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

path = '/content/netflix_titles.csv'
df = pd.read_csv(path)
```

```
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rat
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	P
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV

```
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
```

Basic Analysis

- Handling null values

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

categorical_columns = df.select_dtypes(include=['object', 'category']).columns
for col in categorical_columns:
    df[col].fillna('Unknown ' + col, inplace=True)

continuous_columns = df.select_dtypes(include=['int64', 'float64','datetime64']).columns
for col in continuous_columns:
    df[col].fillna(0, inplace=True)

df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	liste
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown cast	United States	2021-09-25 00:00:00	2020	PG-13	90 min	Documen
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24 00:00:00	2021	TV-MA	2 Seasons	Interna TV Show Drama Mys
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown country	2021-09-24 00:00:00	2021	TV-MA	1 Season	Crin Sl Interna TV Show
3	s4	TV Show	Jailbirds New Orleans	Unknown director	Unknown cast	Unknown country	2021-09-24 00:00:00	2021	TV-MA	1 Season	Docus Reali
4	s5	TV Show	Kota Factory	Unknown director	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24 00:00:00	2021	TV-MA	2 Seasons	Interna TV Sl Romant Shows,

```
missing_count = df.isnull().sum()
missing_count
```

```
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
```

Basic Analysis

- Un-nesting the columns

```
df = df.assign(cast=df['cast'].str.split(',')).explode('cast').reset_index(drop=True)
df = df.assign(listed_in=df['listed_in'].str.split(',')).explode('listed_in').reset_index(drop=True)
```

```
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_i
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown cast	United States	2021-09-25 00:00:00	2020	PG-13	90 min	Documentarie
1	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	2021-09-24 00:00:00	2021	TV-MA	2 Seasons	Internationa TV Show
2	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	2021-09-24 00:00:00	2021	TV-MA	2 Seasons	TV Drama
3	s2	TV Show	Blood & Water	Unknown director	Ama Qamata	South Africa	2021-09-24 00:00:00	2021	TV-MA	2 Seasons	TV Mysterie
4	s2	TV Show	Blood & Water	Unknown director	Khosi Ngema	South Africa	2021-09-24 00:00:00	2021	TV-MA	2 Seasons	Internationa TV Show

Basic Analysis

- Removing Duplicates rows

```
df = df[~df.duplicated(keep='first')]
duplicate_rows = df[df.duplicated()]
duplicate_rows
```

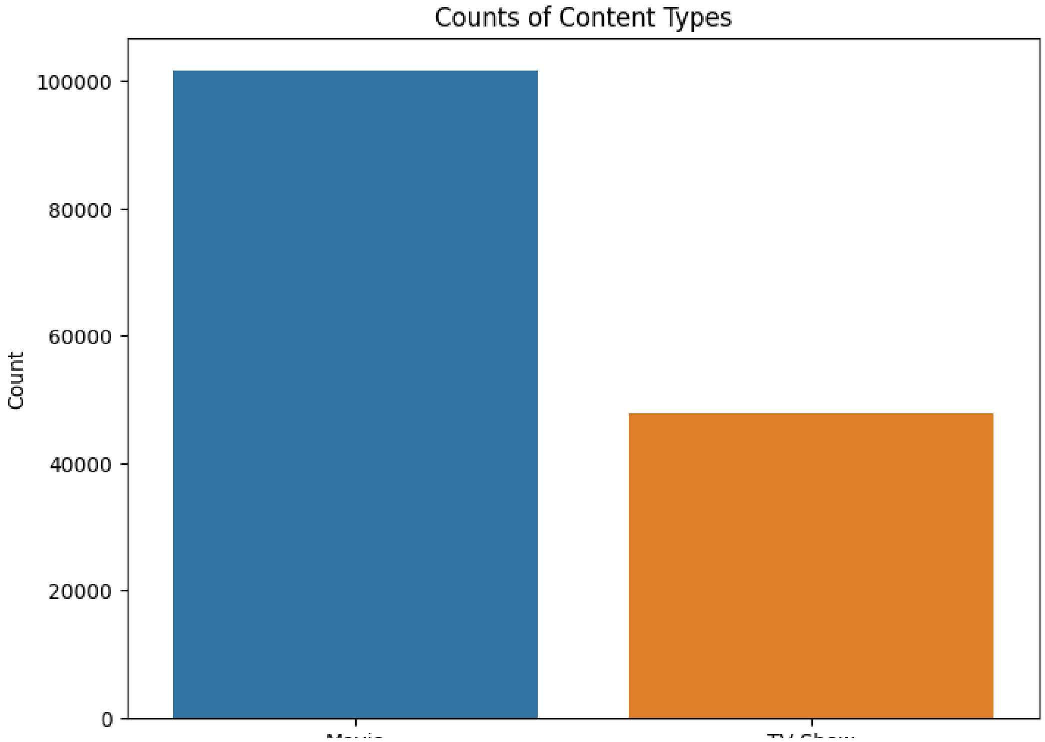
show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descript

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

```
type_counts = df['type'].value_counts()
type_counts
```

```
Movie      101689
TV Show     47820
Name: type, dtype: int64
```

```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='type')
plt.title('Counts of Content Types')
plt.xlabel('Content Type')
plt.ylabel('Count')
plt.show()
```

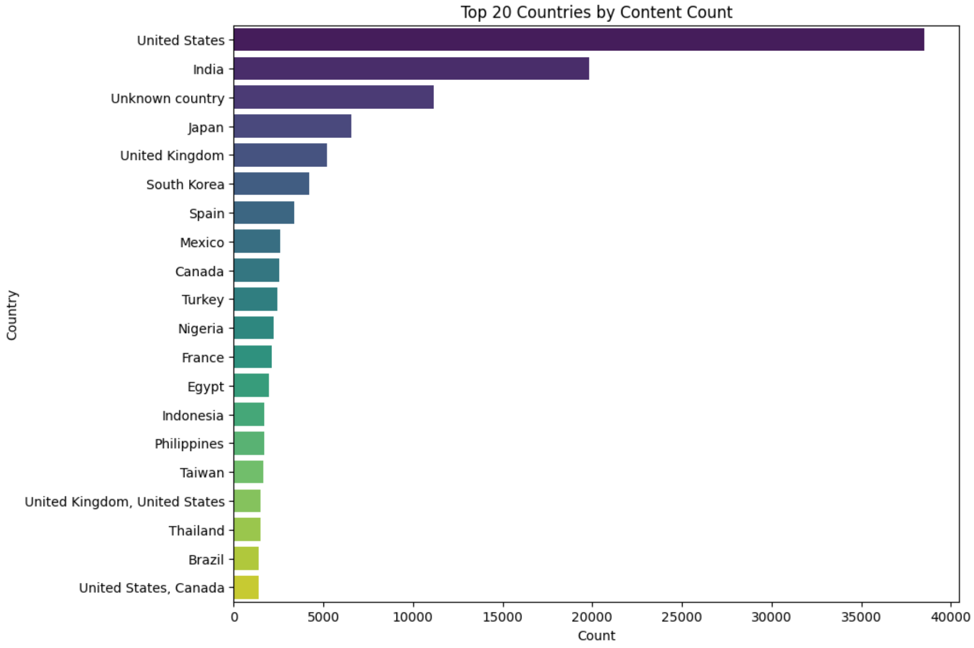


It seems that the majority of the content are movies. TV shows make up a smaller portion. This distribution suggests that there are significantly more movies available compared to TV shows.

```
country_counts = df['country'].value_counts()
country_counts
```

```
United States                                38550
India                                        19816
Unknown country                              11145
Japan                                         6584
United Kingdom                                5180
                                             ...
Germany, United States, Sweden                   1
United States, Botswana                          1
United States, Brazil, Japan, Spain, India       1
United States, Uruguay                           1
France, New Zealand                              1
Name: country, Length: 749, dtype: int64
```

```
top_countries = df['country'].value_counts().nlargest(20).index

df_top_countries = df[df['country'].isin(top_countries)]

plt.figure(figsize=(10, 8))
sns.countplot(data=df_top_countries, y='country', order=top_countries, palette='viridis')
plt.title('Top 20 Countries by Content Count')
plt.xlabel('Count')
plt.ylabel('Country')
plt.show()
```



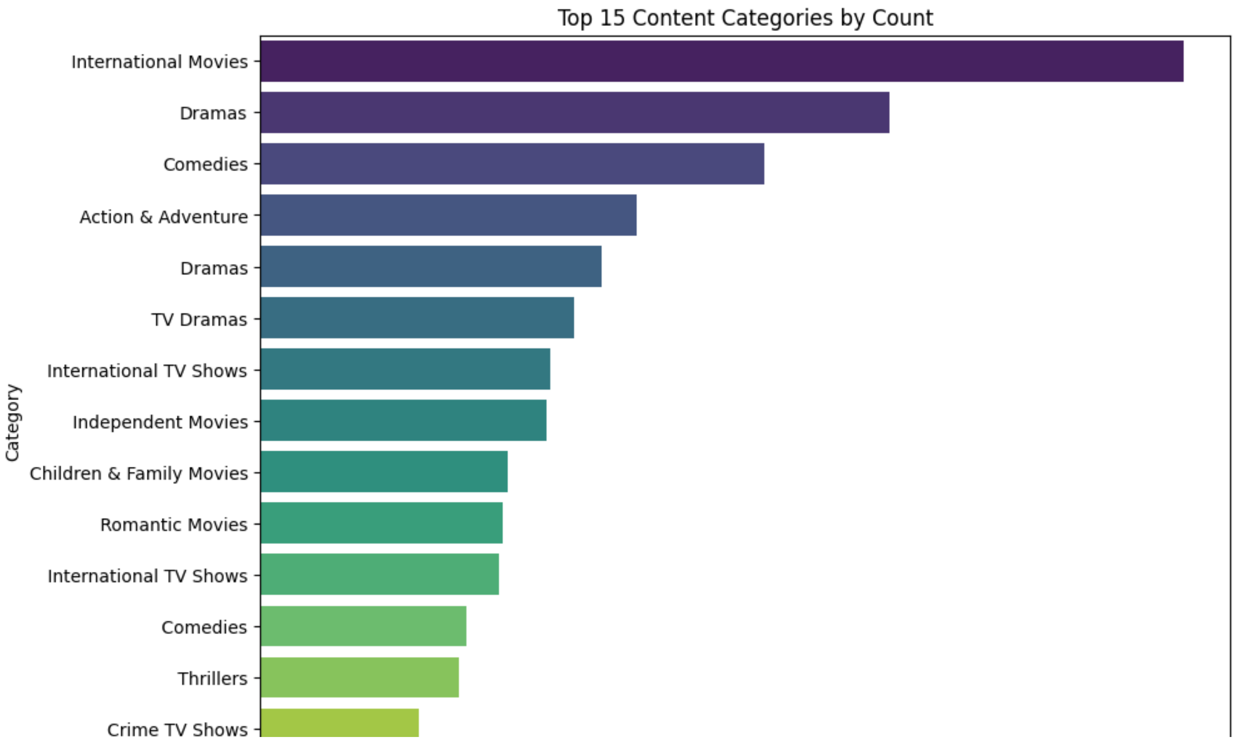
From the above data

- Maximum content is form US followed by India
- Main focus should be on the top countries
- Also we have Large amount of data in the sata set having countries as `Unknown` or `None`

```
listed_in_counts = df['listed_in'].value_counts()
listed_in_counts
```

```
 International Movies    19762
Dramas                   13466
Comedies                 10789
Action & Adventure        8060
 Dramas                   7311
                          ... 
Romantic Movies             20
 Stand-Up Comedy            18
TV Sci-Fi & Fantasy          7
LGBTQ Movies                 5
Sports Movies                3
Name: listed_in, Length: 73, dtype: int64
```

```
top_categories = df['listed_in'].value_counts().nlargest(15).index

df_top_categories = df[df['listed_in'].isin(top_categories)]

plt.figure(figsize=(10, 8))
sns.countplot(data=df_top_categories, y='listed_in', order=top_categories, palette='viridis')
plt.title('Top 15 Content Categories by Count')
plt.xlabel('Count')
plt.ylabel('Category')
plt.show()
```



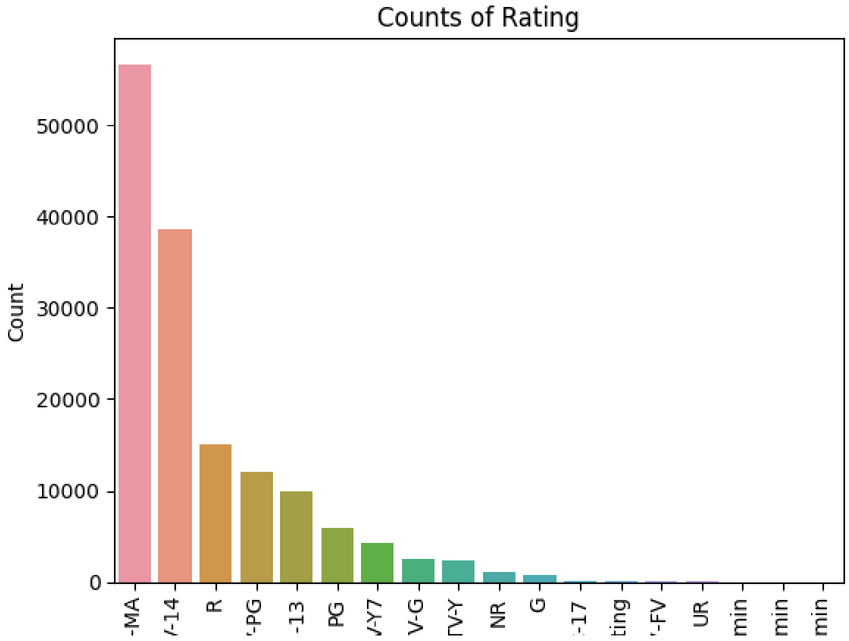
From the above data

- Maximum content is for International Movies followed by Drama then Comedies
- Least content is for categores TV Sci-Fi & Fantasy, LGBTQ Movies, Sports Movies,

```
rating_counts = df['rating'].value_counts()
rating_counts
```

```
     TV-MA              56695
     TV-14              38642
     R                  15151
     TV-PG              11944
     PG-13               9860
     PG                  5955
     TV-Y7               4287
     TV-G                2435
     TV-Y                2407
     NR                  1133
     G                    728
     NC-17                 71
     Unknown rating        67
     TV-Y7-FV              66
     UR                    65
     74 min                 1
     84 min                 1
     66 min                 1
     Name: rating, dtype: int64
```

```
order = df['rating'].value_counts().index
sns.countplot(data=df, x='rating',order=order)
plt.title('Counts of Rating')
plt.xlabel('Ratings')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



2.Comparison of tv shows vs. movies.

1. Find the number of movies produced in each country and pick the top 10 countries.
2. Find the number of Tv-Shows produced in each country and pick the top 10 countries.

```
df_no_unkown_country = df[df['country'] != 'Unknown country']
```

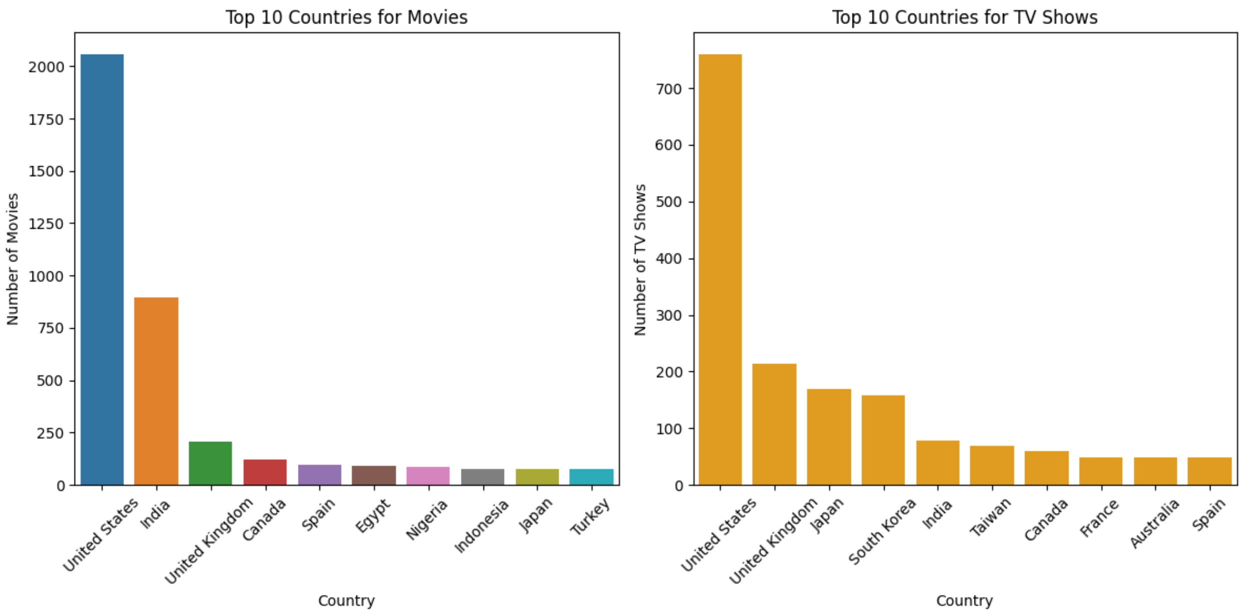
```
movie_countries_count = df_no_unkown_country[df_no_unkown_country['type'] == 'Movie'].groupby('country')['title'].nunique(
top_movie_countries = movie_countries_count.nlargest(10)
top_movie_countries
```

```
     country
     United States     2058
     India              893
     United Kingdom     206
     Canada             122
     Spain               97
     Egypt               92
     Nigeria             86
     Indonesia           77
     Japan               76
     Turkey              76
     Name: title, dtype: int64
```

```
tv_show_countries_count = df_no_unkown_country[df_no_unkown_country['type'] == 'TV Show'].groupby('country')['title'].nunique()
top_tv_show_countries = tv_show_countries_count.nlargest(10)
top_tv_show_countries
```

```
     country
     United States     760
     United Kingdom    213
     Japan             169
     South Korea       158
     India              79
     Taiwan             68
     Canada             59
     France             49
     Australia          48
     Spain              48
     Name: title, dtype: int64
```

```
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.barplot(x=top_movie_countries.index, y=top_movie_countries.values)
plt.title('Top 10 Countries for Movies')
plt.xlabel('Country')
plt.ylabel('Number of Movies')
plt.xticks(rotation=45)

plt.subplot(1, 2, 2)
sns.barplot(x=top_tv_show_countries.index, y=top_tv_show_countries.values, color='orange')
plt.title('Top 10 Countries for TV Shows')
plt.xlabel('Country')
plt.ylabel('Number of TV Shows')
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



From the above data

1. US have the max numbers of moves and TV shows
2. India have 2nd highest number of movies content on netflix
3. UK have 2nd highest TV shows content on Netflix

What is the best time to launch a TV show?

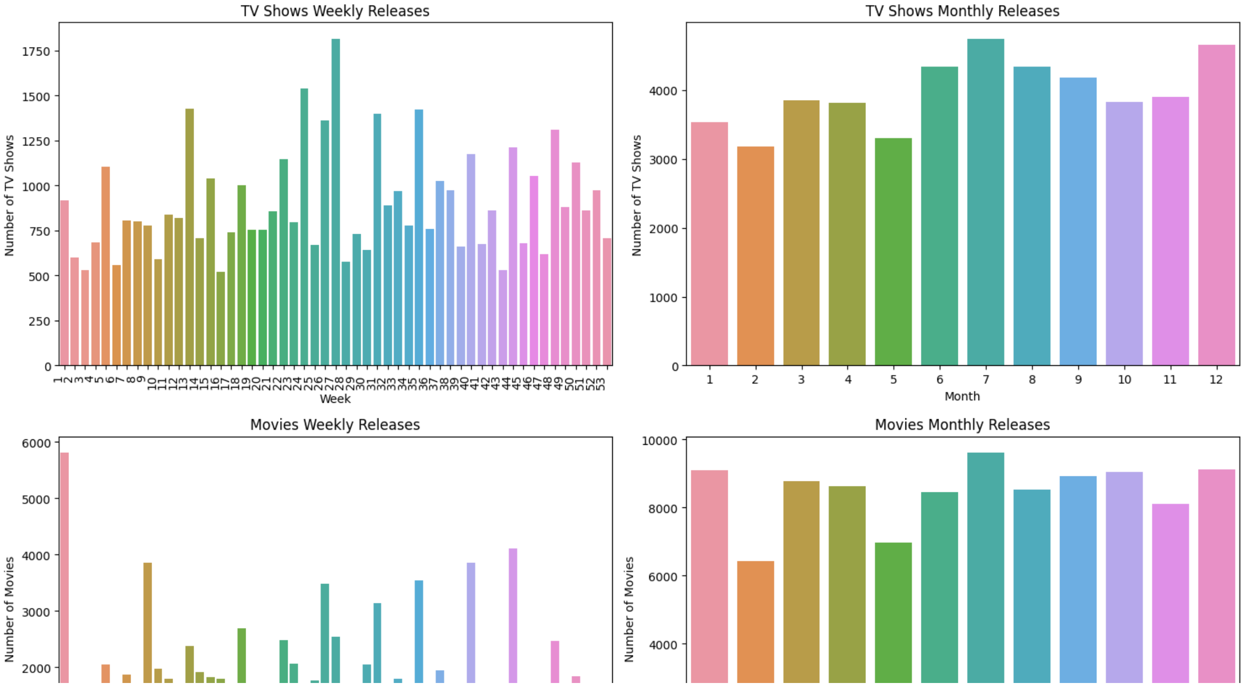
1. Find which is the best week to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies
2. Find which is the best month to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies

```
drop_zero_date_added = df[df['date_added'] != 0]
drop_zero_date_added['date_added'] = pd.to_datetime(drop_zero_date_added['date_added'])

drop_zero_date_added['week'] = drop_zero_date_added['date_added'].dt.week
drop_zero_date_added['month'] = drop_zero_date_added['date_added'].dt.month

tv_shows = drop_zero_date_added[drop_zero_date_added['type'] == 'TV Show']
movies = drop_zero_date_added[drop_zero_date_added['type'] == 'Movie']
```

```
tv_shows_weekly = tv_shows.groupby('week').size()
tv_shows_monthly = tv_shows.groupby('month').size()

movies_weekly = movies.groupby('week').size()
movies_monthly = movies.groupby('month').size()

#best week and month for TV shows
best_tv_week = tv_shows_weekly.idxmax()
best_tv_month = tv_shows_monthly.idxmax()

#best week and month for Movies
best_movie_week = movies_weekly.idxmax()
best_movie_month = movies_monthly.idxmax()

print("Best Week to Release TV Shows:", best_tv_week)
print("Best Month to Release TV Shows:", best_tv_month)
print("Best Week to Release Movies:", best_movie_week)
print("Best Month to Release Movies:", best_movie_month)
```

```
Best Week to Release TV Shows: 27
Best Month to Release TV Shows: 7
Best Week to Release Movies: 1
Best Month to Release Movies: 7
```

```
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
plt.subplots_adjust(hspace=0.4)

# TV Shows Weekly Releases
sns.barplot(x=tv_shows_weekly.index, y=tv_shows_weekly.values, ax=axes[0, 0])
axes[0, 0].set_title('TV Shows Weekly Releases')
axes[0, 0].set_xlabel('Week')
axes[0, 0].set_ylabel('Number of TV Shows')
axes[0, 0].set_xticklabels(axes[0, 0].get_xticklabels(), rotation=90, ha="right")

# TV Shows Monthly Releases
sns.barplot(x=tv_shows_monthly.index, y=tv_shows_monthly.values, ax=axes[0, 1])
axes[0, 1].set_title('TV Shows Monthly Releases')
axes[0, 1].set_xlabel('Month')
axes[0, 1].set_ylabel('Number of TV Shows')

# Movies Weekly Releases
sns.barplot(x=movies_weekly.index, y=movies_weekly.values, ax=axes[1, 0])
axes[1, 0].set_title('Movies Weekly Releases')
axes[1, 0].set_xlabel('Week')
axes[1, 0].set_ylabel('Number of Movies')
axes[1, 0].set_xticklabels(axes[1, 0].get_xticklabels(), rotation=90, ha="right")

# Movies Monthly Releases
sns.barplot(x=movies_monthly.index, y=movies_monthly.values, ax=axes[1, 1])
axes[1, 1].set_title('Movies Monthly Releases')
axes[1, 1].set_xlabel('Month')
axes[1, 1].set_ylabel('Number of Movies')

plt.tight_layout()
plt.show()
```



As per the analysis

- Best Week to Release TV Shows: 27
- Best Month to Release TV Shows: 7
- Best Week to Release Movies: 1
- Best Month to Release Movies: 7

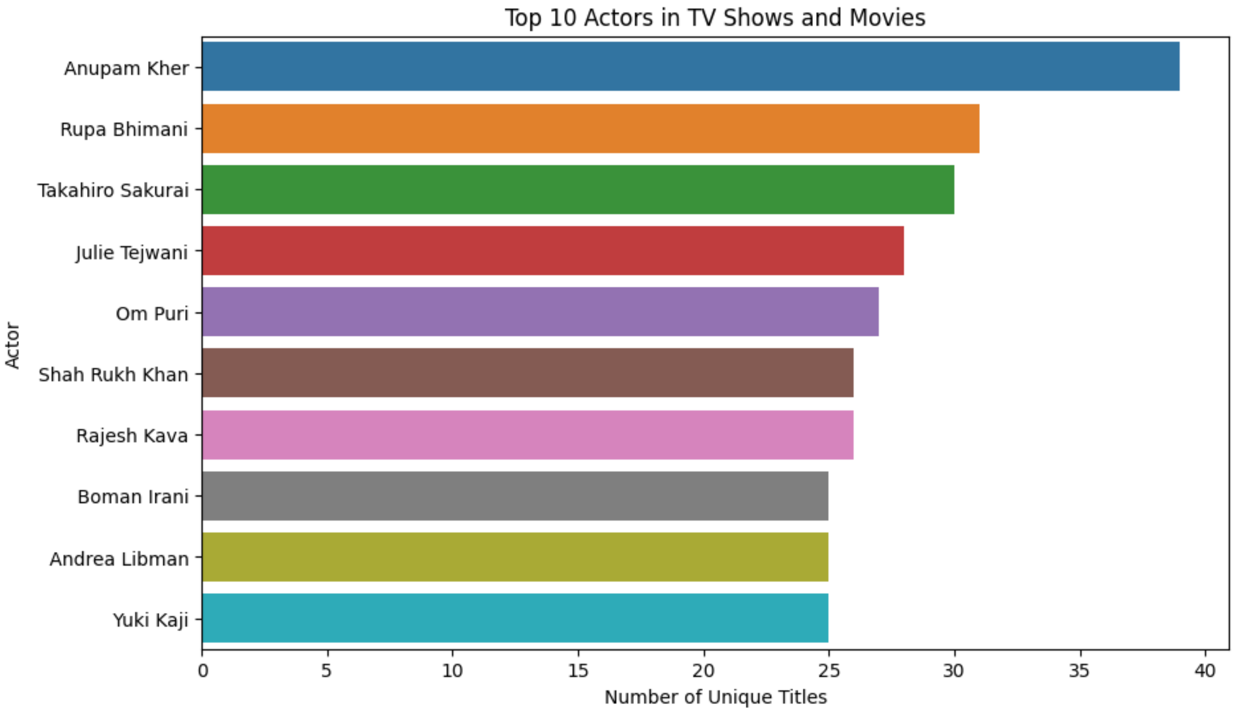
Analysis of actors/directors of different types of shows/movies.

- Identify the top 10 actors who have appeared in most movies or TV shows.

```
df_no_unknown = df[df['cast'] != 'Unknown cast']
actor_tv_counts = df_no_unknown[df_no_unknown['type'] == 'TV Show'].groupby('cast')['title'].nunique()

actor_movie_counts = df_no_unknown[df_no_unknown['type'] == 'Movie'].groupby('cast')['title'].nunique()

actor_total_counts = actor_tv_counts.add(actor_movie_counts, fill_value=0)

top_actors = actor_total_counts.sort_values(ascending=False).head(10)

print("Top 10 Actors in TV Shows and Movies:")
print(top_actors)
```

```
Top 10 Actors in TV Shows and Movies:
cast
 Anupam Kher          39.0
 Rupa Bhimani         31.0
 Takahiro Sakurai     30.0
 Julie Tejwani        28.0
 Om Puri              27.0
Shah Rukh Khan        26.0
 Rajesh Kava          26.0
 Boman Irani          25.0
 Andrea Libman        25.0
 Yuki Kaji            25.0
Name: title, dtype: float64
```

```
plt.figure(figsize=(10, 6))

# Plot top 10 actors
sns.barplot(x=top_actors.values, y=top_actors.index)
plt.title('Top 10 Actors in TV Shows and Movies')
plt.xlabel('Number of Unique Titles')
plt.ylabel('Actor')
plt.show()
```

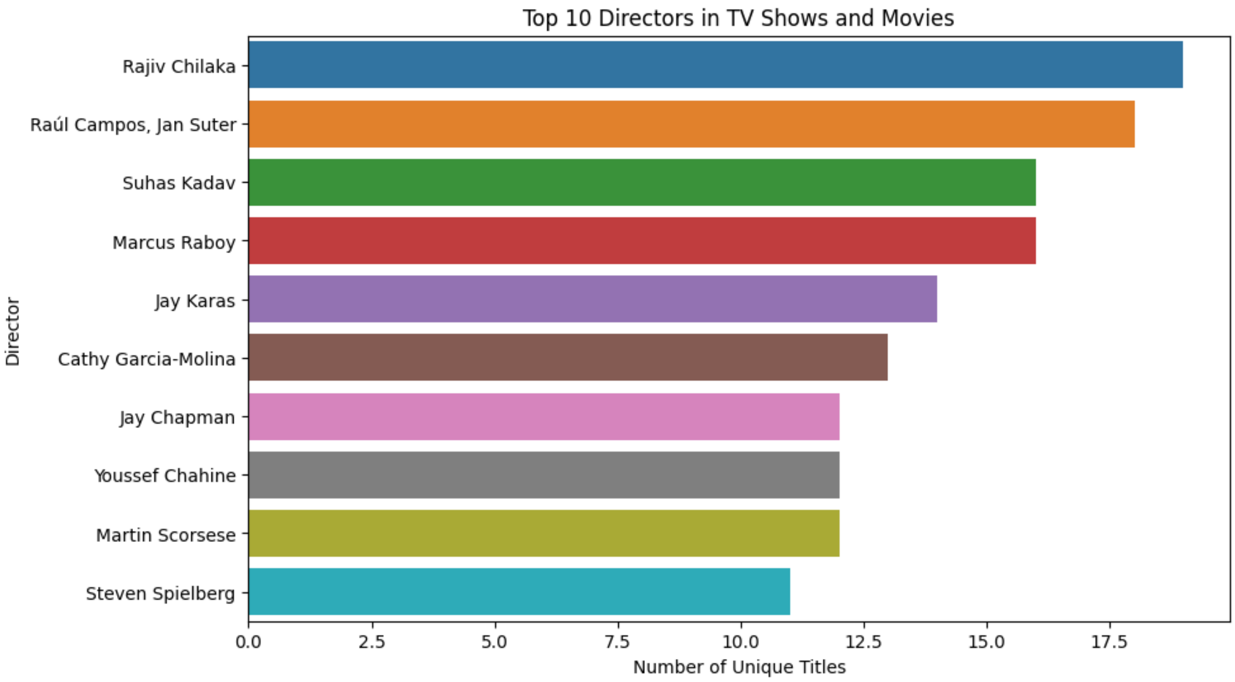


b. Top 10 directors who have directed the most movies or TV shows:

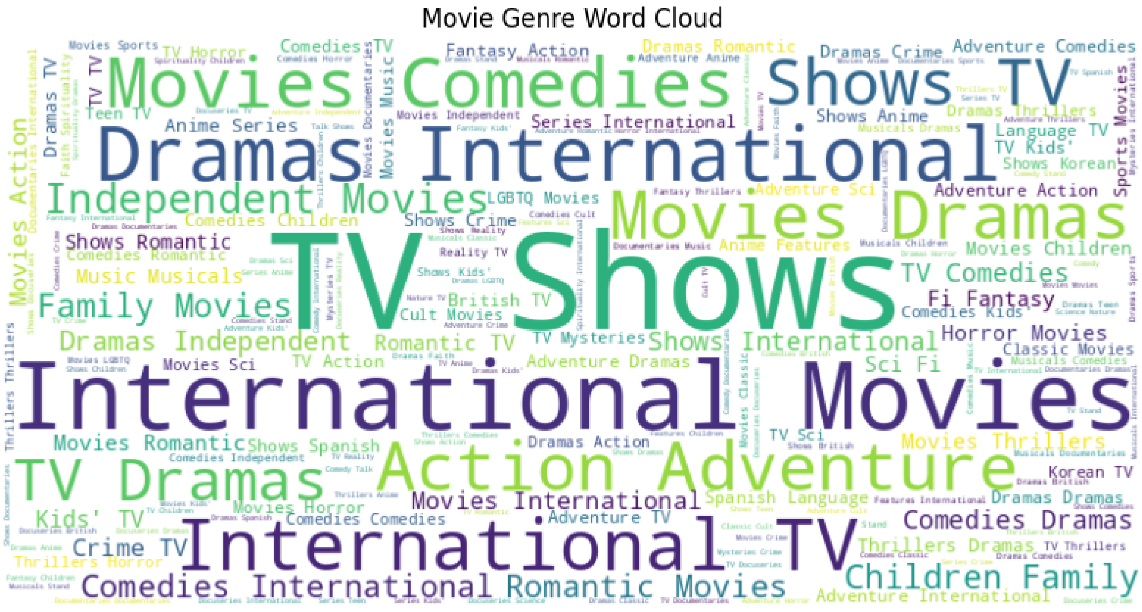
```
df_no_unkown_director = df[df['director'] != 'Unknown director']
director_tv_counts = df_no_unkown_director[df_no_unkown_director['type'] == 'TV Show'].groupby('director')['title'].nunique()

director_movie_counts = df_no_unkown_director[df_no_unkown_director['type'] == 'Movie'].groupby('director')['title'].nunique()

director_total_counts = director_tv_counts.add(director_movie_counts, fill_value=0)

top_directors = director_total_counts.sort_values(ascending=False).head(10)

print("Top 10 Directors in TV Shows and Movies:")
print(top_directors)
```

```
Top 10 Directors in TV Shows and Movies:
director
Rajiv Chilaka             19.0
Raúl Campos, Jan Suter    18.0
Suhas Kadav               16.0
Marcus Raboy              16.0
Jay Karas                 14.0
Cathy Garcia-Molina       13.0
Jay Chapman               12.0
Youssef Chahine           12.0
Martin Scorsese           12.0
Steven Spielberg          11.0
Name: title, dtype: float64
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x=top_directors.values, y=top_directors.index)
plt.title('Top 10 Directors in TV Shows and Movies')
plt.xlabel('Number of Unique Titles')
plt.ylabel('Director')
plt.show()
```



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

all_genres = ' '.join(df['listed_in'])
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(all_genres)

# Display the word cloud using matplotlib
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Movie Genre Word Cloud')
plt.show()
```



As per the alaysis form the above work cloud

- TV shows are the most popular gern
- followed by International movies, International TV and Comedies

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)

```
no_zero_df = df[(df['date_added'] != 0) & (df['release_year'] != 0)]

no_zero_df['date_added'] = pd.to_datetime(no_zero_df['date_added'])

no_zero_df['days_to_add'] = (no_zero_df['date_added'] - pd.to_datetime(no_zero_df['release_year'], format='%Y')).dt.days

movies = no_zero_df[no_zero_df['type'] == 'Movie']

typical_days_to_add = movies['days_to_add'].mode()[0]



print("Typical Days to Add a Movie to Netflix After Release:", typical_days_to_add)
```

```
Typical Days to Add a Movie to Netflix After Release: 424
```

✓ 0s completed at 7:23 PM