Business Case: Netflix - Data Exploration and Visualisation.

The target is to work on content delivery,to increase the subscribers base , to reach out customers satisfactin.This can be achieved by identifying and studying the low falls in content,by understanding the viewers reaction and there preferences regarding various attributes like genre,duration,cast etc and making future recommendation for strategical growth.

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

```
data = pd.read_csv("netflix_file.csv")
```

```
#statistics for dataset

data.describe()
```

	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

```
#Obtaining imformation about dataset
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
```

```
#checking for columns names in dataset
data.columns
```

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

```
data.isna().sum()
```

	0
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

```
#checking for duplicates rows in dataset
data duplicated() sum()
```

```
0
```

```
#checking for shape of the dataset
data.shape
```

```
(8807, 12)
```

```
data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	...	n	o	p	r	s	t	u	v	w	y
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	[Unknown Cast]	United States	2021-09-25	2020	PG-13	90 min	...	1	1	0	1	1	1	1	0	0	0
1	s2	TV Show	Blood & Water	Unknown director	[Ama Qamata, Khosi Ngema, Gail Mabalane, Th...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	...	1	1	0	1	1	1	0	0	1	1
4	s5	TV Show	Kota Factory	Unknown director	[Mayur More, Jitendra Kumar, Ranjan Raj, Al...	India	2021-09-24	2021	TV-MA	2 Seasons	...	1	1	0	1	1	1	0	0	1	0
7	s8	Movie	Sankofa	Haile Gerima	[Kofi Ghanaba, Oyafunmike Ogunlano, Alexandr...	United States, Ghana, Burkina Faso, United Kin...	2021-09-24	1993	TV-MA	125 min	...	1	1	1	1	1	1	0	1	0	0
8	s9	TV Show	The Great British Baking Show	Andy Devonshire	[Mel Giedroyc, Sue Perkins, Mary Berry, Pau...	United Kingdom	2021-09-24	2021	TV-14	9 Seasons	...	0	1	0	1	1	1	0	0	1	1

5 rows × 62 columns

1. Handling null values**

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

   b. Replace with 0 for continuous variables having null values.bold text

```
#Replacing missing values in the director column with "No Data"

data["director"].replace(np.nan , "Unknown director" ,inplace = True)
```

```
#filling missing values in country column with mode vale.
data["country"].fillna(data["country"].mode()[0]).head(5)
```

	country
0	United States
1	South Africa
2	United States
3	United States
4	India

dtype: object

```
#Replacing missing values in the cast column with no data.
data["cast"].replace(np.nan, "Unknown Cast" , inplace = True )
```

```
#finding the mode ratings for movies and tv shows.
movie_rating = data.loc[data["type"] =="Movie","rating"].mode()[0]
tv_rating = data.loc[data["type"] =="TV Show","rating"].mode()[0]
#filling missing rating values based on type of content
data["rating"] = data.apply(lambda x:movie_rating if x["type"]=="Movie" and pd.isna(x["rating"])
                            else tv_rating if x["type"] == "TV Show" and pd.isna(x["rating"])
                            else x["rating"],axis=1)
```

```
#finding the mode duration for movies and Tv shows
movie_duration_mode = data.loc[data["type"]=="Movie","duration"].mode()[0]
tv_duration_mode = data.loc[data["type"]=="TV Show","duration"].mode()[0]
#filling missing duration values based on type of content
data["duration"] = data.apply(lambda x : movie_duration_mode if x["type"]=="Movie" and pd.isna(x["duration"])
                              else tv_duration_mode if x["type"]=="TV Show" and pd.isna(x["duration"])
                              else x["duration"],axis = 1)
```

```
#Droping rows with missin values
data.dropna(inplace= True)
```

```
#converting the "date_added" column to datetime format
data["date_added"] = pd.to_datetime(data["date_added"] , format = "%B %d, %Y" , errors ="coerce")
data["date_added"].fillna(data["date_added"].mode()[0],inplace=True)
```

```
#Extracting month,year,week from "date_added" column
data["month_added"] = data["date_added"].dt.month
data["month_name_added"] = data["date_added"].dt.month_name()
data["year_added"] = data["date_added"].dt.year
data["week_added"] = data["date_added"].dt.isocalendar().week
```

Un-nesting the columns

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

```
#splitting and expanding the cast colimn
data_cast = data["cast"].str.split(", " , expand = True).stack()
data_cast = data_cast.reset_index(level = 1,drop = True).to_frame("cast")
data_cast["show_id"] = data["show_id"]
```

```
#splitting and expanding the country column
data_country  =  data["country"].str.split(",  ",expand=True).stack()
data_country = data_country.reset_index(level = 1,drop = True).to_frame("country")
data_country["show_id"] = data["show_id"]
```

```
#splitting and expanding the listed in column
data_listed_in  =  data["listed_in"].str.split(",  ",expand=True).stack()
data_listed_in = data_listed_in.reset_index(level = 1,drop = True).to_frame("listed_in")
data_country["show_id"] = data["show_id"]
```

```
#splitting and expanding the director column
data_director = data["director"].str.split(",  ",expand=True).stack()
data_director = data_director.reset_index(level = 1,drop = True).to_frame("director")
data_country["show_id"] = data["show_id"]
```

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

a. For Non-graphical Analysis:

```
data_cat = data[["type","country","rating"]].describe()
data_cat
```

	type	country	rating
count	7967	7967	7967
unique	2	748	17
top	Movie	United States	TV-MA
freq	5691	2812	2932

Next steps: Generate code with data_cat | View recommended plots | New interactive sheet

```
value_counts_type = data["type"].value_counts()
value_counts_type
```

type	count
Movie	5691
TV Show	2276

dtype: int64

```
value_counts_country = data["country"].value_counts()
value_counts_country.head(5)
```

country	count
United States	2812
India	972
United Kingdom	418
Japan	244
South Korea	199

dtype: int64

```
value_counts_rating = data["rating"].value_counts()
value_counts_rating.head(5)
```

rating	count
TV-MA	2932
TV-14	1927
R	788
TV-PG	771
PG-13	482

dtype: int64

```
value_counts_release_year = data["release_year"].value_counts()
value_counts_release_year.head()
```

release_year	count
2018	1037
2017	966
2019	913
2020	852
2016	837

dtype: int64

```
unique_type = data["type"].unique()
unique_type
```

```
array(['Movie', 'TV Show'], dtype=object)
```

```
unique_rating = data["rating"].unique()
unique_rating
```

```
array(['PG-13', 'TV-MA', 'TV-14', 'TV-Y7', 'PG', 'R', 'TV-PG', 'TV-Y',
       'TV-G', 'G', 'NC-17', '74 min', '84 min', '66 min', 'NR',
       'TV-Y7-FV', 'UR'], dtype=object)
```

```
unique_release_year= (data["release_year"].unique())
unique_release_year
```

```
array([2020, 2021, 1993, 2018, 1998, 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, 1996, 1984, 1997, 1980, 1961, 1995, 1985, 2000, 1976,
       1959, 1988, 1972, 1981, 1964, 1945, 1954, 1979, 1958, 1956, 1963,
       1970, 1973, 1974, 1960, 1966, 1971, 1962, 1969, 1977, 1967, 1968,
       1965, 1946, 1942, 1955, 1944, 1947, 1943])
```

```
data["type"].value_counts()
```

type	count
Movie	5691
TV Show	2276

dtype: int64

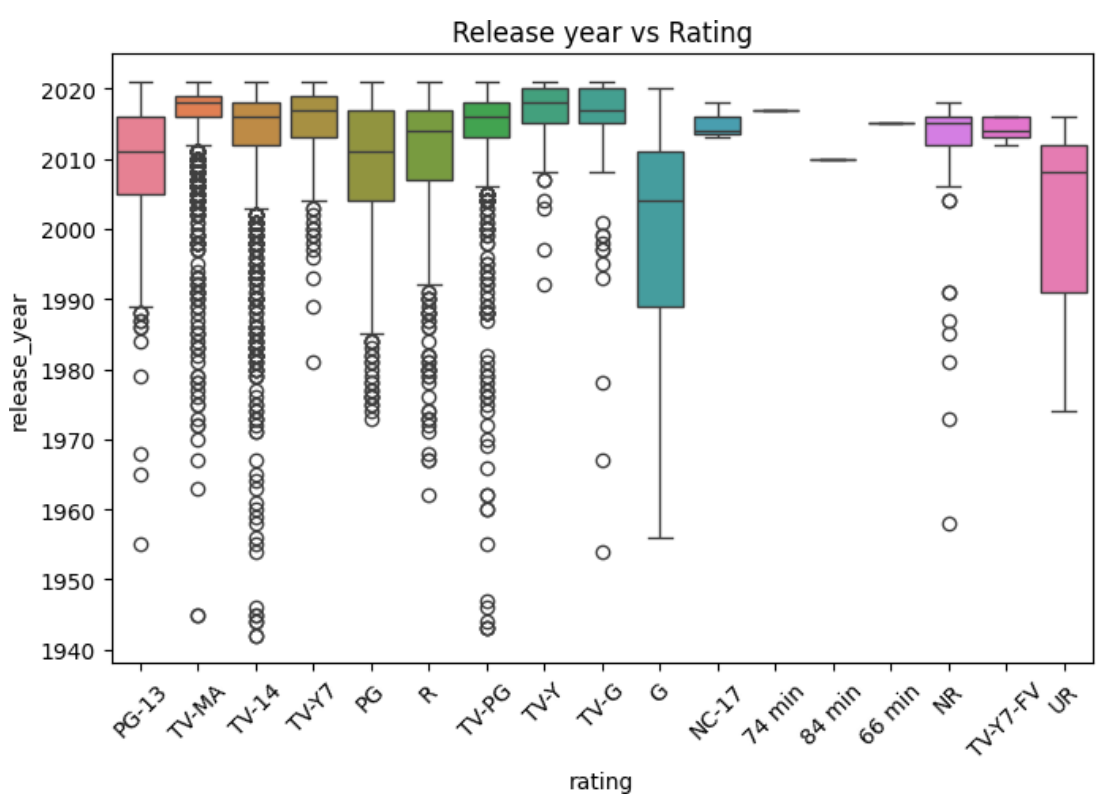
Double-click (or enter) to edit

```
round (data["type"].value_counts()/len(data)*100,2)
```

type	count
Movie	71.43
TV Show	28.57

dtype: float64

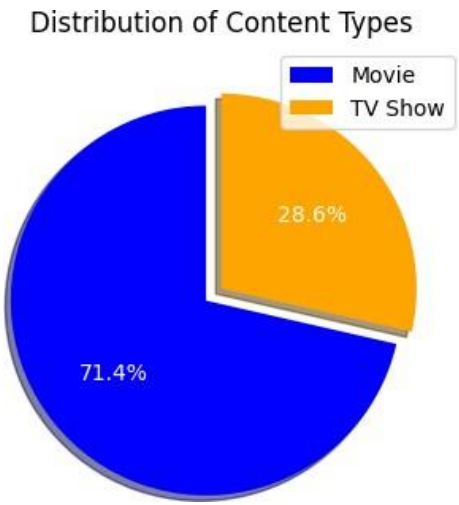
```
plt.figure(figsize = (8,5))
sns.boxplot(data = data ,x="rating" , y= "release_year",hue ="rating",legend=False)
plt.title("Release year vs Rating")
plt.xticks(rotation = 45)
plt.show()
```




```
x=data.groupby(["type"])["type"].count()
y=len(data)
r=((x/y)*100).round(2)

mf_ratio = pd.DataFrame(r)
mf_ratio.rename({"type":"%"},axis  =  1,inplace=True)


plt.figure(figsize=(8,4))
colors = ["blue" , "orange"]
explode = (0.1,0)
plt.pie(mf_ratio["%"],labels=mf_ratio.index,autopct="%1.1f%%",colors=colors,explode=explode,
        shadow=True,startangle=90,textprops={"color":"white"})
plt.legend(loc="upper  right")
plt.title("Distribution of Content Types")
plt.show()
```



Comparison of tv shows vs. movies.

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

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

```
data_1= data[data["type"]=="Movie"]
data_1.groupby("country")["title"].count().sort_values(ascending=False).head(5)
```

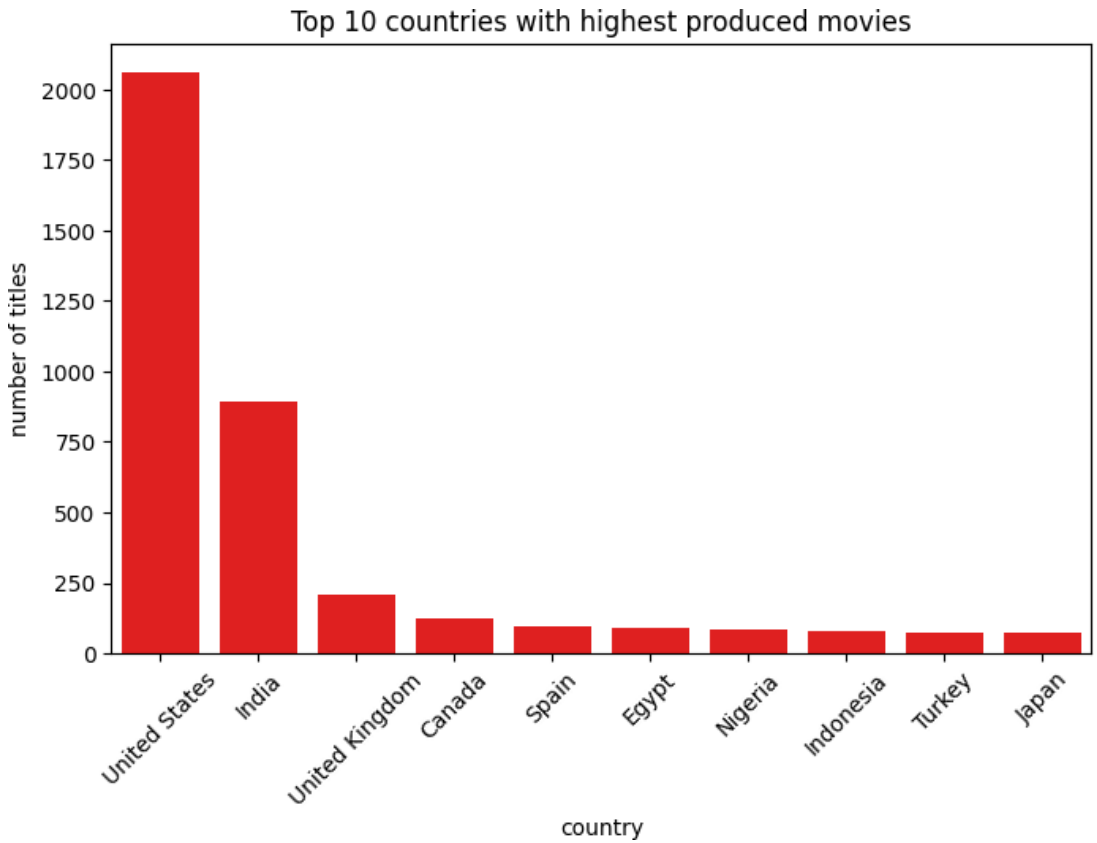
country	title
United States	2058
India	893
United Kingdom	206
Canada	122
Spain	97

dtype: int64

```
data_1=data.loc[(data["type"]=="Movie")]
top_10_countries = data_1.groupby("country")["title"].count().sort_values(ascending=False).head(10)
```

```
plt.figure(figsize=(8,5))
sns.barplot(x=top_10_countries.index,y=top_10_countries.values,color = "red")
plt.xlabel("country")
plt.ylabel("number of titles")
plt.xticks(rotation = 45)
plt.title("Top 10 countries with highest produced movies")

plt.show()
```




```
data_2 = data.loc[(data["type"]=="TV Show")]
data_2.groupby("country")["title"].count().sort_values(ascending=False).head(10)
```

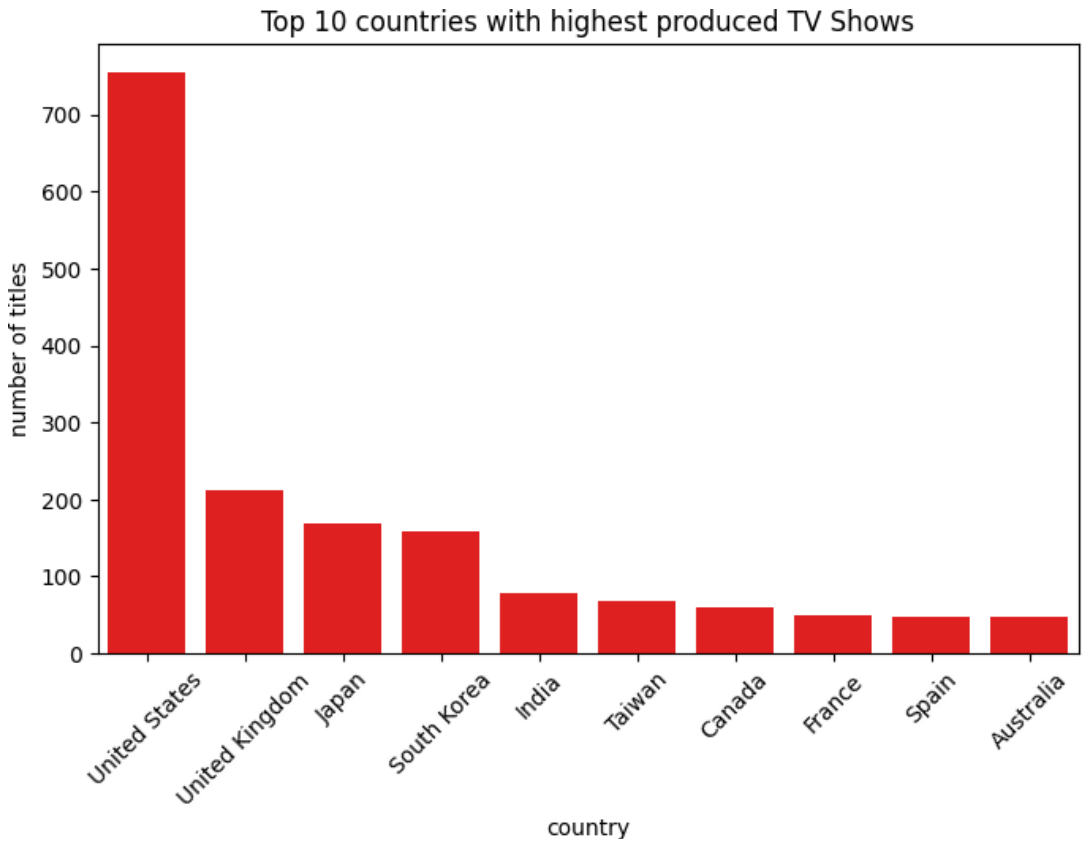
country	title
United States	754
United Kingdom	212
Japan	168
South Korea	158
India	79
Taiwan	68
Canada	59
France	49
Spain	48
Australia	47

dtype: int64

```
data2=data.loc[data["type"]=="TV  Show"]
top_10_countries = data_2.groupby("country")["title"].count().sort_values(ascending=False).head(10)

plt.figure(figsize=(8,5))
sns.barplot(x=top_10_countries.index,y=top_10_countries.values,color  =  "red")

plt.xlabel("country")
plt.ylabel("number of titles")
plt.xticks(rotation=45)
plt.title("Top 10 countries with highest produced TV Shows")
plt.show()
```

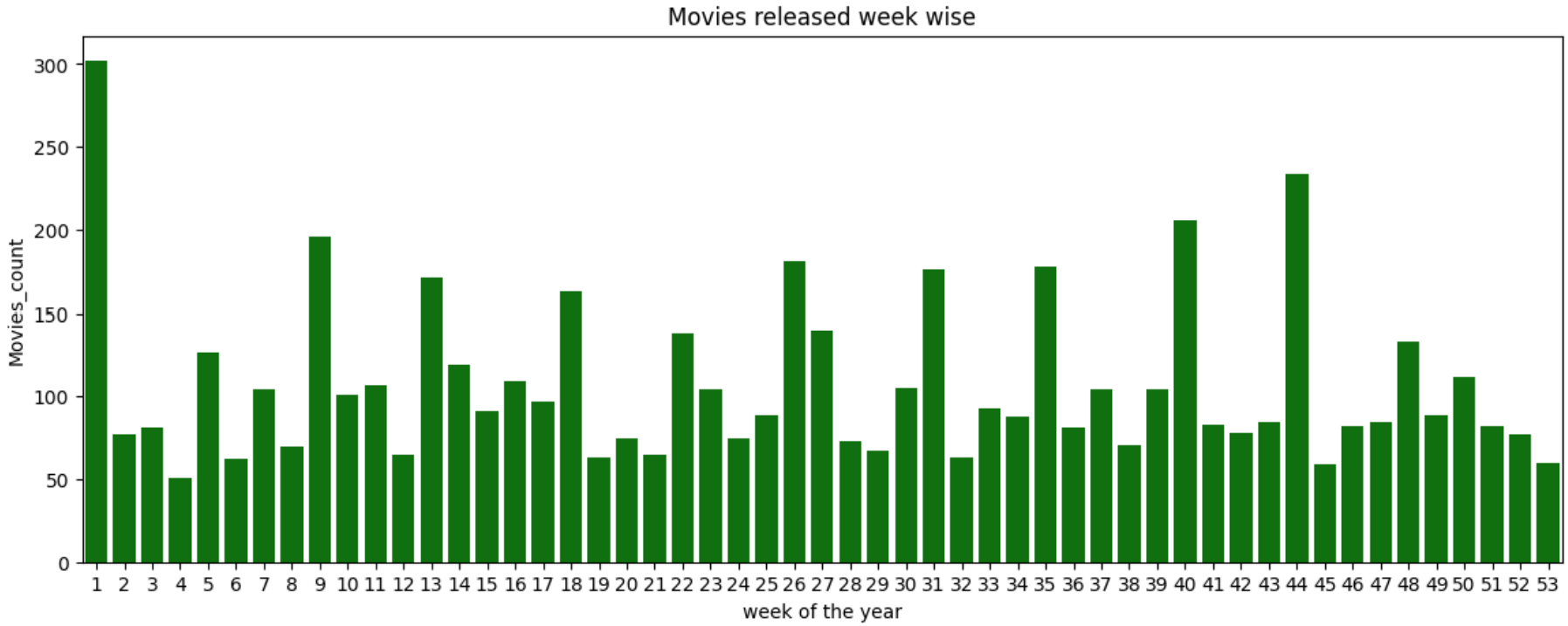


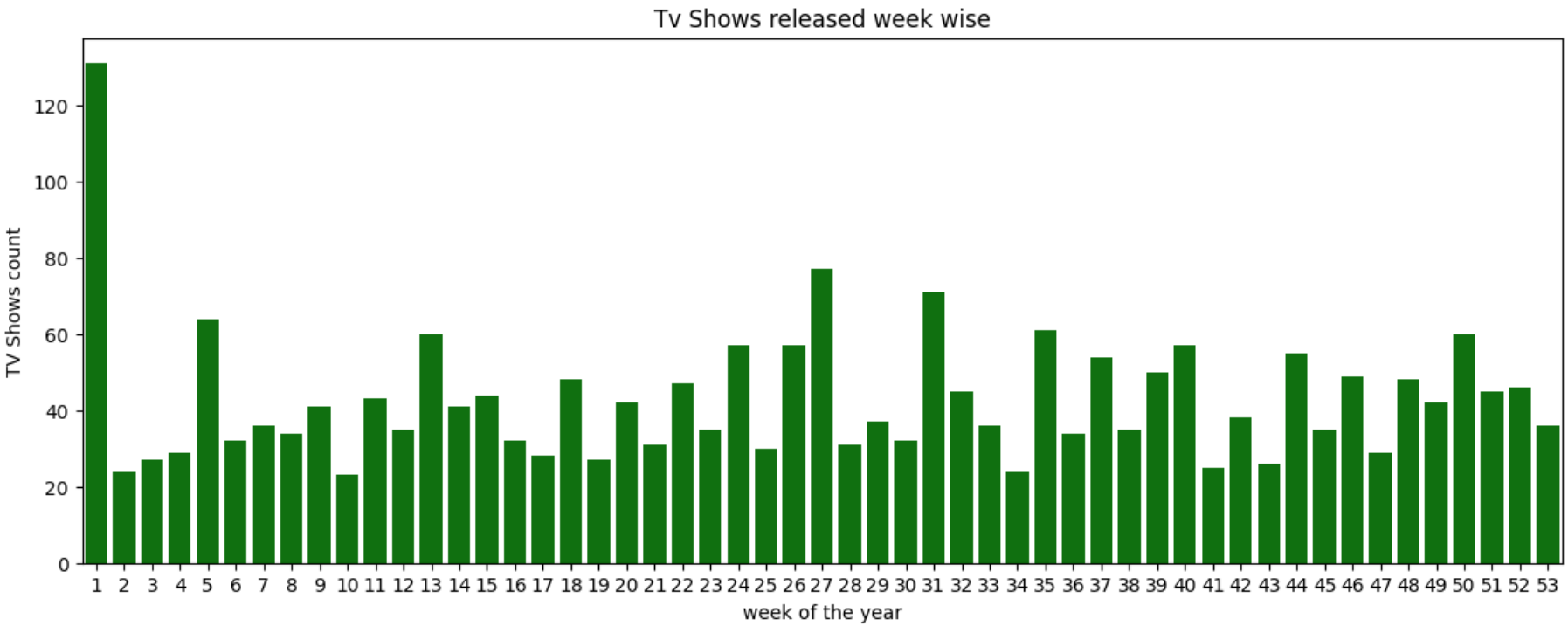
What is the best time to launch a TV show?

```
data_movies = data[data["type"]=="Movie"]
movies_count = data_movies["week_added"].value_counts().sort_index()
plt.figure(figsize=(14,5))
sns.barplot(x=movies_count.index, y=movies_count.values,color = "green")
plt.xlabel("week of the year")
plt.ylabel("Movies_count")
plt.title("Movies released week wise")
plt.show()

data_tv_shows = data[data["type"]=="TV Show"]
tv_shows_count = data_tv_shows["week_added"].value_counts().sort_index()
plt.figure(figsize=(14,5))
sns.barplot(x=tv_shows_count.index , y=tv_shows_count.values,color="green")
plt.xlabel("week of the year")
plt.ylabel("TV Shows count")
```

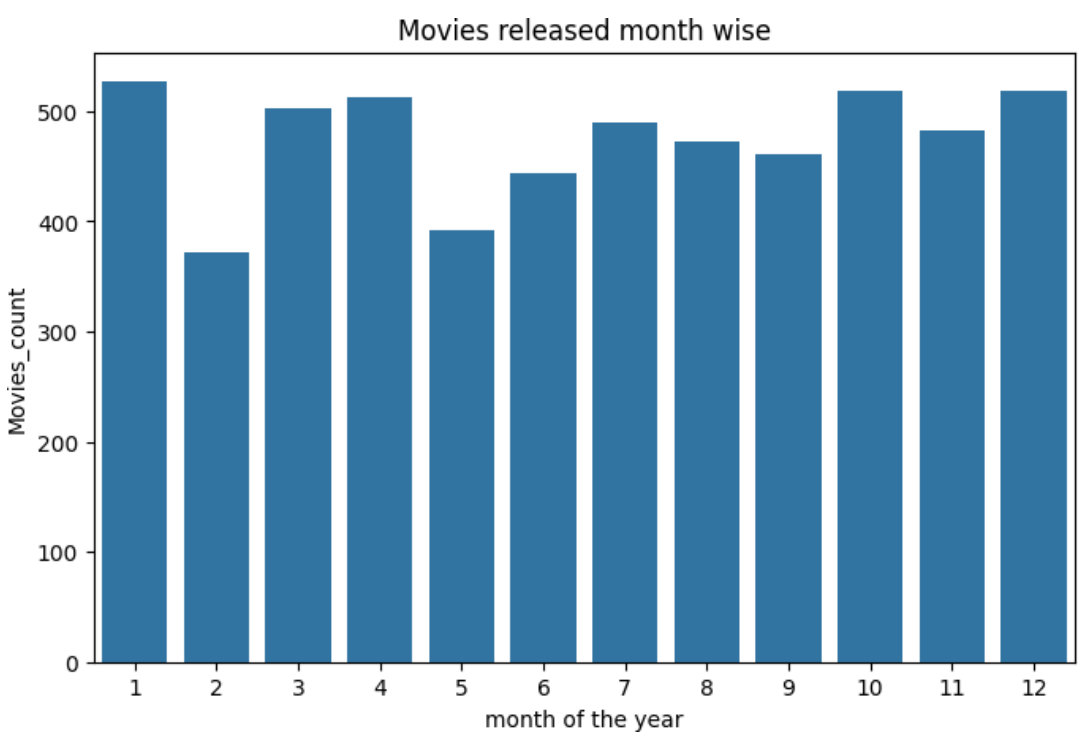
```
plt.title("Tv Shows released week wise")
plt.show()
```


Movies released week wise
Movies_count
week of the year


Tv Shows released week wise
TV Shows count
week of the year

```
data_movies = data[data["type"]=="Movie"]
movies_count = data_movies["month_added"].value_counts().sort_index()

plt.figure(figsize=(8,5))
sns.barplot(x=movies_count.index, y=movies_count.values)
plt.xlabel("month of the year")
plt.ylabel("Movies_count")
plt.title("Movies released month wise")
plt.show()
```

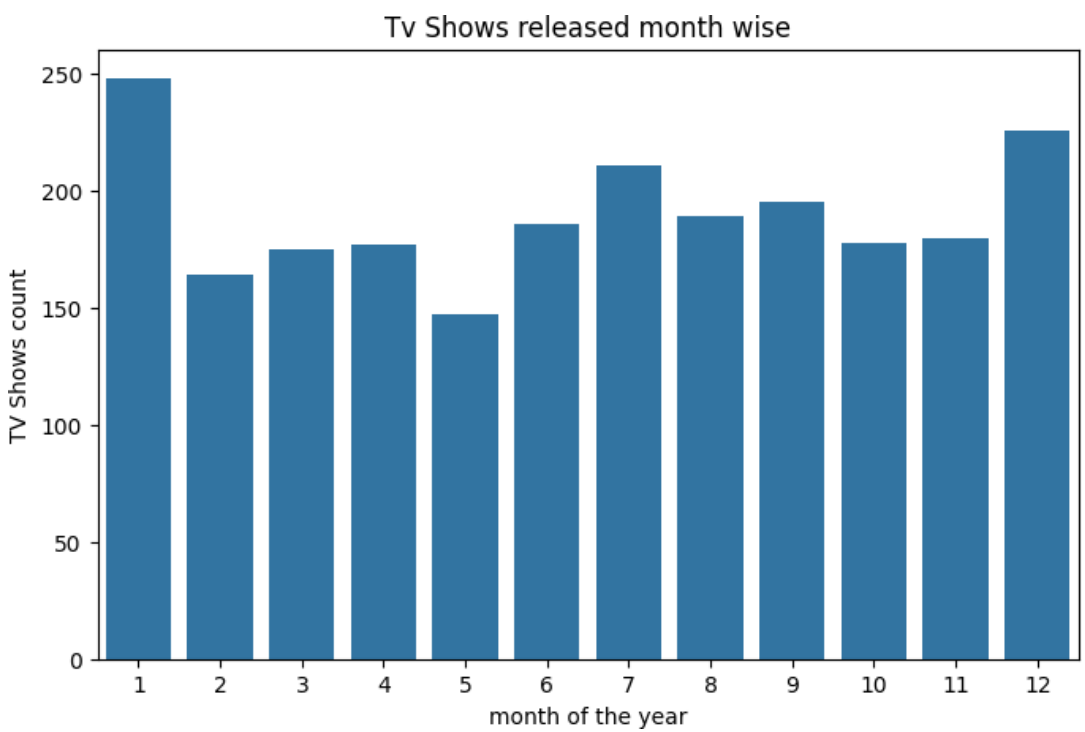



```
data_tv_shows = data[data["type"]=="TV Show"]
tv_shows_count = data_tv_shows["month_added"].value_counts().sort_index()

plt.figure(figsize=(8,5))
sns.barplot(x=tv_shows_count.index  ,  y=tv_shows_count.values)

plt.xlabel("month of the year")
plt.ylabel("TV Shows count")
plt.title("Tv Shows released month wise")
```

```
plt.show()
```



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

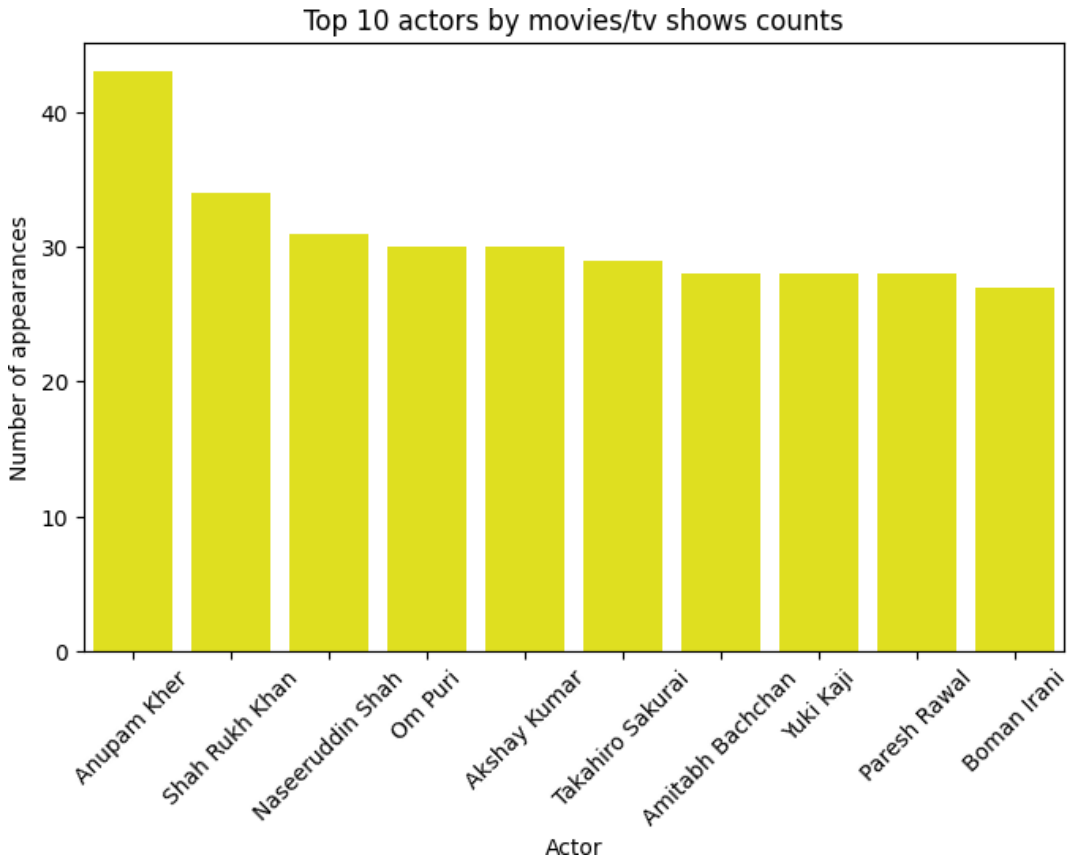
a. Identify the top 10 directors who have appeared in most movies or TV shows.

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

```
cast_counts =data_cast["cast"].value_counts()[1:]

top_10_cast = cast_counts.head(10)

plt.figure(figsize=(8,5))
sns.barplot(x=top_10_cast.index , y=top_10_cast.values,color="yellow")
plt.xlabel("Actor")
plt.ylabel("Number of appearances")
plt.xticks(rotation = 45)
plt.title("Top 10 actors by movies/tv shows counts")
plt.show()
```

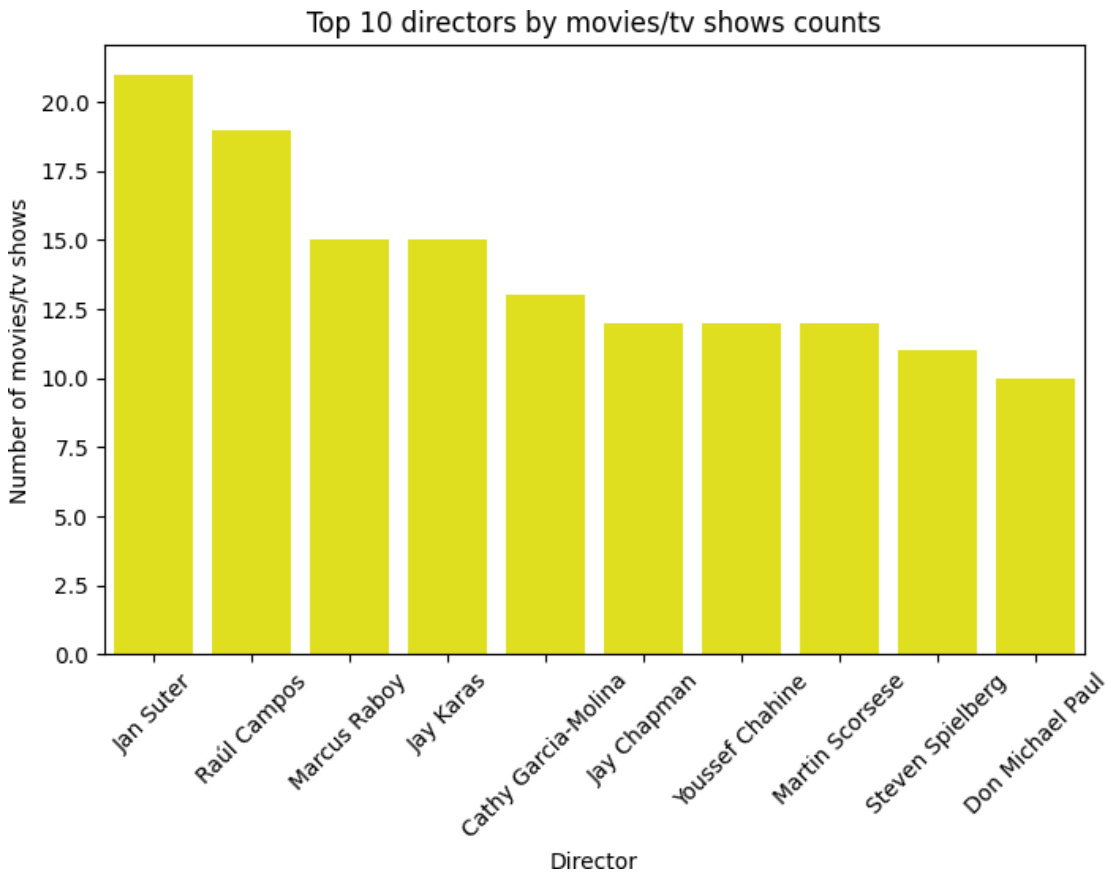


From above it is seen that Anupam Kher is the actor who appeared in most of the movies.

```
director_counts =data_director["director"].value_counts()[1:]

top_10_directors = director_counts.head(10)

plt.figure(figsize=(8,5))
sns.barplot(x=top_10_directors.index , y=top_10_directors.values,color="yellow")
plt.xlabel("Director")
plt.ylabel("Number of movies/tv shows")
plt.xticks(rotation = 45)
plt.title("Top 10 directors by movies/tv shows counts")
plt.show()
```

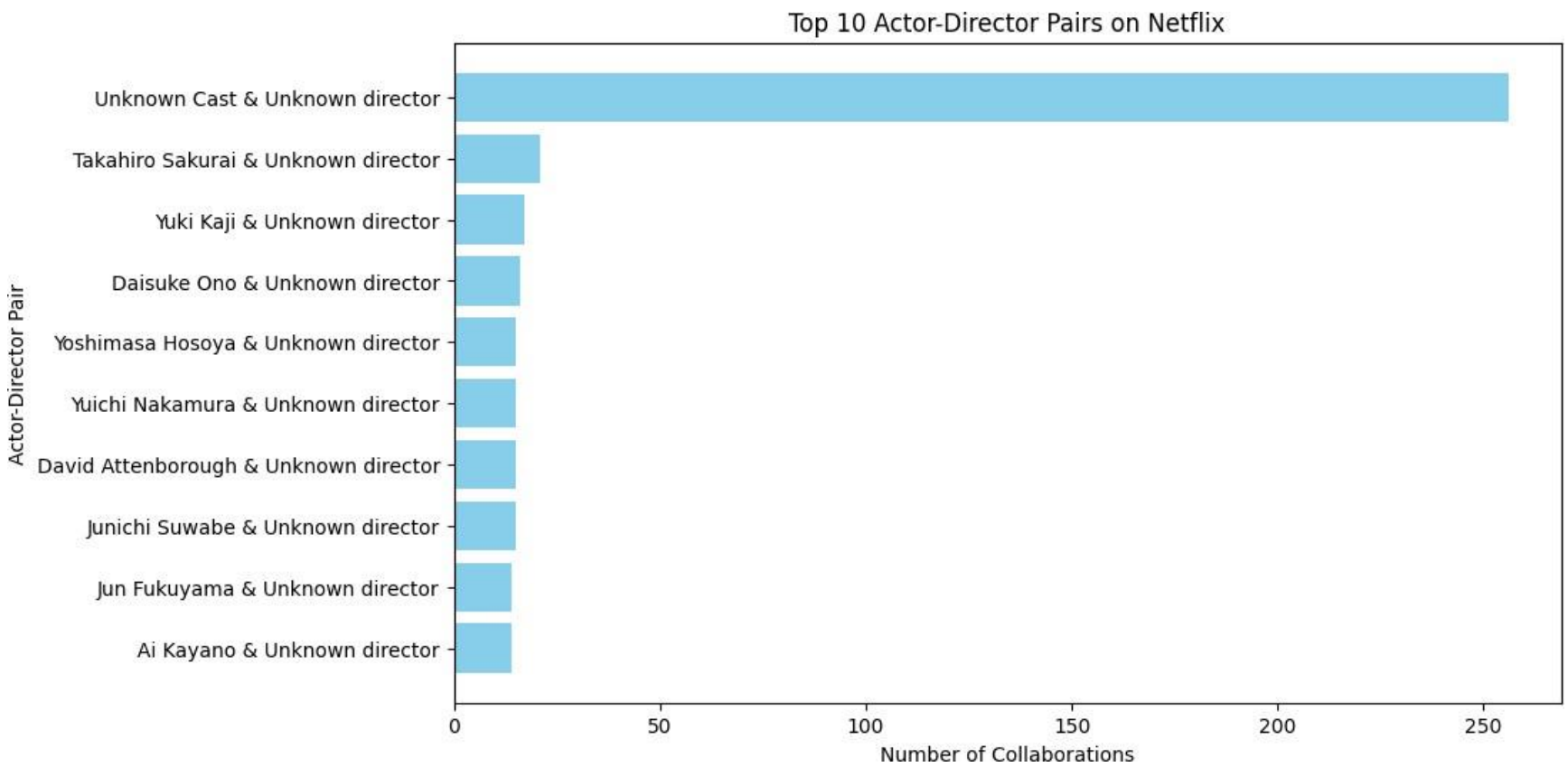


From above graph analysis we can listed top 10 directors.

```
#Top 10actor directors pairs
```

```
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter
data = data.dropna(subset=['cast', 'director'])
data['cast'] = data['cast'].astype(str)
data['cast'] = data['cast'].str.split(',')
actor_director_pairs = []
for index, row in data.iterrows():
    for actor in row['cast']:
        actor_director_pairs.append((actor.strip(), row['director'].strip()))
pair_counts = Counter(actor_director_pairs)
pair_counts_df = pd.DataFrame(pair_counts.items(), columns=['Actor-Director Pair', 'Count'])
```

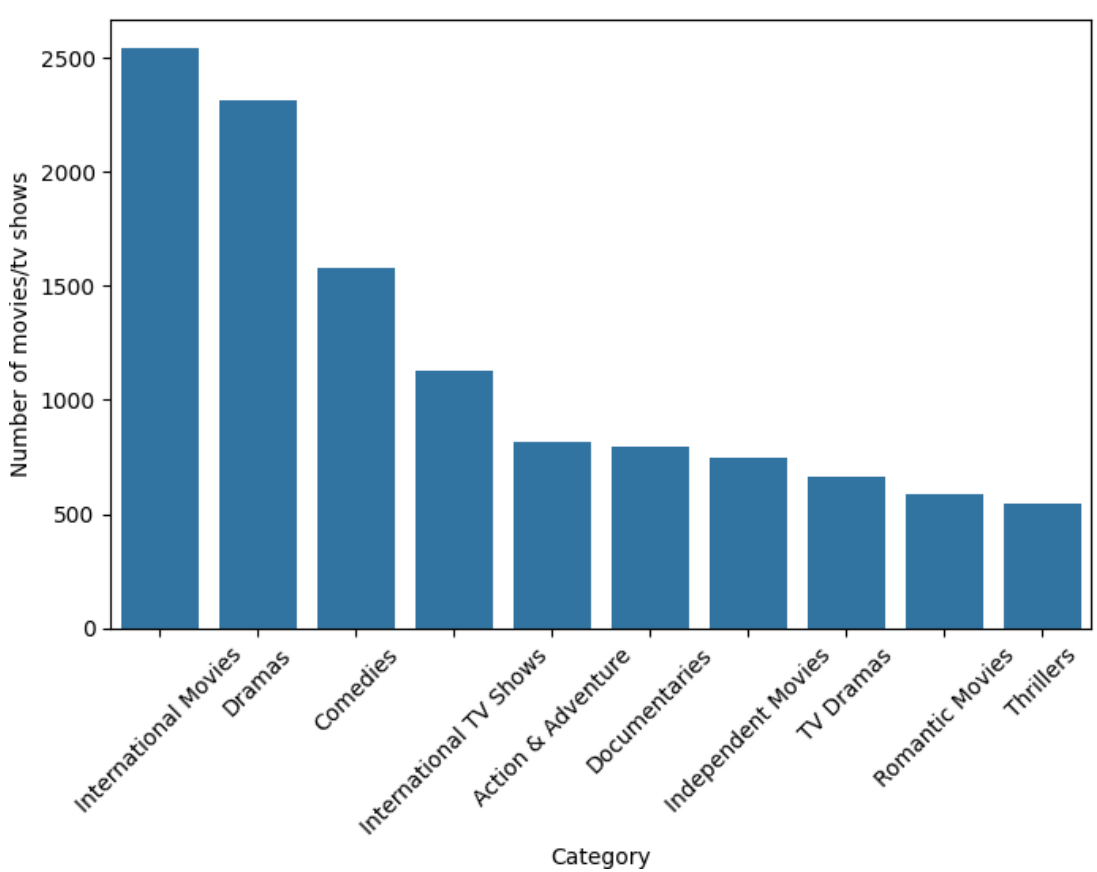
```
pair_counts_df['Actor-Director Pair'] = pair_counts_df['Actor-Director Pair'].apply(lambda x: f'{x[0]} & {x[1]}')
top_10_pairs = pair_counts_df.sort_values(by='Count', ascending=False).head(10)
# Plotting the top 10 actor-director pairs
plt.figure(figsize=(10, 6))
plt.barh(top_10_pairs['Actor-Director Pair'], top_10_pairs['Count'], color='skyblue')
plt.xlabel('Number of Collaborations')
plt.ylabel('Actor-Director  Pair')
plt.title('Top 10 Actor-Director Pairs on Netflix')
plt.gca().invert_yaxis()  # Invert y-axis to have the highest count on top
plt.show()
```



Julie Tejwani and Rajiv Chilaka are on top.

Which genre movies are more popular or produced more

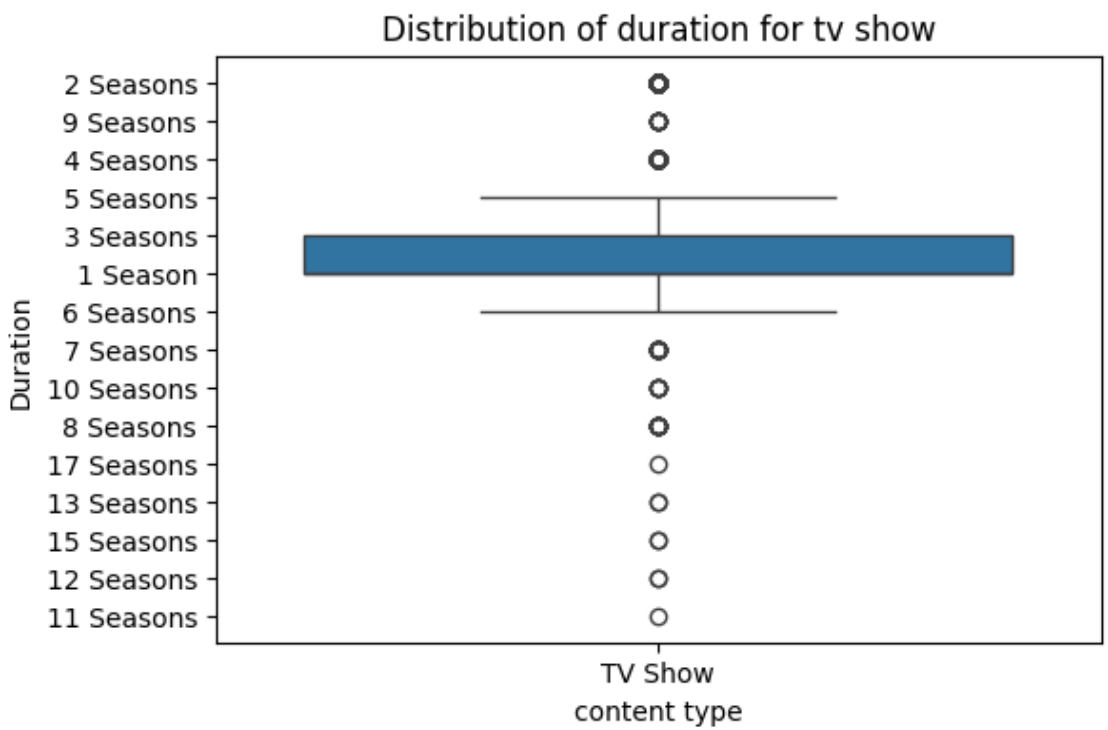
```
data_listed_in["listed_in"] =data_listed_in["listed_in"].str.strip()
listed_in_counts = data_listed_in["listed_in"].value_counts()
top_10_listed_in = listed_in_counts.head(10)
plt.figure(figsize=(8,5))
sns.barplot(x=top_10_listed_in.index,y=top_10_listed_in.values)
plt.xlabel("Category")
plt.ylabel("Number of movies/tv shows")
plt.xticks(rotation = 45)
plt.show()
```



From the above graph it is derived that number one category is international movies followed by dramas and comedies.

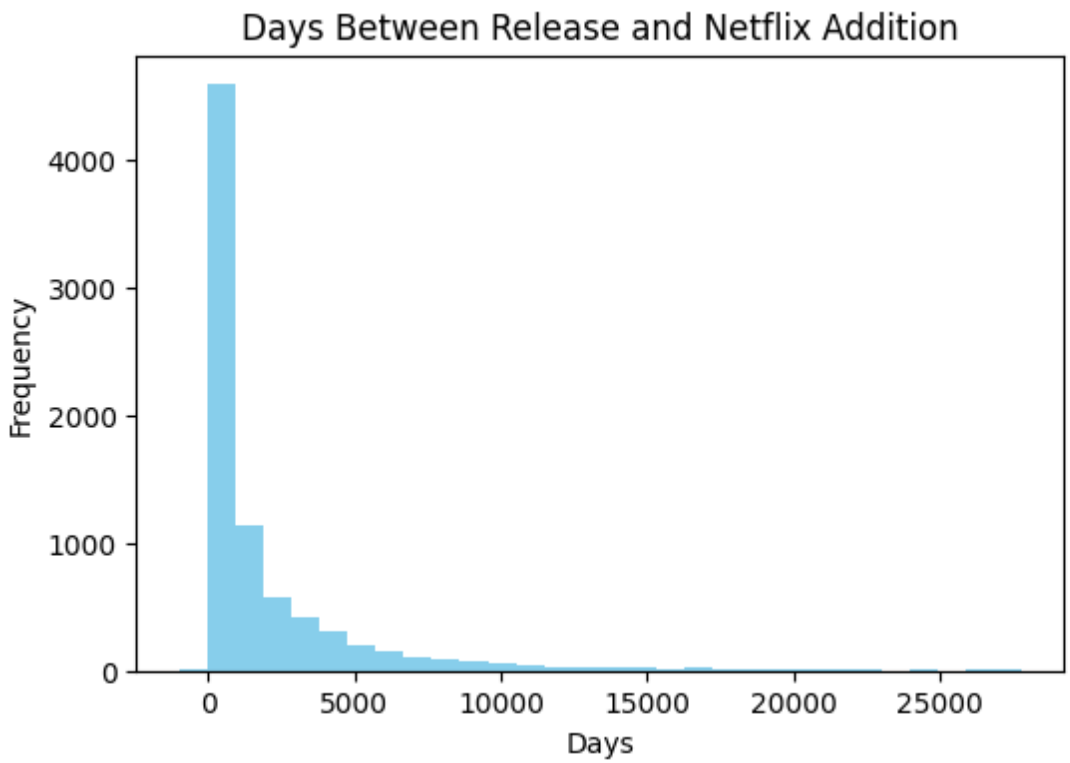
Duration distribution for tv shows

```
data['date_added'] = pd.to_datetime(data['date_added'])
data['release_date'] = pd.to_datetime(data['release_year'].astype(str) + '-01-01')
data['Days to Netflix'] = (data['date_added'] - data['release_date']).dt.days
# Plotting the result
plt.figure(figsize=(6, 4))
plt.hist(data['Days to Netflix'], bins=30, color='skyblue')
plt.title('Days Between Release and Netflix Addition')
plt.xlabel('Days')
plt.ylabel('Frequency')
plt.show()
```

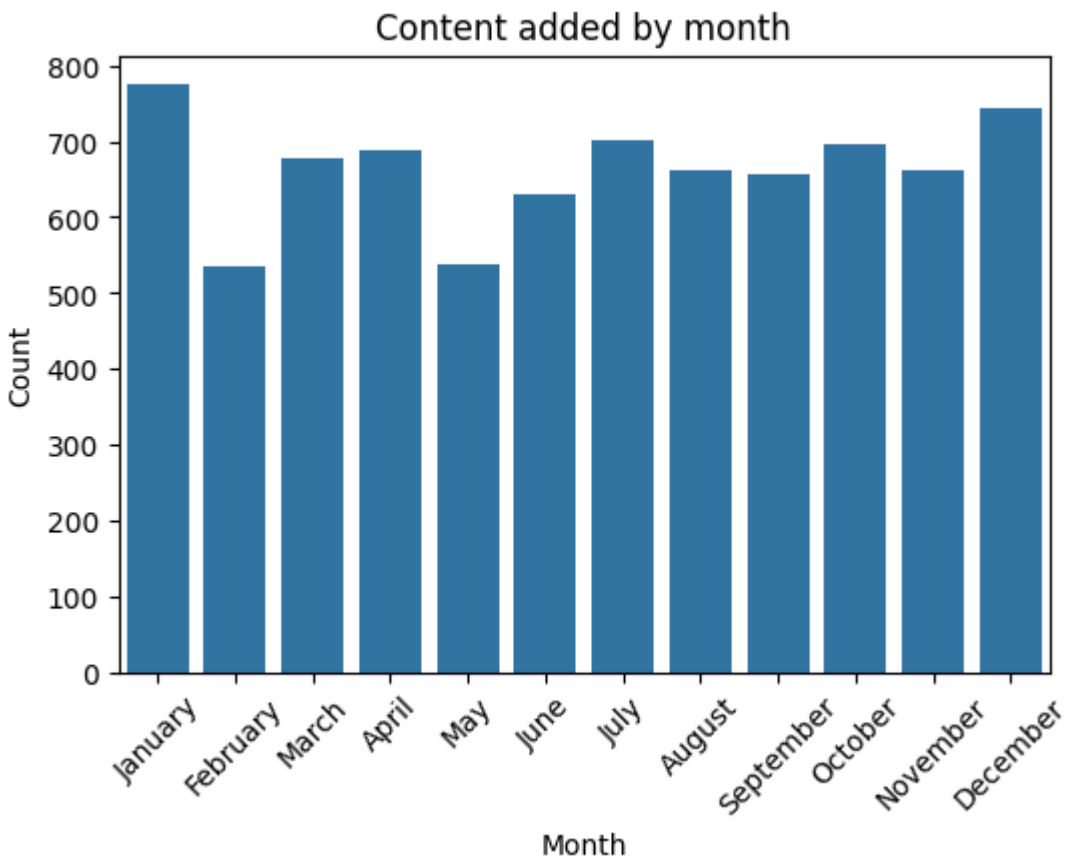


Movies and TV Shows added over time

```
data['date_added'] = pd.to_datetime(data['date_added'])
data['release_date'] = pd.to_datetime(data['release_year'].astype(str) + '-01-01')
data['Days to Netflix'] = (data['date_added'] - data['release_date']).dt.days
# Plotting the result
plt.figure(figsize=(6, 4))
plt.hist(data['Days to Netflix'], bins=30, color='skyblue')
plt.title('Days Between Release and Netflix Addition')
plt.xlabel('Days')
plt.ylabel('Frequency')
plt.show()
```



```
data["month_added"] = pd.to_datetime(data["date_added"]).dt.month_name()
month_order
=["January","February","March","April","May","June","July","August","September","October","November","December"]
monthly_counts = data["month_added"].value_counts().loc[month_order]
max_count = monthly_counts.max()
plt.figure(figsize=(6,4))
sns.barplot(x=monthly_counts.index , y=monthly_counts.values)
plt.xlabel("Month")
plt.ylabel("Count")
plt.title("Content added by month")
plt.xticks(rotation = 45)
plt.show()
```



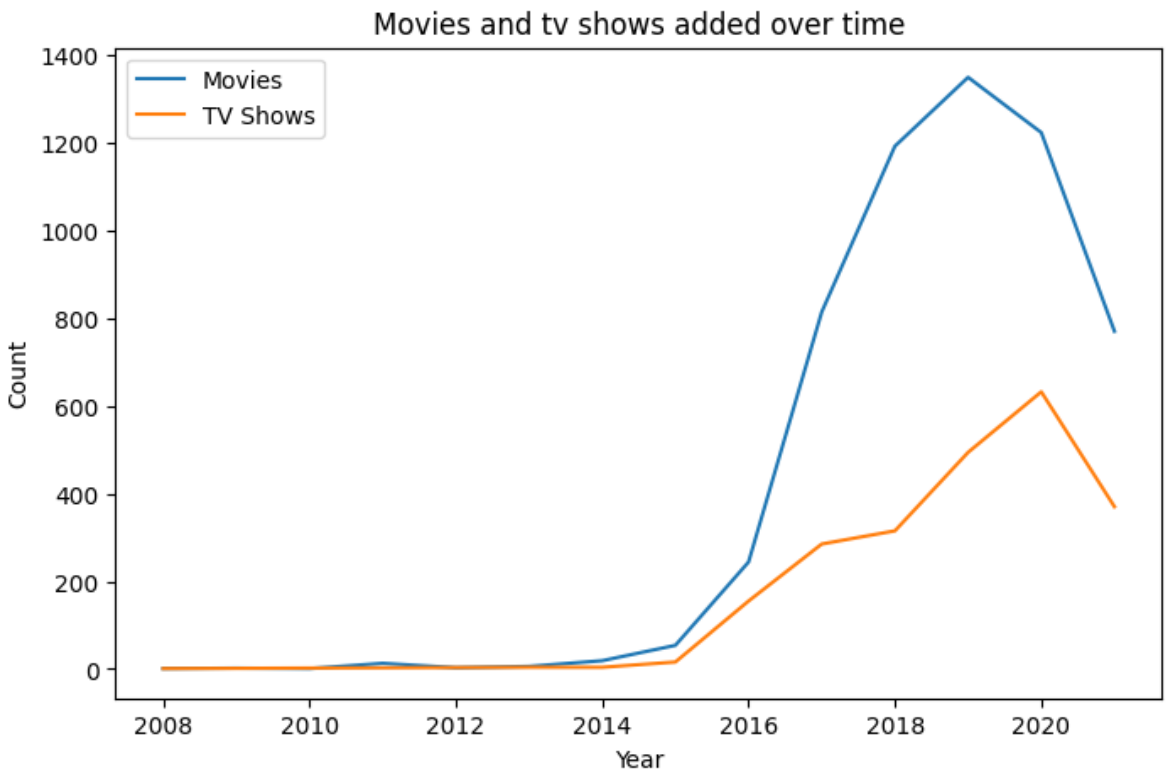
```
data_movies = data[data["type"]=="Movie"]
data_tv_show = data[data["type"]=="TV Show"]

movies_count = data_movies["year_added"].value_counts().sort_index()
tv_shows_count = data_tv_shows["year_added"].value_counts().sort_index()

plt.figure(figsize=(8,5))
plt.plot(movies_count.index,movies_count.values,label="Movies")
plt.plot(tv_shows_count.index,tv_shows_count.values,label="TV Shows")

plt.xlabel("Year")
plt.ylabel("Count")
plt.title("Movies and tv shows added over time")
plt.legend()

plt.show()
```



Insights

A major part of Netflix content has been released in recent years.

Our analysis shows that Netflix had added more movies than TV Shows,with expectation of movies content dominates the tv show content.

Content from the United States, India, and the United Kingdom makes up nearly 50% of the entire Netflix.

Month wise analysis shows the seasonality and gives the strategic approach for content delivery.

Netflix saw its real growth starting from the year 2015 and we can see it added more movies than tv shows over the years also pandemic situaton after 2020 create some adverse effects on growth.

2019 is the year when Netflix added movies count is on the pic.

Most of the TV shows on Netflix have one season,it clearly suggesting that viewers are likely to prefer shorter content.

The ratings TV-MA and TV-14 dominate the content on Netflix.

Netflix primary target is to be mature and teen audiences.This type of content delivery strategies are useful for to increase viewers base and companys growth. .

As in this modern world the streaming industry evolves continuously understanding this patterns and trends becomes increasingly essential for continuous growth and to manage all this data as per ratings.

Recommendation

The pie chart visualization shows that approximately 70% of the content on Netflix consists of film while the remaining 30% are TV hows.Netflix should try to balance both the categories as both are popular.

Viewers watching TV shows which has 1-2 seasons,so should add more content with 1-2 seasons.

As international movies ,dramas ,comedies and international Tv shows contribute about 50%.Netflix should focus more on adding content related to these as they are most popular and watched more.

With content available from 748 different countries, Netflix has the opportunity to further customize its offerings based on regional popularity. This could lead to an increase in local subscriptions and customer satisfaction.

Netflix could diversify its portfolio by exploring underrepresented genres and ratings to attract a more diverse audience.

By understanding seasonal trend, Netflix could focus on releasing highly anticipated new seasons or exclusive content during these months to capitalize or to increase viewership.

Recognizing the popularity of shorter TV series, Netflix should continue focusing on producing limited series and shorter season formats.

Thank You…