=df, order=df.release_year.value_counts().index[0:15])
```


release_year
2018
2017
2019
2020
2016
2021
2015
2014
2013
2012
2010
2011
2009
2008
2006
0
200
400
600
800
1000
1200
count

Heighest Releases in 2018 followed by 2017 and 2019

In [19]:
```
df.director.value_counts().head(10)
```

Out[19]:
```
Rajiv Chilaka             19
Raúl Campos, Jan Suter    18
Marcus Raboy              16
Suhas Kadav               16
Jay Karas                 14
Cathy Garcia-Molina       13
Martin Scorsese           12
Youssef Chahine           12
Jay Chapman               12
Steven Spielberg          11
Name: director, dtype: int64
```

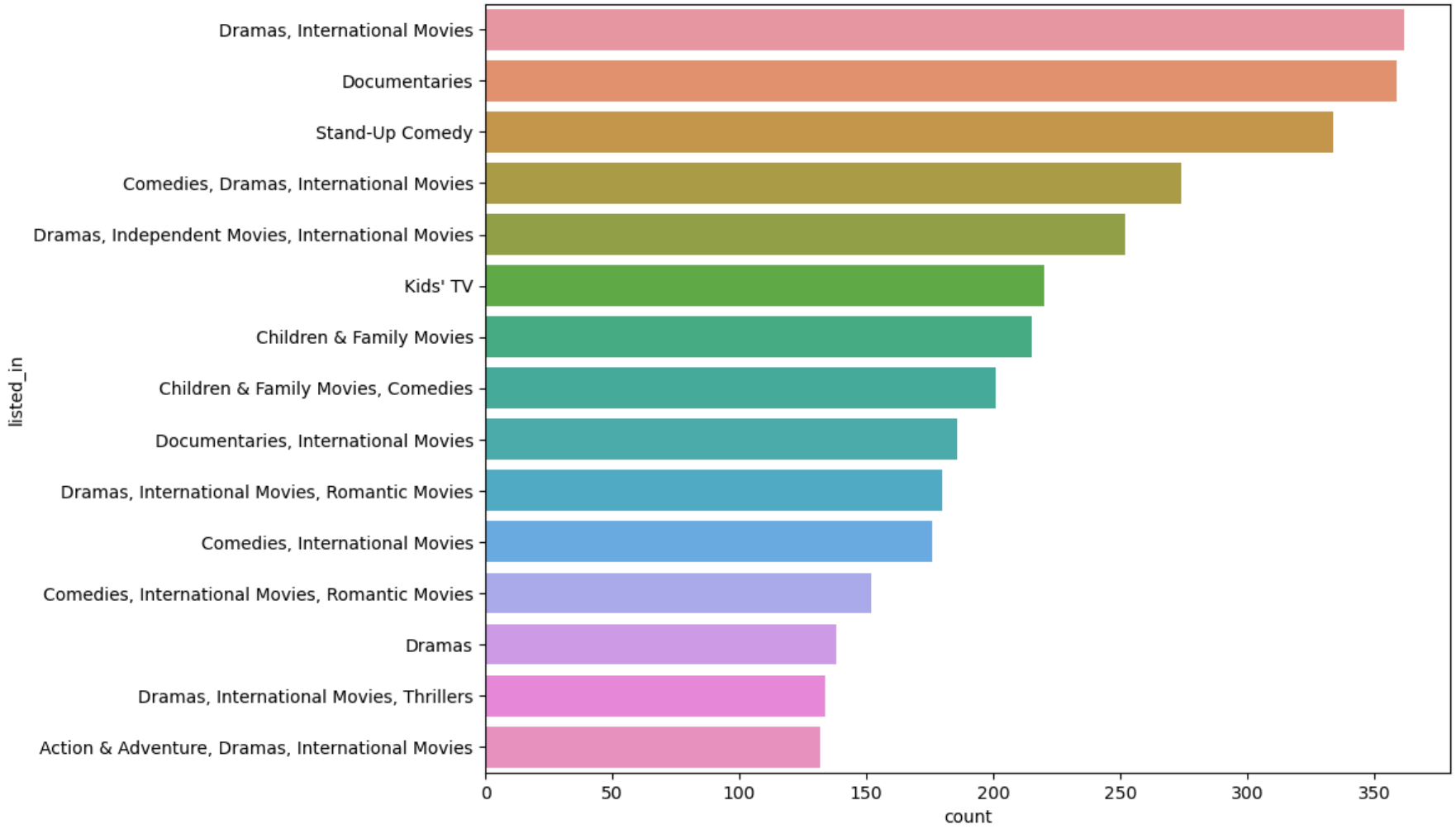
In [20]:
```
df.listed_in.value_counts().head(10)
```

Out[20]:
```
Dramas, International Movies                          362
Documentaries                                         359
Stand-Up Comedy                                       334
Comedies, Dramas, International Movies                274
Dramas, Independent Movies, International Movies      252
Kids' TV                                              220
Children & Family Movies                              215
Children & Family Movies, Comedies                    201
Documentaries, International Movies                   186
Dramas, International Movies, Romantic Movies         180
Name: listed_in, dtype: int64
```

In [21]:
```
df.listed_in.value_counts().tail(10)
```

Out[21]:
```
Docuseries, Reality TV, Teen TV Shows                1
Crime TV Shows, International TV Shows, Reality TV   1
Anime Features, Romantic Movies                      1
Anime Features, Music & Musicals                     1
British TV Shows, Kids' TV, TV Thrillers             1
Kids' TV, TV Action & Adventure, TV Dramas           1
TV Comedies, TV Dramas, TV Horror                    1
Children & Family Movies, Comedies, LGBTQ Movies     1
Kids' TV, Spanish-Language TV Shows, Teen TV Shows   1
Cult Movies, Dramas, Thrillers                       1
Name: listed_in, dtype: int64
```

In [22]:
```
plt.figure(figsize=(10,8))
ax = sns.countplot (y="listed_in", data=df, order=df.listed_in.value_counts().index[0:15])
```



Handling Missing Values

In [23]:
```
round(df.isnull().sum() / df.shape[0] * 100,2).sort_values(ascending=False)
```

Out[23]:
```
director        29.91
country          9.44
cast             9.37
date_added       0.11
rating           0.05
duration         0.03
show_id          0.00
type             0.00
title            0.00
release_year     0.00
listed_in        0.00
description      0.00
dtype: float64
```

In [24]:
```
round(df.isnull().sum())
```

Out[24]:
```
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 [25]:
```
#Dropping rows for small percentages of null
df.dropna(subset=["rating", "duration"],axis=0, inplace=True)
```

In [26]:
```
df.shape
```

Out[26]:
```
(8800, 12)
```

In [27]:
```
round(df.isnull().sum() / df.shape[0] * 100,2).sort_values(ascending=False)
```

Out[27]:
```
director        29.90
country          9.43
cast             9.38
date_added       0.11
show_id          0.00
type             0.00
title            0.00
release_year     0.00
rating           0.00
duration         0.00
listed_in        0.00
description      0.00
dtype: float64
```

In [28]:
```
df.dropna (subset=["date_added"],axis=0, inplace=True)
```

In [29]:
```
round(df.isnull().sum() / df.shape[0] * 100,2).sort_values(ascending=False)
```

Out[29]:
```
director        29.82
country          9.43
cast             9.39
show_id          0.00
type             0.00
title            0.00
date_added       0.00
release_year     0.00
rating           0.00
duration         0.00
listed_in        0.00
description      0.00
dtype: float64
```

In [30]:
```
### replace missing values in country with "Unknown"
df["country"].replace(np.NaN, "Unknown", inplace=True)
```

In [31]:
```
round(df.isnull().sum() / df.shape[0] * 100,2).sort_values(ascending=False)
```

Out[31]:
```
director        29.82
cast             9.39
show_id          0.00
type             0.00
title            0.00
country          0.00
date_added       0.00
release_year     0.00
rating           0.00
duration         0.00
listed_in        0.00
description      0.00
dtype: float64
```

In [32]:
```
df.country.value_counts().head()
```

Out[32]:
```
United States     2809
India              972
Unknown            829
United Kingdom     418
Japan              243
Name: country, dtype: int64
```

In [33]:
```
df.cast.value_counts().head()
```

Out[33]:
```
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
Name: cast, dtype: int64
```

In [34]:
```
### replace missing values in Cast with "No Cast"
df["cast"].replace(np. NaN, "No Cast", inplace=True)
```

In [35]:
```
round(df.isnull().sum() / df.shape[0] * 100,2).sort_values(ascending=False)
```

Out[35]:
```
director        29.82
show_id          0.00
type             0.00
title            0.00
cast             0.00
country          0.00
date_added       0.00
release_year     0.00
rating           0.00
duration         0.00
listed_in        0.00
description      0.00
dtype: float64
```

In [36]:
```
### replace missing values in Director with "No Director"
df ["director"].replace(np.NaN, "No Director", inplace=True)
```

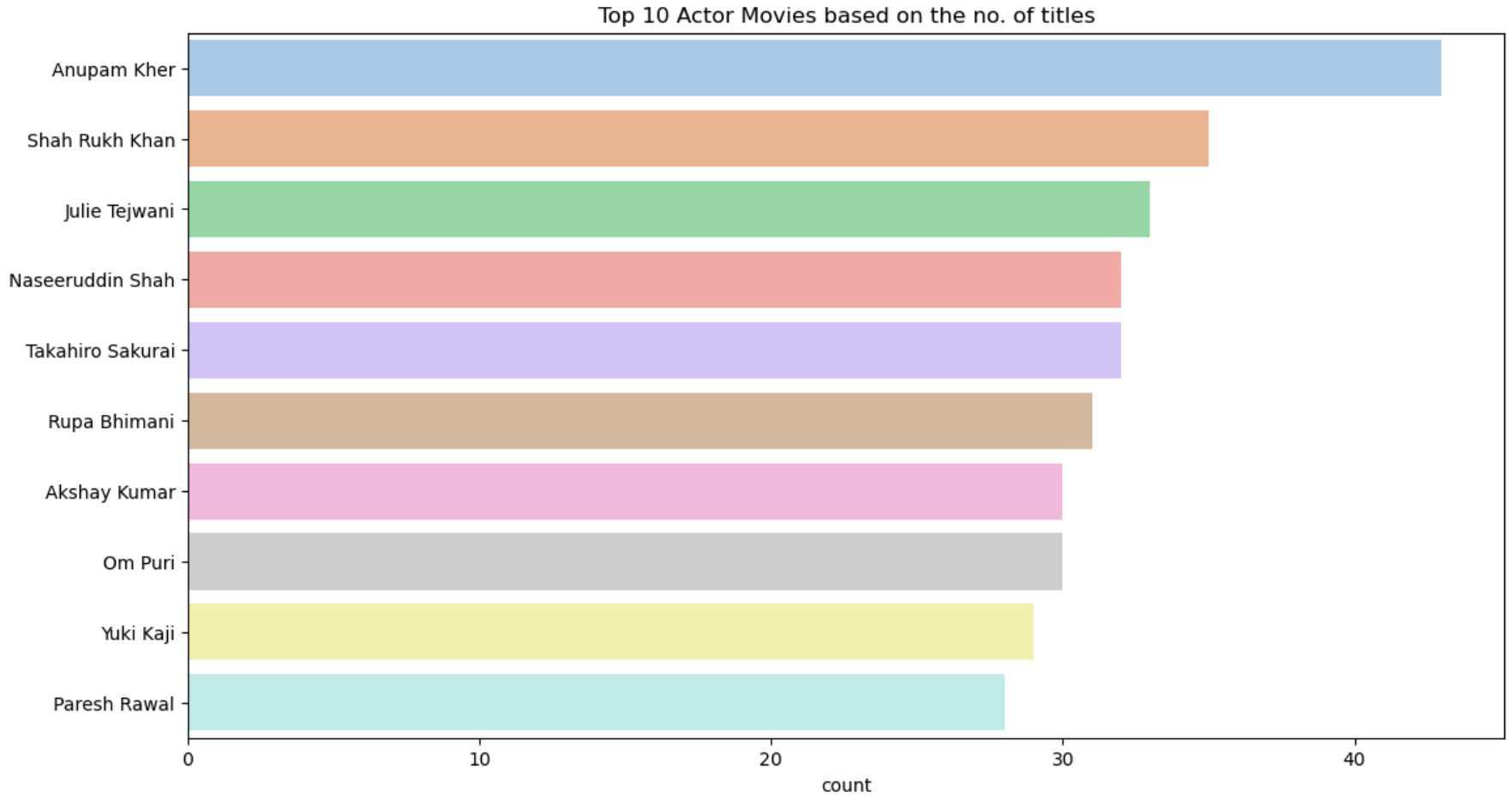
In [37]:
```
round(df.isnull().sum() / df.shape[0] * 100,2).sort_values(ascending=False)
```

Out[37]:
```
show_id         0.0
type            0.0
title           0.0
director        0.0
cast            0.0
country         0.0
date_added      0.0
release_year    0.0
rating          0.0
duration        0.0
listed_in       0.0
description     0.0
dtype: float64
```

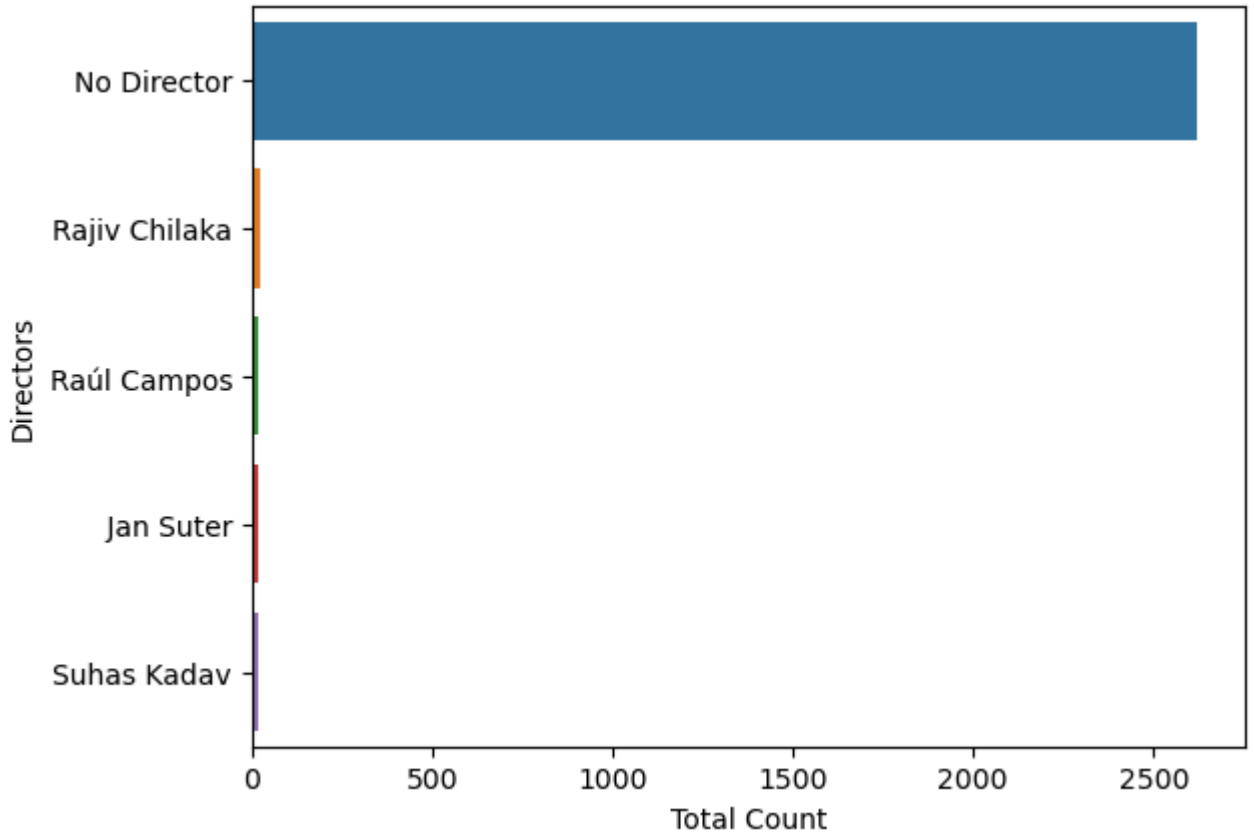
In [38]:
```
df["title"]
```

Out[38]:
```
0        Dick Johnson Is Dead
1               Blood & Water
2                   Ganglands
3       Jailbirds New Orleans
4                Kota Factory
                ...
8802                   Zodiac
8803             Zombie Dumb
8804              Zombieland
8805                    Zoom
8806                  Zubaan
Name: title, Length: 8790, dtype: object
```

In [39]:
```
cast_shows = df[df.cast != "No Cast"].set_index("title").cast.str.split(", ", expand=True).stack().reset_index(level
plt.figure(figsize=(13,7))
plt.title("Top 10 Actor Movies based on the no. of titles")
sns.countplot(y = cast_shows, order = cast_shows.value_counts().index[:10], palette = "pastel")
plt.show()
```



In [40]:
```
cast_df = pd.DataFrame()
cast_df = df['director'].str.split(',', expand=True).stack()
cast_df = cast_df.to_frame()
cast_df.columns = ['Directors']
directors=cast_df.groupby(['Directors']).size().reset_index(name = 'Total Count')
directors=directors [directors. Directors != 'Unknown']
directors=directors.sort_values (by= [ 'Total Count'], ascending=False)
top5directors=directors.head()
barChart2 = sns.barplot (top5directors, x= 'Total Count', y='Directors')
```



1. Top 5 Popular Cast: Anupam Kher, Rupa Bhimani, Takahiro Sakurai, Julie Tejwani,Om Puri
2. Top 5 Popular Directors: Rajiv Chilaka, Jan Suter, Raúl Campos, Suhas Kadav, Marcus Raboy

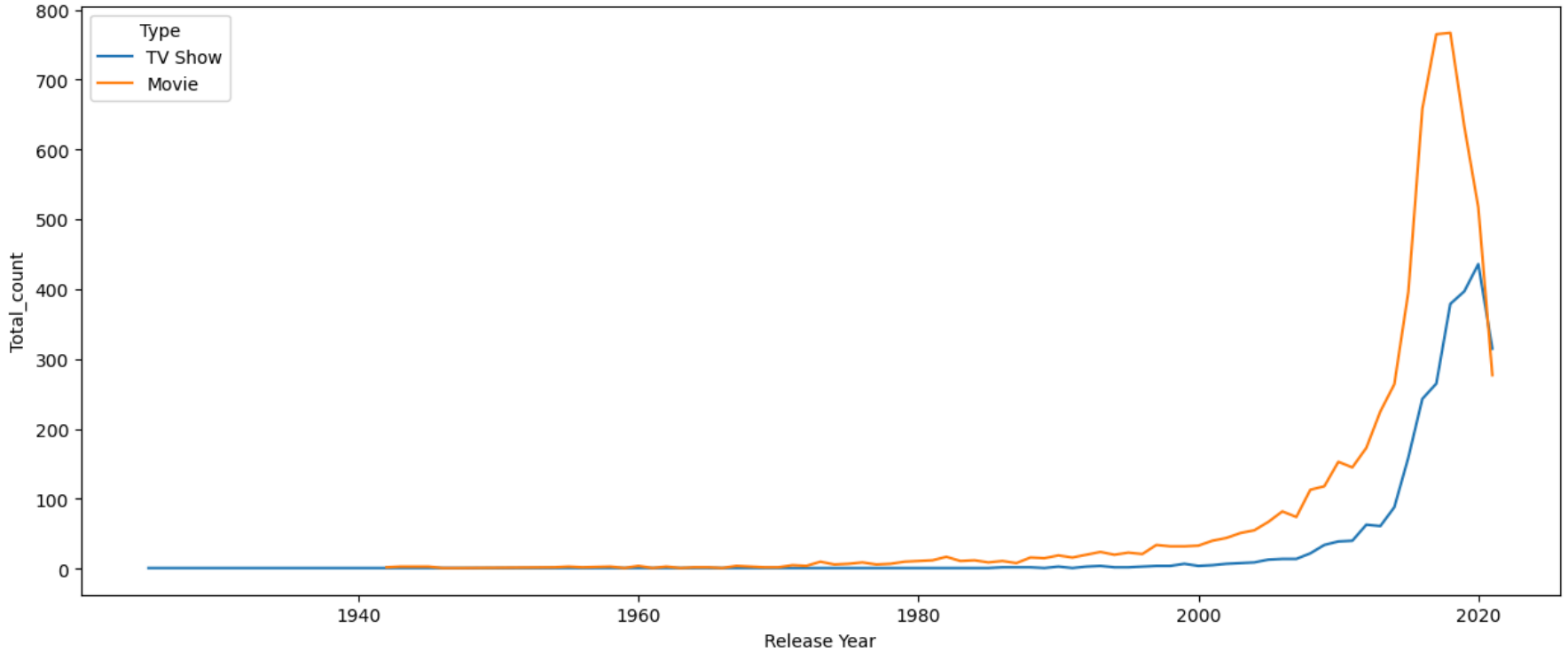
In [41]:
```
df1 = df[['type', 'release_year']]
df1 = df1.rename (columns = {"release_year": "Release Year", "type": "Type"})
df2 = df1.groupby(['Release Year', 'Type']).size().reset_index (name = 'Total_count')
df2
```

Out[41]:

	Release Year	Type	Total_count
0	1925	TV Show	1
1	1942	Movie	2
2	1943	Movie	3
3	1944	Movie	3
4	1945	Movie	3
...	...	...	...
114	2019	TV Show	397
115	2020	Movie	517
116	2020	TV Show	436
117	2021	Movie	277
118	2021	TV Show	315

119 rows × 3 columns

In [42]:
```
plt.figure(figsize=(15, 6))
graph = sns.lineplot(df2, x = "Release Year", y="Total_count", hue = "Type")
```



In [43]:
```
df['duration']
```

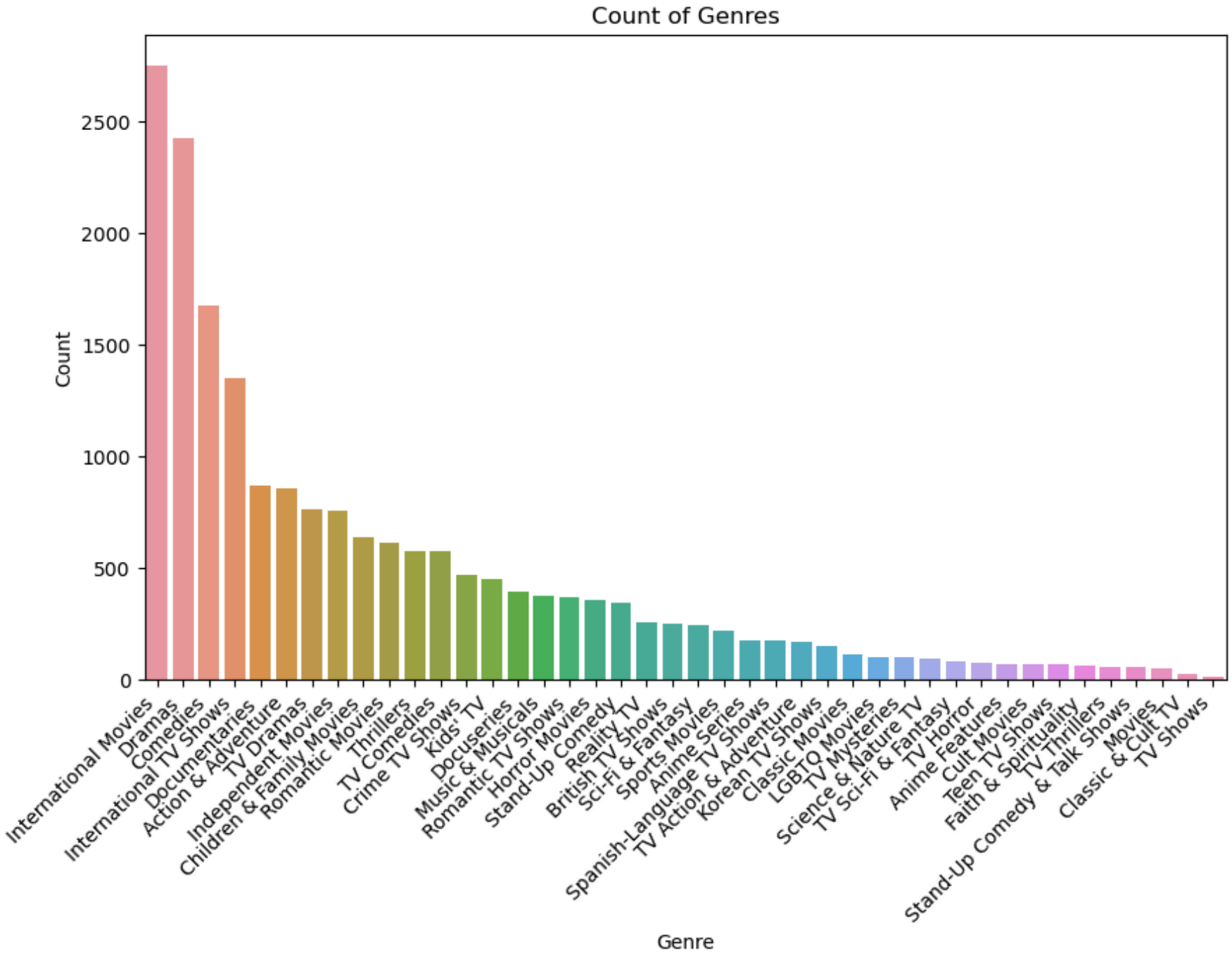
Out[43]:
```
0          90 min
1       2 Seasons
2        1 Season
3        1 Season
4       2 Seasons
           ...
8802      158 min
8803    2 Seasons
8804       88 min
8805       88 min
8806      111 min
Name: duration, Length: 8790, dtype: object
```

In [44]:
```
df['duration_min'] = df[df['type'] == 'Movie']['duration'].str.extract('(\d+)').astype(float)
df['duration_seasons'] = df[df['type'] == 'TV Show']['duration'].str.extract('(\d+)').astype(float) # fill NaN value
df[['duration_min', 'duration_seasons' ]] = df[['duration_min', 'duration_seasons']].fillna(0)
df = df.drop('duration', axis= 1)
```

In [45]:
```
df['duration_min']
df['duration_seasons']
```

Out[45]:
```
0       0.0
1       2.0
2       1.0
3       1.0
4       2.0
       ...
8802    0.0
8803    2.0
8804    0.0
8805    0.0
8806    0.0
Name: duration_seasons, Length: 8790, dtype: float64
```

In [46]:
```
genre_counts = df['listed_in'].str.split(', ').explode().value_counts()
# Plot the count of genres using a bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x=genre_counts.index, y=genre_counts.values)
plt.xlabel('Genre')
plt.ylabel('Count')
plt.title('Count of Genres')
plt.xticks(rotation=45, ha='right')
plt.show()
```


Count of Genres
Count
0
500
1000
1500
2000
2500
Genre
International Movies
Dramas
Comedies
International TV Shows
Documentaries
Action & Adventure
TV Dramas
Independent Movies
Children & Family Movies
Romantic Movies
Thrillers
TV Comedies
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
Stand-Up Comedy & Talk Shows
Movies
Classic & Cult TV
TV Shows

international movies, dramas, comedies, international TV shows are very popular genres.

In [47]:
```
movies_df=df.loc[(df['type']=="Movie")]
movies_df.head (2)
```

Out[47]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	listed_in	description	duration_min	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No Cast	United States	September 25, 2021	2020	PG-13	Documentaries	As her father nears the end of his life, filmm...	90.0	
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	Unknown	September 24, 2021	2021	PG	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...	91.0	

In [48]:
```
show_df=df.loc[(df['type']=="TV Show")]
show_df.head (2)
```

Out[48]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	listed_in	description	duration_min	duration_s
1	s2	TV Show	Blood & Water	No Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	0.0	
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown	September 24, 2021	2021	TV-MA	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	0.0	

In [50]:
```
movies_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6126 entries, 0 to 8806
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   show_id           6126 non-null   object 
 1   type              6126 non-null   object 
 2   title             6126 non-null   object 
 3   director          6126 non-null   object 
 4   cast              6126 non-null   object 
 5   country           6126 non-null   object 
 6   date_added        6126 non-null   object 
 7   release_year      6126 non-null   int64  
 8   rating            6126 non-null   object 
 9   listed_in         6126 non-null   object 
 10  description       6126 non-null   object 
 11  duration_min      6126 non-null   float64
 12  duration_seasons  6126 non-null   float64
dtypes: float64(2), int64(1), object(10)
memory usage: 670.0+ KB
```

Bussiness Insights

1. Netflix present more movies than TV shows. Most movies last for 90-120 minutes. Most TV series are new with only 1 or 2 seasons. There was an increase since 2015 in both TV shows and movies. The number of movies was greater than TV shows before the decrease. Netflix content experienced a sharp decrease in 2019, and the number of TV shows exceeded movies for the first time.
2. Anupam Kher, Rupa Bhimani, Takahiro Sakurai, Julie Tejwani,Om Puri are popular casts and Rajiv Chilaka, Jan Suter, Raúl Campos, Suhas Kadav, Marcus Raboy are popular directors.
3. United States, India and United Kingdom are top countries for movies and TV shows.
4. TV-MA, TV-14, TV-PG, R are top ratings in Netflix contents.
5. International movies, Dramas, Comedies, International TV shows are very popular genres.

Recommendations

1. According to above analysis,it is advisable for Netflix to prioritize future collaborations with renowned directors, casts, genres and contents.
2. Rajiv Chilaka, Jan Suter are considered as best directors according to more number of Movie releases i.e 22, 21 respectively. So these directors movies could help in grow bussiness for netflix
3. International Movies, Dramas, Comedies, are mostly viewed genre in Netflix, So movies with these genre has more demand.
4. United States, India, United Kingdom these countries have the highest releases. So their is no risk of amount burn either in marketing or sales because its already done.
5. Movies having duration between 90-120 mins are recommended.