```
!pip install matplotlib
```

```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv -O netflix.csv
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
To: /content/netflix.csv
100% 3.40M/3.40M [00:00<00:00, 4.62MB/s]
```

```
#importing libraries
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import copy
```

```
#reading/loading the dataset netflix.csv
data = pd.read_csv('netflix.csv')
```

```
data.head(3)
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	year_added	month_added	wee
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown cast	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	2021.0	9.0	
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...	2021.0	9.0	
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...	2021.0	9.0	

```
data.shape
```

```
(8807, 12)
```

```
data.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
```

```
data.describe(include = 'object')
```

	show_id	type	title	director	cast	country	date_added	rating	duration	listed_in	descri
count	8807	8807	8807	6173	7982	7976	8797	8803	8804	8807	
unique	8807	2	8807	4528	7692	748	1767	17	220	514	
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 abandoned p
freq	1	6131	1	19	19	2818	109	3207	1793	362	

```
data.duplicated().value_counts()
```

```
False    8807
Name: count, dtype: int64
```

Basic Analysis

1. *Un-nesting the columns*

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

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

data.head(3)
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	year_added	month_added	wee
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown cast	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	2021.0	9.0	
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...	2021.0	9.0	
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...	2021.0	9.0	

2. Handling null values

   a. For categorical variables with null values, update those rows as unknown_column_name. Example : Replace missing value with Unknown Actor for missing value in Actors column.

```
data['director'].fillna('Unknown director',inplace = True)
data['cast'].fillna('Unknown cast',inplace = True)
data['country'].fillna('Unknown country',inplace = True)
data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	de
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown cast	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	A
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	pa
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy	Unknown country	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act	To fa

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

```
# checking the value counts for columns
for i in ['rating','duration']:
 print('Value count in',i,'column are :-')
 print(data[i].value_counts())
 print('-'*70)

#replace unknown values in ratings columns to nan
data['rating'].replace({'74 min' : np.nan, '84 min' : np.nan, '66 min' : np.nan, '0' : np.nan}, inplace = True)

#Fill nan values to unknown rating
data['rating'].fillna('Unknown rating',inplace = True)
data['rating'].value_counts()
```

```
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
Unknown rating       7
TV-Y7-FV             6
NC-17                3
UR                   3
Name: count, dtype: int64
```

```
data['duration'].fillna(0,inplace = True)
data.head(3)
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descr
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown cast	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As he ne en life,
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows	c pa party, T
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	TV Dramas	c pa party, T

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

```
a. For Non-graphical Analysis:
```

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

```
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 : Upon checking the above data, we can see that there are top 4 categories listed in; International Movies(2752), Dramas (2427), Comedies(1674), International TV Shows(1351) and least watched categories are; Classic & Cult TV(28), TV Shows(16).

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

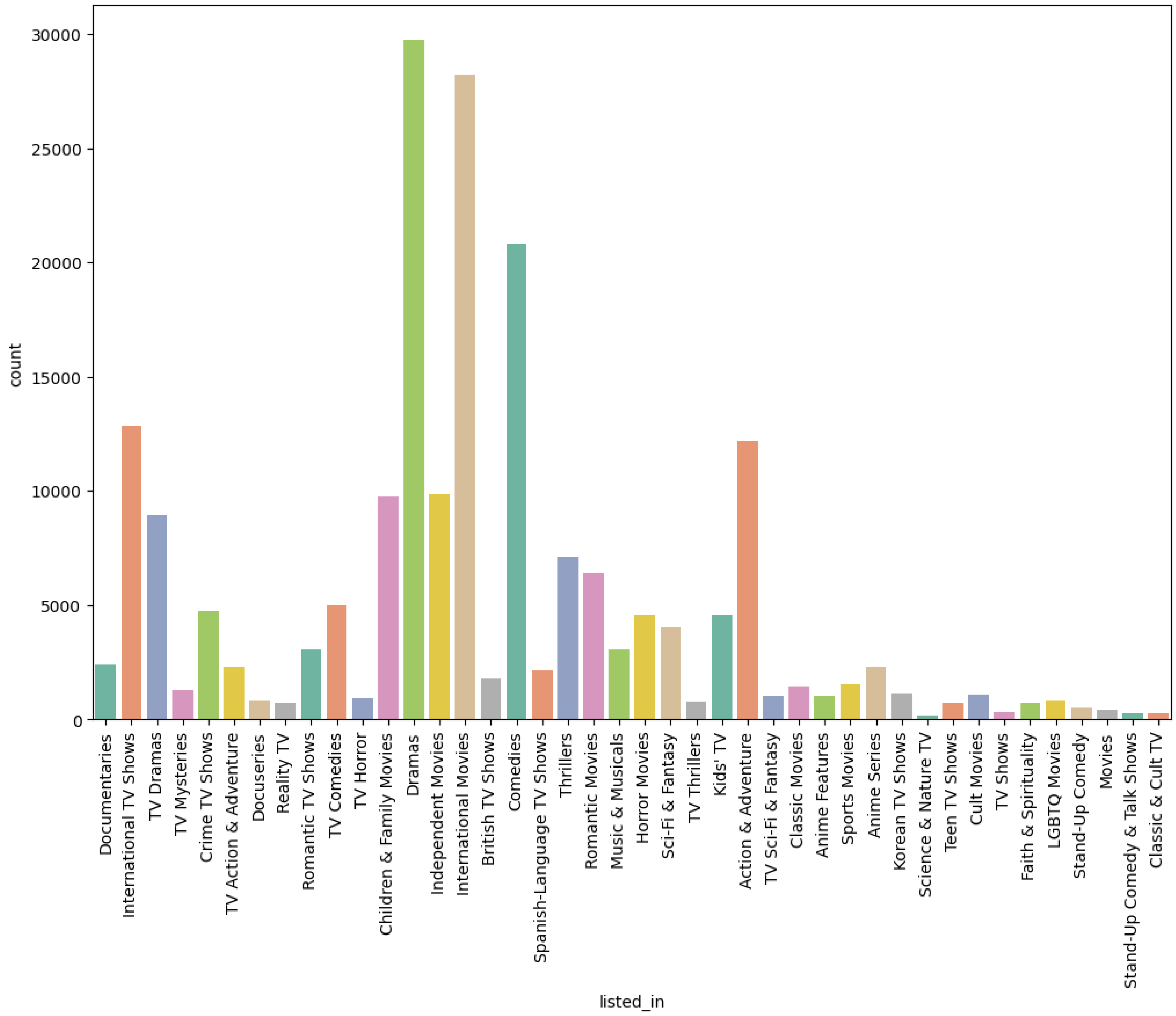
```
rating
TV-MA             3207
TV-14             2160
TV-PG              863
R                  799
PG-13              490
TV-Y7              334
TV-Y               307
PG                 287
TV-G               220
NR                  80
G                   41
Unknown rating       7
TV-Y7-FV             6
NC-17                3
UR                   3
Name: title, dtype: int64
```

Analysis: Top ratings which people has given are: TV-MA(3207), TV-14(2160), TV-PG(863), R(799).

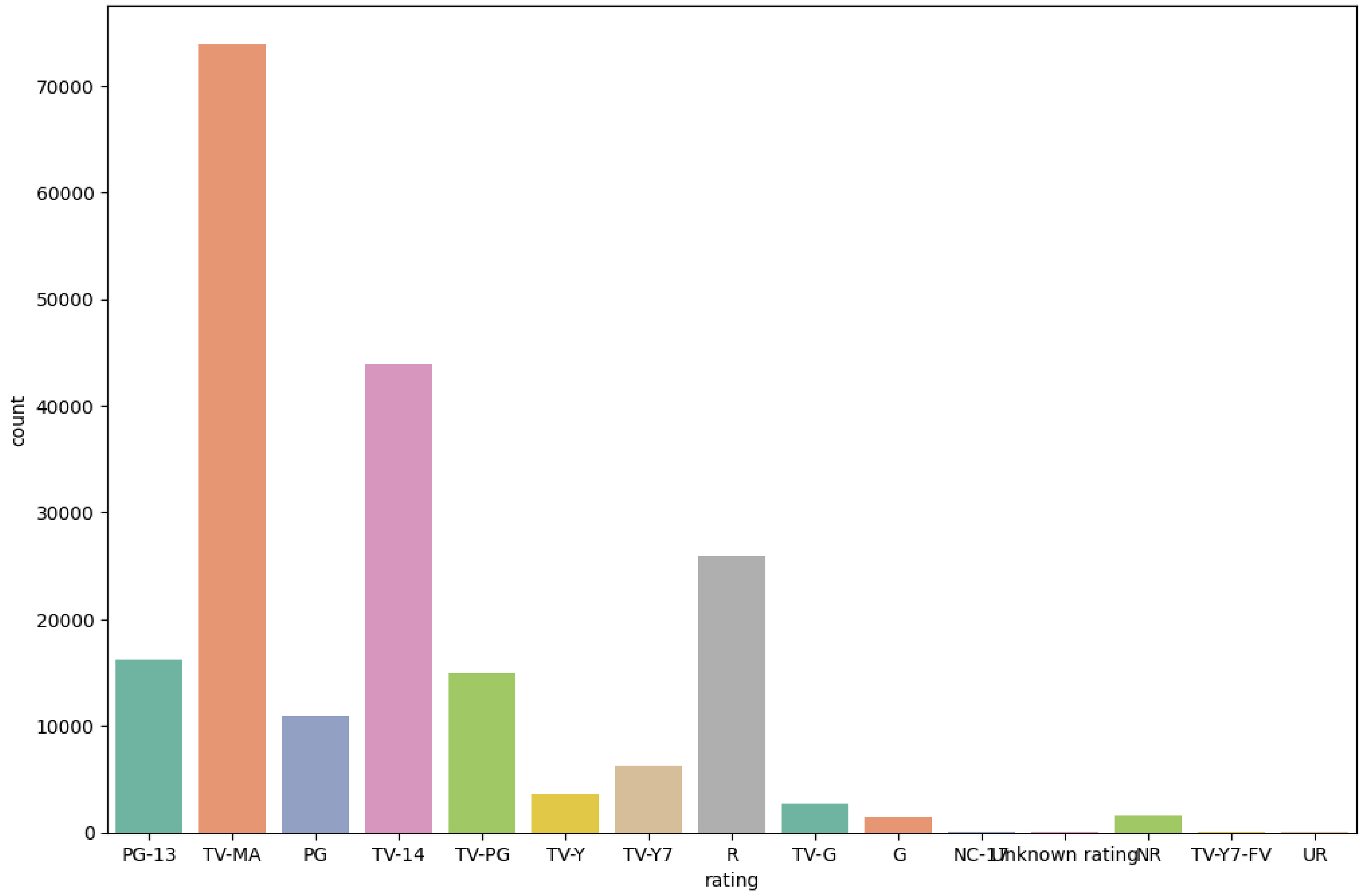
```
b. For graphical analysis:
```

```
import warnings
warnings.filterwarnings("ignore")

#listed_in
plt.figure(figsize=(12, 8))
sns.countplot(x='listed_in', data=data, palette='Set2')
plt.xticks(rotation = 90)
plt.show()
```



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



2. Comparison of tv shows vs. movies.

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

```
#Movies
movies = data[data['type'] == 'Movie']
numberofmovies = movies.groupby('country').size().reset_index(name = 'Number_of_Movies')
numberofmovies.sort_values(by='Number_of_Movies', ascending=False).head(10)
```

	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: Most of the people watch movies are from UNITED STATES(40811) AND INDIA(20109). They have majority of movie watchers compartively from other countries.

```
b. Find the number of Tv-Shows produced in each country and pick the top 10 countries.
```

```
#TV shows
tv_shows = data[data['type']== 'TV Show']
numberoftv_shows = tv_shows.groupby('country').size().reset_index(name='Number_of_Tv_shows')
numberoftv_shows.sort_values(by='Number_of_Tv_shows', ascending=False).head(10)
```

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

Most watched duration for movies and tv-shows

```
#Tv shows
tv_shows = data[data['type'] == 'TV Show']
tv_shows['duration'].value_counts()
```

```
duration
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: count, dtype: int64
```

Analysis: People would prefer to watch 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.

```
#Movies
movies = data[data['type'] == 'Movie']
movies['duration'].value_counts().head(10)
```

```
duration
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: count, dtype: int64
```

```
movies['duration'].value_counts().tail()
```

```
duration
5 min     3
9 min     2
3 min     2
11 min    2
8 min     1
Name: count, dtype: int64
```

Analysis: 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
```

```
#converting date_added col to date time format and creating three new columns; Year, Month and Week
data['date_added'] = pd.to_datetime(data['date_added'],errors='coerce')
data['year_added'] = data['date_added'].dt.year
data['month_added'] = data['date_added'].dt.month
data['week_added'] = data['date_added'].dt.isocalendar().week
```

```
#TV Shows
tv_shows = data.loc[data['type']== 'TV Show']
tv_show_counts = tv_shows['week_added'].value_counts().reset_index()
tv_show_counts.columns = ['week_added', 'Number_of_TV_Shows']
best_tv_show_week = tv_show_counts.loc[tv_show_counts['Number_of_TV_Shows'].idxmax()]
print(best_tv_show_week)
```

```
week_added              27
Number_of_TV_Shows    1945
Name: 0, dtype: Int64
```

Analysis : According to the above analysis, the best week to release a TV show is 'week 27'.

```
#Movies
movies = data.loc[data['type']== 'Movie']
movies_counts = movies['week_added'].value_counts().reset_index()
movies_counts.columns = ['week_added', 'Number_of_movies']
best_movies_week = movies_counts.loc[movies_counts['Number_of_movies'].idxmax()]
print(best_movies_week)
```

```
week_added            1
Number_of_movies    8456
Name: 0, dtype: Int64
```

Analysis : According to the data above, the best week to release a movie is 'week 01'.

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

```
#TV Shows
tv_shows = data.loc[data['type']== 'TV Show']
tv_show_counts = tv_shows['month_added'].value_counts().reset_index()
tv_show_counts.columns = ['month_added', 'Number_of_TV_Shows']
best_tv_show_month = tv_show_counts.loc[tv_show_counts['Number_of_TV_Shows'].idxmax()]
print(best_tv_show_month)
```

```
month_added             12.0
Number_of_TV_Shows    5341.0
Name: 0, dtype: float64
```

Analysis : The best month to release a TV show would be last month as there will be a moderate traffic.

```
#Movies
movies = data.loc[data['type']== 'Movie']
movies_counts = movies['month_added'].value_counts().reset_index()
movies_counts.columns = ['month_added', 'Number_of_movies']
best_movies_month = movies_counts.loc[movies_counts['Number_of_movies'].idxmax()]
print(best_movies_month)
```

```
month_added             7.0
Number_of_movies    15049.0
Name: 0, dtype: float64
```

Analysis : The best month to release a TV show would be 7th month as there will be a moderate traffic.

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.
```

```
data.groupby('director')['title'].nunique().sort_values(ascending = False)[0:10].reset_index(name = 'count_of_director')
```

	director	count_of_director
0	Unknown director	2634
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Marcus Raboy	16
5	Suhas Kadav	16
6	Jay Karas	15
7	Cathy Garcia-Molina	13
8	Jay Chapman	12
9	Martin Scorsese	12

Analysis : From above data we could identify the top 10 directors who have appeared in most movies or TV shows. Rajiv Chilaka(22), Jan Suter(21), Raul Campos(19) are the top 3.

```
b. Identify the top 10 Actors who have appeared in most movies or TV shows.
```

```
data.groupby('cast')['title'].nunique().sort_values(ascending = False)[0:10].reset_index(name = 'Count_of_Actors')
```

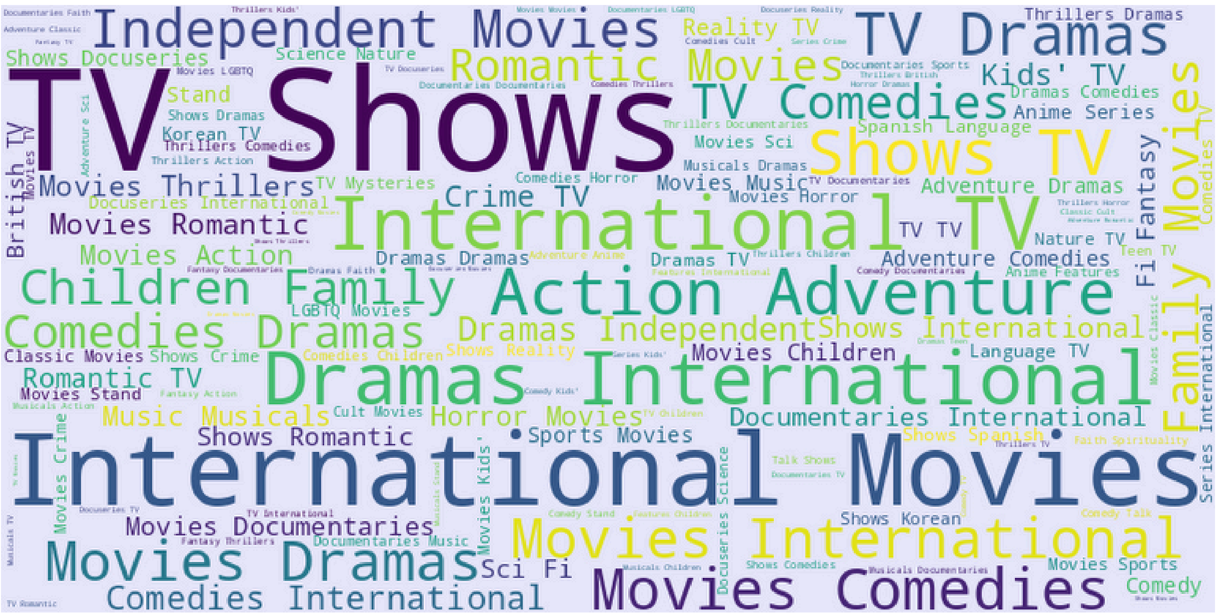
	cast	Count_of_Actors
0	Unknown cast	825
1	Anupam Kher	43
2	Shah Rukh Khan	35
3	Julie Tejwani	33
4	Naseeruddin Shah	32
5	Takahiro Sakurai	32
6	Rupa Bhimani	31
7	Om Puri	30
8	Akshay Kumar	30
9	Yuki Kaji	29

Analysis : From above data we could identify the top 10 actors who have appeared in most movies or TV shows. Anupam Kher(43), Shah Rukh Khan(35), Julie Tejwani(33) are the top 3.

5. Which genre movies are more popular or produced more

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

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



Analysis: The best genre of movies are Comedies, International movies, International TV, Romantic movies, Action Adventure, Family movies, Dramas.

Note: The plot is in the text editor saved as an image but this code can be run.

6. 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).

```
import pandas as pd

# Read the CSV file
data = pd.read_csv('netflix.csv')

# Strip leading and trailing spaces from the 'date_added' column
data['date_added'] = data['date_added'].str.strip()

# Convert 'date_added' column to datetime format
data['date_added'] = pd.to_datetime(data['date_added'], format='%B %d, %Y')

# Extract the year from 'date_added'
data['year'] = data['date_added'].dt.year

# Calculate the delay in years between 'date_added' and 'release_year'
data['delay'] = data['year'] - data['release_year']

data['delay']
```

```
0        1.0
1        0.0
2        0.0
3        0.0
4        0.0
        ...
8802    12.0
8803     1.0
8804    10.0
8805    14.0
8806     4.0
Name: delay, Length: 8807, dtype: float64
```

```
# Find the mode of the delay
mode_delay = data['delay'].mode()
print(mode_delay.values[0])
```

```
0.0
```

Analysis : Majority of movies/ tv shows are added and released in the same year itself.

Understanding what content is available in different countries.

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

```
country       listed_in
              Dramas                     8
              Independent Movies         8
              International Movies       8
              International TV Shows     4
              TV Dramas                  4
                                        ..
West Germany  Thrillers                 10
Zimbabwe      Comedies                  12
              Documentaries              3
              International Movies      15
              Romantic Movies           12
Name: title, Length: 1464, dtype: int64
```