In [66]:

```
import pandas as pd
data = pd.read_csv('netflix.csv')
```

In [67]:

```
data
```

Out[67]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duratio
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 mi
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	Season
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	Seaso
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	Seaso
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	Season
...	...	...	...	...	...	...	...	...	...	.
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 mi
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	Season
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 mi
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 mi
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 mi

8807 rows × 12 columns

Basic Analysis

#1. Un-nesting the columns #a. Un-nest the columns those have cells with multiple comma separated values by #creating multiple rows

In [68]:

```
columns_to_unnest= ['cast','listed_in','country']
for col in columns_to_unnest:
    data[col] = data[col].str.split(', ')
    data = data.explode(col)
```

In [69]:
```
data
```

Out[69]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
1	s2	TV Show	Blood & Water	NaN	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
...	...	...	...	...	...	...	...	...	...	...
8806	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min

186325 rows × 12 columns

2. Handling null values

#a. For categorical variables with null values, update those rows as #unknown_column_name

```
In [70]: columns_to_update = ['director','cast','country']
         for col in columns_to_update:
             data[col] = data[col].fillna("unknown" + ' '+  col)
```

In [71]:
```
data
```

Out[71]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	unknown cast	United States	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	unknown director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
1	s2	TV Show	Blood & Water	unknown director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
1	s2	TV Show	Blood & Water	unknown director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
1	s2	TV Show	Blood & Water	unknown director	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
...	...	...	...	...	...	...	...	...	...	...
8806	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min

186325 rows × 12 columns

In [72]:

```
#b. Replace with 0 for continuous variables having null values
```

In [73]:

```
columns_to_update = ['rating','duration']
for col in columns_to_update:
    data[col] = data[col].fillna(0)
```

In [74]:
```
data
```

Out[74]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	unknown cast	United States	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	unknown director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
1	s2	TV Show	Blood & Water	unknown director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
1	s2	TV Show	Blood & Water	unknown director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
1	s2	TV Show	Blood & Water	unknown director	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
...	...	...	...	...	...	...	...	...	...	...
8806	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min

186325 rows × 12 columns

```
In [75]: data[data['rating']==0]
```

```
Out[75]:
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	dur
5989	s5990	Movie	13TH: A Conversation with Oprah Winfrey & Ava ...	unknown director	Oprah Winfrey	unknown country	January 26, 2017	2017	0	3
5989	s5990	Movie	13TH: A Conversation with Oprah Winfrey & Ava ...	unknown director	Ava DuVernay	unknown country	January 26, 2017	2017	0	3
6827	s6828	TV Show	Gargantia on the Verdurous Planet	unknown director	Kaito Ishikawa	Japan	December 1, 2016	2013	0	Se
6827	s6828	TV Show	Gargantia on the Verdurous Planet	unknown director	Kaito Ishikawa	Japan	December 1, 2016	2013	0	Se
6827	s6828	TV Show	Gargantia on the Verdurous Planet	unknown director	Hisako Kanemoto	Japan	December 1, 2016	2013	0	Se
...	...	...	...	...	...	...	...	...	...	
7537	s7538	Movie	My Honor Was Loyalty	Alessandro Pepe	Francesco Migliore	Italy	March 1, 2017	2015	0	11
7537	s7538	Movie	My Honor Was Loyalty	Alessandro Pepe	Albrecht Weimer	Italy	March 1, 2017	2015	0	11
7537	s7538	Movie	My Honor Was Loyalty	Alessandro Pepe	Giulia Dichiaro	Italy	March 1, 2017	2015	0	11
7537	s7538	Movie	My Honor Was Loyalty	Alessandro Pepe	Alessandra Oriti Niosi	Italy	March 1, 2017	2015	0	11
7537	s7538	Movie	My Honor Was Loyalty	Alessandro Pepe	Andreas Segeritz	Italy	March 1, 2017	2015	0	11

67 rows × 12 columns

What does ‘good’ look like?

1. Find the counts of each categorical variable both using graphical and non- graphical analysis.

a. For Non-graphical Analysis:

In [76]:

```
data.groupby('listed_in').nunique()['title'].sort_values(ascending = False)
```

Out[76]:

```
listed_in
International Movies            2752
Dramas                          2427
Comedies                        1674
International TV Shows          1351
Documentaries                    869
Action & Adventure               859
TV Dramas                        763
Independent Movies               756
Children & Family Movies         641
Romantic Movies                  616
TV Comedies                      581
Thrillers                        577
Crime TV Shows                   470
Kids' TV                         451
Docuseries                       395
Music & Musicals                 375
Romantic TV Shows                370
Horror Movies                    357
Stand-Up Comedy                  343
Reality TV                       255
British TV Shows                 253
Sci-Fi & Fantasy                 243
Sports Movies                    219
Anime Series                     176
Spanish-Language TV Shows        174
TV Action & Adventure            168
Korean TV Shows                  151
Classic Movies                   116
LGBTQ Movies                     102
TV Mysteries                      98
Science & Nature TV               92
TV Sci-Fi & Fantasy               84
TV Horror                         75
Anime Features                    71
Cult Movies                       71
Teen TV Shows                     69
Faith & Spirituality              65
TV Thrillers                      57
Movies                            57
Stand-Up Comedy & Talk Shows      56
Classic & Cult TV                 28
TV Shows                          16
Name: title, dtype: int64
```

Analysis: For each categorical variable (listed in and ratings) *My analysis is International Movies 2752 Dramas 2427 Comedies 1674 International TV Shows 1351 These categories for movies/tvshows is been watched more, These are the top 4 genre which people prefer to watch. but there is a very less demand or watchers who watch Classic & Cult TV 28 TV Shows 16.

In [77]:

```
#rating
```

In [78]:

```
data.groupby('rating').nunique()['title'].sort_values(ascending = False)
```

Out[78]:

```
rating
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
0              4
NC-17          3
UR             3
66 min         1
84 min         1
74 min         1
Name: title, dtype: int64
```

Top ratings which people has given are for these. TV-MA 3207 TV-14 2160 TV-PG 863 R 799

In [79]:

```
#b. For graphical analysis:
```

In [80]:

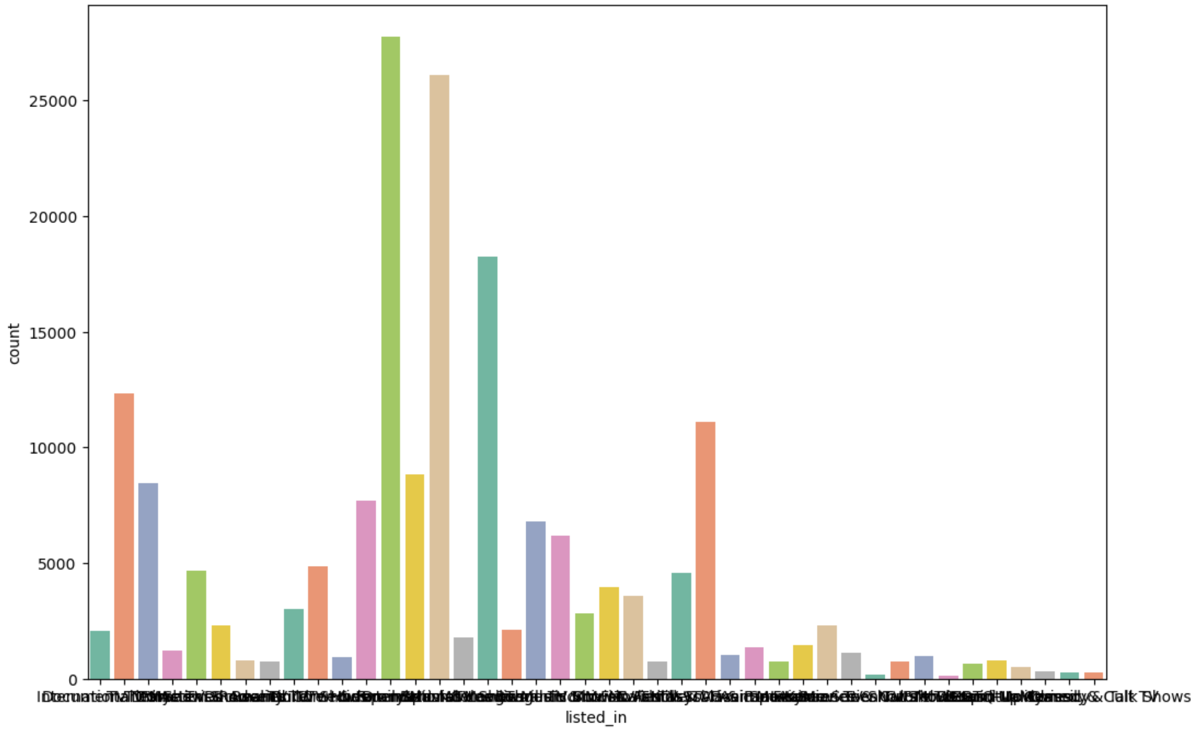
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [81]:

```
plt.figure(figsize=(12, 8))
sns.countplot(x='listed_in', data=data, palette='Set2')
```

Out[81]:

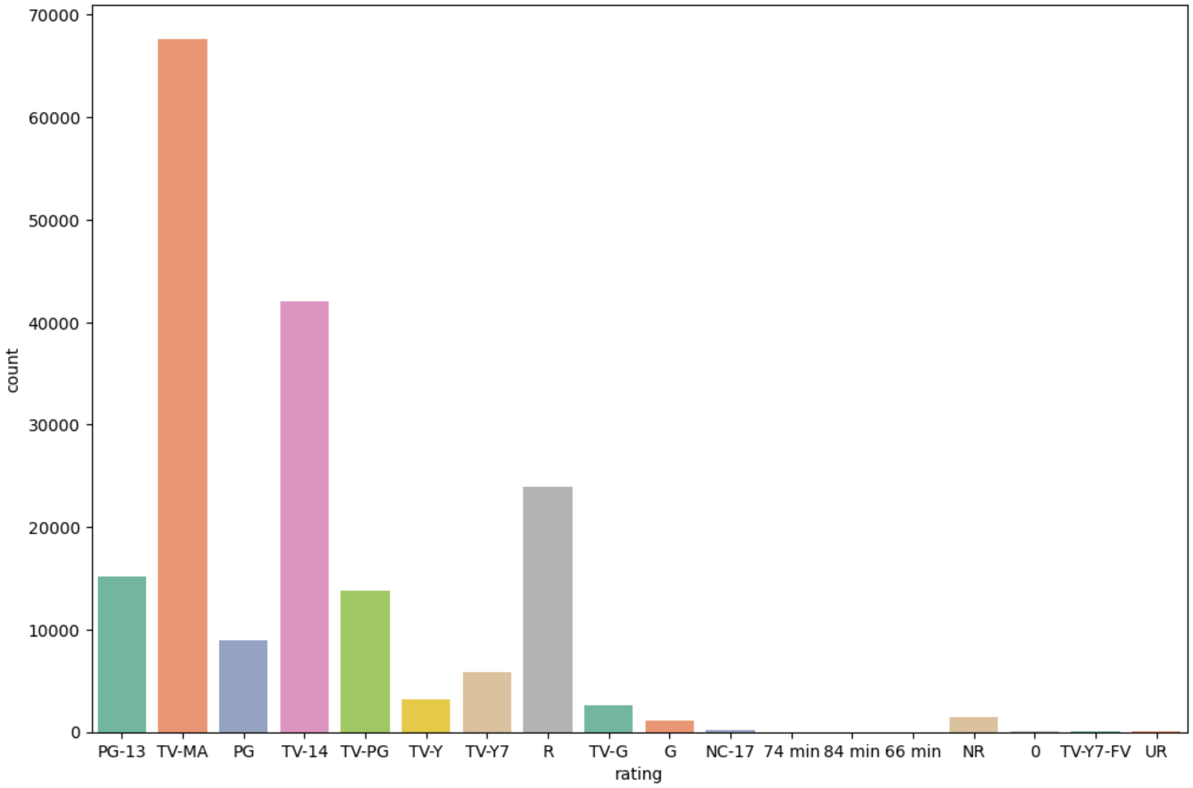
```
<Axes: xlabel='listed_in', ylabel='count'>
```



rating

In [82]:
```
plt.figure(figsize=(12, 8))
sns.countplot(x='rating', data=data, palette='Set2')
```

Out[82]:
```
<Axes: xlabel='rating', ylabel='count'>
```



2. Comparison of tv shows vs. movies.

a. Find the number of movies produced in each country and pick the top 10 countries.

In [83]:
```
movies = data[data['type'] == 'Movie']
numofmovies= movies.groupby('country').size().reset_index(name='Number_of_Movies')
```

In [84]:
```
numofmovies.sort_values(by='Number_of_Movies', ascending=False).head(10)
```

Out[84]:

	country	Number_of_Movies
114	United States	40811
43	India	20109
112	United Kingdom	8118
34	France	5872
122	unknown country	5708
20	Canada	5035
100	Spain	3250
36	Germany	3149
51	Japan	2803
75	Nigeria	2186

Analysis: The most people watch movies are from UNITED STATES AND INDIA. They have majority of movie watchers compartively from other countries.

In [85]:
```
#b. Find the number of Tv-Shows produced in each country and pick the top 10
#countries.
```

In [86]:
```
tv_shows = data[data['type']== 'TV Show']
```

In [87]:
```
numoftv_shows = tv_shows.groupby('country').size().reset_index(name='Number_of_Tv_show
```

In [88]:
```
numoftv_shows.sort_values(by='Number_of_Tv_shows', ascending=False).head(10)
```

Out[88]:

	country	Number_of_Tv_shows
63	United States	13408
66	unknown country	5437
30	Japan	5137
62	United Kingdom	4286
52	South Korea	3682
8	Canada	2133
38	Mexico	2018
53	Spain	1798
19	France	1542
57	Taiwan	1446

Analysis: Tv-shows - Majority of people from united states prefer watching Tv shows and followed by other countries.

Most watched duration for movies and tv-shows

In [89]:
```
tv_shows = data[data['type']== 'TV Show']
```

In [90]:
```
tv_shows['duration'].value_counts()
```

Out[90]:
```
1 Season      33444
2 Seasons      9470
3 Seasons      5084
4 Seasons      2134
5 Seasons      1698
7 Seasons       843
6 Seasons       633
8 Seasons       286
9 Seasons       257
10 Seasons      220
13 Seasons      132
12 Seasons      111
15 Seasons       96
17 Seasons       30
11 Seasons       30
Name: duration, dtype: int64
```

Analysis:

People prefer watching 1-2 seasons for a tv show and do not prefer no. of season in just 1 tv show. As the Number of seasons increases the watchers decreases, There might be multiple reasons such as it gets boring further or they loose interest etcc..

In [91]:
```
movies = data[data['type']== 'Movie']
```

In [92]:
```
movies['duration'].value_counts().head(10)
```

Out[92]:
```
94 min     3591
97 min     3434
93 min     3356
95 min     3192
106 min    3052
90 min     2948
102 min    2912
96 min     2911
105 min    2903
107 min    2886
Name: duration, dtype: int64
```

In [93]:
```
movies['duration'].value_counts().tail()
```

Out[93]:
```
5 min     3
9 min     2
3 min     2
11 min    2
8 min     1
Name: duration, dtype: int64
```

People are generally fine with watching around 1.5-1.7 hours of movies.it shouldn't be too short or too long in terms of duration.

3. What is the best time to launch a TV show?

a. Find which is the best week to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies

In [94]:

```
tv_shows = data[data['type']== 'TV Show']
```

In [95]:

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

```
C:\Users\asus\AppData\Local\Temp\ipykernel_14536\1784707428.py:1: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pand
as-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  tv_shows['date_added'] = pd.to_datetime(tv_shows['date_added'])
```

In [96]:

```
tv_shows['week'] = tv_shows['date_added'].dt.week
```

```
C:\Users\asus\AppData\Local\Temp\ipykernel_14536\2951684841.py:1: FutureWarning: Seri
es.dt.weekofyear and Series.dt.week have been deprecated. Please use Series.dt.isocal
endar().week instead.
  tv_shows['week'] = tv_shows['date_added'].dt.week
C:\Users\asus\AppData\Local\Temp\ipykernel_14536\2951684841.py:1: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pand
as-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  tv_shows['week'] = tv_shows['date_added'].dt.week
```

In [97]:

```
bestweek = tv_shows.groupby('week').size().reset_index(name='countof_Tvshows')
```

In [98]:

```
bestweek.sort_values(by='countof_Tvshows',ascending = True).head(5)
```

Out[98]:

	week	countof_Tvshows
15	16.0	549
42	43.0	564
27	28.0	586
2	3.0	590
5	6.0	611

In [99]:

```
bestweek.sort_values(by='countof_Tvshows',ascending = True).tail(5)
```

Out[99]:

	week	countof_Tvshows
25	26.0	1530
12	13.0	1554
23	24.0	1702
34	35.0	1942
26	27.0	1977

analysis: According to my analysis the week 16 is a week where most of the tv-shows are added but i feel the week 13, week 31, week 27 are the week where tv show should be added as least movies are added and if in that time movies are added they will be more watched because of less no. of traffic of movies . And in weeks where most ofthe no. of shows are added there will be too much of confusion and none of the shows will get the hype.

In [100]:

```
#Movies
```

In [101]:

```
movies = data[data['type']== 'Movie']
```

In [102]:

```
movies['date_added'] = pd.to_datetime(movies['date_added'])
movies['week'] = movies['date_added'].dt.week
```

```
C:\Users\asus\AppData\Local\Temp\ipykernel_14536\1550269020.py:1: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pand
as-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  movies['date_added'] = pd.to_datetime(movies['date_added'])
C:\Users\asus\AppData\Local\Temp\ipykernel_14536\1550269020.py:2: FutureWarning: Seri
es.dt.weekofyear and Series.dt.week have been deprecated. Please use Series.dt.isocal
endar().week instead.
  movies['week'] = movies['date_added'].dt.week
C:\Users\asus\AppData\Local\Temp\ipykernel_14536\1550269020.py:2: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pand
as-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  movies['week'] = movies['date_added'].dt.week
```

In [103]:

```
bestweek_movies = movies.groupby('week').size().reset_index(name='countof_movies')
```

In [104]:

```
bestweek_movies.sort_values(by = 'countof_movies', ascending = True).tail(5)
```

Out[104]:

	week	countof_movies
39	40	4573
34	35	4589
8	9	4671
43	44	5275
0	1	7838

analysis: Best time to launch or add is 40th, 1, 44th week .

In [105]:

```
#b. Find which is the best month to release the Tv-show or the movie. Do the
#analysis separately for Tv-shows and Movies
```

In [106]:

```
tv_shows = data[data['type']== 'TV Show']
```

In [107]:

```
tv_shows['date_added'] = pd.to_datetime(tv_shows['date_added'])
tv_shows['month'] = tv_shows['date_added'].dt.month
```

```
C:\Users\asus\AppData\Local\Temp\ipykernel_14536\3019805930.py:1: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pand
as-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  tv_shows['date_added'] = pd.to_datetime(tv_shows['date_added'])
C:\Users\asus\AppData\Local\Temp\ipykernel_14536\3019805930.py:2: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pand
as-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  tv_shows['month'] = tv_shows['date_added'].dt.month
```

In [108]:

```
bestweek_tvshows = tv_shows.groupby('month').size().reset_index(name='countof_tvshows'
```

In [109]:

```
bestweek_tvshows.sort_values(by = 'countof_tvshows', ascending = True).head(5)
```

Out[109]:

	month	countof_tvshows
1	2.0	3691
4	5.0	3886
9	10.0	4220
0	1.0	4296
2	3.0	4324

Analysis: The best month to upload will be between 2nd , 3rd, 10th month as there will be moderate traffic and people will be able to watch every movie that's been released. And would rate them accoridngly. In the peak months a lot of shows are added because of which shows are left out and are not watched because of multiple options.

4 Analysis of actors/directors of different types of shows/movies.

a. Identify the top 10 directors who have appeared in most movies or TV shows.

In [110]:
```
data.groupby('director').nunique()['title'].reset_index(name = 'count_of_director').so
```

Out[110]:

	director	count_of_director
4516	unknown director	2634
3392	Rajiv Chilaka	19
3443	Raúl Campos, Jan Suter	18
2598	Marcus Raboy	16
4046	Suhas Kadav	16
1790	Jay Karas	14
685	Cathy Garcia-Molina	13
2671	Martin Scorsese	12
1787	Jay Chapman	12
4480	Youssef Chahine	12

The top Directors are Rajiv Chilaka , Raul campos, Marcus, Raboy , Suhas kadav these are top directors which has the most no. of occurences

In [111]:
```
#Identify the top 10 actors who have appeared in most movies or TV shows.
```

In [112]:
```
data.groupby('cast').nunique()['title'].reset_index(name = 'count_of_actor').sort_valu
```

Out[112]:

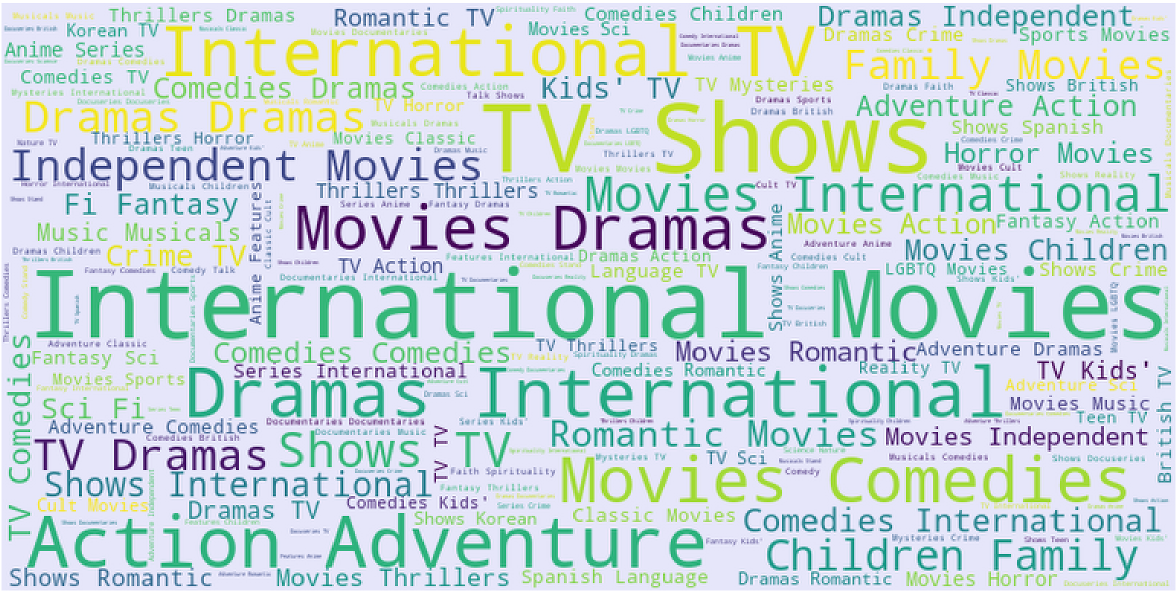
	cast	count_of_actor
36318	unknown cast	825
2833	Anupam Kher	43
30489	Shah Rukh Khan	35
16697	Julie Tejwani	33
24215	Naseeruddin Shah	32
32591	Takahiro Sakurai	32
28974	Rupa Bhimani	31
25424	Om Puri	30
846	Akshay Kumar	30
35880	Yuki Kaji	29

Analysis: There is an unknown actor , for the data is a NAN value has occured the most no. of times .Additionally Anupam kher , shahrukh khan , Julie tejwani, naseeruddin shah , takahiro and so on are the top famous cast which has been in most of the movies/ tv shows.T hey has been casted by a lot of famous directors major number of the times.

In [113]:
```
# Which genre movies are more popular or produced more
```

In [114]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

In [115]:
```
text = ' '.join(data['listed_in'])
wordcloud = WordCloud(width=800, height=400, background_color='lavender').generate(tex
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='lanczos')
plt.axis('off')
plt.show()
```



The top genre of people who prefer watching movies / tv shows are Comedies , International TV, Romantic movies, Action Adventure, Family movies, Dramas.

In [116]:
```
#Find After how many days the movie will be added to Netflix after the release of
#the movie (you can consider the recent past data)
```

In [117]:
```
datanew = pd.read_csv('netflix.csv')
```

In [118]:
```
datanew['date_added'] = pd.to_datetime(datanew['date_added'])
```

In [119]:
```
datanew['year']= datanew['date_added'].dt.year
```

In [120]:
```
datanew['year']= datanew['year'].fillna(datanew['release_year']).astype(int)
```

In [121]:
```
diffofdates= datanew['year'] - datanew['release_year']
```

```
In [122]: diffofdates.reset_index(drop = True, name='diff')
```

```
Out[122]: 0        1
          1        0
          2        0
          3        0
          4        0
                  ..
          8802    12
          8803     1
          8804    10
          8805    14
          8806     4
          Length: 8807, dtype: int64
```

```
In [123]: diffofdates.value_counts()
```

```
Out[123]:  0     3251
           1     1585
           2      714
           3      491
           4      367
                 ...
          -2        1
           93       1
           60       1
           70       1
           63       1
          Length: 75, dtype: int64
```

Majority of movies/ tv shows are added and released in the same year itself . And few of them are added after an year or two. People also prefer watching a freshly made up movies because to intact the excitement to watch it.

Understanding what content is available in different countries

```
In [124]:
          data.groupby(by = ['country','listed_in']).count()['title']
```

```
Out[124]: country          listed_in
                           Dramas                      8
                           Independent Movies          8
                           International Movies        8
                           International TV Shows      4
                           TV Dramas                   4
                                                     ...
          unknown country  TV Sci-Fi & Fantasy        61
                           TV Shows                   43
                           TV Thrillers               27
                           Teen TV Shows              21
                           Thrillers                 235
          Name: title, Length: 1464, dtype: int64
```

My analysis is that majority of countries prefer watching international movies. Australia prefers bristish shows , anime series, action and adventure . Argentina prefers family and childrens. Austria prefers crime tv shows and documentaries. bahamas only watch action and adventure. Cambodia prefers Dramas and International Movies Cameroon prefers Dramas and International Movies Cuba prefers Crime TV Shows, International TV Shows and Spanish-Language TV Shows Cyprus prefers Kids' TV and TV Comedies Dominican Republic prefer Horror Movies , International Movies ,Thrillers East Germany prefers Children

In [ ]: