In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import random

import warnings
warnings.filterwarnings("ignore")

sns.set_style("darkgrid", {"grid.color": ".6", "grid.linestyle": ":"})

```

In [2]:
```
df = pd.read_csv("Netflix Data Exploration and Vizualization.csv")

```

Problem Statement

a. Which type of shows/movies to produce: Understanding the preferences and trends of viewers to create content that attracts more subscribers and retains existing ones.

b. How to grow the business in different countries: Identifying opportunities to expand the platform's reach in various countries

Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary

In [3]:
```
df.shape
```

Out[3]: (8807, 12)

In [4]:
```
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 [5]:
```
df.head(10)
```

Out[5]:

	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...
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	September 24, 2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
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	125 min	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...
8	s9	TV Show	The Great British Baking Show	Andy Devonshire	Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho...	United Kingdom	September 24, 2021	2021	TV-14	9 Seasons	British TV Shows, Reality TV	A talented batch of amateur bakers face off in...
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	2021	PG-13	104 min	Comedies, Dramas	A woman adjusting to life after a loss contend...

1. We can convert `date_added` to datetime, then extract yearly, monthly, weekly columns
2. Convert Duration into numerical column
3. Need to unnest the cast, director, country and listed_in columns
4. We can drop Description and Title column as much analysis cannot be gathered from these two columns

In [6]:
```
df.head()
```

Out[6]:

	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...

Let's check how much missing data is present:

In [7]:
```
df.isna().sum()
```

Out[7]:
```
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 [8]:
```
(df.isna().sum()/len(df))*100
```

Out[8]:
```
show_id          0.000000
type             0.000000
title            0.000000
director        29.908028
cast             9.367549
country          9.435676
date_added       0.113546
release_year     0.000000
rating           0.045418
duration         0.034064
listed_in        0.000000
description      0.000000
dtype: float64
```

We can see almost 30% of director data and approx 10% of both cast and country rare missing, Except the above mentioned columns date_added, duration and rating has some missing values but they don't amount to much

Let's check if any row is duplicated?

In [9]:
```
df.duplicated().sum()
```

Out[9]:
```
0
```

Let's check some statistical data

In [10]:
```
df.describe()
```

Out[10]:

	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

Min value of `release_year` is 1925, so some TV Shows or Movies are present that are almost 95 years old

Only 25% of records that are present in this dataset were released before 2013. So,we have a lot of data that were released in the past decade

In [11]:
```
df.describe(include = 'object')
```

Out[11]:

	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

Rajiv Chilaka is has directed most Movies or TV Shows

Most of the TV Shows or Movies were available in United States

David Attenborough has worked in most Movies or TV Shows

Even this particular `"Paranormal activity at a lush...."` description has been repeated four times in Movies/TV Shows. It can suspected that other descrptions are also be repeated

One thing to Note as we have not yet unnested the data these above basic insights might not hold true

In [12]:
```
df.loc[df.duplicated('description',keep = False)].sort_values('description')
```

...

Description column helped to find the repeated Movies/TV Shows or the Movies/TV Shows that were released in other languages

Handling Missing Data

In [13]:
```
df1 = df.copy()
```

In [14]:
```
df.isna().sum()
```

Out[14]:
```
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 [15]:
```
df['country']=df['country'].fillna('Unknown Country')
df['cast']=df['cast'].fillna('Unknown Actor')
df['director'] = df['director'].fillna('Unknown Director')
df['listed_in']  = df['listed_in'].fillna('Unknown Genre')
df['rating'] = df['rating'].fillna('Unknown Rating')
df['duration'] = df['duration'].fillna(0)
df['date_added'] = df['date_added'].fillna(df['date_added'].mode()[0])
```

In [16]:
```
df.isna().sum()
```

Out[16]:
```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

Unnesting the Columns

In [17]:
```
final_df = df.copy()
```

In [18]:
```
def remove_spaces(x):
    if x != x:
        return np.nan
    return x.strip()

def unnesting (new_df,col):

    dataframe =new_df.copy()
    dataframe[col] = dataframe[col].str.split(',')
    dataframe = dataframe.explode(col)
    dataframe[col] = dataframe[col].apply(remove_spaces)
    return dataframe
```

In [19]:
```
final_df.shape
```

Out[19]:
```
(8807, 12)
```

In [20]:
```
%%time
final_df = unnesting(final_df,'cast')
print('After splitting cast into muliple rows', final_df.shape)
final_df = unnesting(final_df,'country')
print('After splitting country into muliple rows', final_df.shape)
final_df = unnesting(final_df,'listed_in')
print('After splitting listed_in into muliple rows', final_df.shape)
final_df = unnesting(final_df,'director')
print('After splitting listed_in into muliple rows', final_df.shape)


final_df = final_df.reset_index(drop = True)
```

```
After splitting cast into muliple rows (64951, 12)
After splitting country into muliple rows (81741, 12)
After splitting listed_in into muliple rows (186399, 12)
After splitting listed_in into muliple rows (202065, 12)
CPU times: total: 1.02 s
Wall time: 1.09 s
```

Feature Engineering

Converted Date added to DateTime column and extracted dayname, day, month, year and week of the year

In [21]:
```
final_df['date_added'] = pd.to_datetime(final_df['date_added'])
final_df['dayname'] = final_df['date_added'].dt.day_name()
final_df['day'] = final_df['date_added'].dt.day
final_df['month'] = final_df['date_added'].dt.month_name()
final_df['year'] = final_df['date_added'].dt.year
final_df['week'] = final_df['date_added'].dt.isocalendar().week
final_df['year_diff'] = final_df['year'] - final_df['release_year']
final_df.drop(columns=['date_added'],inplace = True)
```

Converted Duration column from object to numerical column

In [22]:
```
#converting the duration from object type to float
final_df['duration'] = final_df['duration'].str.split(' ',expand = True)[0].astype('float')
```

Dividing the dataset into two categories Movies and Shows

In [23]:
```
movies = final_df[final_df['type'] =='Movie']
shows = final_df[final_df['type'] == 'TV Show']
```

Data types of all the attributes of unnested data:

In [24]:
```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 202065 entries, 0 to 202064
Data columns (total 17 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   show_id       202065 non-null  object 
 1   type          202065 non-null  object 
 2   title         202065 non-null  object 
 3   director      202065 non-null  object 
 4   cast          202065 non-null  object 
 5   country       202065 non-null  object 
 6   release_year  202065 non-null  int64  
 7   rating        202065 non-null  object 
 8   duration      202062 non-null  float64
 9   listed_in     202065 non-null  object 
 10  description   202065 non-null  object 
 11  dayname       202065 non-null  object 
 12  day           202065 non-null  int64  
 13  month         202065 non-null  object 
 14  year          202065 non-null  int64  
 15  week          202065 non-null  UInt32 
 16  year_diff     202065 non-null  int64  
dtypes: UInt32(1), float64(1), int64(4), object(11)
memory usage: 25.6+ MB
```

Missing values in unnested data

In [25]:
```
final_df.isna().sum()
```

Out[25]:
```
show_id         0
type            0
title           0
director        0
cast            0
country         0
release_year    0
rating          0
duration        3
listed_in       0
description     0
dayname         0
day             0
month           0
year            0
week            0
year_diff       0
dtype: int64
```

Statistical Summary in unnested data:

In [26]:
```
1 final_df.describe()
```

Out[26]:

	release_year	duration	day	year	week	year_diff
count	202065.000000	202062.000000	202065.000000	202065.000000	202065.0	202065.000000
mean	2013.448950	77.687873	12.174686	2018.966248	26.678217	5.517299
std	9.013616	51.482097	9.848247	1.551317	15.062558	9.064940
min	1925.000000	1.000000	1.000000	2008.000000	1.0	-3.000000
25%	2012.000000	4.000000	1.000000	2018.000000	14.0	0.000000
50%	2016.000000	95.000000	12.000000	2019.000000	27.0	2.000000
75%	2019.000000	112.000000	20.000000	2020.000000	39.0	7.000000
max	2021.000000	312.000000	31.000000	2021.000000	53.0	93.000000

In [27]:
```
1 final_df.describe(include = 'object')
```

Out[27]:

	show_id	type	title	director	cast	country	rating	listed_in	description	dayname	month
count	202065	202065	202065	202065	202065	202065	202065	202065	202065	202065	202065
unique	8807	2	8807	4994	36440	124	18	42	8775	7	12
top	s7165	Movie	Kahlil Gibran's The Prophet	Unknown Director	Unknown Actor	United States	TV-MA	Dramas	A troubled young girl and her mother find sola...	Friday	July
freq	700	145917	700	50643	2149	59350	73915	29806	700	58028	20302

Here we cannot derive much inferences as due to nesting many records are duplicated

In [28]:
```
1 movies.describe()
```

Out[28]:

	release_year	duration	day	year	week	year_diff
count	145917.000000	145914.000000	145917.000000	145917.000000	145917.0	145917.000000
mean	2012.130663	106.840385	11.544419	2018.956612	26.280427	6.825949
std	9.816535	24.709395	9.819765	1.539319	15.097225	9.973533
min	1942.000000	3.000000	1.000000	2008.000000	1.0	-1.000000
25%	2010.000000	93.000000	1.000000	2018.000000	13.0	1.000000
50%	2016.000000	104.000000	10.000000	2019.000000	27.0	3.000000
75%	2018.000000	119.000000	19.000000	2020.000000	39.0	9.000000
max	2021.000000	312.000000	31.000000	2021.000000	53.0	75.000000

In [29]:
```
1 shows.describe()
```

Out[29]:

	release_year	duration	day	year	week	year_diff
count	56148.000000	56148.000000	56148.00000	56148.000000	56148.0	56148.000000
mean	2016.874902	1.928101	13.81262	2018.991291	27.711993	2.116389
std	5.069625	1.811729	9.73295	1.581810	14.922722	4.604624
min	1925.000000	1.000000	1.00000	2008.000000	1.0	-3.000000
25%	2016.000000	1.000000	5.00000	2018.000000	15.0	0.000000
50%	2018.000000	1.000000	14.00000	2019.000000	27.0	0.000000
75%	2020.000000	2.000000	22.00000	2020.000000	40.0	2.000000
max	2021.000000	17.000000	31.00000	2021.000000	53.0	93.000000

Non-Graphical Analysis: Value counts and unique attributes

In [30]:
```
1 df.columns
```

Out[30]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

In [31]:
```
# this function is to bold python output
def bold_text(text):
    bold_start = '\033[1m'
    bold_end = '\033[0m'
    return bold_start + text + bold_end
```

In [32]:
```
cols_list = ['type','director','cast','country','release_year','rating','duration','listed_in']
```

Value counts and unique attributes in original data

In [33]:
```
for i in cols_list:
    print(bold_text(i.upper()+':'))
    print(f'Number of unique elements in {i} is:\n {df[i].nunique()}\n')
    print(f'Unique elements present in {i} column is:\n {df[i].unique()}\n')
    print(f'Value Counts of {i} columns is:\n{df[i].value_counts()}\n\n\n')
```

TYPE:

```
Number of unique elements in type is:
 2

Unique elements present in type column is:
 ['Movie' 'TV Show']

Value Counts of type columns is:
Movie      6131
TV Show    2676
Name: type, dtype: int64
```

DIRECTOR:

```
Number of unique elements in director is:
 4529

Unique elements present in director column is:
 ['Kirsten Johnson' 'Unknown Director' 'Julien Leclercq' ...
 'Majid Al Ansari' 'Peter Hewitt' 'Mozez Singh']

Value Counts of director columns is:
Unknown Director                  2634
Rajiv Chilaka                       19
Raúl Campos, Jan Suter              18
Suhas Kadav                         16
Marcus Raboy                        16
                                  ...
Raymie Muzquiz, Stu Livingston       1
Joe Menendez                         1
Eric Bross                           1
Will Eisenberg                       1
Mozez Singh                          1
Name: director, Length: 4529, dtype: int64
```

CAST:

```
Number of unique elements in cast is:
 7693

Unique elements present in cast column is:
 ['Unknown Actor'
 'Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmore Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake Ka-Ncube, Odwa Gwanya, Mekaila Mathys, Sandi Schultz, Duane Williams, Shamilla Miller, Patrick Mofokeng'
 'Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabiha Akkari, Sofia Lesaffre, Salim Kechiouche, Noureddine Farihi, Geert Van Rampelberg, Bakary Diombera'
 ...
 'Jesse Eisenberg, Woody Harrelson, Emma Stone, Abigail Breslin, Amber Heard, Bill Murray, Derek Graf'
 'Tim Allen, Courteney Cox, Chevy Chase, Kate Mara, Ryan Newman, Michael Cassidy, Spencer Breslin, Rip Torn, Kevin Zegers'
 'Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna Malik, Malkeet Rauni, Anita Shabdish, Chittaranjan Tripathy']

Value Counts of cast columns is:
Unknown Actor
825
David Attenborough
19
Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mousam, Swapnil
14
Samuel West
10
Jeff Dunham
7

...
Nick Lachey, Vanessa Lachey
1
Takeru Sato, Kasumi Arimura, Haru, Kentaro Sakaguchi, Takayuki Yamada, Kendo Kobayashi, Ken Yasuda, Arata Furuta, 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 Adebayo, Bayray McNwizu, Biodun Stephen                                                 1
Neeraj Kabi, Geetanjali Kulkarni, Danish Husain, Sheeba Chaddha, Paras Priyadarshan, Anshul Chauhan, Anud Singh Dhaka, Shirin Sewani, Mihir Ahuja, Vasundhara Rajput                            1
Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna Malik, Malkeet Rauni, Anita Shabdish, Chittaranjan Tripathy                                                          1
Name: cast, Length: 7693, dtype: int64
```

COUNTRY:

```
Number of unique elements in country is:
 749
```

```
Unique elements present in country column is:
 ['United States' 'South Africa' 'Unknown Country' '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' 'Belgium' 'United Kingdom, United States'
 'United States, United Kingdom' 'France, United States' 'South Korea'
 'Spain' 'United States, Singapore' 'United Kingdom, Australia, France'
 'United Kingdom, Australia, France, United States'
 'United States, Canada' 'Germany, United States'
 'South Africa, United States' 'United States, Mexico'
 'United States, Italy, France, Japan'
 'United States, Italy, Romania, United Kingdom'
 'Australia, United States' 'Argentina, Venezuela'
 'United States, United Kingdom, Canada' 'China, Hong Kong' 'Russia'
 'Canada' 'Hong Kong' 'United States, China, Hong Kong'
 'Italy, United States' 'United States, Germany'
 'United Kingdom, Canada, United States' ', South Korea' 'Ireland'
 'India, Nepal' 'New Zealand, Australia, France, United States' 'Italy'
 'Italy, Brazil, Greece' 'Argentina' 'Jordan' 'Colombia'
 'United States, Japan' 'Belgium, United Kingdom'
 'Switzerland, United Kingdom, Australia' 'Israel, United States'
 'Canada, United States' 'Brazil' 'Argentina, Spain' 'Taiwan'
 'United States, Nigeria' 'Bulgaria, United States'
 'Spain, United Kingdom, United States' 'United States, China'
 'United States, France' 'Spain, France, United Kingdom, United States'
 ', France, Algeria' 'Poland' 'Germany'
 'France, Israel, Germany, United States, United Kingdom' 'New Zealand'
 'Saudi Arabia' 'Thailand' 'Indonesia' 'Egypt, Denmark, Germany'
 'United States, Switzerland' 'Hong Kong, Canada, United States'
 'Kuwait, United States' 'France, Canada, United States, Spain'
 'France, Netherlands, Singapore' 'France, Belgium'
 'Ireland, United States, United Kingdom' 'Egypt' 'Malaysia' 'Israel'
 'Australia, New Zealand' 'United Kingdom, Germany' 'Belgium, Netherlands'
 'South Korea, Czech Republic' 'Australia, Germany' 'Vietnam'
 'United Kingdom, Belgium' 'United Kingdom, Australia, United States'
 'France, Japan, United States'
 'United Kingdom, Germany, Spain, United States'
 'United Kingdom, United States, France, Italy'
 'United States, Germany, Canada'
 'United States, France, Italy, United Kingdom'
 'United States, United Kingdom, Germany, Hungary'
 'United States, New Zealand' 'Sweden' 'China' 'Lebanon' 'Romania'
 'Finland, Germany' 'Lebanon, Syria' 'Philippines' 'Iceland' 'Denmark'
 'United States, India' 'Philippines, Singapore, Indonesia'
 'China, United States, Canada' 'Lebanon, United Arab Emirates'
 'Canada, United States, Denmark' 'United Arab Emirates'
 'Mexico, France, Colombia' 'Netherlands' 'Germany, United States, France'
 'United States, Bulgaria'
 'United Kingdom, France, Germany, United States' 'Norway, Denmark'
 'Syria, France, Lebanon, Qatar' 'United States, Czech Republic'
 'Mauritius' 'Canada, South Africa' 'Austria' 'Mexico, Brazil'
 'Germany, France' 'Mexico, United States'
 'United Kingdom, France, Spain, United States' 'United States, Australia'
 'United States, United Kingdom, France' 'United States, Russia'
 'United States, United Kingdom, New Zealand' 'Australia, United Kingdom'
 'Canada, Nigeria, United States'
 'France, United States, United Kingdom, Canada' 'France, United Kingdom'
 'India, United Kingdom' 'Canada, United States, Mexico'
 'United Kingdom, Germany, United States'
 'Czech Republic, United Kingdom, United States' 'China, United Kingdom'
 'Italy, United Kingdom' 'China, Taiwan'
 'United States, Brazil, Japan, Spain, India'
 'United States, China, United Kingdom' 'Cameroon'
 'Lebanon, Palestine, Denmark, Qatar' 'Japan, United States'
 'Uruguay, Germany' 'Egypt, Saudi Arabia'
 'United Kingdom, France, Poland, Germany, United States'
 'Ireland, Switzerland, United Kingdom, France, United States'
 'United Kingdom, South Africa, France'
 'Ireland, United Kingdom, France, Germany' 'Russia, United States'
 'United Kingdom, United States, France' 'United Kingdom,'
 'United States, India, United Kingdom' 'Kenya' 'Spain, Argentina'
 'India, United Kingdom, France, Qatar' 'Belgium, France'
 'Argentina, Chile' 'United States, Thailand' 'Chile, Brazil'
 'United States, Colombia' 'Canada, United States, United Kingdom'
 'Uruguay' 'Luxembourg' 'United States, Cambodia, Romania' 'Bangladesh'
 'Spain, Belgium, United States'
 'United Kingdom, United States, Australia'
 'Canada, United States, France' 'Portugal, United States'
 'Portugal, Spain' 'India, United States' 'United Kingdom, Ireland'
 'United Kingdom, Spain, United States' 'Hungary, United States'
 'United States, South Korea' 'Canada, United States, Cayman Islands'
 'India, France' 'France, Canada' 'Canada, Hungary, United States'
 'Norway' 'Canada, United Kingdom, United States'
 'United Kingdom, Germany, France, United States' 'Denmark, United States'
```

```
'Senegal' 'France, Algeria'
'United Kingdom, Finland, Germany, United States, Australia, Japan, France, Ireland'
'Philippines, Canada, United Kingdom, United States'
'Ireland, France, Iceland, United States, Mexico, Belgium, United Kingdom, Hong Kong'
'Singapore' 'Kuwait' 'United States, France, Serbia'
'United States, Italy' 'Spain, Italy'
'United States, Ireland, United Kingdom, India'
'United Kingdom, Singapore' 'Hong Kong, United States'
'United States, Malta, France, United Kingdom'
'United States, China, Canada' 'Canada, United States, Ireland'
'Lebanon, Canada, France' 'Japan, Canada, United States'
'Spain, France, Canada' 'Denmark, Singapore, Canada, United States'
'United States, France, Denmark' 'United States, China, Colombia'
'Spain, Thailand, United States' 'Mexico, Spain'
'Ireland, Luxembourg, Belgium' 'China, United States' 'Canada, Belgium'
'Canada, United Kingdom'
'Lebanon, United Arab Emirates, France, Switzerland, Germany'
'France, Belgium, Italy' 'Lebanon, United States, United Arab Emirates'
'Lebanon, France' 'France, Lebanon' 'France, Lebanon, United Kingdom'
'France, Norway, Lebanon, Belgium'
'Sweden, Czech Republic, United Kingdom, Denmark, Netherlands'
'United States, United Kingdom, India' 'Indonesia, Netherlands'
'Turkey, South Korea' 'Serbia, United States' 'Namibia'
'United Kingdom, Kenya' 'United Kingdom, France, Germany, Spain'
'United Kingdom, France, United States, Belgium, Luxembourg, China, Germany'
'Thailand, United States' 'United States, France, Canada, Belgium'
'United Kingdom, China' 'Germany, China, United Kingdom'
'Australia, New Zealand, United States'
'Hong Kong, Iceland, United States' 'France, Australia, Germany'
'United States, Belgium, Canada, France' 'South Africa, Angola'
'United States, Philippines'
'United States, United Kingdom, Canada, China'
'United States, Canada, United Kingdom' 'Turkey, United States'
'Peru, Germany, Norway' 'Mozambique' 'Brazil, France'
'China, Spain, South Korea, United States' 'Spain, Germany'
'Hong Kong, China' 'France, Belgium, Luxembourg, Cambodia,'
'United Kingdom, Australia' 'Belarus' 'Indonesia, United Kingdom'
'Switzerland, France, Belgium, United States' 'Ghana'
'Spain, France, Canada, United States' 'Chile, Italy'
'United Kingdom, Nigeria' 'Chile' 'France, Egypt' 'Egypt, France'
'France, Brazil, Spain, Belgium' 'Egypt, Algeria'
'Canada, South Korea, United States' 'Nigeria, United Kingdom'
'United States, France, Canada' 'Poland, United States'
'United Arab Emirates, Jordan, Lebanon, Saudi Arabia'
'United States, Mexico, Spain, Malta'
'Saudi Arabia, United Arab Emirates' 'Zimbabwe'
'United Kingdom, Germany, United Arab Emirates, New Zealand'
'Romania, United States' 'Canada, Nigeria'
'Saudi Arabia, Netherlands, Germany, Jordan, United Arab Emirates, United States'
'United Kingdom, Spain' 'Finland, France'
'United Kingdom, Germany, United States, France'
'India, United Kingdom, China, Canada, Japan, South Korea, United States'
'Italy, United Kingdom, France' 'United States, Mexico, Colombia'
'Turkey, India' 'Italy, Turkey' 'United Kingdom, United States, Japan'
'France, Belgium, United States' 'Puerto Rico, United States, Colombia'
'Uruguay, Argentina' 'United States, United Kingdom, Japan'
'United States, Argentina' 'United Kingdom, Italy'
'Ireland, United Kingdom'
'United Kingdom, France, Belgium, Canada, United States'
'Netherlands, Germany, Denmark, United Kingdom' 'Hungary'
'Austria, Germany' 'Taiwan, China'
'United Kingdom, United States, Ireland' 'South Korea, United States'
'Brazil, United Kingdom' 'Pakistan, United States'
'Romania, France, Switzerland, Germany' 'Romania, United Kingdom'
'France, Malta, United States' 'Cyprus'
'United Kingdom, France, Belgium, Ireland, United States'
'United States, Norway, Canada' 'Kenya, United States'
'France, South Korea, Japan, United States' 'Taiwan, Malaysia'
'Uruguay, Argentina, Germany, Spain'
'United States, United Kingdom, France, Germany, Japan'
'United States, France, Japan' 'United Kingdom, France, United States'
'Spain, France, United States' 'Indonesia, South Korea, Singapore'
'United States, Spain' 'Netherlands, Germany, Italy, Canada'
'Spain, Germany, Denmark, United States' 'Norway, Sweden'
'South Korea, Canada, United States, China' 'Argentina, Uruguay, Serbia'
'France, Japan' 'Mauritius, South Africa' 'United States, Poland'
'United Kingdom, United States, Germany, Denmark, Belgium, Japan'
'India, Germany' 'India, United Kingdom, Canada, United States'
'Philippines, United States' 'Romania, Bulgaria, Hungary'
'Uruguay, Guatemala' 'France, Senegal, Belgium' 'United Kingdom, Canada'
'Mexico, United States, Spain, Colombia' 'Canada, Norway'
'Singapore, United States' 'Finland, Germany, Belgium'
'United Kingdom, France' 'United States, Chile'
'United Kingdom, Japan, United States' 'Spain, United Kingdom'
'Argentina, United States, Mexico' 'United States, South Korea, Japan'
'Canada, Australia' 'United Kingdom, Hungary, Australia' 'Italy, Belgium'
```

```
'United States, United Kingdom, Germany' 'Switzerland'
'Singapore, Malaysia'
'France, Belgium, Luxembourg, Romania, Canada, United States'
'South Africa, Nigeria' 'Spain, France' 'United Kingdom, Hong Kong'
'Pakistan' 'Brazil, United States'
'Denmark, Brazil, France, Portugal, Sweden' 'India, Turkey'
'Malaysia, Singapore, Hong Kong' 'Philippines, Singapore'
'Australia, Canada' 'Taiwan, China, France, United States'
'Germany, Italy' 'Colombia, Peru, United Kingdom'
'Thailand, China, United States' 'Argentina, United States'
'Sweden, United States' 'Uruguay, Spain, Mexico'
'France, Luxembourg, Canada' 'Denmark, Spain' 'Chile, Argentina'
'United Kingdom, Belgium, Sweden' 'Canada, Brazil' 'Italy, France'
'Canada, Germany' 'Pakistan, United Arab Emirates' 'Ghana, United States'
'Mexico, Finland' 'United Arab Emirates, United Kingdom, India'
'Netherlands, Belgium' 'United States, Taiwan'
'Austria, Iraq, United States' 'United Kingdom, Malawi'
'Paraguay, Argentina' 'United Kingdom, Russia, United States'
'India, Pakistan' 'Indonesia, Singapore' 'Spain, Belgium'
'Iceland, Sweden, Belgium' 'Croatia' 'Uruguay, Argentina, Spain'
'United Kingdom, Ireland, United States'
'Canada, Germany, France, United States' 'United Kingdom, Japan'
'Norway, Denmark, Netherlands, Sweden' 'Hong Kong, China, United States'
'Ireland, Canada' 'Italy, Switzerland, France, Germany'
'Mexico, Netherlands' 'United States, Sweden' 'Germany, France, Russia'
'France, Iran, United States' 'United Kingdom, India'
'Russia, Poland, Serbia' 'Spain, Portugal' 'Peru' 'Mexico, Argentina'
'United Kingdom, Canada, United States, Cayman Islands'
'Indonesia, United States'
'United States, Israel, United Kingdom, Canada'
'Norway, Iceland, United States' 'Czech Republic, United States'
'United Kingdom, India, United States' 'United Kingdom, West Germany'
'India, Australia' 'United States,'
'Belgium, United Kingdom, United States' 'India, Germany, Austria'
'United States, Brazil, South Korea, Mexico, Japan, Germany'
'Spain, Mexico' 'China, Japan' 'Argentina, France'
'China, United States, United Kingdom'
'France, Luxembourg, United States' 'China, United States, Australia'
'Colombia, Mexico' 'United States, Canada, Ireland' 'Chile, Peru'
'Argentina, Italy' 'Canada, Japan, United States'
'United Kingdom, Canada, United States, Germany'
'Italy, Switzerland, Albania, Poland' 'United States, Japan, Canada'
'Cambodia' 'Italy, United States, Argentina'
'Saudi Arabia, Syria, Egypt, Lebanon, Kuwait'
'United States, Canada, Indonesia, United Kingdom, China, Singapore'
'Spain, Colombia'
'United Kingdom, South Africa, Australia, United States' 'Bulgaria'
'Argentina, Brazil, France, Poland, Germany, Denmark'
'United Kingdom, Spain, United States, Germany' 'Philippines, Qatar'
'Netherlands, Belgium, Germany, Jordan'
'United Arab Emirates, United States' 'Norway, Germany, Sweden'
'South Korea, China' 'Georgia' 'Soviet Union, India'
'Australia, United Arab Emirates' 'Canada, Germany, South Africa'
'South Korea, China, United States' 'India, Soviet Union' 'India, Mexico'
'Georgia, Germany, France' 'United Arab Emirates, Romania'
'India, Malaysia' 'Germany, Jordan, Netherlands'
'Turkey, France, Germany, Poland' 'Greece, United States'
'France, United Kingdom, United States' 'Norway, Germany'
'France, Morocco' 'Cambodia, United States' 'United States, Denmark'
'United States, Colombia, Mexico'
'United Kingdom, Italy, Israel, Peru, United States'
'Argentina, Uruguay, Spain, France'
'United Kingdom, France, United States, Belgium'
'France, Canada, China, Cambodia'
'United Kingdom, France, Belgium, United States' 'Chile, France'
'Netherlands, United States' 'France, United Kingdom, India'
'Czech Republic, Slovakia' 'Singapore, France' 'Spain, Switzerland'
'United States, Australia, China' 'South Africa, United States, Germany'
'United States, United Kingdom, Australia' 'Spain, Italy, Argentina'
'Chile, Spain, Argentina, Germany' 'West Germany'
'Austria, Czech Republic' 'Lebanon, Qatar'
'United Kingdom, Jordan, Qatar, Iran' 'France, South Korea, Japan'
'Israel, Germany, France' 'Canada, Japan, Netherlands'
'United States, Hungary' 'France, Germany' 'France, Qatar'
'United Kingdom, Germany, Canada' 'Ireland, South Africa'
'Chile, United States, France' 'Belgium, France, Netherlands'
'United Kingdom, Ukraine, United States'
'Germany, Australia, France, China' 'Norway, United States'
'United States, Bermuda, Ecuador'
'United States, Hungary, Ireland, Canada'
'United Kingdom, Egypt, United States'
'United States, France, United Kingdom' 'Spain, Mexico, France'
'United States, South Africa' 'Hong Kong, China, Singapore'
'South Africa, China, United States' 'Denmark, France, Poland'
'New Zealand, United Kingdom' 'Netherlands, Denmark, South Africa'
'Iran, France' 'United Kingdom, United States, France, Germany'
```

```
'Australia, France' 'Ireland, United Kingdom, United States'
'United Kingdom, France, Germany' 'Canada, Luxembourg'
'Brazil, Netherlands, United States, Colombia, Austria, Germany'
'France, Canada, Belgium' 'Canada, France'
'Bulgaria, United States, Spain, Canada' 'Sweden, Netherlands'
'France, United States, Mexico'
'Australia, United Kingdom, United Arab Emirates, Canada'
'Australia, Armenia, Japan, Jordan, Mexico, Mongolia, New Zealand, Philippines, South Africa, Sweden, United States, Uruguay'
'India, Iran' 'France, Belgium, Spain'
'Denmark, Sweden, Israel, United States' 'United States, Iceland'
'United Kingdom, Russia' 'United States, Israel, Italy, South Africa'
'Netherlands, Denmark, France, Germany' 'South Korea, Japan'
'United Kingdom, Pakistan' 'France, New Zealand'
'United Kingdom, Czech Republic, United States, Germany, Bahamas'
'China, Germany, India, United States' 'Germany, Sri Lanka'
'United States, India, Bangladesh' 'United States, Canada, France'
'Brazil, France, Germany' 'Germany, United States, Hong Kong, Singapore'
'France, Germany, Switzerland'
'Germany, France, Luxembourg, United Kingdom, United States'
'United Kingdom, Canada, Italy' 'Czech Republic, France'
'Taiwan, Hong Kong, United States, China' 'Germany, Australia'
'United Kingdom, Poland, United States' 'Denmark, Zimbabwe'
'United Kingdom, South Africa' 'Finland, Sweden, Norway, Latvia, Germany'
'South Africa, United States, New Zealand, Canada'
'United States, Italy, United Kingdom, Liechtenstein'
'Denmark, France, Belgium, Italy, Netherlands, United States, United Kingdom'
'United States, Australia, Mexico'
'United Kingdom, Czech Republic, Germany, United States'
'France, China, Japan, United States' 'United States, South Korea, China'
'Germany, Belgium' 'Pakistan, Norway, United States'
'United States, Canada, Belgium, United Kingdom' 'Venezuela'
'Canada, France, Italy, Morocco, United States' 'Canada, Spain, France'
'United States, Indonesia' 'Spain, France, Italy'
'United Arab Emirates, United States, United Kingdom'
'United Kingdom, Israel, Russia' 'Spain, Cuba' 'United States, Brazil'
'United States, France, Mexico' 'United States, Nicaragua'
'United Kingdom, United States, Spain, Germany, Greece, Canada'
'Italy, Canada, France' 'United Kingdom, Denmark, Canada, Croatia'
'Italy, Germany' 'United States, France, United Kingdom, Japan'
'United States, United Kingdom, Denmark, Sweden'
'United States, United Kingdom, Italy'
'United States, France, Canada, Spain' 'Russia, United States, China'
'United States, Canada, Germany' 'Ireland, United States'
'United States, United Arab Emirates' 'United States, Ireland'
'Ireland, United Kingdom, Italy, United States' 'Poland,'
'Slovenia, Croatia, Germany, Czech Republic, Qatar'
'Canada, United Kingdom, Netherlands' 'United States, Spain, Germany'
'India, Japan' 'China, South Korea, United States'
'United Kingdom, France, Belgium' 'Canada, Ireland, United States'
'United Kingdom, United States, Dominican Republic'
'United States, Senegal' 'Germany, United Kingdom, United States'
'South Africa, Germany, Netherlands, France'
'Canada, United States, United Kingdom, France, Luxembourg'
'Ireland, United States, France' 'Germany, United States, Canada'
'United Kingdom, Germany, Canada, United States'
'United States, France, Canada, Lebanon, Qatar'
'Netherlands, Belgium, United Kingdom, United States'
'France, Belgium, China, United States' 'United States, Chile, Israel'
'United Kingdom, Norway, Denmark, Germany, Sweden'
'Norway, Denmark, Sweden' 'China, India, Nepal'
'Colombia, Mexico, United States' 'United Kingdom, South Korea'
'Denmark, China' 'United States, Greece, Brazil' 'South Korea, France'
'United States, Australia, Samoa, United Kingdom'
'Germany, United Kingdom' 'Argentina, Chile, Peru' 'Turkey, Azerbaijan'
'Poland, West Germany' 'Germany, United States, Sweden' 'Canada, Spain'
'United States, Cambodia' 'United States, Greece'
'Norway, United Kingdom, France, Ireland' 'United Kingdom, Poland'
'Israel, Sweden, Germany, Netherlands' 'Switzerland, France'
'Italy, India' 'United States, Botswana'
'Chile, Argentina, France, Spain, United States'
'United States, India, South Korea, China'
'Denmark, Germany, Belgium, United Kingdom, France'
'Denmark, Germany, Belgium, United Kingdom, France, Sweden'
'France, Switzerland, Spain, United States, United Arab Emirates'
'Brazil, India, China, United States'
'Denmark, France, United States, Sweden' 'Australia, Iraq'
'China, Morocco, Hong Kong' 'Canada, United States, Germany'
'United Kingdom, Thailand' 'Venezuela, Colombia'
'Colombia, United States' 'France, Germany, Czech Republic, Belgium'
'Switzerland, Vatican City, Italy, Germany, France'
'Portugal, France, Poland, United States'
'United States, New Zealand, Japan'
'United States, Netherlands, Japan, France' 'India, Switzerland'
'Canada, India' 'United States, Morocco' 'Singapore, Japan, France'
'Canada, Mexico, Germany, South Africa'
'United Kingdom, United States, Canada'
```

```
 'Germany, France, United States, Canada, United Kingdom'
 'United States, Uruguay' 'India, Canada'
 'Ireland, Canada, United Kingdom, United States'
 'United States, Germany, Australia' 'Australia, France, Ireland'
 'Australia, India' 'United States, United Kingdom, Canada, Japan'
 'Sweden, United Kingdom, Finland' 'Hong Kong, Taiwan'
 'United States, United Kingdom, Spain, South Korea' 'Guatemala' 'Ukraine'
 'Italy, South Africa, West Germany, Australia, United States'
 'United States, Germany, United Kingdom, Australia'
 'Italy, France, Switzerland' 'Canada, France, United States'
 'Switzerland, United States' 'Thailand, Canada, United States'
 'China, Hong Kong, United States' 'United Kingdom, New Zealand'
 'Czech Republic, United Kingdom, France'
 'Australia, United Kingdom, Canada' 'Jamaica, United States'
 'Australia, United Kingdom, United States, New Zealand, Italy, France'
 'France, United States, Canada'
 'United Kingdom, France, Canada, Belgium, United States'
 'Denmark, United Kingdom, Sweden' 'United States, Hong Kong'
 'United States, Kazakhstan'
 'Argentina, France, United States, Germany, Qatar'
 'United States, Germany, United Kingdom'
 'United States, Germany, United Kingdom, Italy'
 'United States, New Zealand, United Kingdom' 'Finland, United States'
 'Spain, France, Uruguay' 'France, Canada, United States'
 'United States, Canada, China'
 'Ireland, Canada, Luxembourg, United States, United Kingdom, Philippines, India'
 'United States, Czech Republic, United Kingdom' 'Israel, Germany'
 'Mexico, France'
 'Israel, Germany, Poland, Luxembourg, Belgium, France, United States'
 'Austria, United States' 'United Kingdom, Lithuania'
 'United States, Greece, United Kingdom'
 'United Kingdom, China, United States, India'
 'United States, Sweden, Norway' 'United Kingdom, United States, Morocco'
 'United States, United Kingdom, Morocco' 'Spain, Canada, United States'
 'United States, India, United Arab Emirates'
 'United Kingdom, Canada, France, United States' 'India, Germany, France'
 'Belgium, Ireland, Netherlands, Germany, Afghanistan'
 'France, Canada, Italy, United States, China'
 'Ireland, United Kingdom, Greece, France, Netherlands'
 'Denmark, Indonesia, Finland, Norway, United Kingdom, Israel, France, United States, Germany, Netherlands'
 'New Zealand, United States'
 'United States, Australia, South Africa, United Kingdom'
 'United States, Germany, Mexico'
 'Somalia, Kenya, Sudan, South Africa, United States'
 'United States, Canada, Japan, Panama' 'United Kingdom, Spain, Belgium'
 'Serbia, South Korea, Slovenia'
 'Denmark, United Kingdom, South Africa, Sweden, Belgium'
 'Germany, Canada, United States'
 'Ireland, Canada, United States, United Kingdom'
 'New Zealand, United Kingdom, Australia'
 'United Kingdom, Australia, Canada, United States'
 'Germany, United States, Italy' 'United States, Venezuela'
 'United Kingdom, Canada, Japan'
 'United Kingdom, United States, Czech Republic'
 'United Kingdom, China, United States' 'United Kingdom, Brazil, Germany'
 'United Kingdom, Namibia, South Africa, Zimbabwe, United States'
 'Canada, United States, India, United Kingdom'
 'Switzerland, United Kingdom, United States'
 'United Kingdom, India, Sweden'
 'United States, Brazil, India, Uganda, China'
 'Peru, United States, United Kingdom'
 'Germany, United States, United Kingdom, Canada'
 'Canada, India, Thailand, United States, United Arab Emirates'
 'United States, East Germany, West Germany'
 'France, Netherlands, South Africa, Finland'
 'Egypt, Austria, United States' 'Russia, Spain'
 'Croatia, Slovenia, Serbia, Montenegro' 'Japan, Canada'
 'United States, France, South Korea, Indonesia'
 'United Arab Emirates, Jordan']

Value Counts of country columns is:
United States                              2818
India                                       972
Unknown Country                             831
United Kingdom                              419
Japan                                       245
                                           ...
Romania, Bulgaria, Hungary                    1
Uruguay, Guatemala                            1
France, Senegal, Belgium                      1
Mexico, United States, Spain, Colombia        1
United Arab Emirates, Jordan                  1
Name: country, Length: 749, dtype: int64
```

```
RELEASE_YEAR:
Number of unique elements in release_year is:
 74

Unique elements present in release_year column is:
 [2020 2021 1993 2018 1996 1998 1997 2010 2013 2017 1975 1978 1983 1987
 2012 2001 2014 2002 2003 2004 2011 2008 2009 2007 2005 2006 1994 2015
 2019 2016 1982 1989 1990 1991 1999 1986 1992 1984 1980 1961 2000 1995
 1985 1976 1959 1988 1981 1972 1964 1945 1954 1979 1958 1956 1963 1970
 1973 1925 1974 1960 1966 1971 1962 1969 1977 1967 1968 1965 1946 1942
 1955 1944 1947 1943]

Value Counts of release_year columns is:
2018    1147
2017    1032
2019    1030
2020     953
2016     902
        ...
1959       1
1925       1
1961       1
1947       1
1966       1
Name: release_year, Length: 74, dtype: int64



RATING:
Number of unique elements in rating is:
 18

Unique elements present in rating column is:
 ['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 Rating' 'TV-Y7-FV' 'UR']

Value Counts of rating columns is:
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
Unknown Rating       4
NC-17                3
UR                   3
74 min               1
84 min               1
66 min               1
Name: rating, dtype: int64



DURATION:
Number of unique elements in duration is:
 221

Unique elements present in duration column is:
 ['90 min' '2 Seasons' '1 Season' '91 min' '125 min' '9 Seasons' '104 min'
 '127 min' '4 Seasons' '67 min' '94 min' '5 Seasons' '161 min' '61 min'
 '166 min' '147 min' '103 min' '97 min' '106 min' '111 min' '3 Seasons'
 '110 min' '105 min' '96 min' '124 min' '116 min' '98 min' '23 min'
 '115 min' '122 min' '99 min' '88 min' '100 min' '6 Seasons' '102 min'
 '93 min' '95 min' '85 min' '83 min' '113 min' '13 min' '182 min' '48 min'
 '145 min' '87 min' '92 min' '80 min' '117 min' '128 min' '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' '107 min' '129 min' '135 min'
 '136 min' '165 min' '150 min' '133 min' '70 min' '84 min' '140 min'
 '78 min' '7 Seasons' '64 min' '59 min' '139 min' '69 min' '148 min'
 '189 min' '141 min' '130 min' '138 min' '81 min' '132 min' '10 Seasons'
 '123 min' '65 min' '68 min' '66 min' '62 min' '74 min' '131 min' '39 min'
 '46 min' '38 min' '8 Seasons' '17 Seasons' '126 min' '155 min' '159 min'
 '137 min' '12 min' '273 min' '36 min' '34 min' '77 min' '60 min' '49 min'
 '58 min' '72 min' '204 min' '212 min' '25 min' '73 min' '29 min' '47 min'
 '32 min' '35 min' '71 min' '149 min' '33 min' '15 min' '54 min' '224 min'
 '162 min' '37 min' '75 min' '79 min' '55 min' '158 min' '164 min'
 '173 min' '181 min' '185 min' '21 min' '24 min' '51 min' '151 min'
 '42 min' '22 min' '134 min' '177 min' '13 Seasons' '52 min' '14 min'
 '53 min' '8 min' '57 min' '28 min' '50 min' '9 min' '26 min' '45 min'
```

```
 '171 min' '27 min' '44 min' '146 min' '20 min' '157 min' '17 min'
 '203 min' '41 min' '30 min' '194 min' '15 Seasons' '233 min' '237 min'
 '230 min' '195 min' '253 min' '152 min' '190 min' '160 min' '208 min'
 '180 min' '144 min' '5 min' '174 min' '170 min' '192 min' '209 min'
 '187 min' '172 min' '16 min' '186 min' '11 min' '193 min' '176 min'
 '56 min' '169 min' '40 min' '10 min' '3 min' '168 min' '312 min'
 '153 min' '214 min' '31 min' '163 min' '19 min' '12 Seasons' 0 '179 min'
 '11 Seasons' '43 min' '200 min' '196 min' '167 min' '178 min' '228 min'
 '18 min' '205 min' '201 min' '191 min']

Value Counts of duration columns is:
1 Season     1793
2 Seasons     425
3 Seasons     199
90 min        152
94 min        146
             ...
189 min         1
10 min          1
3 min           1
229 min         1
191 min         1
Name: duration, Length: 221, dtype: int64
```

LISTED_IN:

```
Number of unique elements in listed_in is:
 514

Unique elements present in listed_in column is:
 ['Documentaries' 'International TV Shows, TV Dramas, TV Mysteries'
 'Crime TV Shows, International TV Shows, TV Action & Adventure'
 'Docuseries, Reality TV'
 'International TV Shows, Romantic TV Shows, TV Comedies'
 'TV Dramas, TV Horror, TV Mysteries' 'Children & Family Movies'
 'Dramas, Independent Movies, International Movies'
 'British TV Shows, Reality TV' 'Comedies, Dramas'
 'Crime TV Shows, Docuseries, International TV Shows'
 'Dramas, International Movies' 'Children & Family Movies, Comedies'
 'British TV Shows, Crime TV Shows, Docuseries' 'TV Comedies, TV Dramas'
 'Documentaries, International Movies'
 'Crime TV Shows, Spanish-Language TV Shows, TV Dramas' 'Thrillers'
 'International TV Shows, Spanish-Language TV Shows, TV Action & Adventure'
 'International TV Shows, TV Action & Adventure, TV Dramas'
 'Comedies, International Movies'
 'Comedies, International Movies, Romantic Movies'
 'Docuseries, International TV Shows, Reality TV'
 'Comedies, International Movies, Music & Musicals' 'Comedies'
 'Horror Movies, Sci-Fi & Fantasy' 'TV Comedies'
 'British TV Shows, International TV Shows, TV Comedies'
 'International TV Shows, TV Dramas, TV Thrillers' "Kids' TV"
 'Dramas, International Movies, Thrillers'
 'Action & Adventure, Dramas, International Movies'
 "Kids' TV, TV Comedies" 'Action & Adventure, Dramas'
 "Kids' TV, TV Sci-Fi & Fantasy"
 'Action & Adventure, Classic Movies, Dramas'
 'Dramas, Horror Movies, Thrillers'
 'Action & Adventure, Horror Movies, Thrillers' 'Action & Adventure'
 'Dramas, Thrillers' 'International TV Shows, TV Dramas'
 'International TV Shows, TV Dramas, TV Sci-Fi & Fantasy'
 'Action & Adventure, Anime Features, International Movies' 'Reality TV'
 'Docuseries, International TV Shows'
 'Documentaries, International Movies, Sports Movies'
 'International TV Shows, Reality TV, Romantic TV Shows'
 'British TV Shows, Docuseries, International TV Shows'
 'Anime Series, International TV Shows'
 'Comedies, Dramas, International Movies'
 'Crime TV Shows, TV Comedies, TV Dramas'
 'Action & Adventure, Comedies, Dramas' "Anime Series, Kids' TV"
 'International Movies, Thrillers' "Kids' TV, Korean TV Shows"
 'Documentaries, Sports Movies' 'Sci-Fi & Fantasy, Thrillers'
 'Dramas, International Movies, Romantic Movies'
 'Documentaries, Music & Musicals'
 "Kids' TV, TV Comedies, TV Sci-Fi & Fantasy" "British TV Shows, Kids' TV"
 'Docuseries, Science & Nature TV' 'Children & Family Movies, Dramas'
 "Kids' TV, TV Dramas, Teen TV Shows"
 'Crime TV Shows, International TV Shows, Spanish-Language TV Shows'
 'Docuseries, International TV Shows, Spanish-Language TV Shows' 'Dramas'
 'Comedies, Romantic Movies' 'Dramas, Romantic Movies'
 'Comedies, Dramas, Independent Movies'
 'Crime TV Shows, TV Action & Adventure, TV Comedies'
 'Children & Family Movies, Music & Musicals'
 'Action & Adventure, Classic Movies, Cult Movies'
 'International TV Shows, TV Action & Adventure, TV Comedies'
 'Action & Adventure, Sci-Fi & Fantasy' 'Action & Adventure, Comedies'
```

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

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

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

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

```
 'Action & Adventure, Classic Movies, Sci-Fi & Fantasy'
 'TV Action & Adventure, TV Comedies'
 'Classic Movies, Comedies, Music & Musicals' 'Independent Movies'
 'Documentaries, Horror Movies'
 'Classic & Cult TV, TV Horror, TV Mysteries'
 'Comedies, Faith & Spirituality, International Movies'
 'Dramas, Horror Movies, Sci-Fi & Fantasy'
 'British TV Shows, TV Dramas, TV Sci-Fi & Fantasy'
 'Comedies, Cult Movies, Horror Movies'
 'Comedies, Cult Movies, Sports Movies' 'Classic Movies, Documentaries'
 'Action & Adventure, Faith & Spirituality, Sci-Fi & Fantasy'
 'Action & Adventure, Children & Family Movies'
 'International TV Shows, Reality TV, TV Action & Adventure'
 'Docuseries, Science & Nature TV, TV Dramas' 'Anime Features'
 'Action & Adventure, Horror Movies, Independent Movies'
 'Action & Adventure, Classic Movies, International Movies'
 'Cult Movies, Independent Movies, Thrillers'
 'Crime TV Shows, TV Comedies'
 'Classic Movies, Cult Movies, Documentaries'
 "Classic & Cult TV, Kids' TV, TV Comedies"
 'Classic Movies, Dramas, LGBTQ Movies'
 'Classic Movies, Dramas, Sports Movies' 'Action & Adventure, Cult Movies'
 'Action & Adventure, Comedies, Music & Musicals'
 'Classic Movies, Horror Movies, Thrillers'
 'Classic Movies, Comedies, Independent Movies'
 'Children & Family Movies, Classic Movies, Dramas'
 'Dramas, Faith & Spirituality, Sports Movies'
 'Classic Movies, Comedies, Romantic Movies'
 'Dramas, Horror Movies, Music & Musicals'
 'Classic Movies, Independent Movies, Thrillers'
 'Children & Family Movies, Faith & Spirituality'
 'Classic Movies, Comedies, Sports Movies'
 'Comedies, Dramas, Sports Movies'
 'Action & Adventure, Romantic Movies, Sci-Fi & Fantasy'
 'Classic & Cult TV, TV Sci-Fi & Fantasy'
 'Comedies, Cult Movies, LGBTQ Movies'
 'Comedies, Horror Movies, Sci-Fi & Fantasy'
 'Action & Adventure, Comedies, Horror Movies'
 'Classic & Cult TV, Crime TV Shows, TV Dramas'
 'Action & Adventure, Documentaries, Sports Movies'
 'International Movies, LGBTQ Movies, Romantic Movies'
 'Cult Movies, Dramas, Thrillers']

Value Counts of listed_in columns is:
Dramas, International Movies                          362
Documentaries                                         359
Stand-Up Comedy                                       334
Comedies, Dramas, International Movies                274
Dramas, Independent Movies, International Movies      252
                                                     ...
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, Length: 514, dtype: int64
```

Value counts and unique attributes in unnested data

In [34]:
```
1 final_df.columns
```

Out[34]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country',
       'release_year', 'rating', 'duration', 'listed_in', 'description',
       'dayname', 'day', 'month', 'year', 'week', 'year_diff'],
      dtype='object')
```

In [35]:
```
1 cols_list = ['type','rating','director','cast','country','listed_in','release_year','year','week','month','day','dayname']
```

In [36]:

```
for i in cols_list:
    print(bold_text(i.upper()+':'))
    print(f'Number of unique elements in {i} is:\n {final_df[i].nunique()}\n')
    print(f'Unique elements present in {i} column is:\n {final_df[i].unique()}\n')
    print(f'Value Counts of {i} columns is:\n{final_df[i].value_counts()}\n\n\n')
```

```
TYPE:
Number of unique elements in type is:
 2

Unique elements present in type column is:
 ['Movie' 'TV Show']

Value Counts of type columns is:
Movie      145917
TV Show     56148
Name: type, dtype: int64
```

```
RATING:
Number of unique elements in rating is:
 18

Unique elements present in rating column is:
 ['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 Rating' 'TV-Y7-FV' 'UR']

Value Counts of rating columns is:
TV-MA             73915
TV-14             43957
R                 25860
PG-13             16246
TV-PG             14926
PG                10919
TV-Y7              6304
TV-Y               3665
TV-G               2779
NR                 1573
G                  1530
NC-17               149
TV-Y7-FV             86
UR                   86
Unknown Rating       67
74 min                1
84 min                1
66 min                1
Name: rating, dtype: int64
```

```
DIRECTOR:
Number of unique elements in director is:
 4994

Unique elements present in director column is:
 ['Kirsten Johnson' 'Unknown Director' 'Julien Leclercq' ...
 'Majid Al Ansari' 'Peter Hewitt' 'Mozez Singh']

Value Counts of director columns is:
Unknown Director       50643
Martin Scorsese          419
Youssef Chahine          409
Cathy Garcia-Molina      356
Steven Spielberg         355
                       ...
Brendon Marotta            1
Charlie Siskel             1
Adam Bolt                  1
Anthony Palmer             1
Kirsten Johnson            1
Name: director, Length: 4994, dtype: int64
```

```
CAST:
Number of unique elements in cast is:
 36440

Unique elements present in cast column is:
 ['Unknown Actor' 'Ama Qamata' 'Khosi Ngema' ... 'Malkeet Rauni'
 'Anita Shabdish' 'Chittaranjan Tripathy']

Value Counts of cast columns is:
Unknown Actor        2149
Liam Neeson           161
Alfred Molina         160
John Krasinski        139
Salma Hayek           130
                     ...
Peter Dunning           1
Benjamin Bradley        1
```

```
Drew Ray Tanner         1
Rena Strober            1
Hrishikesh Hirway       1
Name: cast, Length: 36440, dtype: int64
```

COUNTRY:

```
Number of unique elements in country is:
 124

Unique elements present in country column is:
 ['United States' 'South Africa' 'Unknown Country' 'India' 'Ghana'
 'Burkina Faso' 'United Kingdom' 'Germany' 'Ethiopia' 'Czech Republic'
 'Mexico' 'Turkey' 'Australia' 'France' 'Finland' 'China' 'Canada' 'Japan'
 'Nigeria' 'Spain' 'Belgium' 'South Korea' 'Singapore' 'Italy' 'Romania'
 'Argentina' 'Venezuela' 'Hong Kong' 'Russia' '' 'Ireland' 'Nepal'
 'New Zealand' 'Brazil' 'Greece' 'Jordan' 'Colombia' 'Switzerland'
 'Israel' 'Taiwan' 'Bulgaria' 'Algeria' 'Poland' 'Saudi Arabia' 'Thailand'
 'Indonesia' 'Egypt' 'Denmark' 'Kuwait' 'Netherlands' 'Malaysia' 'Vietnam'
 'Hungary' 'Sweden' 'Lebanon' 'Syria' 'Philippines' 'Iceland'
 'United Arab Emirates' 'Norway' 'Qatar' 'Mauritius' 'Austria' 'Cameroon'
 'Palestine' 'Uruguay' 'Kenya' 'Chile' 'Luxembourg' 'Cambodia'
 'Bangladesh' 'Portugal' 'Cayman Islands' 'Senegal' 'Serbia' 'Malta'
 'Namibia' 'Angola' 'Peru' 'Mozambique' 'Belarus' 'Zimbabwe' 'Puerto Rico'
 'Pakistan' 'Cyprus' 'Guatemala' 'Iraq' 'Malawi' 'Paraguay' 'Croatia'
 'Iran' 'West Germany' 'Albania' 'Georgia' 'Soviet Union' 'Morocco'
 'Slovakia' 'Ukraine' 'Bermuda' 'Ecuador' 'Armenia' 'Mongolia' 'Bahamas'
 'Sri Lanka' 'Latvia' 'Liechtenstein' 'Cuba' 'Nicaragua' 'Slovenia'
 'Dominican Republic' 'Samoa' 'Azerbaijan' 'Botswana' 'Vatican City'
 'Jamaica' 'Kazakhstan' 'Lithuania' 'Afghanistan' 'Somalia' 'Sudan'
 'Panama' 'Uganda' 'East Germany' 'Montenegro']

Value Counts of country columns is:
United States      59350
India              22814
United Kingdom     12965
Unknown Country    11897
Japan               8679
                   ...
Panama                 2
Mongolia               2
Kazakhstan             1
Nicaragua              1
Uganda                 1
Name: country, Length: 124, dtype: int64
```

LISTED_IN:

```
Number of unique elements in listed_in is:
 42

Unique elements present in listed_in column is:
 ['Documentaries' 'International TV Shows' 'TV Dramas' 'TV Mysteries'
 'Crime TV Shows' 'TV Action & Adventure' 'Docuseries' 'Reality TV'
 'Romantic TV Shows' 'TV Comedies' 'TV Horror' 'Children & Family Movies'
 'Dramas' 'Independent Movies' 'International Movies' 'British TV Shows'
 'Comedies' 'Spanish-Language TV Shows' 'Thrillers' 'Romantic Movies'
 'Music & Musicals' 'Horror Movies' 'Sci-Fi & Fantasy' 'TV Thrillers'
 "Kids' TV" 'Action & Adventure' 'TV Sci-Fi & Fantasy' 'Classic Movies'
 'Anime Features' 'Sports Movies' 'Anime Series' 'Korean TV Shows'
 'Science & Nature TV' 'Teen TV Shows' 'Cult Movies' 'TV Shows'
 'Faith & Spirituality' 'LGBTQ Movies' 'Stand-Up Comedy' 'Movies'
 'Stand-Up Comedy & Talk Shows' 'Classic & Cult TV']

Value Counts of listed_in columns is:
Dramas                          29806
International Movies            28243
Comedies                        20829
International TV Shows          12845
Action & Adventure              12216
Independent Movies               9834
Children & Family Movies         9771
TV Dramas                        8942
Thrillers                        7107
Romantic Movies                  6412
TV Comedies                      4963
Crime TV Shows                   4733
Horror Movies                    4571
Kids' TV                         4568
Sci-Fi & Fantasy                 4037
Music & Musicals                 3077
Romantic TV Shows                3049
Documentaries                    2409
Anime Series                     2313
```

```
TV Action & Adventure            2288
Spanish-Language TV Shows        2126
British TV Shows                 1808
Sports Movies                    1531
Classic Movies                   1443
TV Mysteries                     1281
Korean TV Shows                  1122
Cult Movies                      1077
TV Sci-Fi & Fantasy              1045
Anime Features                   1045
TV Horror                         941
Docuseries                        845
LGBTQ Movies                      838
TV Thrillers                      768
Teen TV Shows                     742
Reality TV                        735
Faith & Spirituality              719
Stand-Up Comedy                   540
Movies                            412
TV Shows                          337
Classic & Cult TV                 272
Stand-Up Comedy & Talk Shows      268
Science & Nature TV               157
Name: listed_in, dtype: int64
```

RELEASE_YEAR:

```
Number of unique elements in release_year is:
 74

Unique elements present in release_year column is:
 [2020 2021 1993 2018 1996 1998 1997 2010 2013 2017 1975 1978 1983 1987
 2012 2001 2014 2002 2003 2004 2011 2008 2009 2007 2005 2006 1994 2015
 2019 2016 1982 1989 1990 1991 1999 1986 1992 1984 1980 1961 2000 1995
 1985 1976 1959 1988 1981 1972 1964 1945 1954 1979 1958 1956 1963 1970
 1973 1925 1974 1960 1966 1971 1962 1969 1977 1967 1968 1965 1946 1942
 1955 1944 1947 1943]

Value Counts of release_year columns is:
2018    24441
2019    21931
2017    20516
2020    19697
2016    18465
        ...
1947        8
1946        6
1942        6
1943        5
1925        1
Name: release_year, Length: 74, dtype: int64
```

YEAR:

```
Number of unique elements in year is:
 14

Unique elements present in year column is:
 [2021 2020 2019 2018 2017 2016 2015 2014 2013 2012 2011 2009 2008 2010]

Value Counts of year columns is:
2019    46997
2020    46209
2021    36534
2018    35785
2017    25209
2016     8569
2015     1560
2014      452
2011      438
2013      207
2012       36
2009       30
2010       20
2008       19
Name: year, dtype: int64
```

WEEK:

```
Number of unique elements in week is:
 53

Unique elements present in week column is:
```

```
 <IntegerArray>
[38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20,
 19, 18, 17, 16, 15, 14, 13, 12, 11, 10,  9,  8,  7,  6,  5,  4,  3,  2,  1,
 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39]
Length: 53, dtype: UInt32

Value Counts of week columns is:
1     9632
35    6993
44    6943
26    6593
40    6267
31    6034
9     5920
27    5785
48    5250
13    5057
18    5050
5     4534
22    4387
15    4313
39    4245
23    4128
30    3993
37    3686
50    3645
7     3637
24    3622
33    3598
17    3491
51    3470
36    3465
25    3464
14    3437
28    3330
10    3258
49    3255
38    3214
11    3170
34    3166
29    3158
43    3087
42    3005
52    2970
16    2877
19    2811
20    2768
46    2674
21    2650
3     2632
41    2571
8     2514
53    2484
12    2433
2     2430
47    2418
6     2262
45    2253
32    2201
4     1835
Name: week, dtype: Int64
```

MONTH:

```
Number of unique elements in month is:
 12

Unique elements present in month column is:
 ['September' 'August' 'July' 'June' 'May' 'April' 'March' 'February'
 'January' 'December' 'November' 'October']

Value Counts of month columns is:
July         20302
January      18412
December     18266
September    18120
October      17796
August       17086
April        17081
June         16659
March        15859
November     15597
May          13827
February     13060
Name: month, dtype: int64
```

```
DAY:
Number of unique elements in day is:
 31

Unique elements present in day column is:
 [25 24 23 22 21 20 19 17 16 15 14 11 10  9  8  7  6  5  4  3  2  1 31 29
 28 27 26 18 13 12 30]

Value Counts of day columns is:
1     53594
15    15709
16     8159
2      7113
19     5854
5      5631
20     5322
8      5286
6      5204
31     5194
27     5076
26     5035
30     4953
25     4616
10     4577
7      4480
14     4367
23     4298
22     4270
18     4195
4      4186
21     4151
28     3898
9      3643
17     3565
24     3517
13     3421
12     3415
11     3384
3      3036
29     2916
Name: day, dtype: int64


DAYNAME:
Number of unique elements in dayname is:
 7

Unique elements present in dayname column is:
 ['Saturday' 'Friday' 'Thursday' 'Wednesday' 'Tuesday' 'Monday' 'Sunday']

Value Counts of dayname columns is:
Friday       58028
Thursday     33432
Wednesday    29092
Tuesday      24002
Saturday     19278
Sunday       19246
Monday       18987
Name: dayname, dtype: int64
```

In [ ]:
```
1
```

In [37]:
```
1 cols_list = ['type','rating','director','cast','country','listed_in','release_year','year','week','month','day','dayname','d
```

In [38]:

```
for i in cols_list:
    print(bold_text(i.upper()+':'))
    print(f'Number of unique elements in {i} is:\n {movies[i].nunique()}\n')
    print(f'Unique elements present in {i} column is:\n {movies[i].unique()}\n')
    print(f'Value Counts of {i} columns is:\n{movies[i].value_counts()}\n\n\n')
```

```
TYPE:
Number of unique elements in type is:
 1

Unique elements present in type column is:
 ['Movie']

Value Counts of type columns is:
Movie    145917
Name: type, dtype: int64



RATING:
Number of unique elements in rating is:
 18

Unique elements present in rating column is:
 ['PG-13' 'PG' 'TV-MA' 'TV-PG' 'TV-14' 'TV-Y' 'R' 'TV-G' 'TV-Y7' 'G'
 'NC-17' '74 min' '84 min' '66 min' 'NR' 'Unknown Rating' 'TV-Y7-FV' 'UR']

Value Counts of rating columns is:
TV-MA             44009
TV-14             29266
R                 25806
PG-13             16246
PG                10919
TV-PG             10312
TV-Y7              2486
TV-Y               1878
TV-G               1738
G                  1530
NR                 1418
NC-17               149
UR                   86
TV-Y7-FV             62
Unknown Rating        9
74 min                1
84 min                1
66 min                1
Name: rating, dtype: int64



DIRECTOR:
Number of unique elements in director is:
 4778

Unique elements present in director column is:
 ['Kirsten Johnson' 'Robert Cullen' 'José Luis Ucha' ... 'Majid Al Ansari'
 'Peter Hewitt' 'Mozez Singh']

Value Counts of director columns is:
Unknown Director       1285
Martin Scorsese         419
Youssef Chahine         409
Cathy Garcia-Molina     356
Steven Spielberg        355
                       ...
John Smithson             1
Alex Coletti              1
Michael Govier            1
Sabaah Folayan            1
Kirsten Johnson           1
Name: director, Length: 4778, dtype: int64



CAST:
Number of unique elements in cast is:
 25952

Unique elements present in cast column is:
 ['Unknown Actor' 'Vanessa Hudgens' 'Kimiko Glenn' ... 'Malkeet Rauni'
 'Anita Shabdish' 'Chittaranjan Tripathy']

Value Counts of cast columns is:
Unknown Actor      1331
Liam Neeson         161
Alfred Molina       157
John Krasinski      138
Salma Hayek         130
                   ...
Jillian Estell        1
Kim Dickens           1
W. Earl Brown         1
```

```
Willie Taylor         1
Sam Robert Muik       1
Name: cast, Length: 25952, dtype: int64
```

COUNTRY:

```
Number of unique elements in country is:
 119

Unique elements present in country column is:
 ['United States' 'Unknown Country' 'Ghana' 'Burkina Faso' 'United Kingdom'
 'Germany' 'Ethiopia' 'Czech Republic' 'India' 'France' 'China' 'Canada'
 'South Africa' 'Japan' 'Nigeria' 'Spain' 'Australia' 'Mexico' 'Italy'
 'Romania' 'Argentina' 'Venezuela' 'Hong Kong' 'Nepal' 'New Zealand'
 'Brazil' 'Greece' 'Colombia' 'Belgium' 'Switzerland' 'Bulgaria' ''
 'Algeria' 'Poland' 'Israel' 'Saudi Arabia' 'Thailand' 'Indonesia' 'Egypt'
 'Denmark' 'Kuwait' 'Netherlands' 'Singapore' 'Malaysia' 'South Korea'
 'Vietnam' 'Hungary' 'Lebanon' 'Syria' 'Philippines'
 'United Arab Emirates' 'Sweden' 'Qatar' 'Mauritius' 'Austria' 'Turkey'
 'Russia' 'Taiwan' 'Cameroon' 'Palestine' 'Ireland' 'Kenya' 'Chile'
 'Uruguay' 'Cambodia' 'Bangladesh' 'Portugal' 'Cayman Islands' 'Norway'
 'Iceland' 'Serbia' 'Malta' 'Luxembourg' 'Namibia' 'Angola' 'Peru'
 'Mozambique' 'Jordan' 'Zimbabwe' 'Pakistan' 'Guatemala' 'Senegal'
 'Finland' 'Iraq' 'Malawi' 'Paraguay' 'Iran' 'Albania' 'Georgia'
 'Soviet Union' 'Morocco' 'Slovakia' 'West Germany' 'Ukraine' 'Bermuda'
 'Ecuador' 'Armenia' 'Mongolia' 'Bahamas' 'Sri Lanka' 'Latvia'
 'Liechtenstein' 'Nicaragua' 'Croatia' 'Slovenia' 'Dominican Republic'
 'Samoa' 'Botswana' 'Vatican City' 'Jamaica' 'Kazakhstan' 'Lithuania'
 'Afghanistan' 'Somalia' 'Sudan' 'Panama' 'Uganda' 'East Germany'
 'Montenegro']

Value Counts of country columns is:
United States      45817
India              21411
United Kingdom      8580
France              6607
Unknown Country     6199
                   ...
Armenia                2
Palestine              2
Kazakhstan             1
Nicaragua              1
Uganda                 1
Name: country, Length: 119, dtype: int64
```

LISTED_IN:

```
Number of unique elements in listed_in is:
 20

Unique elements present in listed_in column is:
 ['Documentaries' 'Children & Family Movies' 'Dramas' 'Independent Movies'
 'International Movies' 'Comedies' 'Thrillers' 'Romantic Movies'
 'Music & Musicals' 'Horror Movies' 'Sci-Fi & Fantasy'
 'Action & Adventure' 'Classic Movies' 'Anime Features' 'Sports Movies'
 'Cult Movies' 'Faith & Spirituality' 'LGBTQ Movies' 'Stand-Up Comedy'
 'Movies']

Value Counts of listed_in columns is:
Dramas                      29806
International Movies        28243
Comedies                    20829
Action & Adventure          12216
Independent Movies           9834
Children & Family Movies     9771
Thrillers                    7107
Romantic Movies              6412
Horror Movies                4571
Sci-Fi & Fantasy             4037
Music & Musicals             3077
Documentaries                2409
Sports Movies                1531
Classic Movies               1443
Cult Movies                  1077
Anime Features               1045
LGBTQ Movies                  838
Faith & Spirituality          719
Stand-Up Comedy               540
Movies                        412
Name: listed_in, dtype: int64
```

RELEASE_YEAR:

```
Number of unique elements in release_year is:
 73

Unique elements present in release_year column is:
 [2020 2021 1993 1996 1998 1997 2010 2013 2017 1975 1978 1983 1987 2012
 2001 2002 2003 2004 2011 2008 2009 2007 2005 2006 2018 2019 1994 2015
 1982 1989 2014 1990 1991 1999 2016 1986 1984 1980 1961 2000 1995 1985
 1992 1976 1959 1988 1981 1972 1964 1954 1979 1958 1956 1963 1970 1973
 1960 1974 1966 1971 1962 1969 1977 1967 1968 1965 1945 1946 1942 1955
 1944 1947 1943]

Value Counts of release_year columns is:
2018    17060
2017    15069
2016    14075
2019    11926
2015    10612
        ...
1959       12
1947        8
1942        6
1943        5
1946        2
Name: release_year, Length: 73, dtype: int64
```

YEAR:

```
Number of unique elements in year is:
 14

Unique elements present in year column is:
 [2021 2020 2019 2018 2017 2016 2015 2014 2013 2012 2011 2009 2008 2010]

Value Counts of year columns is:
2019    34473
2020    32488
2018    28050
2021    25709
2017    18252
2016     4858
2015     1125
2011      438
2014      345
2013       75
2012       36
2009       30
2010       20
2008       18
Name: year, dtype: int64
```

WEEK:

```
Number of unique elements in week is:
 53

Unique elements present in week column is:
 <IntegerArray>
[38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20,
 19, 18, 17, 16, 15, 14, 13, 12, 11, 10,  9,  8,  7,  6,  5,  4,  3,  2,  1,
 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39]
Length: 53, dtype: UInt32

Value Counts of week columns is:
1     8456
44    5563
9     5094
35    5048
26    4931
40    4905
31    4388
27    3808
48    3737
18    3686
13    3503
39    3502
30    3262
22    3237
23    3164
5     3148
15    3083
28    2744
7     2636
17    2627
14    2609
```

```
36    2586
25    2568
37    2559
43    2521
10    2515
50    2463
33    2418
29    2361
34    2332
16    2323
51    2276
11    2243
49    2181
42    2105
38    2086
3     2031
24    1920
52    1840
20    1829
41    1807
47    1740
6     1649
19    1630
2     1618
21    1606
8     1538
46    1519
12    1431
53    1413
45    1398
32    1233
4     1047
Name: week, dtype: Int64
```

MONTH:

```
Number of unique elements in month is:
 12

Unique elements present in month column is:
 ['September' 'August' 'July' 'June' 'May' 'April' 'March' 'February'
 'January' 'December' 'November' 'October']

Value Counts of month columns is:
July         15075
January      13947
October      13541
September    13220
December     12768
April        12538
August       11924
June         11616
March        11507
November     11065
May           9579
February      9137
Name: month, dtype: int64
```

DAY:

```
Number of unique elements in day is:
 31

Unique elements present in day column is:
 [25 24 23 22 21 20 19 17 16 15 14 11 10  9  8  7  6  5  4  3  2  1 31 28
 27 18 13 12 30 29 26]

Value Counts of day columns is:
1     43527
15    11330
16     6488
2      5840
19     4297
5      4086
26     3823
20     3694
27     3383
31     3339
7      3311
8      3235
25     3150
6      3085
21     3036
10     2952
```

```
18     2933
22     2860
28     2856
14     2747
9      2710
4      2670
30     2665
23     2616
12     2412
11     2305
24     2260
13     2110
17     2086
3      2085
29     2026
Name: day, dtype: int64
```

DAYNAME:

```
Number of unique elements in dayname is:
 7

Unique elements present in dayname column is:
 ['Saturday' 'Friday' 'Thursday' 'Wednesday' 'Tuesday' 'Monday' 'Sunday']

Value Counts of dayname columns is:
Friday       36902
Thursday     24878
Wednesday    22500
Tuesday      17398
Sunday       15944
Monday       14681
Saturday     13614
Name: dayname, dtype: int64
```

DURATION:

```
Number of unique elements in duration is:
 205

Unique elements present in duration column is:
 [ 90.  91. 125. 104. 127.  67.  94. 161.  61. 166. 147. 103.  97. 106.
 111. 110. 105.  96. 124. 116.  98.  23. 115. 122.  99.  88. 100. 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.  64.  59. 139.  69. 148. 189. 141. 130. 138.  81. 132. 123.
  65.  68.  66.  62.  74. 131.  39.  46.  38. 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.   8.  57.  28.  50.   9.  26.  45. 171.  27.  44. 146.  20. 157.
  17. 203.  41.  30. 194. 233. 237. 230. 195. 253. 152. 190. 160. 208.
 180. 144.   5. 174. 170. 192. 209. 187. 172.  16. 186.  11. 193. 176.
  56. 169.  40.  10.   3. 168. 312. 153. 214.  31. 163.  19.  nan 179.
  43. 200. 196. 167. 178. 228.  18. 205. 201. 191.]

Value Counts of duration columns is:
94.0     4343
106.0    4040
97.0     3624
95.0     3560
96.0     3511
         ...
20.0        4
5.0         3
9.0         2
8.0         2
11.0        2
Name: duration, Length: 205, dtype: int64
```

In [39]:

```
for i in cols_list:
    print(bold_text(i.upper()+':'))
    print(f'Number of unique elements in {i} is:\n {shows[i].nunique()}\n')
    print(f'Unique elements present in {i} column is:\n {shows[i].unique()}\n')
    print(f'Value Counts of {i} columns is:\n{shows[i].value_counts()}\n\n\n')
```

```
TYPE:
Number of unique elements in type is:
 1

Unique elements present in type column is:
 ['TV Show']

Value Counts of type columns is:
TV Show    56148
Name: type, dtype: int64



RATING:
Number of unique elements in rating is:
 10

Unique elements present in rating column is:
 ['TV-MA' 'TV-14' 'TV-Y7' 'TV-PG' 'TV-Y' 'TV-G' 'R' 'NR' 'Unknown Rating'
 'TV-Y7-FV']

Value Counts of rating columns is:
TV-MA             29906
TV-14             14691
TV-PG              4614
TV-Y7              3818
TV-Y               1787
TV-G               1041
NR                  155
Unknown Rating       58
R                    54
TV-Y7-FV             24
Name: rating, dtype: int64



DIRECTOR:
Number of unique elements in director is:
 300

Unique elements present in director column is:
 ['Unknown Director' 'Julien Leclercq' 'Mike Flanagan' 'Andy Devonshire'
 'Kongkiat Komesiri' 'Olivier Megaton' 'Bunmi Ajakaiye' 'Jason Hehir'
 'Luis Alfaro' 'Javier Gómez Santander' 'Gerhard Mostert' 'Kemi Adetiba'
 'Brad Anderson' 'Mauricio Dias' 'Tatiana Villela' 'Juan Carlos Medina'
 'Inma Torrente' 'Sakon Tiacharoen' 'Bejoy Nambiar' 'Priyadarshan'
 'Karthik Narain' 'Vasanth Sai' 'Karthik Subbaraj' 'Arvind Swamy'
 'Rathindran R Prasad' 'Sarjun' 'Gautham Vasudev Menon' 'Billy Corben'
 'Hsu Fu-chun' 'Glen Winter' 'Royale Watkins' 'Rich Schlansker'
 'Garrett Bradley' 'Jan Holoubek' 'Malik Nejer' 'Sidharta Tata'
 'Aco Tenriyagelli' 'Dian Sastrowardoyo' 'Ifa Isfansyah' 'Jason Iskandar'
 'Shen Leping' 'John Dower' 'Srijit Mukherji' 'Vasan Bala'
 'Abhishek Chaubey' 'Fernando Moro' 'Ivan Andrew Payawal'
 'Rob Seidenglanz' 'Simon Frederick' 'Sarawut Wichiensarn' 'Michael Simon'
 'Joe Berlinger' 'Bruce Sinofsky' 'Jay Oliva' 'Ally Pankiw' 'Chiaki Kon'
 'Takuya Igarashi' 'José Larraza' 'Marc Pons' 'Gary Sing' 'Daniel Minahan'
 'Uzodinma Okpechi' 'Cai Cong' 'Vijay Roche' 'Joshua Zeman' 'Oriol Paulo'
 'Kongkiat Khomsiri' 'Glenn Weiss' 'Pass Patthanakumjon' 'Moyoung Jin'
 'Caroline Sá' 'Deepti Kakkar' 'Fahad Mustafa' 'Hikaru Toda'
 'Chico Pereira' 'Elaine McMillion Sheldon' 'Adrián García Bogliano'
 'Jung-ah Im' 'Alain Brunard' 'Ange Basterga' 'Nicolas Lopez'
 'Maribel Sánchez-Maroto' 'Jared Hess' 'Tyler Measom' 'Sunny Lau'
 'Tharun Bhascker Dhaassyam' 'B. V. Nandini Reddy' 'Nag Ashwin'
 'Sankalp Reddy' 'Richard Arellano' 'Tiller Russell' 'Juliet May'
 'Maite Ruiz De Austri' 'Tony Collingwood' 'Cosima Spender'
 'Sudha Kongara' 'Vignesh Shivan' 'Vetri Maaran' 'Alexx Media'
 'Andrucha Waddington' 'Pedro Waddington' 'Picky Talarico' 'Ellena Wood'
 'Jesse Vile' 'Lai Chun-yu' 'Jalil Lespert' 'Su I-Hsuan'
 'Chuang Shiang-an' 'Liu Yi' 'BB Sasore' 'Mateo Gil' 'Carla Barros'
 'Steven Bognar' 'Julia Reichert' 'Alejandro Hartmann' 'Juliana Vicente'
 'Chang Chin-jung' 'Chen Rong-hui' 'Thierry Demaizière' 'Alban Teurlai'
 'Manolo Caro' 'Pavel Kostomarov' 'Philippa Lowthorpe' 'Kenny Ortega'
 'Pepe Mora' 'Greg Tiernan' 'Hong Won-ki' 'Nopparoj Chotmunkongsit'
 'Alex Gibney' 'Cassia Dian' 'Wouter Bouvijn' 'Rachit Kusonkoonsiri'
 'Kyran Kelly' 'Phil Sgriccia' 'Jesse Warn' 'Norm Hiscock' 'Gary Howsam'
 'Mike Smith' 'John Paul Tremblay' 'Robb Wells' 'Marcus Raboy'
 'Ziad Doueiri' 'Sivaroj Kongsakul' 'Damien Chazelle' 'Houda Benyamina'
 'Laïla Marrakchi' 'Alan Poul' 'He Xiaofeng' 'Hernán Guerschuny'
 'Sion Sono' 'David Chuang' 'Chen Kuan-chung' 'Ryan Polito'
 'Adrien Lagier' 'Ousmane Ly' 'Stefano Lodovichi' 'Vijay S. Bhanushali'
 'Eric Goode' 'Rebecca Chaiklin' 'Lynn Novick' 'Kim Seong-hun'
 'Carlos Sedes' 'Ali Kalthami' 'Meshal Aljaser' 'Faris Godus'
 'Mohamed al Salman' 'Faisal al Amer' 'Mick Grogan' 'Tosin Coker'
 'Pantham Thongsang' 'Cheewatan Pusitsuksa' 'James Bamford'
 'Julia Willoughby Nason' 'Jenner Furst' 'Thomas Astruc' 'Quek Shio-chuan'
 'Soumendra Padhi' 'Justin Webster' 'Lee Toland Krieger' 'Mark Lewis'
```

```
 'Obi Emelonye' 'Tsutomu Mizushima' 'Michel Tikhomiroff' 'Robert Kenner'
 'Taki Oldham' 'DJ Chen' 'Park Joon-hwa' 'Ahmet Katıksız' 'Lee Yoon-jung'
 'Danny Cannon' 'Mariano Barroso' 'Kiko Cruz Claverol' 'Patricia Font'
 'Hayato Date' 'Vikramaditya Motwane' 'Anurag Kashyap' 'Cho Li'
 'Chen Hung-yi' 'Weica Wang' 'Liu Bang-yao' 'Lin Guan-fu' 'Shen Chi'
 'YC Tom Lee' 'Jesse Moss' 'Shin Won-ho' 'Jerry Seinfeld'
 'Luis Alberto Restrepo' 'Andrés Beltrán' 'Jaime Rayo' 'Elías León'
 'Diego Enrique Osorno' 'Felipe Cano' 'Quentin Tarantino' 'Lee Kyoungmi'
 'Yim Pilsung' 'Jeon Go-woon' 'Kim Jong-kwan' 'Lars Kaalund' 'Seren Yüce'
 'Noam Murro' 'Joel Gallen' 'Tig Notaro' 'Mat King' 'Padraic McKinley'
 'Peter Ho' 'Patrick Graham' 'Han Qing' 'Lee Eung-bok' 'Trey Borzillieri'
 'Barbara Schroeder' 'Dawn Porter' 'Dan Lucchesi' 'Kazuya Murata'
 'Dan Forrer' 'Kobun Shizuno' 'Hiroyuki Seshita' 'Masaaki Yuasa'
 'Yasuhiro Irie' 'Stan Lathan' 'Daniel Kontur' 'Olivier Jean-Marie'
 'David Ayer' 'Go Koga' 'Pali Yahya' 'Vanessa Roth' 'Eric Abrams'
 'Stefan Brogren' 'Serdar Akar' 'Iginio Straffi' 'Chang-Min Lee'
 'Jill Bauer' 'Ronna Gradus' 'Rashida Jones' 'Laurent Bouzereau'
 'Gordon Anderson' 'James Hawes' 'David Schalko' 'Tensai Okamura'
 'Eli Roth' 'Nizar Shafi' 'Seung-uk Jo' 'Sharon Grimberg' 'Miguel Conde'
 'Cecilia Peck' 'Carlos Bolado' 'James Lee' 'Ian Barber'
 'Alastair Fothergill' 'Guy Vasilovich' 'Neslihan Yesilyurt' 'Ah Loong'
 'Stuart Orme' 'Toby Haynes' 'Ken Burns' 'Steve Greenwood'
 'Alejandro Lozano' 'Andrew Tan' 'Stephen Murray' 'Onur Ünlü'
 'Jakob Verbruggen' 'Alessandro Angulo' 'Everardo Gout' 'Rachel Bell'
 'Oliver Stone' 'Peter McDonnell' 'Oscar Micheaux' 'Spencer Williams'
 'Richard E. Norman' 'Richard Maurice' 'Bhavik Thakore' 'Jani Lachauer'
 'Jakob Schuh' 'Jay Chandrasekhar' 'Ehtesham Uddin' 'Dheeraj Berry'
 'Estela Renner' 'Bumpy' 'Michael Samuels' 'Mark Tonderai'
 'Michael Cumming']

Value Counts of director columns is:
Unknown Director    49358
Noam Murro            189
Thomas Astruc         160
Houda Benyamina       104
Damien Chazelle       104
                     ...
Rashida Jones           1
Sharon Grimberg         1
Garrett Bradley         1
Alex Gibney             1
Padraic McKinley        1
Name: director, Length: 300, dtype: int64



CAST:
Number of unique elements in cast is:
 14864

Unique elements present in cast column is:
 ['Ama Qamata' 'Khosi Ngema' 'Gail Mabalane' ... 'Waseem Abbas'
 'Javed Sheikh' 'Hina Khawaja Bayat']

Value Counts of cast columns is:
Unknown Actor         818
David Attenborough     82
Takahiro Sakurai       56
Yuki Kaji              45
Ai Kayano              41
                     ...
Jimmy O. Yang           1
Diana Silvers           1
John Malkovich          1
Sassy Bermudez          1
Telma Hopkins           1
Name: cast, Length: 14864, dtype: int64



COUNTRY:
Number of unique elements in country is:
 67

Unique elements present in country column is:
 ['South Africa' 'Unknown Country' 'India' 'United Kingdom' 'United States'
 'Mexico' 'Turkey' 'Australia' 'Finland' 'Nigeria' 'Japan' 'Belgium'
 'France' 'South Korea' 'Spain' 'Singapore' 'Russia' '' 'Ireland' 'Italy'
 'Argentina' 'Jordan' 'Colombia' 'Israel' 'Taiwan' 'Germany' 'Canada'
 'Poland' 'Thailand' 'New Zealand' 'Netherlands' 'Sweden' 'China'
 'Iceland' 'Denmark' 'Philippines' 'Indonesia' 'United Arab Emirates'
 'Norway' 'Czech Republic' 'Lebanon' 'Brazil' 'Uruguay' 'Egypt'
 'Luxembourg' 'Senegal' 'Saudi Arabia' 'Kuwait' 'Belarus' 'Chile' 'Malta'
 'Puerto Rico' 'Austria' 'Cyprus' 'Malaysia' 'Mauritius' 'Hong Kong'
 'Croatia' 'West Germany' 'Syria' 'Hungary' 'Cuba' 'Greece' 'Pakistan'
 'Azerbaijan' 'Ukraine' 'Switzerland']
```

```
Value Counts of country columns is:
United States      13533
Unknown Country     5698
Japan               5154
United Kingdom      4385
South Korea         3754
                   ...
Hungary               10
                       8
Greece                 6
Belarus                6
Uruguay                3
Name: country, Length: 67, dtype: int64
```

LISTED_IN:

```
Number of unique elements in listed_in is:
 22

Unique elements present in listed_in column is:
 ['International TV Shows' 'TV Dramas' 'TV Mysteries' 'Crime TV Shows'
 'TV Action & Adventure' 'Docuseries' 'Reality TV' 'Romantic TV Shows'
 'TV Comedies' 'TV Horror' 'British TV Shows' 'Spanish-Language TV Shows'
 'TV Thrillers' "Kids' TV" 'TV Sci-Fi & Fantasy' 'Anime Series'
 'Korean TV Shows' 'Science & Nature TV' 'Teen TV Shows' 'TV Shows'
 'Stand-Up Comedy & Talk Shows' 'Classic & Cult TV']

Value Counts of listed_in columns is:
International TV Shows          12845
TV Dramas                        8942
TV Comedies                      4963
Crime TV Shows                   4733
Kids' TV                         4568
Romantic TV Shows                3049
Anime Series                     2313
TV Action & Adventure            2288
Spanish-Language TV Shows        2126
British TV Shows                 1808
TV Mysteries                     1281
Korean TV Shows                  1122
TV Sci-Fi & Fantasy              1045
TV Horror                         941
Docuseries                        845
TV Thrillers                      768
Teen TV Shows                     742
Reality TV                        735
TV Shows                          337
Classic & Cult TV                 272
Stand-Up Comedy & Talk Shows      268
Science & Nature TV               157
Name: listed_in, dtype: int64
```

RELEASE_YEAR:

```
Number of unique elements in release_year is:
 46

Unique elements present in release_year column is:
 [2021 2020 2018 2014 1994 2015 2013 2019 2017 2016 2012 1992 2002 2009
 2011 2005 2008 2010 2007 2001 2006 1993 1997 2003 1945 1999 1998 2000
 2004 1986 1995 1925 1972 1974 1988 1991 1977 1979 1990 1996 1981 1946
 1985 1967 1989 1963]

Value Counts of release_year columns is:
2020    10089
2019    10005
2018     7381
2021     6467
2017     5447
2016     4390
2015     3516
2014     1454
2013     1223
2012     1077
2010      709
2008      622
2011      599
2009      581
2006      318
1999      278
2007      259
2003      242
2005      205
1997      187
2004      175
2002      164
1990       98
1986       84
1998       79
2000       78
1996       61
1993       49
2001       46
1995       43
1992       38
1977       24
1979       24
1972       24
1981       20
1974       18
1989       18
1985       16
1994       15
1967       12
1988        4
1946        4
1963        2
1991        1
1925        1
1945        1
Name: release_year, dtype: int64
```

YEAR:

```
Number of unique elements in year is:
 10

Unique elements present in year column is:
 [2021 2020 2019 2018 2017 2016 2015 2014 2013 2008]

Value Counts of year columns is:
2020    13721
2019    12524
2021    10825
2018     7735
2017     6957
2016     3711
2015      435
2013      132
2014      107
2008        1
Name: year, dtype: int64
```

WEEK:

```
Number of unique elements in week is:
 53
```

```
Unique elements present in week column is:
 <IntegerArray>
[38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20,
 19, 18, 17, 16, 15, 14, 13, 12, 11, 10,  9,  8,  7,  6,  5,  4,  3,  2,  1,
 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39]
Length: 53, dtype: UInt32

Value Counts of week columns is:
27    1977
35    1945
24    1702
26    1662
31    1646
13    1554
48    1513
5     1386
44    1380
18    1364
40    1362
15    1230
51    1194
50    1182
19    1181
33    1180
1     1176
46    1155
22    1150
52    1130
38    1128
37    1127
49    1074
53    1071
21    1044
12    1002
7     1001
8      976
32     968
23     964
20     939
11     927
42     900
25     896
36     879
17     864
45     855
34     834
14     828
9      826
2      812
29     797
4      788
41     764
10     743
39     743
30     731
47     678
6      613
3      601
28     586
43     566
16     554
Name: week, dtype: Int64
```

MONTH:

```
Number of unique elements in month is:
 12

Unique elements present in month column is:
 ['September' 'August' 'July' 'June' 'May' 'April' 'March' 'February'
 'January' 'December' 'November' 'October']

Value Counts of month columns is:
December     5498
July         5227
August       5162
June         5043
September    4900
April        4543
November     4532
January      4465
March        4352
October      4255
May          4248
February     3923
```

```
Name: month, dtype: int64
```

DAY:

```
Number of unique elements in day is:
 31

Unique elements present in day column is:
 [24 23 22 21 17 16 15 14 10  9  8  7  6  5  3  2  1 31 29 28 27 26 25 20
 18 13 12 11  4 30 19]

Value Counts of day columns is:
1     10067
15     4379
30     2288
6      2119
8      2051
31     1855
27     1693
23     1682
16     1671
20     1628
10     1625
14     1620
19     1557
5      1545
4      1516
17     1479
25     1466
22     1410
13     1311
2      1273
18     1262
24     1257
26     1212
7      1169
21     1115
11     1079
28     1042
12     1003
3       951
9       933
29      890
Name: day, dtype: int64
```

DAYNAME:

```
Number of unique elements in dayname is:
 7

Unique elements present in dayname column is:
 ['Friday' 'Thursday' 'Wednesday' 'Tuesday' 'Monday' 'Sunday' 'Saturday']

Value Counts of dayname columns is:
Friday       21126
Thursday      8554
Tuesday       6604
Wednesday     6592
Saturday      5664
Monday        4306
Sunday        3302
Name: dayname, dtype: int64
```

DURATION:

```
Number of unique elements in duration is:
 15

Unique elements present in duration column is:
 [ 2.  1.  9.  4.  5.  3.  6.  7. 10.  8. 17. 13. 15. 12. 11.]

Value Counts of duration columns is:
1.0     35035
2.0      9559
3.0      5084
4.0      2134
5.0      1698
7.0       843
6.0       633
8.0       286
9.0       257
10.0      220
13.0      132
```

```
12.0      111
15.0       96
17.0       30
11.0       30
Name: duration, dtype: int64
```

In [40]:
```
df[df['type'] == 'Movie'].shape
```

Out[40]: (6131, 12)

In [41]:
```
df[df['type'] == 'TV Show'].shape
```

Out[41]: (2676, 12)

In [42]:
```
len(set(movies['director'].unique()).intersection(shows['director'].unique()))
```

Out[42]: 84

In [43]:
```
len(set(movies['cast'].unique()).intersection(shows['cast'].unique()))
```

Out[43]: 4376

Insights from Non Graphical Analysis:

Type:

There are Only Two types of Show -> Movies and TV Shows

b. Out of 8807 shows 6131 shows are Movies and 2676 shows are TV Shows

Rating:

a. There were a total of 17 ratings present for movies. Only 9 of which are ratings used in TV Shows

Director:

a. There were a total of 4528 directors in original dataset

b. There are a total of 4993 directors in the unnested dataset. Out of which 4777 directors worked in movies and only 299 directors worked in TV shows. Only 84 directors worked both in Movies and TV Shows

Cast:

a. There were a total of 7692 actors in original dataset

b. There are a total of 36439 casted actors/actress present in the unnested dataset. Out of which 25951 worked in movies and 14863 worked in TV Shows. Only 4376 worked both in Movies and TV Shows

Country:

a. There were a total of 748 different values of clubbe country in original dataset

b. There are a total of 123 countries where these shows were available. Movies were accessible in 118 different countries and only 66 countries for TV Shows

Genre/Listed_in:

a. There are a total of 42 genres values of present in the dataset. Out of which 20 belong to Movies and 22 belong the TV shows

b. There are a total of 123 countries where these shows were available

Years:

a. These movies/TV Shows were released in 74 different years starting from 1925. First TV Shows that was realeased in the dataset was in year 1925 and Movie was in year 1942.

b. 75% of movies were released in the last decade and 75% of Shows were released in last 7 years.

c. Only from 2008 these tv shows/movies were added in Netflix. Most of the tv shows/movies were added in July following by December
Most of the tv shows/movies were released in Friday followed by Thursday

In [44]:
```
movies.sort_values('listed_in')['listed_in'].unique()
```

Out[44]:
```
array(['Action & Adventure', 'Anime Features', 'Children & Family Movies',
       'Classic Movies', 'Comedies', 'Cult Movies', 'Documentaries',
       'Dramas', 'Faith & Spirituality', 'Horror Movies',
       'Independent Movies', 'International Movies', 'LGBTQ Movies',
       'Movies', 'Music & Musicals', 'Romantic Movies',
       'Sci-Fi & Fantasy', 'Sports Movies', 'Stand-Up Comedy',
       'Thrillers'], dtype=object)
```

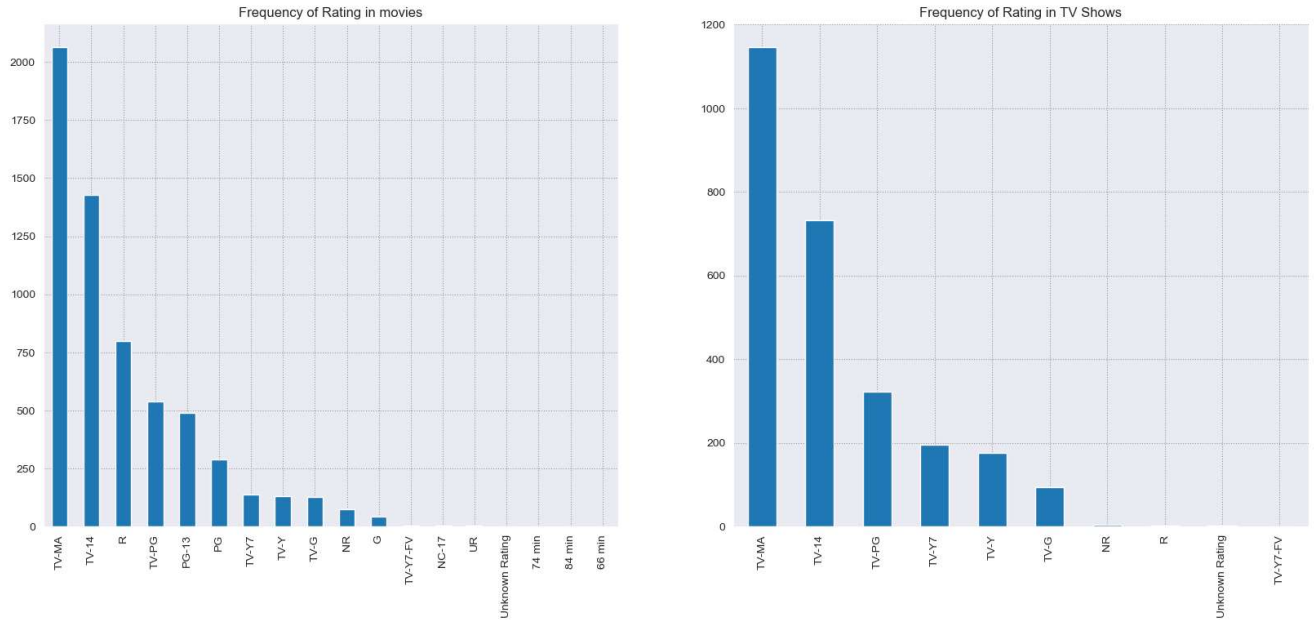
In [45]:
```
shows.sort_values('listed_in')['listed_in'].unique()
```

Out[45]:
```
array(['Anime Series', 'British TV Shows', 'Classic & Cult TV',
       'Crime TV Shows', 'Docuseries', 'International TV Shows',
       "Kids' TV", 'Korean TV Shows', 'Reality TV', 'Romantic TV Shows',
       'Science & Nature TV', 'Spanish-Language TV Shows',
       'Stand-Up Comedy & Talk Shows', 'TV Action & Adventure',
       'TV Comedies', 'TV Dramas', 'TV Horror', 'TV Mysteries',
       'TV Sci-Fi & Fantasy', 'TV Shows', 'TV Thrillers', 'Teen TV Shows'],
      dtype=object)
```

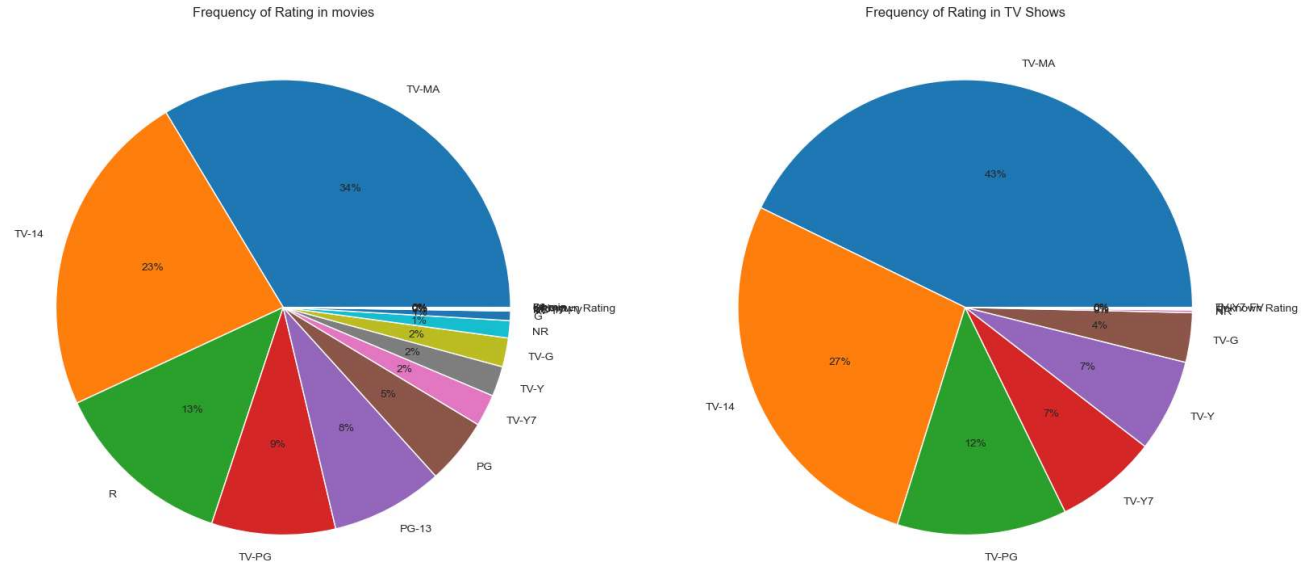
Visual Analysis - Univariate, Bivariate after pre-processing of the data

show_id,title,description will have unique values so we will not be considering these two cols

In [46]:
```
plt.figure(figsize = (20,8))


plt.subplot(1,2,1)
movies[['show_id','rating']].drop_duplicates(keep = 'first')['rating'].value_counts().plot(kind = 'bar')
plt.title('Frequency of Rating in movies')


plt.subplot(1,2,2)
shows[['show_id','rating']].drop_duplicates(keep = 'first')['rating'].value_counts().plot(kind = 'bar')
plt.title('Frequency of Rating in TV Shows')

plt.show()
```



In [47]:
```
plt.figure(figsize = (20,15))

plt.subplot(1,2,1)
mpie = movies[['show_id','rating']].drop_duplicates(keep = 'first')['rating'].value_counts()
plt.pie(mpie, labels= mpie.index, autopct='%.0f%%')
plt.title('Frequency of Rating in movies')


plt.subplot(1,2,2)
tpie = shows[['show_id','rating']].drop_duplicates(keep = 'first')['rating'].value_counts()
plt.pie(tpie, labels= tpie.index, autopct='%.0f%%')
plt.title('Frequency of Rating in TV Shows')

plt.show()
```



Inferences from Rating:

a. Netlix caters to a lot of Mature audience, 34% of movies and 48% of tv shows that are avaiable content is for mature

b. 23% and 27% movies and tv shows rated respectively as TV-14 i.e. children under age of 14 are not suitable to watch, target audience been mid and late teens

c. There are around 13% R Rated movies.

d.There are only 4% movies and 14% of TV Shows available for kids(TV-Y and TV-Y7)

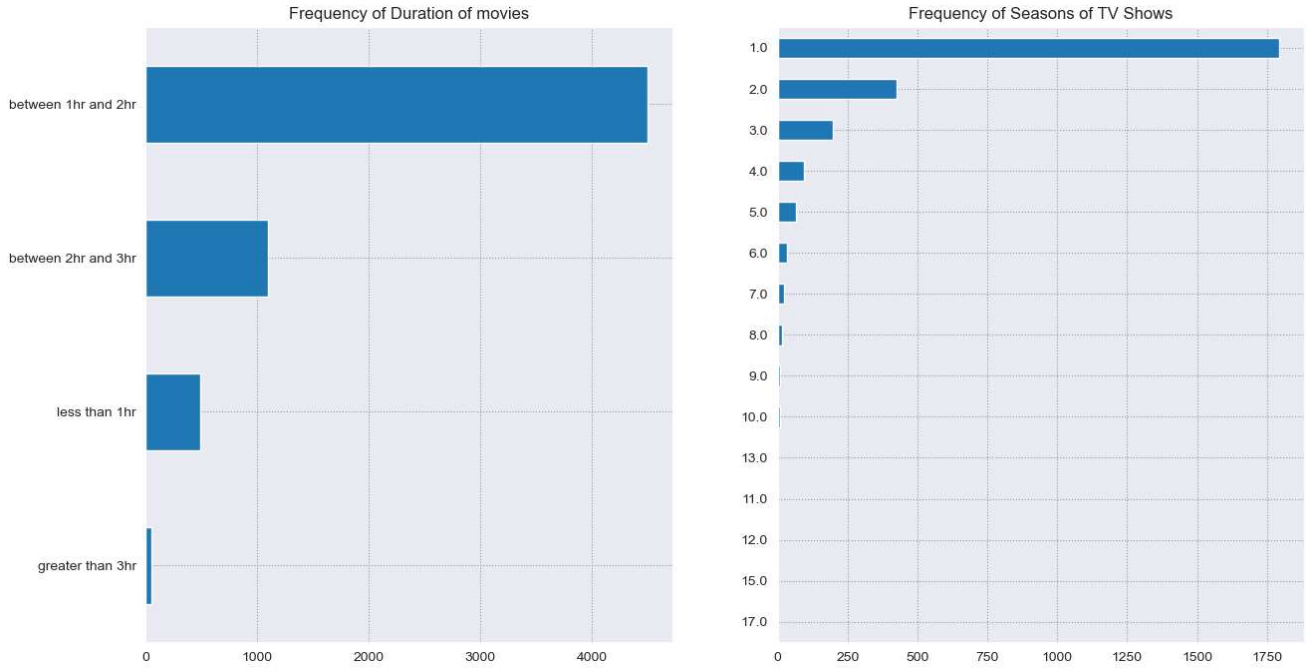
In [48]:
```
label = ['less than 1hr', 'between 1hr and 2hr','between 2hr and 3hr','greater than 3hr']
movies_duration = movies.drop_duplicates(subset=['show_id','duration'], keep='first')['duration']
(pd.cut(movies.drop_duplicates(subset=['show_id','duration'], keep='first')['duration'],
                bins=[1,60,120,180,1000],
                labels = label
).value_counts()/len(movies_duration))*100
```

Out[48]:
```
between 1hr and 2hr    73.381178
between 2hr and 3hr    17.860055
less than 1hr           7.943239
greater than 3hr        0.766596
Name: duration, dtype: float64
```

In [49]:
```
shows_duration = shows[['show_id','duration']].drop_duplicates(keep = 'first')['duration']
shows_duration.value_counts()#/len(shows_duration)*100

```

Out[49]:
```
1.0     1793
2.0      425
3.0      199
4.0       95
5.0       65
6.0       33
7.0       23
8.0       17
9.0        9
10.0       7
13.0       3
15.0       2
12.0       2
11.0       2
17.0       1
Name: duration, dtype: int64
```

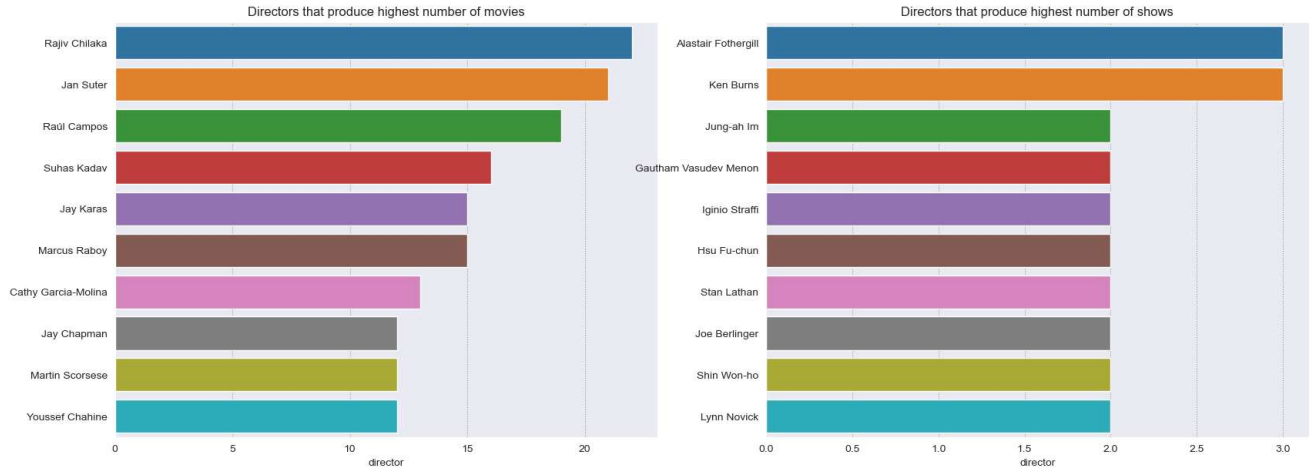
In [50]:
```
plt.figure(figsize =(15,8))
plt.subplot(1,2,1)
label = ['less than 1hr', 'between 1hr and 2hr','between 2hr and 3hr','greater than 3hr']
plt.title('Frequency of Duration of movies')
pd.cut(movies.drop_duplicates(subset=['show_id','duration'], keep='first')['duration'],
               bins=[1,60,120,180,1000],
               labels = label
).value_counts(ascending = True).plot(kind = 'barh')

plt.subplot(1,2,2)
shows[['show_id','duration']].drop_duplicates(keep = 'first')['duration'].value_counts(ascending = True).plot(kind = 'barh')
plt.title('Frequency of Seasons of TV Shows')


plt.show()

```



Inferences for Duration:

a. 4499(~73%) movies are between 1hr and 2hr. 1095 Movies are between 2hr and 3hr.
b. 487 movies are less than 1hr. Only 47 movies are greater than 3hr.
c. TV Shows are mostly of only one season around 65%. There's one such TV Show which has 17 seasons.
d. There are only 26 such TV shows which have more than 8 seasons

In [51]:
```
plt.figure(figsize = (20,7))

plt.subplot(1,2,1)
mask = movies['director'] == 'Unknown Director'
movies_director= movies.loc[~mask,['show_id','director']].drop_duplicates(keep = 'first')['director'].value_counts().head(10
sns.barplot(x = movies_director, y = movies_director.index )
plt.title('Directors that produce highest number of movies')




plt.subplot(1,2,2)
mask = shows['director'] == 'Unknown Director'
shows_director= shows.loc[~mask,['show_id','director']].drop_duplicates(keep = 'first')['director'].value_counts().head(10)
sns.barplot(x = shows_director, y = shows_director.index )
plt.title('Directors that produce highest number of shows')

plt.show()
```

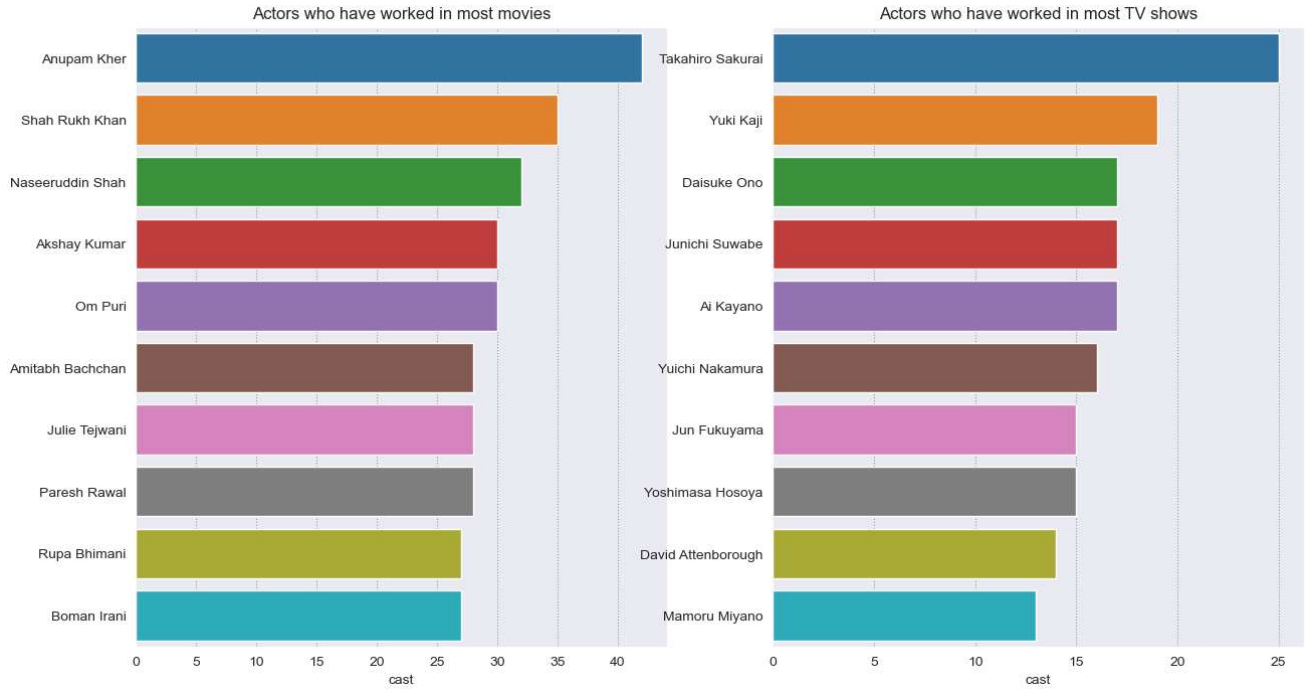


Inferences for Directors:

a. Rajiv Chilaka directed highest number of movies.

b. Alaistar Fothergill directed highest number of TV Shows.

In [52]:
```
plt.figure(figsize =(15,8))

plt.subplot(1,2,1)
mask = movies['cast'] == 'Unknown Actor'
casts = movies.loc[~mask,['show_id','cast']].drop_duplicates(keep = 'first')['cast'].value_counts().head(10)
sns.barplot(x=casts,y = casts.index)
plt.title('Actors who have worked in most movies')

plt.subplot(1,2,2)
mask = shows['cast'] == 'Unknown Actor'
casts = shows.loc[~mask,['show_id','cast']].drop_duplicates(keep = 'first')['cast'].value_counts().head(10)
sns.barplot(x=casts,y = casts.index)
plt.title('Actors who have worked in most TV shows')

plt.show()
```

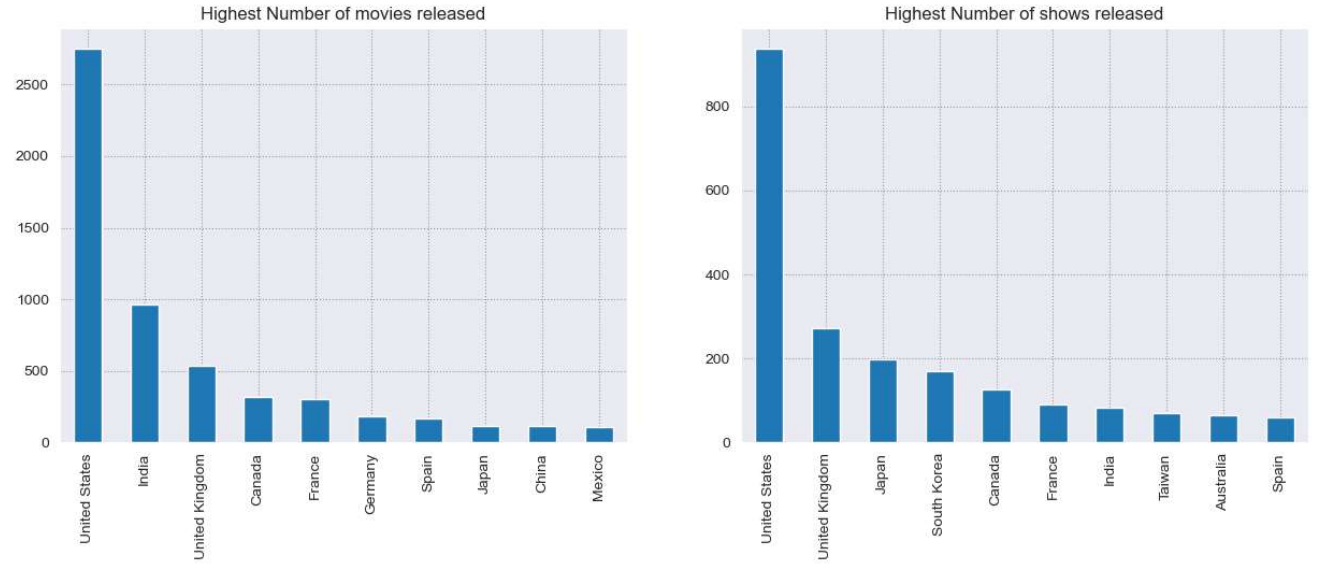


Inferences from Cast:

a. Anupam Kher has appeared in most of movies.

b. Takahiko Sakurai has apperead in most of TV Shows.

In [53]:
```
plt.figure(figsize=(15,5))

plt.subplot(1,2,1)
mask = movies['country'] == 'Unknown Country'
movies.loc[~mask,['show_id','country']].drop_duplicates(keep = 'first')['country'].value_counts().head(10).plot(kind = 'bar'
plt.title('Highest Number of movies released')


plt.subplot(1,2,2)
mask = shows['country'] == 'Unknown Country'
shows.loc[~mask,['show_id','country']].drop_duplicates(keep = 'first')['country'].value_counts().head(10).plot(kind = 'bar')
plt.title('Highest Number of shows released')

plt.show()
```



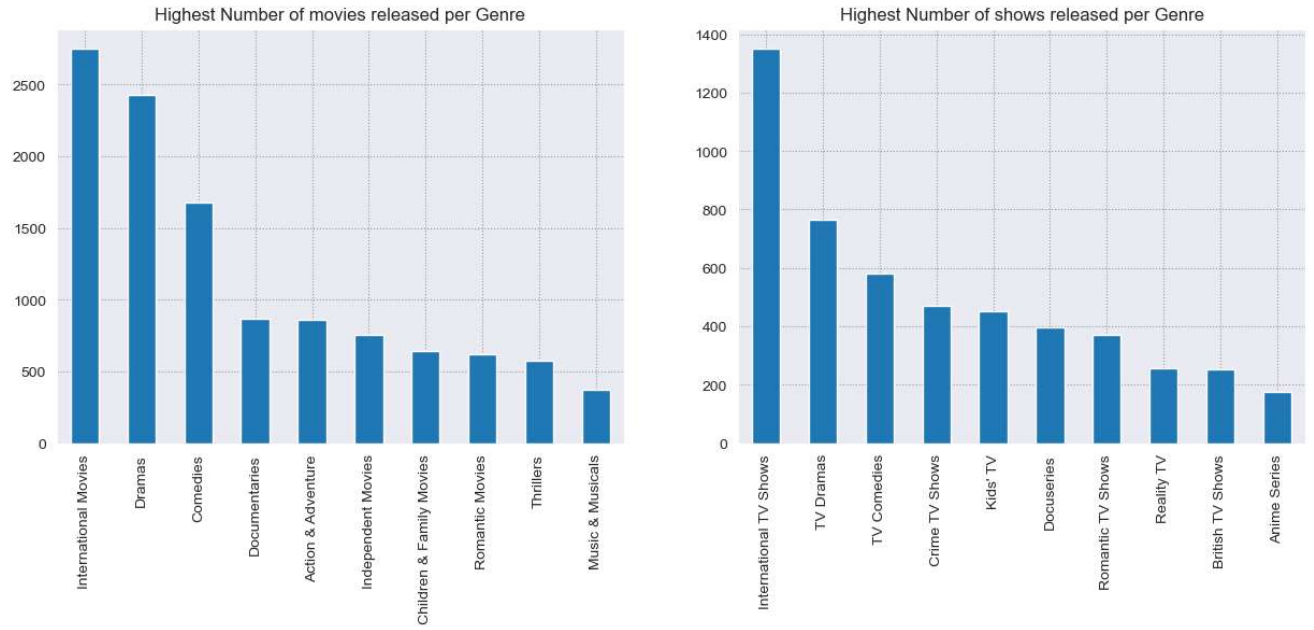
In [ ]:
```
1
```

Inferences from Country:

a. Highest number of movies were released in United States Followed by India and Uk.

b. Highest number of TV Shows were released in United States followed by UK and Japan.

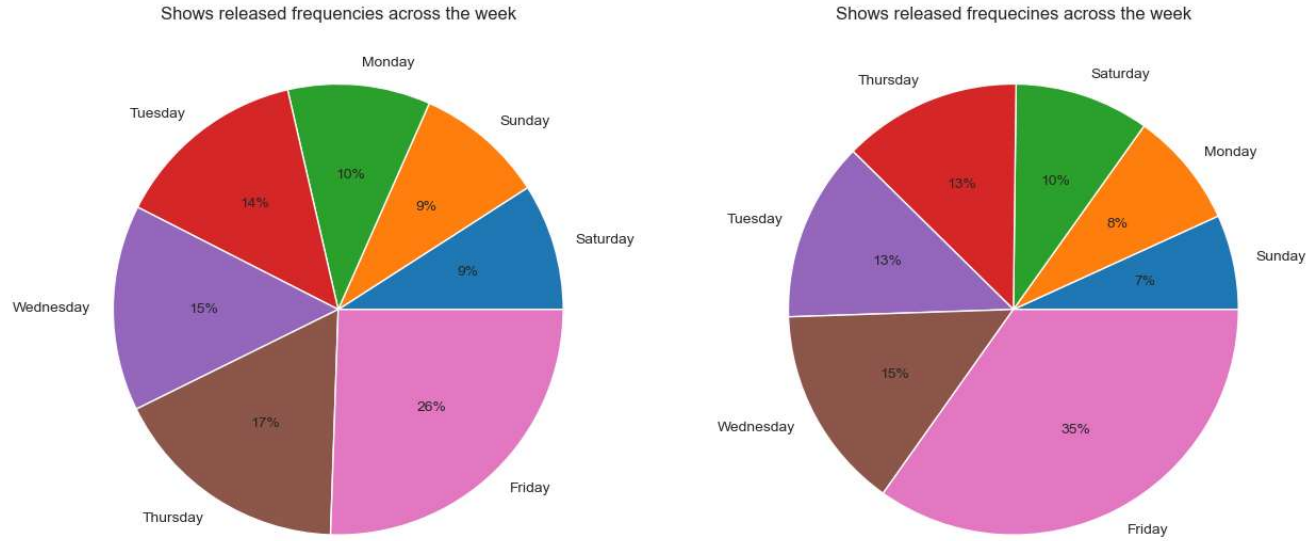
In [54]:
```
plt.figure(figsize =(15,5))
plt.subplot(1,2,1)
movies[['show_id','listed_in']].drop_duplicates(keep = 'first')['listed_in'].value_counts().head(10).plot(kind = 'bar')
plt.title('Highest Number of movies released per Genre')


plt.subplot(1,2,2)
shows[['show_id','listed_in']].drop_duplicates(keep = 'first')['listed_in'].value_counts().head(10).plot(kind = 'bar')
plt.title('Highest Number of shows released per Genre')
plt.show()
```



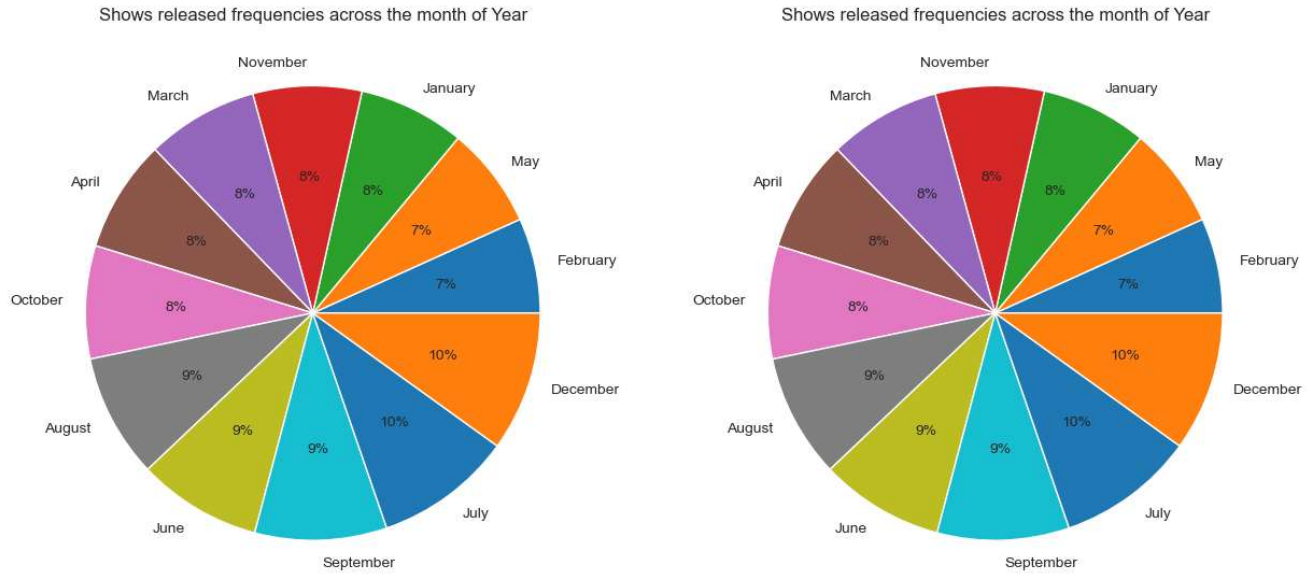
Observations from Genres:

a. Highest Number of Movies/TV Shows are from International Movies, Dramas and Comedy Shows.

In [55]:
```
plt.figure(figsize =(15,8))

plt.subplot(1,2,1)
day_name = movies[['show_id','dayname']].drop_duplicates(keep = 'first')['dayname'].value_counts(ascending = True)
plt.pie(day_name, labels= day_name.index, autopct='%.0f%%')
plt.title('Shows released frequencies across the week')

plt.subplot(1,2,2)
day_name = shows[['show_id','dayname']].drop_duplicates(keep = 'first')['dayname'].value_counts(ascending = True)
plt.pie(day_name, labels= day_name.index, autopct='%.0f%%')
plt.title('Shows released frequecines across the week')
plt.show()
```



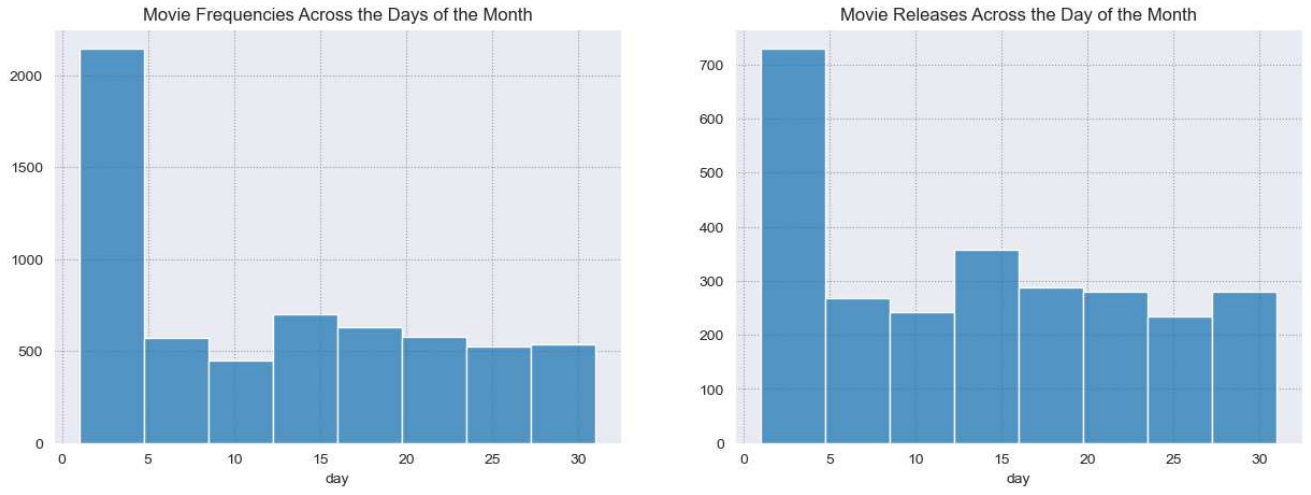
In [56]:
```
plt.figure(figsize =(15,8))

plt.subplot(1,2,1)
month_name = shows[['show_id','month']].drop_duplicates(keep = 'first')['month'].value_counts(ascending = True)
plt.pie(month_name, labels= month_name.index, autopct='%.0f%%')
plt.title('Shows released frequencies across the month of Year')

plt.subplot(1,2,2)
month_name = shows[['show_id','month']].drop_duplicates(keep = 'first')['month'].value_counts(ascending = True)
plt.pie(month_name, labels= month_name.index, autopct='%.0f%%')
plt.title('Shows released frequencies across the month of Year')

plt.show()
```



Observations:

a. Most of the TV Shows/Movies are added in Netflix in December or July

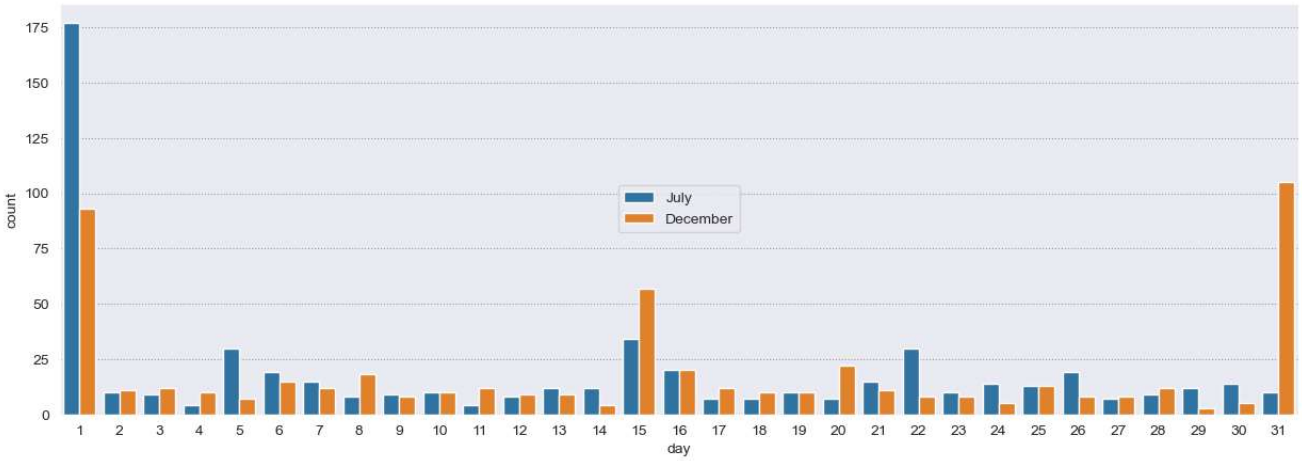
In [57]:
```
plt.figure(figsize =(15,5))

plt.subplot(1,2,1)
days = movies[['show_id','day']].drop_duplicates(keep = 'first')['day']
sns.histplot(days,bins = 8)
plt.title('Movie Frequencies Across the Days of the Month')
plt.ylabel('')

plt.subplot(1,2,2)
days = shows[['show_id','day']].drop_duplicates(keep = 'first')['day']
sns.histplot(days,bins = 8)
plt.title('Movie Releases Across the Day of the Month')
plt.ylabel('')
plt.show()
```



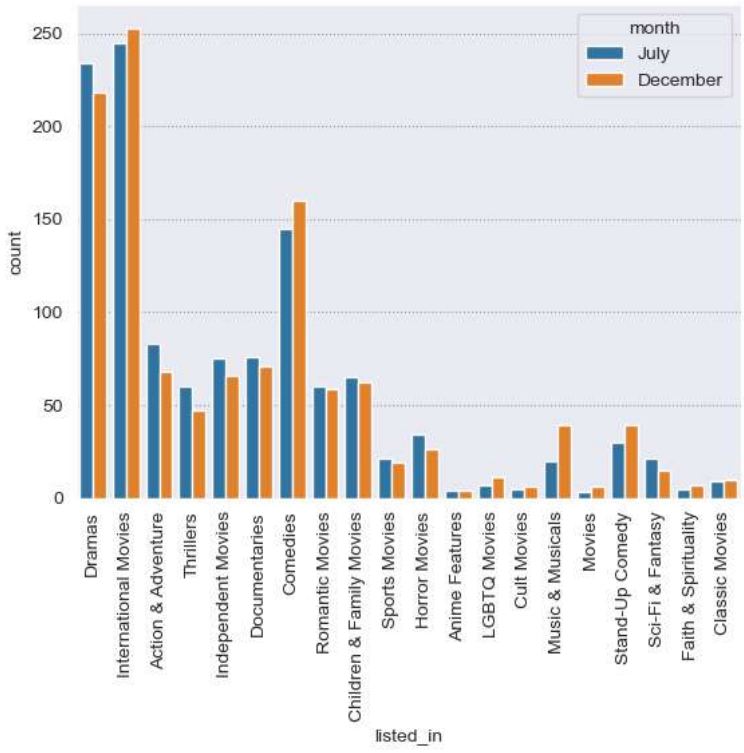
Observations:

a. Most of the TV Shows/Movies are added in Netflix in the first week

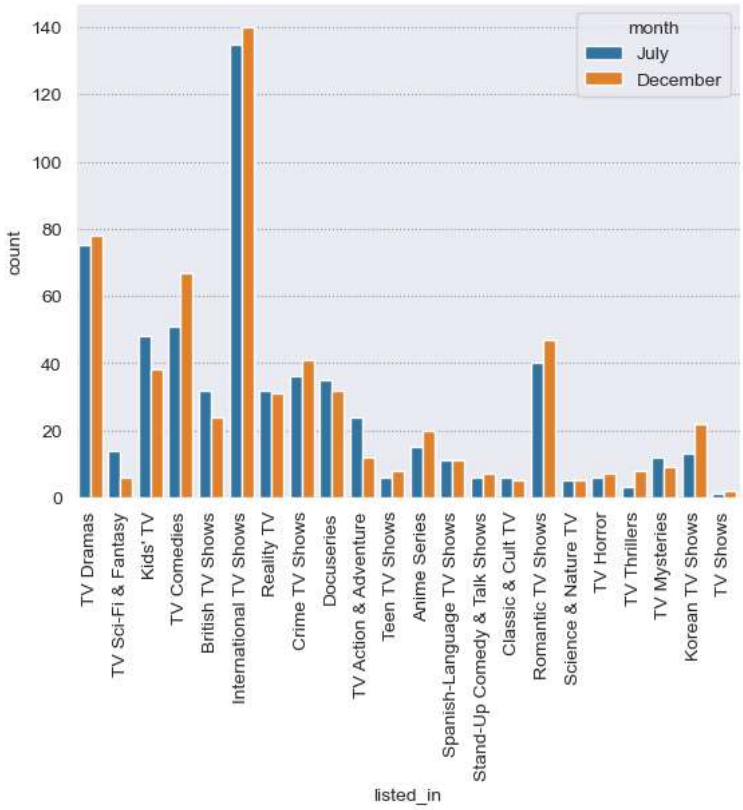
In [58]:
```
mon_list = np.array(['December','July'])
mon_movies = movies.loc[movies['month'].isin(mon_list),['show_id','day','month']].drop_duplicates(keep = 'first')
plt.figure(figsize = (15,5))
sns.countplot(data = mon_movies,x = 'day',hue = 'month')
plt.legend(loc='center')
plt.show()
```



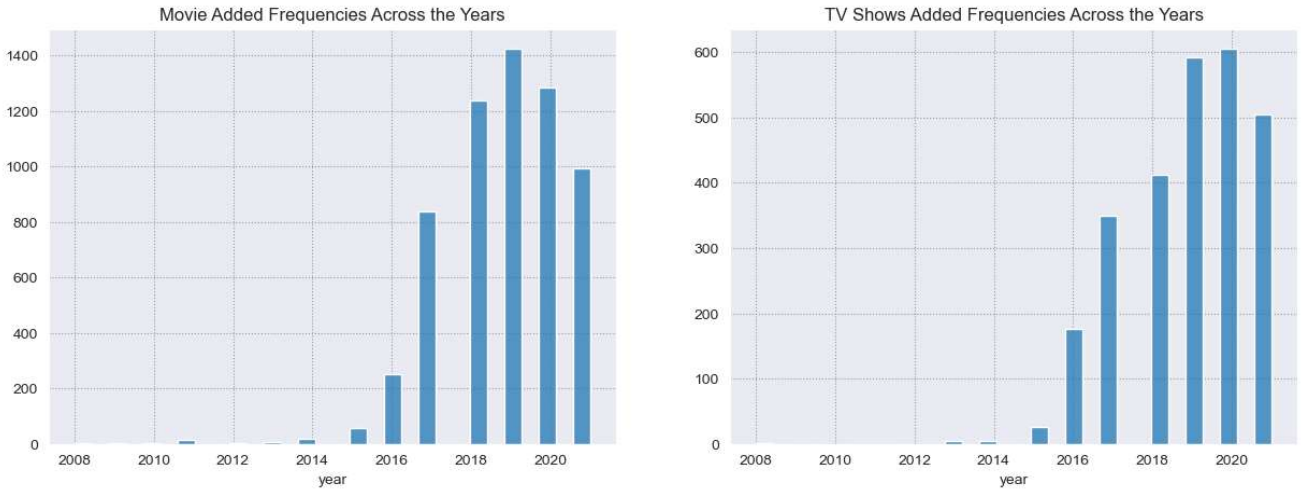
In [59]:
```
mon_list = np.array(['December','July'])
mon_movies = movies.loc[movies['month'].isin(mon_list),['show_id','listed_in','month']].drop_duplicates(keep = 'first')[['mo
sns.countplot(data = mon_movies,x = 'listed_in',hue = 'month')
plt.xticks(rotation  = 90)
plt.show()
```



In [60]:
```
mon_list = np.array(['December','July'])
mon_movies = shows.loc[shows['month'].isin(mon_list),['show_id','listed_in','month']].drop_duplicates(keep = 'first')[['mont
sns.countplot(data = mon_movies,x = 'listed_in',hue = 'month')
plt.xticks(rotation  = 90)
plt.show()
```



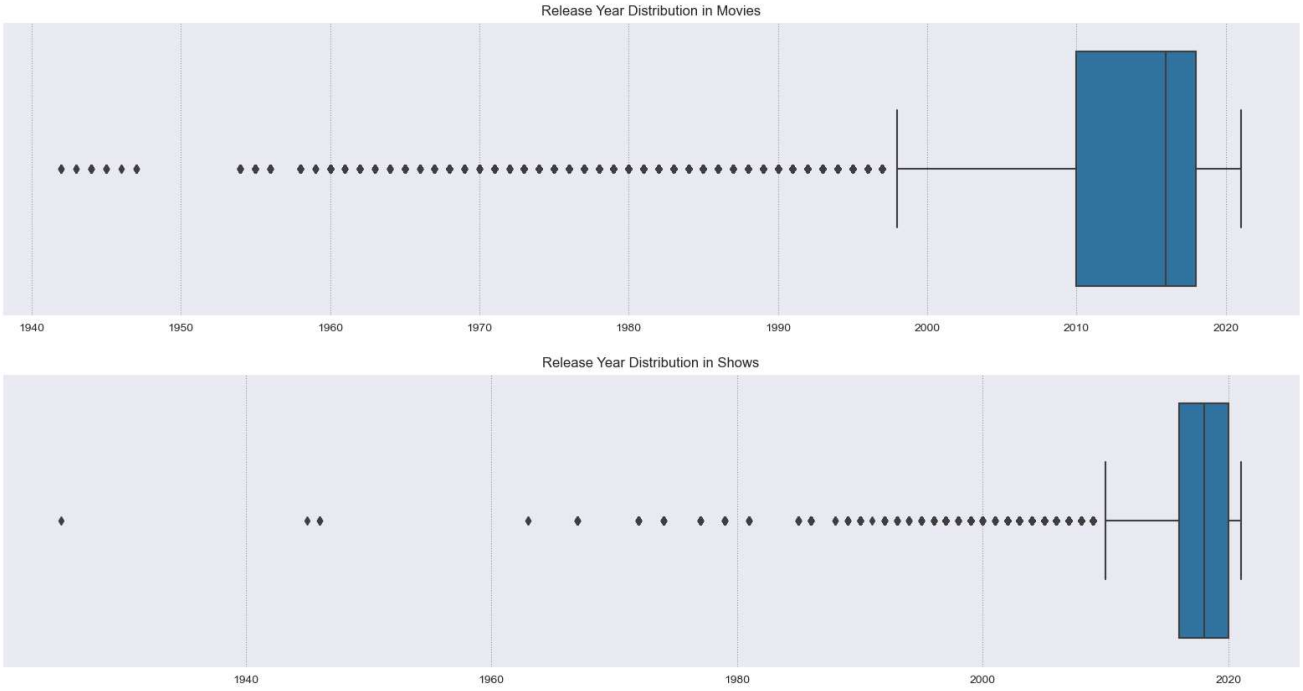
In [61]:
```
plt.figure(figsize =(15,5))

plt.subplot(1,2,1)
days = movies[['show_id','year']].drop_duplicates(keep = 'first')['year']
sns.histplot(days,bins = 30)
plt.title('Movie Added Frequencies Across the Years')
plt.ylabel('')

plt.subplot(1,2,2)
days = shows[['show_id','year']].drop_duplicates(keep = 'first')['year']
sns.histplot(days,bins = 30)
plt.title('TV Shows Added Frequencies Across the Years')
plt.ylabel('')
plt.show()
```



Inferences from Date Added :

a. Most of the TV Shows/Movies are added in Netflix in December or July
b. Most of the TV Shows/Movies are added in Netflix in the first week
c. Most of the movies are added in Month of December or July in the first week or last week
d. Most of the movies are added in Month of December or July have genres Dramas International Movies and Comedies
e. Most of the TV Shows are added in Month of December or July in the first week or last week
f. Most of the TV Shows are added in Month of December or July have genres Dramas International Movies and Comedies
g. Range of Year Added in 13 years

In [62]:
```
plt.figure(figsize = (20,10))
plt.subplot(2,1,1)
sns.boxplot(data = movies,x= 'release_year')
plt.title('Release Year Distribution in Movies')
plt.xlabel('')

plt.subplot(2,1,2)
sns.boxplot(data = shows,x = 'release_year')
plt.title('Release Year Distribution in Shows')
plt.xlabel('')

plt.show()

```



In [63]:
```
df[df['type'] == 'Movie'].describe()
```

Out[63]:

	release_year
count	6131.000000
mean	2013.121514
std	9.678169
min	1942.000000
25%	2012.000000
50%	2016.000000
75%	2018.000000
max	2021.000000

In [64]:
```
df[df['type'] == 'TV Show'].describe()
```

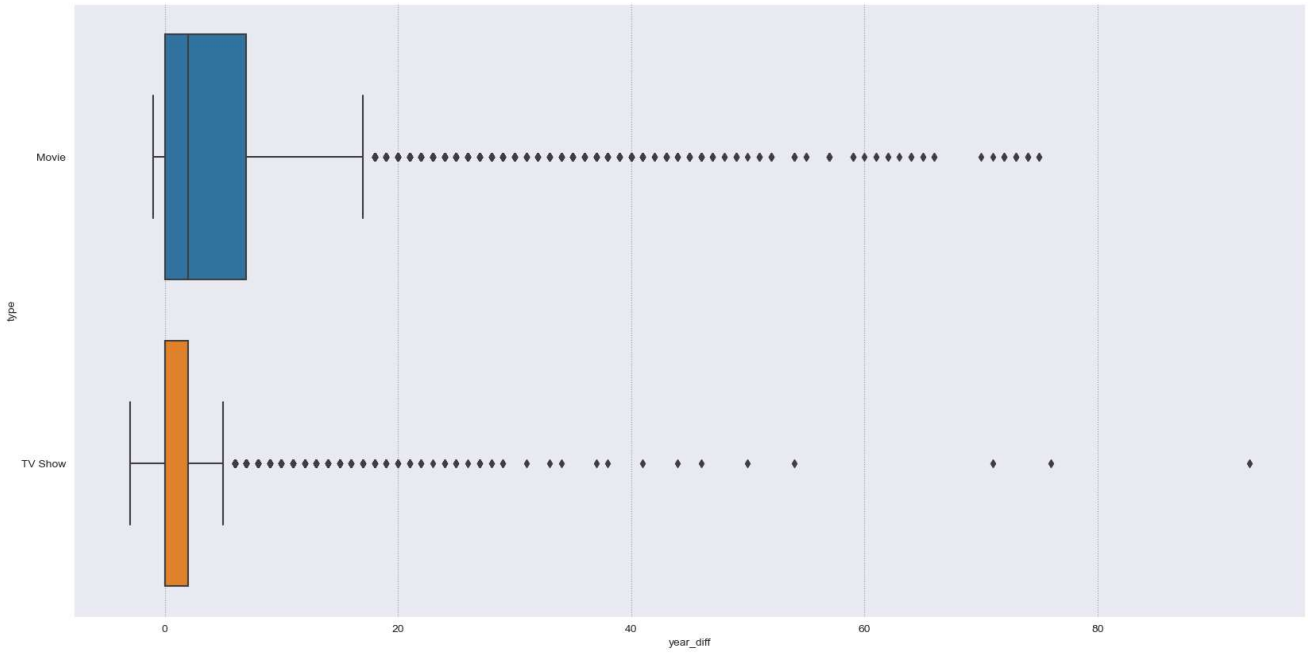
Out[64]:

	release_year
count	2676.000000
mean	2016.605755
std	5.740138
min	1925.000000
25%	2016.000000
50%	2018.000000
75%	2020.000000
max	2021.000000

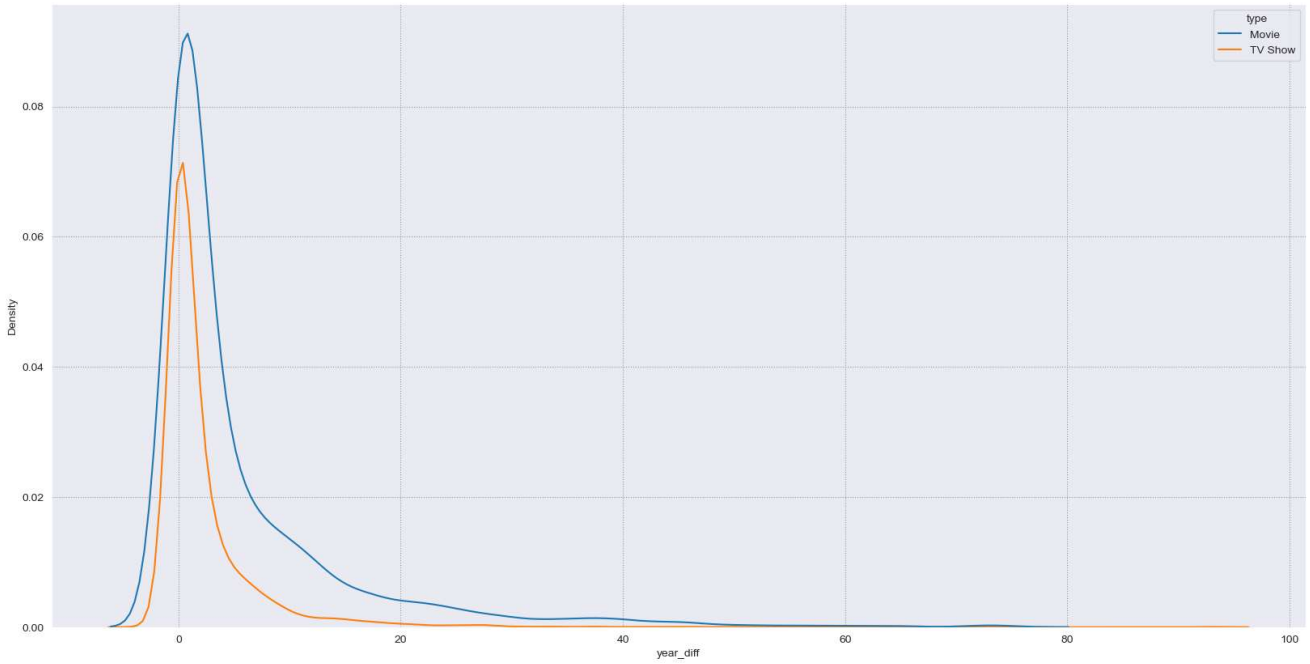
Inferences from Release Year:

a. Very few movies were released before 2000 that are present in this dataset
b. Very few TV Shows were released before 2010 that are present in this dataset
c. Most of the movies were released between 2012 to 2018 that are present in this dataset
d. Very few TV Shows were released between 2016 to 2020 that are present in this dataset
e. Range of Release Year for Movies is equal to 79 years, for TV Shows it is equal to 96 years

In [65]:
```
plt.figure(figsize = (20,10))
box = final_df[['show_id','type','year_diff']].drop_duplicates()
sns.boxplot(data = box,x='year_diff',y  = 'type')
plt.show()
```



In [66]:
```
plt.figure(figsize = (20,10))
box = final_df[['show_id','type','year_diff']].drop_duplicates()
sns.kdeplot(data = box,x='year_diff',hue= 'type')
plt.show()

```



In [67]:
```
box[box['type'] == 'Movie'].max()
```

Out[67]:
```
show_id       s999
type         Movie
year_diff       75
dtype: object
```

In [68]:
```
box[box['type'] == 'TV Show'].max()
```

Out[68]:
```
show_id         s998
type         TV Show
year_diff         93
dtype: object
```

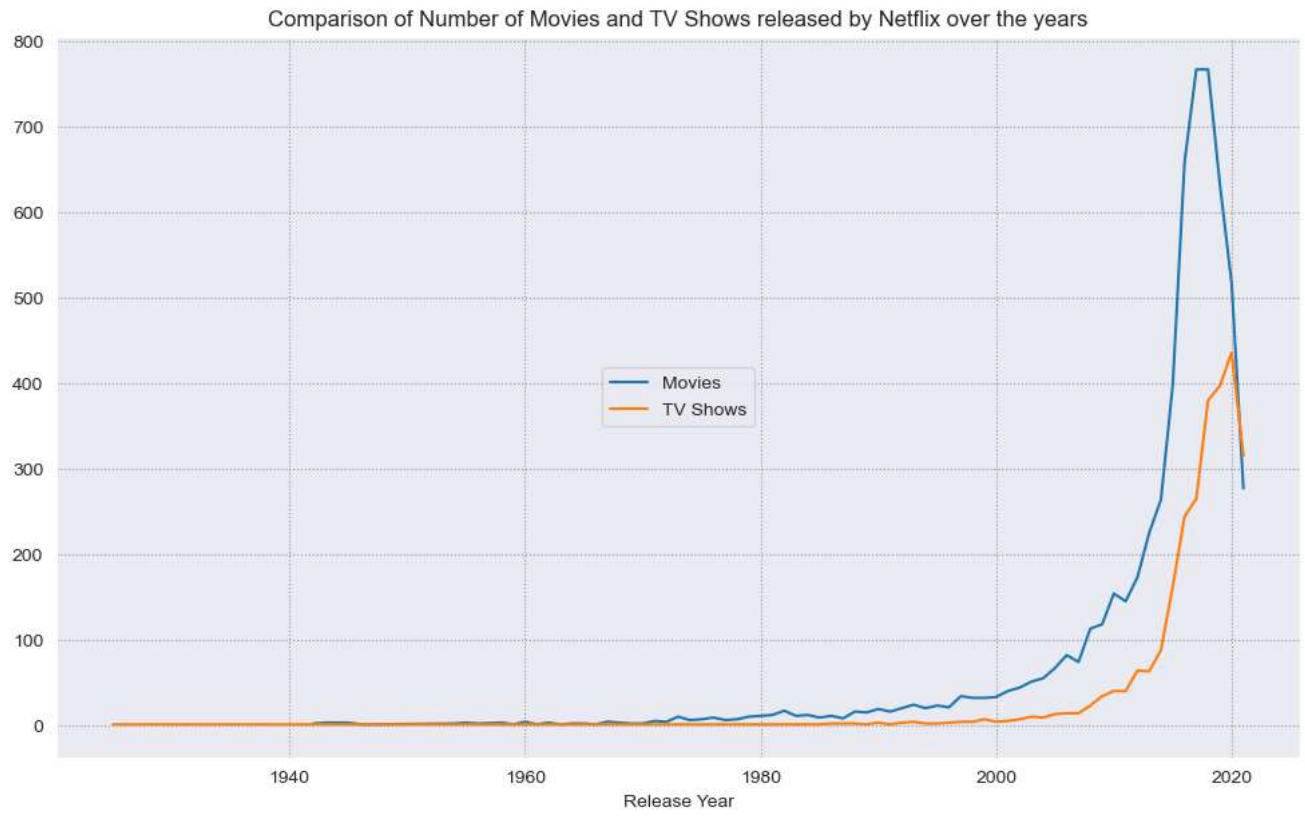
Inferences from difference between year added and year released:

a. Most of the movies/tv shows were added to Netflix in the same year as it was released

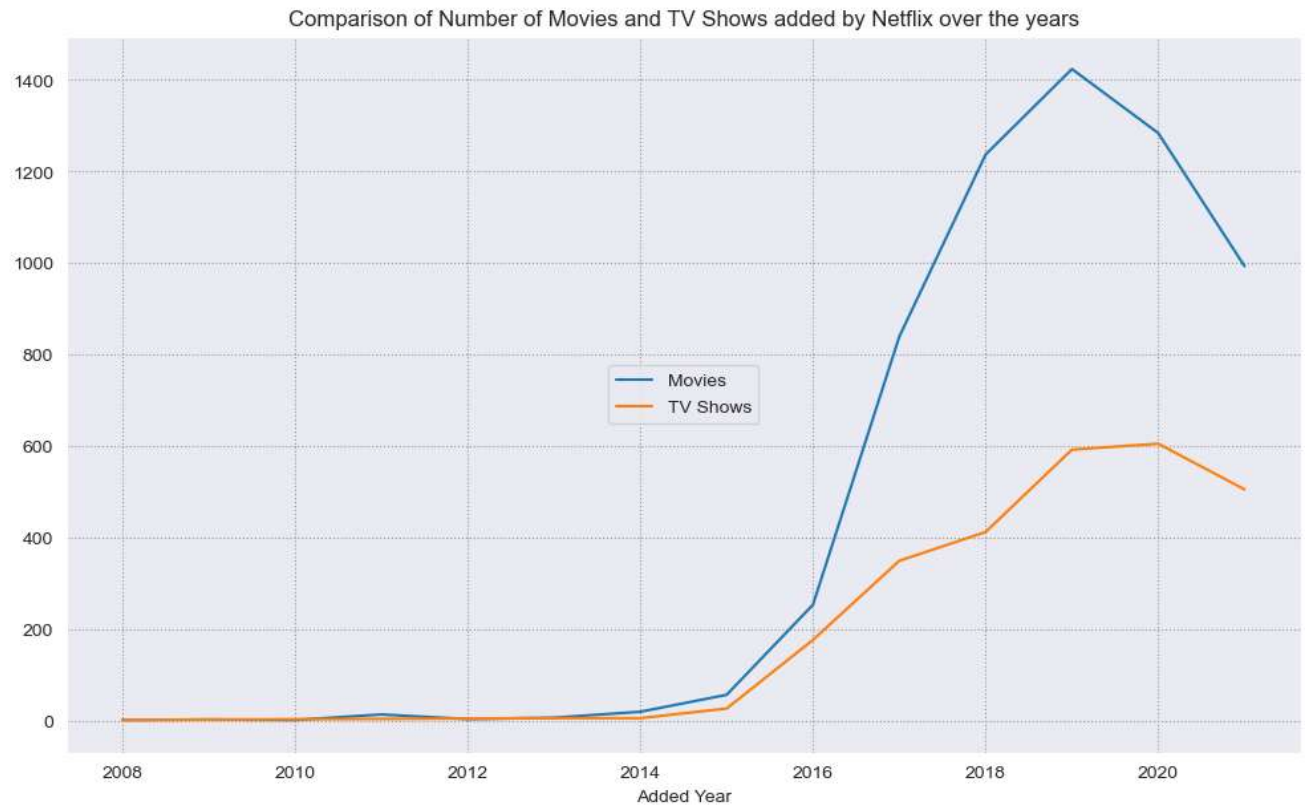
b. Highest year difference between when it was released and when it was added in Netflix is 75 and 93 for movies and TV Shows respectively

In [69]:

```
plt.figure(figsize = (12,7))

movies_released_per_year = df.loc[df['type']=='Movie','release_year'].value_counts().sort_index()
sns.lineplot(x = movies_released_per_year.index,y = movies_released_per_year,label = 'Movies')

shows_released_per_year = df.loc[df['type']=='TV Show','release_year'].value_counts().sort_index()
sns.lineplot(x =shows_released_per_year.index,y = shows_released_per_year,label = 'TV Shows')

plt.xlabel('Release Year')
plt.ylabel('')
plt.title('Comparison of Number of Movies and TV Shows released by Netflix over the years')
plt.legend(loc = 'center')

plt.show()
```



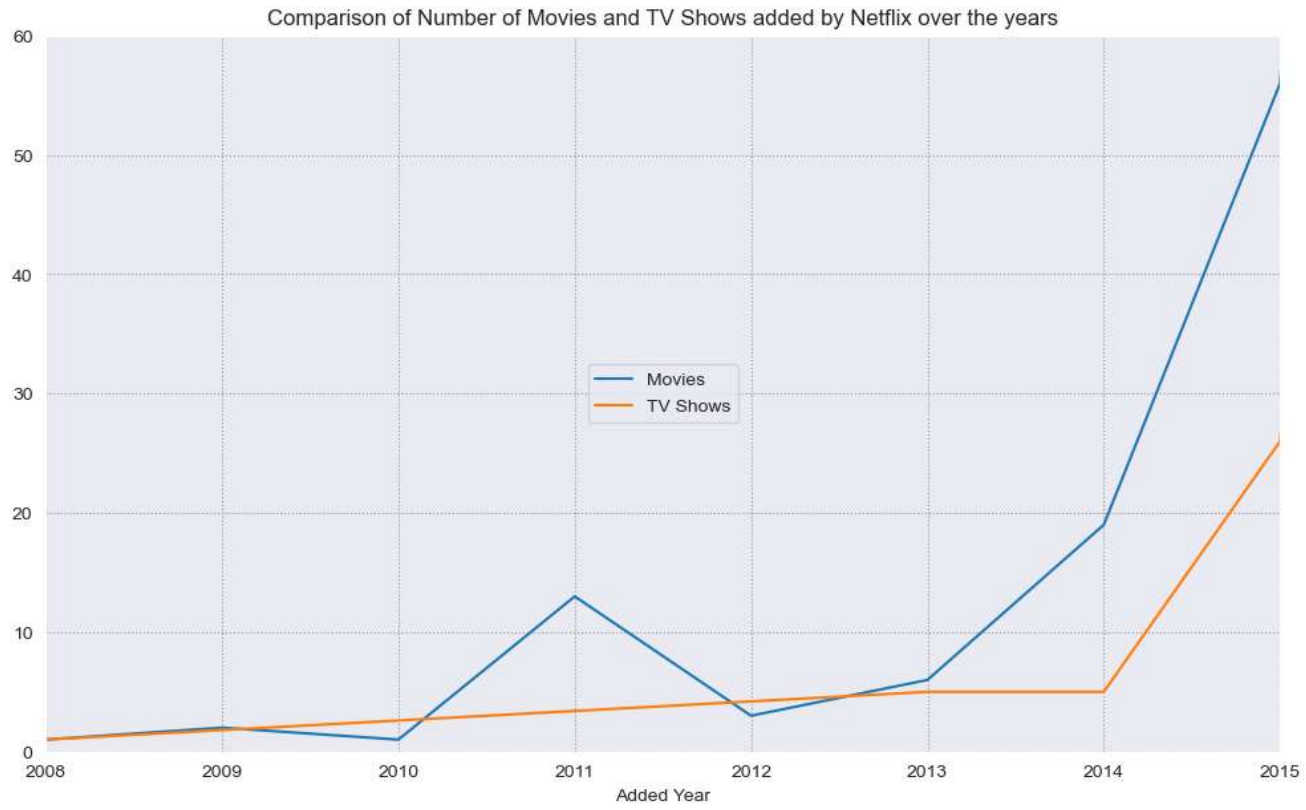
In [70]:
```
plt.figure(figsize = (12,7))

movies_added_per_year = movies.groupby('year')['show_id'].nunique()
sns.lineplot(x = movies_added_per_year.index,y = movies_added_per_year,label = 'Movies')

shows_added_per_year = shows.groupby('year')['show_id'].nunique()
sns.lineplot(x =shows_added_per_year.index,y = shows_added_per_year,label = 'TV Shows')

plt.xlabel('Added Year')
plt.ylabel('')
plt.title('Comparison of Number of Movies and TV Shows added by Netflix over the years')
plt.legend(loc = 'center')

plt.show()
```



Number of Shows Released Across the Years :

a. In the recent years we can there has been a drop in release as well as drop in addition of Movies and Tv Shows. This maybe due to lack of data. As we donot have data we cannot conclude the above statement as true

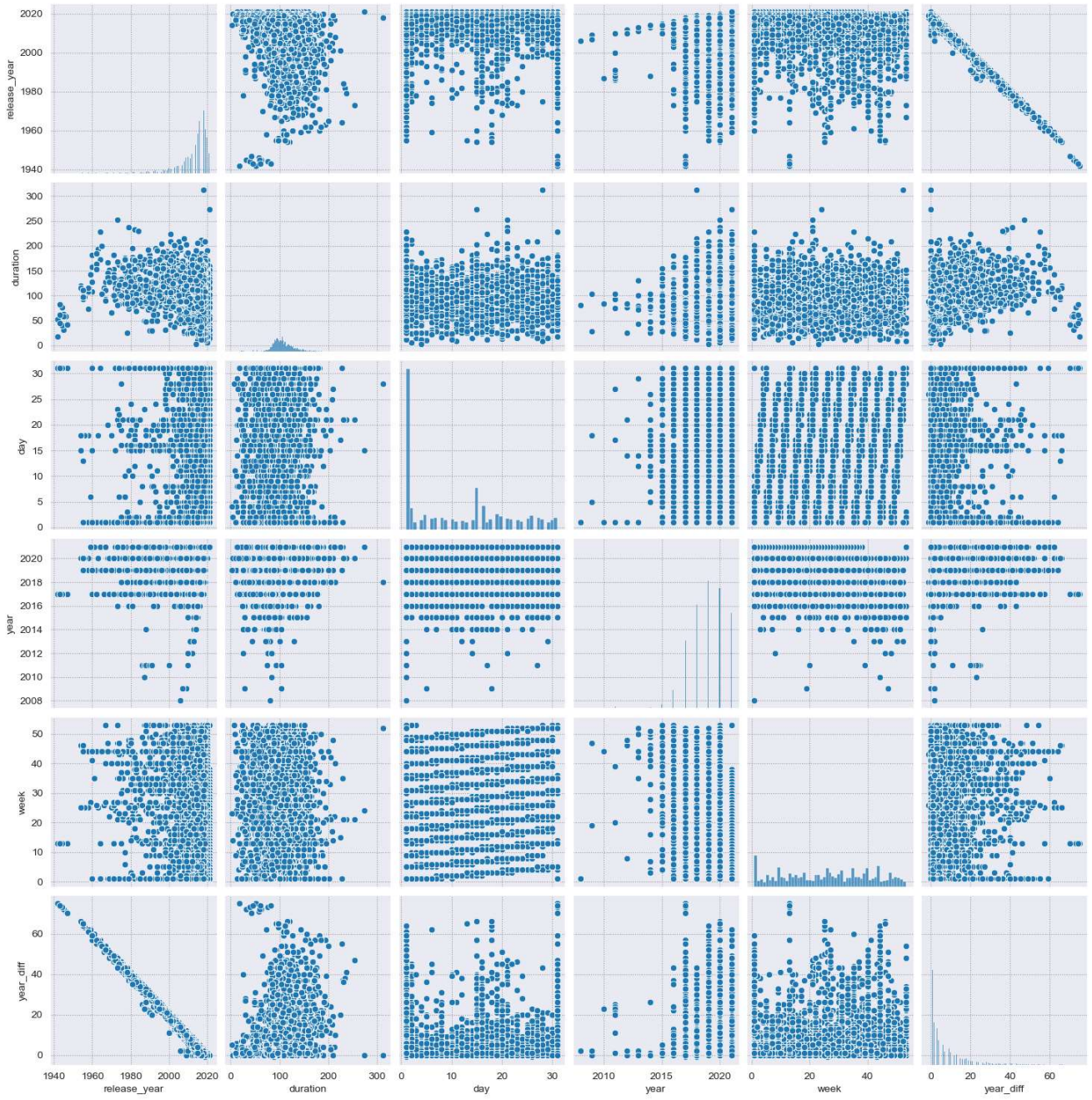
In [71]:
```
plt.figure(figsize = (12,7))

movies_added_per_year = movies.groupby('year')['show_id'].nunique()
sns.lineplot(x = movies_added_per_year.index,y = movies_added_per_year,label = 'Movies')

shows_added_per_year = shows.groupby('year')['show_id'].nunique()
sns.lineplot(x =shows_added_per_year.index,y = shows_added_per_year,label = 'TV Shows')

plt.xlabel('Added Year')
plt.ylabel('')
plt.title('Comparison of Number of Movies and TV Shows added by Netflix over the years')
plt.legend(loc = 'center')
plt.xlim(2008,2015)
plt.ylim(0,60)

plt.show()
```



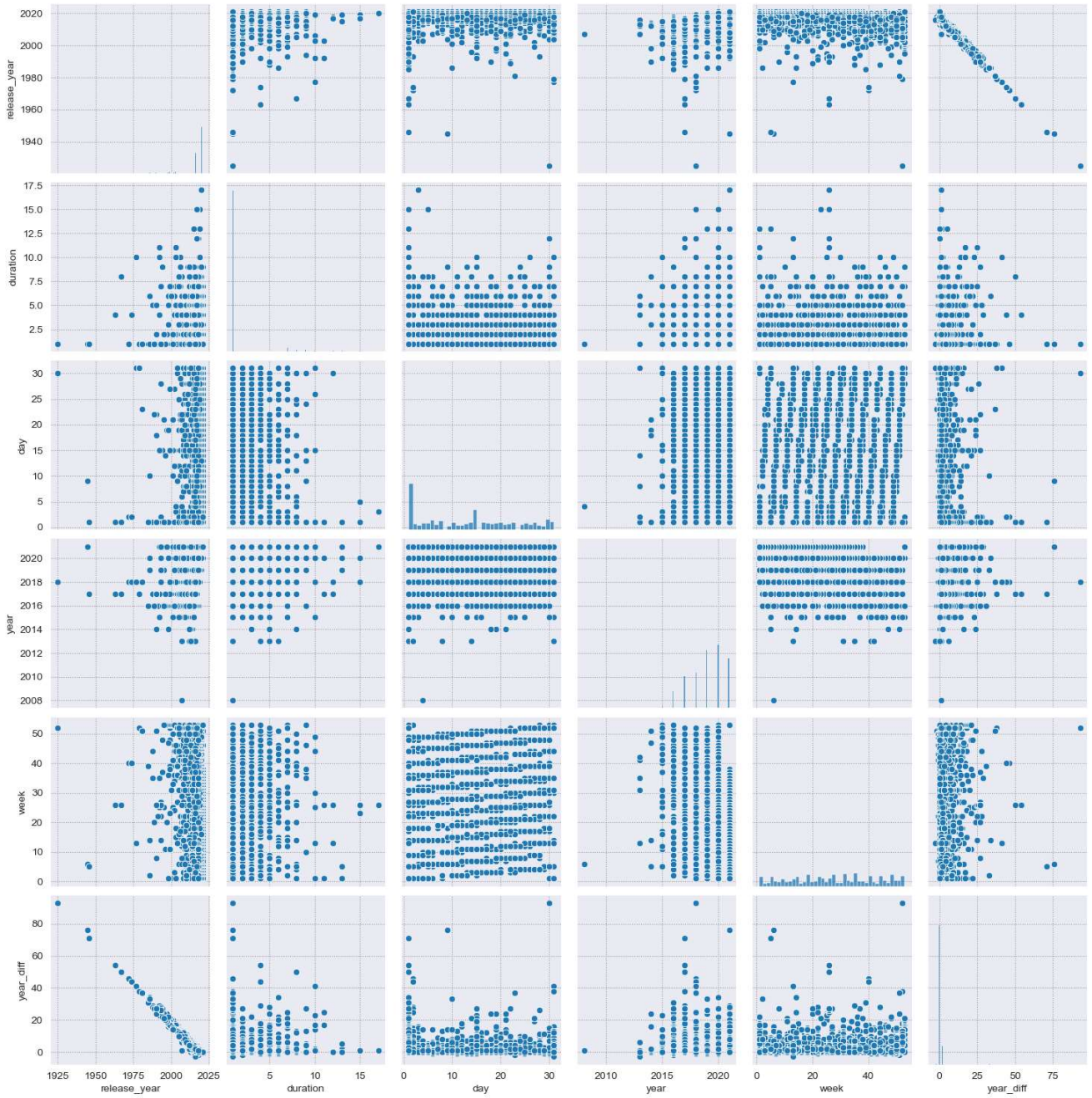
Number of Shows Added across the years:

a. There has been spike in addtion of Movies and spike in addtion of TV Shows from 2013 and 2014 respectively.

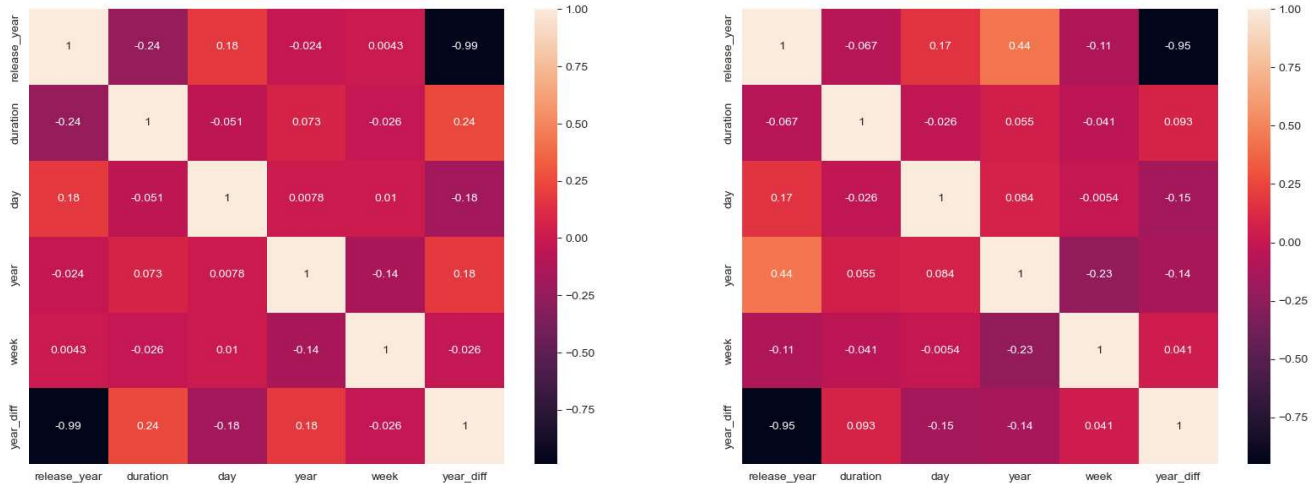
In [72]:
```
1 sns.pairplot(data = movies)
2 plt.show()
```



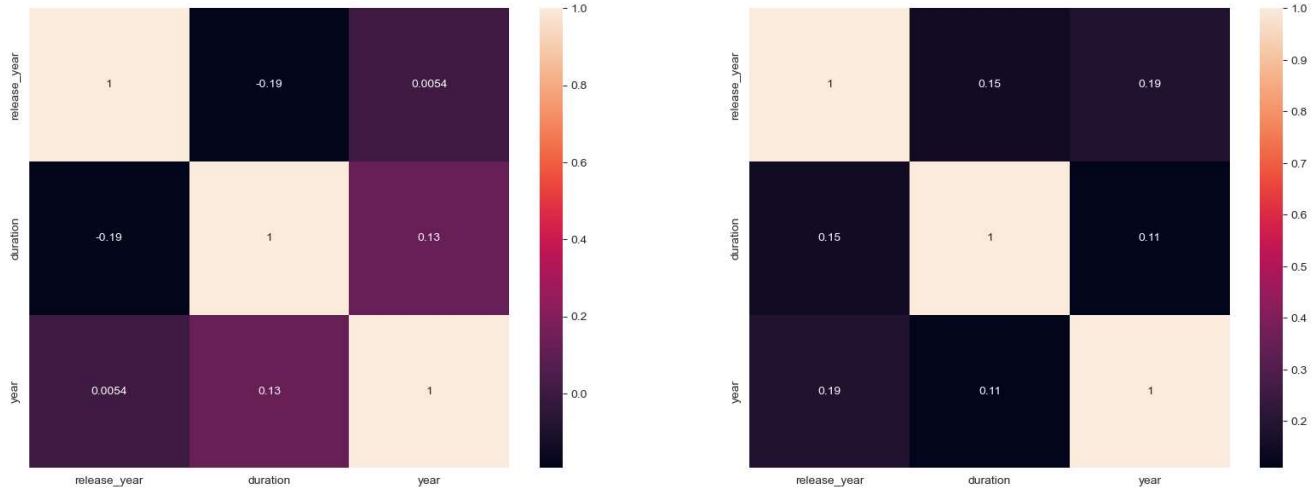
In [73]:
```
sns.pairplot(data = shows)
plt.show()
```



In [74]:
```
plt.figure(figsize=(20,7))
plt.subplot(1,2,1)
sns.heatmap(movies.corr(),annot = True)


plt.subplot(1,2,2)
sns.heatmap(shows.corr(),annot = True)
plt.show()
```



In [75]:
```
plt.figure(figsize=(20,7))
plt.subplot(1,2,1)
corr_mov_data = movies[['release_year','duration','year']].drop_duplicates()
sns.heatmap(corr_mov_data.corr(),annot = True)


plt.subplot(1,2,2)
corr_shows_data = shows[['release_year','duration','year']].drop_duplicates()
sns.heatmap(corr_shows_data.corr(),annot = True)
plt.show()
```

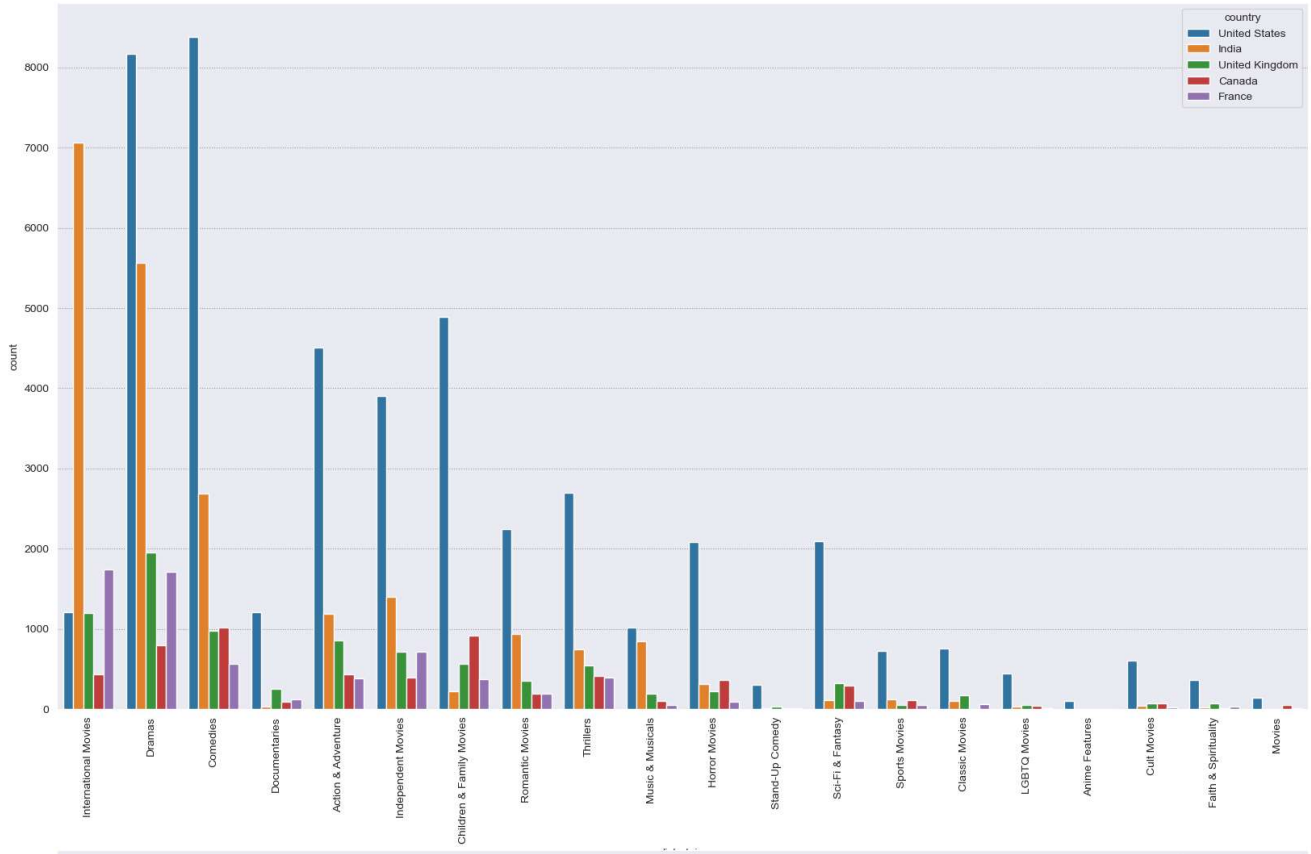


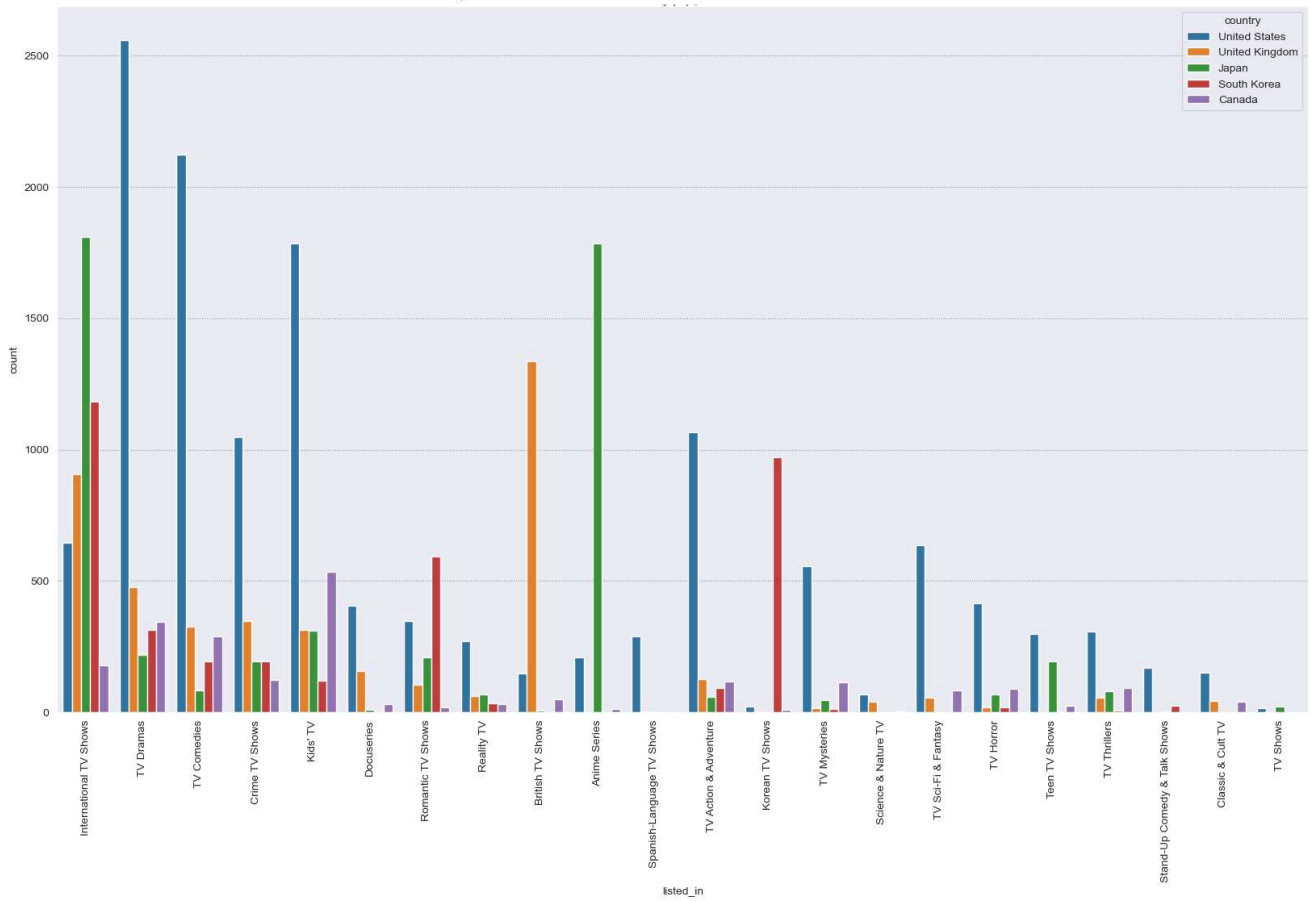
Observations:

a. Except for release_year and year_diff, any clear correlation between any other columns cannot been seen.

In [76]:

```
mask = movies['country'] == 'Unknown Country'
mov_country_list = movies.loc[~mask,['show_id','country']].drop_duplicates(keep = 'first')['country'].value_counts().head(5)

mask = shows['country'] == 'Unknown Country'
show_country_list = shows.loc[~mask,['show_id','country']].drop_duplicates(keep = 'first')['country'].value_counts().head(5)


mov_cg = movies[movies['country'].isin(mov_country_list)]
show_cg = shows[shows['country'].isin(show_country_list)]

mov_order = movies[['show_id','listed_in']].drop_duplicates(keep = 'first')['listed_in'].value_counts().index.tolist()
show_order = shows[['show_id','listed_in']].drop_duplicates(keep = 'first')['listed_in'].value_counts().index.tolist()


plt.figure(figsize = (20,25))
plt.subplot(2,1,1)
sns.countplot(data = mov_cg,x = 'listed_in',hue = 'country',order = mov_order,hue_order=mov_country_list)
plt.xticks(rotation = 90)

plt.subplot(2,1,2)
sns.countplot(data = show_cg,x = 'listed_in',hue = 'country',order = show_order,hue_order=show_country_list)
plt.xticks(rotation = 90)
plt.show()
```



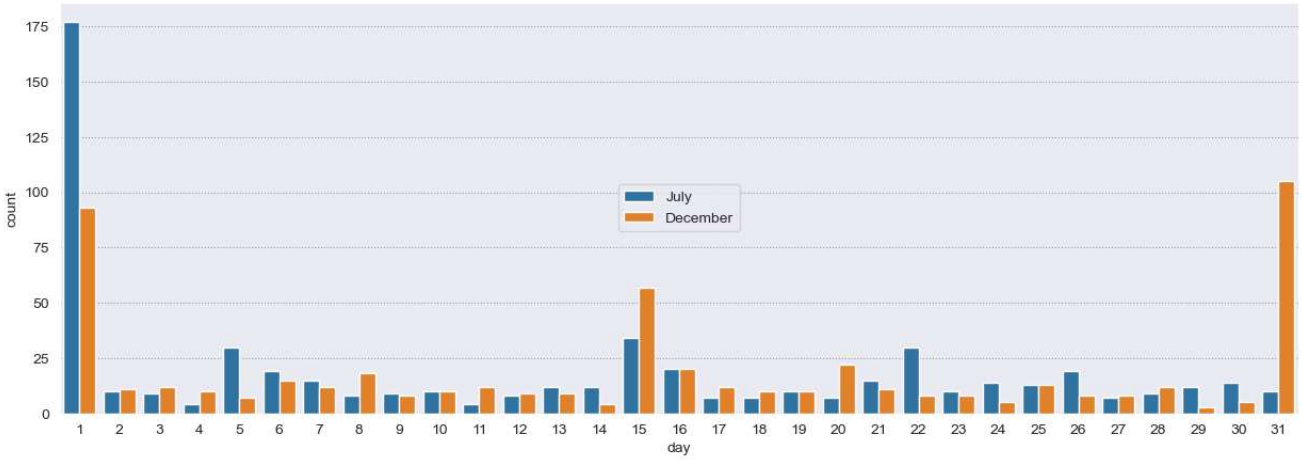


Infernences from Top 5 Countries and Genres:

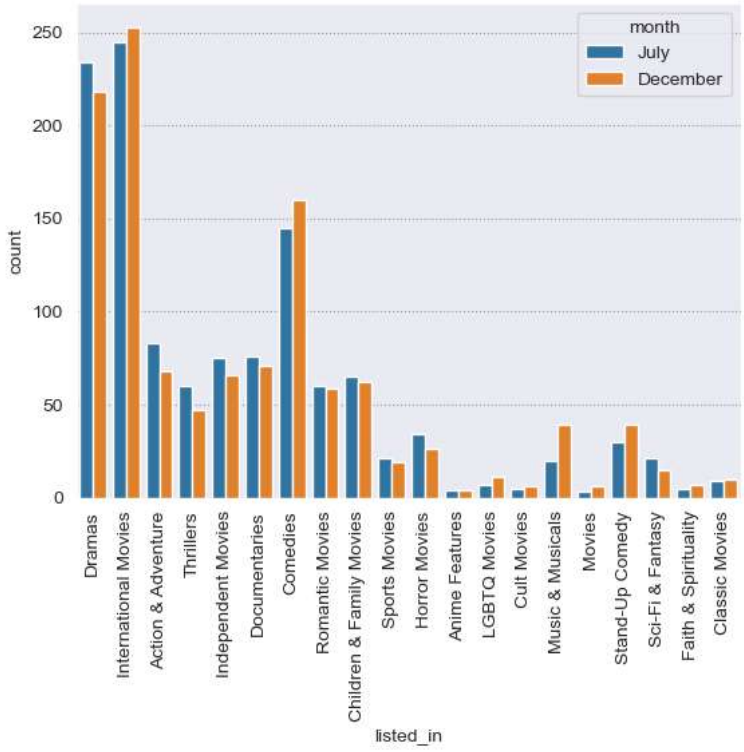
a. Most TV shows in United States are of Dramas, Comedy and Kids Genre.
b. Most TV Shows in United Kingdom are of British TV shows, International Shows and Dramas.
c. Most TV shows in Japan are of International Shows and Anime Series.
d. Most TV Shows in South Korea are of International Shows, Korean TV shows and Romantic TV Shows.

e. Most Movies in United States are of Dramas, Comedy and Children & Family Genre.
f. Most Movies in United Kingdom are of International Movies, Dramas and Comedy Genre.
g. Most Movies in India are of International Movies, Dramas and Comedy Genre.
h. Most Movies in France are of International Movies and Dramas

In [77]:
```
mon_list = np.array(['December','July'])
mon_movies = movies.loc[movies['month'].isin(mon_list),['show_id','day','month']].drop_duplicates(keep = 'first')
plt.figure(figsize = (15,5))
sns.countplot(data = mon_movies,x = 'day',hue = 'month')
plt.legend(loc='center')
plt.show()
```



In [78]:
```
mon_list = np.array(['December','July'])
mon_movies = movies.loc[movies['month'].isin(mon_list),['show_id','listed_in','month']].drop_duplicates(keep = 'first')[['mo
sns.countplot(data = mon_movies,x = 'listed_in',hue = 'month')
plt.xticks(rotation  = 90)
plt.show()
```



In [ ]:
```
```

Buisness Insights

Type:

a. There are Only Two types of Show -> Movies and TV Shows

b. Out of 8807 shows 6131 shows are Movies and 2676 shows are TV Shows

Rating:

a. There were a total of 17 ratings present for movies. Only 9 of which are ratings used in TV Shows

b. Netlix caters to a lot of Mature audience, 34% of movies and 48% of tv shows that are avaiable content is for mature

c. 23% and 27% movies and tv shows rated respectively as TV-14 i.e. children under age of 14 are not suitable to watch, target audience been mid and late teens

d. There are around 13% R Rated movies.

e.There are only 4% movies and 14% of TV Shows available for kids(TV-Y and TV-Y7)

Duration:

a. 4499(~73%) movies are between 1hr and 2hr. 1095 Movies are between 2hr and 3hr.

b. 487 movies are less than 1hr. Only 47 movies are greater than 3hr.

c. TV Shows are mostly of only one season around 65%. There's one such TV Show which has 17 seasons.

d. There are only 26 such TV shows which have more than 8 seasons

Director:

a. There were a total of 4528 directors in original dataset

b. There are a total of 4993 directors in the unnested dataset. Out of which 4777 directors worked in movies and only 299 directors worked in TV shows. Only 84 directors worked both in Movies and TV Shows

c. Rajiv Chilaka directed highest number of movies.

d. Alaistar Fothergill directed highest number of TV Shows.

Cast:

a. There were a total of 7692 actors in original dataset

b. There are a total of 36439 casted actors/actress present in the unnested dataset. Out of which 25951 worked in movies and 14863 worked in TV Shows. Only 4376 worked both in Movies and TV Shows

c. Anupam Kher has appeared in most of movies.

d. Takahiko Sakurai has apperead in most of TV Shows.

Country:

a. There were a total of 748 different values of clubbe country in original dataset

b. There are a total of 123 countries where these shows were available. Movies were accessible in 118 different countries and only 66 countries for TV Shows

c. Highest number of movies were released in United States Followed by India and Uk.

d. Highest number of TV Shows were released in United States followed by UK and Japan.

Genre/Listed_in:

a. There are a total of 42 genres values of present in the dataset. Out of which 20 belong to Movies and 22 belong the TV shows

b. There are a total of 123 countries where these shows were available

c. Highest Number of Movies/TV Shows are from International Movies, Dramas and Comedy Shows.

Years:

a. These movies/TV Shows were released in 74 different years starting from 1925. First TV Shows that was realeased in the dataset was in year 1925 and Movie was in year 1942.

b. 75% of movies were released in the last decade and 75% of Shows were released in last 7 years.

c. Only from 2008 these tv shows/movies were added in Netflix. Most of the tv shows/movies were added in July following by December

d. Most of the tv shows/movies were released in Friday followed by Thursday e. Most of the TV Shows/Movies are added in Netflix in December or July

f. Most of the TV Shows/Movies are added in Netflix in the first week

g. Most of the movies are added in Month of December or July in the first week or last week

h. Most of the movies are added in Month of December or July have genres Dramas International Movies and Comedies

i. Most of the TV Shows are added in Month of December or July in the first week or last week

j. Most of the TV Shows are added in Month of December or July have genres Dramas International Movies and Comedies

k. Range of Year Added in 13 years

l. Very few movies were released before 2000 that are present in this dataset

m. Very few TV Shows were released before 2010 that are present in this dataset

n. Most of the movies were released between 2012 to 2018 that are present in this dataset

o. Very few TV Shows were released between 2016 to 2020 that are present in this dataset

p. Range of Release Year for Movies is equal to 79 years, for TV Shows it is equal to 96 years

a. Most TV shows in United States are of Dramas, Comedy and Kids Genre.
b. Most TV Shows in United Kingdom are of British TV shows, International Shows and Dramas.
c. Most TV shows in Japan are of International Shows and Anime Series.
d. Most TV Shows in South Korea are of International Shows, Korean TV shows and Romantic TV Shows.

e. Most Movies in United States are of Dramas, Comedy and Children & Family Genre.
f. Most Movies in United Kingdom are of International Movies, Dramas and Comedy Genre.
g. Most Movies in India are of International Movies, Dramas and Comedy Genre.
h. Most Movies in France are of International Movies and Dramas.

Inferences from Top 5 Countries and Genres:

a. Most TV shows in United States are of Dramas, Comedy and Kids Genre.
b. Most TV Shows in United Kingdom are of British TV shows, International Shows and Dramas.
c. Most TV shows in Japan are of International Shows and Anime Series.
d. Most TV Shows in South Korea are of International Shows, Korean TV shows and Romantic TV Shows.

e. Most Movies in United States are of Dramas, Comedy and Children & Family Genre.
f. Most Movies in United Kingdom are of International Movies, Dramas and Comedy Genre.
g. Most Movies in India are of International Movies, Dramas and Comedy Genre.
h. Most Movies in France are of International Movies and Dramas.

Other Inferences:

a. Most of the movies/tv shows were added to Netflix in the same year as it was released
b. Highest year difference between when it was released and when it was added in Netflix is 75 and 93 for movies and TV Shows respectively
c. In the recent years we can there has been a drop in release as well as drop in addition of Movies and Tv Shows.
d. There has been spike in addtion of Movies and spike in addtion of TV Shows from 2013 and 2014 respectively.

Recommendations

1. While catering to mature audiences is essential, consider diversifying content genres to attract a broader range of viewers. Produce a mix of genres, including drama, comedy, action, romance, and documentary, to cater to varied tastes.
2. Given the popularity of TV-14 rated content, create more shows and movies tailored for the late teens demographic.
3. Due to kids less attention span, shows of length 15-20 mins should be available more. Side by Side it is also very important to implement a robust parental control and ensure that the content is suitable for this age group
4. Focus on producing movies that fall within the popular 1-hour to 2-hour duration range.
5. We have observed a gradual decline in the number of TV shows with more than one season. A strategic approach is to develop TV shows spanning 3-5 seasons, with each season having a compelling cliffhanger. This will captivate viewers interest and anticipation, making them to eagerly await for the next season.
6. Additionally we can create brief glimpses of behind the screens or share entertaining bloopers, providing a relatable and authentic connection to our audience.
7. We can bring some of the most entertaining old movies, that were released before 2010, which will help to cater the elderly audience. It will work especially in Japan due higher older demographic.
8. Can Experiment with other genres like Sci-Fi, Fantasy, Thriller, and Documentaries.
9. Leverage the trend of adding most TV shows and movies in December and July. Plan the release of highly anticipated original content during these months to attract maximum viewership.

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```