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

df=pd.read_csv('netflix_data.csv')
df.sample(5)
```

Out[1]:

	show_id	type	title	director	country	date_added	release_year	rating	du
5566	s7602	Movie	NOVA: Building Chernobyl's MegaTomb	Martin Gorst	United States	7/1/2019	2017	TV-PG	5
7631	s3681	TV Show	Free Rein	Not Given	United States	7/6/2019	2019	TV-G	Se
767	s623	Movie	Lying and Stealing	Matt Aselton	United States	6/30/2021	2019	R	10
6629	s292	TV Show	SHAMAN KING	Not Given	Japan	8/9/2021	2021	TV-14	1 S
6247	s8456	Movie	The Pirate Fairy	Peggy Holmes	United States	6/15/2014	2014	G	1

In [107…
```
df.isnull().sum()
# Hence no null values are present
```

Out[107…
```
show_id         0
type            0
title           0
director        0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
dtype: int64
```

In [3]:
```
#view the dataset info
print('information of datastet:\n',df.info())

#checkinjg any duplicated rows
print('\nchecking duplicated values present or not :\n',df.duplicated().sum()) # no
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8790 non-null   object
 1   type          8790 non-null   object
 2   title         8790 non-null   object
 3   director      8790 non-null   object
 4   country       8790 non-null   object
 5   date_added    8790 non-null   object
 6   release_year  8790 non-null   int64 
 7   rating        8790 non-null   object
 8   duration      8790 non-null   object
 9   listed_in     8790 non-null   object
dtypes: int64(1), object(9)
memory usage: 686.8+ KB
information of datastet:
 None

checking duplicated values present or not :
 0
```

In [107…

```
# creating a new column 'genre' which would include the 'listed_in' item with type
df['genre']=df['listed_in'].str.split(',',expand=True)[0]
df['genre'].value_counts()
```

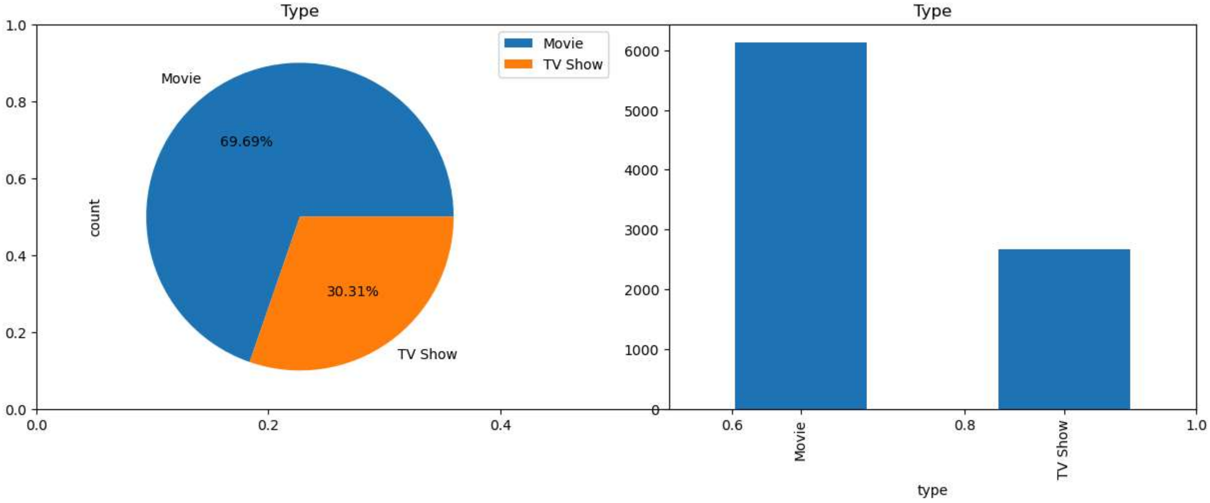
```
Out[107…  genre
          Dramas                          1599
          Comedies                        1210
          Action & Adventure               859
          Documentaries                    829
          International TV Shows           773
          Children & Family Movies         605
          Crime TV Shows                   399
          Kids' TV                         385
          Stand-Up Comedy                  334
          Horror Movies                    275
          British TV Shows                 252
          Docuseries                       220
          Anime Series                     174
          International Movies             128
          Reality TV                       120
          TV Comedies                      119
          Classic Movies                    80
          TV Dramas                         67
          Thrillers                         65
          Movies                            53
          TV Action & Adventure             39
          Stand-Up Comedy & Talk Shows      34
          Romantic TV Shows                 32
          Anime Features                    21
          Independent Movies                20
          Classic & Cult TV                 20
          Music & Musicals                  18
          TV Shows                          16
          Sci-Fi & Fantasy                  13
          Cult Movies                       12
          TV Horror                         11
          Romantic Movies                    3
          Spanish-Language TV Shows          2
          LGBTQ Movies                       1
          TV Sci-Fi & Fantasy                1
          Sports Movies                      1
          Name: count, dtype: int64
```

Data visualizing

```
In [5]:  print('\n',df['type'].value_counts())

         #plotting the show type and movies
         plt.subplots(figsize=(15,5))
         plt.subplot(121)
         df['type'].value_counts().plot(kind='pie',autopct='%.2f%%')
         plt.legend(loc='upper left',bbox_to_anchor=(1,1))
         plt.title('Type')
         plt.subplot(122)
         df['type'].value_counts().plot(kind='bar')
         plt.title('Type')
         plt.show()
```

```
 type
Movie      6126
TV Show    2664
Name: count, dtype: int64
```



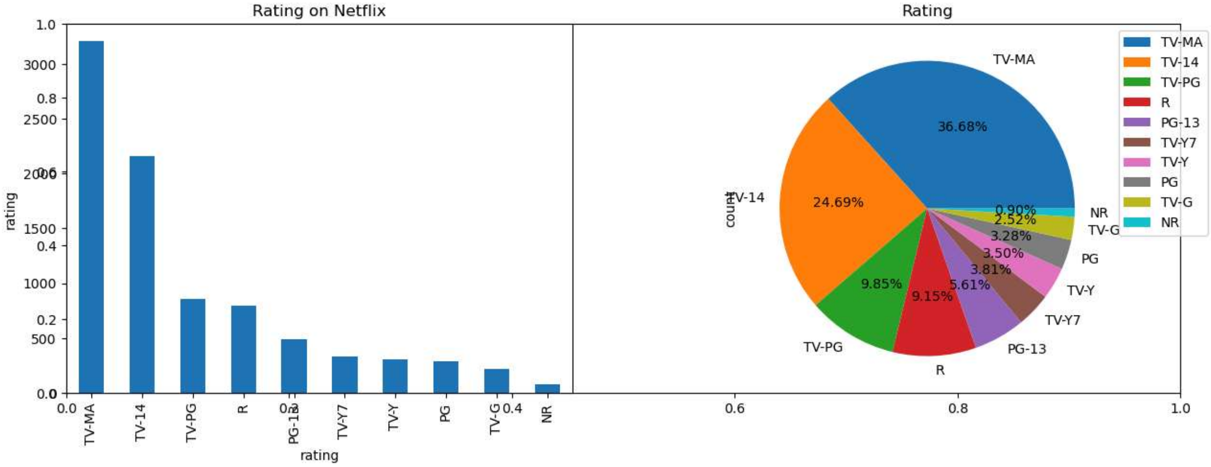
- TOP 10 RATING ON THE NETFLIX

In [13]:

```
print('Top Rating on Netflix:\n',df['rating'].value_counts()[0:10])

# plotting the rating chart
plt.subplots(figsize=(15,5))
plt.subplot(121)
df['rating'].value_counts()[:10].plot(kind='bar') # plotting the top 10 rating in b
plt.ylabel('rating')
plt.title('Rating on Netflix')
plt.subplot(122)
df['rating'].value_counts()[:10].plot(kind='pie',autopct='%.2f%%') # plotting the t
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.title('Rating')
plt.show()
```

```
Top Rating on Netflix:
 rating
TV-MA    3205
TV-14    2157
TV-PG     861
R         799
PG-13     490
TV-Y7     333
TV-Y      306
PG        287
TV-G      220
NR         79
Name: count, dtype: int64
```



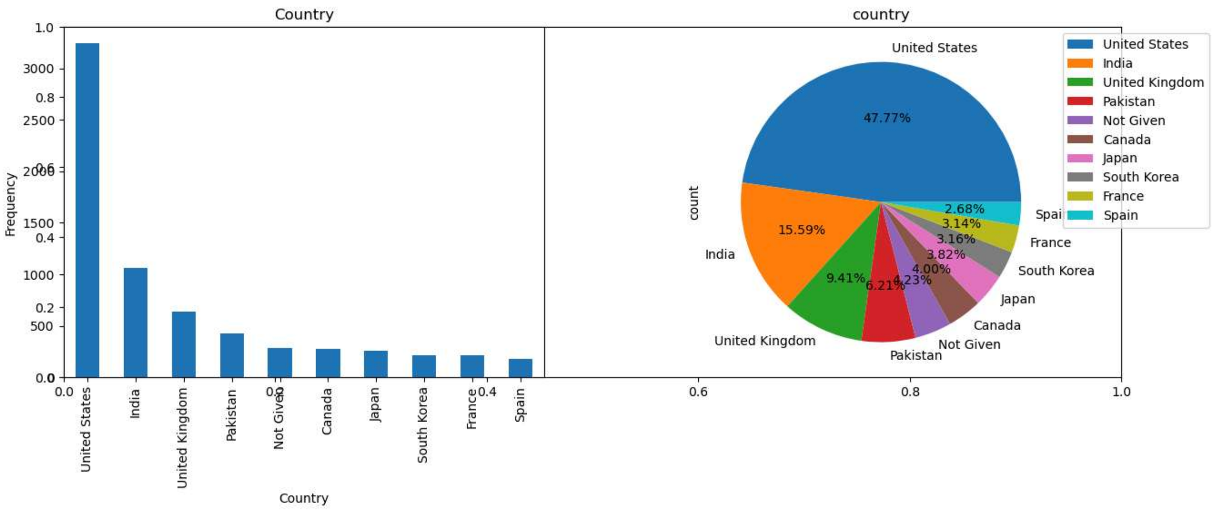
- TOP COUNTRIES WHOSE MOVIES ARE ADDED IN THE NETFLIX

In [111…

```
print('Top 10 countries to release movies and shows:\n',df['country'].value_counts(

#plotting the top 10 countries
plt.subplots(figsize=(15,5))
plt.subplot(121)
df['country'].value_counts()[:10].plot(kind='bar')
plt.xlabel('Country')
plt.ylabel('Frequency')
plt.title('Country')
plt.subplot(122)
df['country'].value_counts()[:10].plot(kind='pie',autopct='%.2f%%') # plotting the
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.title('country')
plt.show()
```

```
Top 10 countries to release movies and shows:
 country
United States     3240
India             1057
United Kingdom     638
Pakistan           421
Not Given          287
Canada             271
Japan              259
South Korea        214
France             213
Spain              182
Name: count, dtype: int64
```



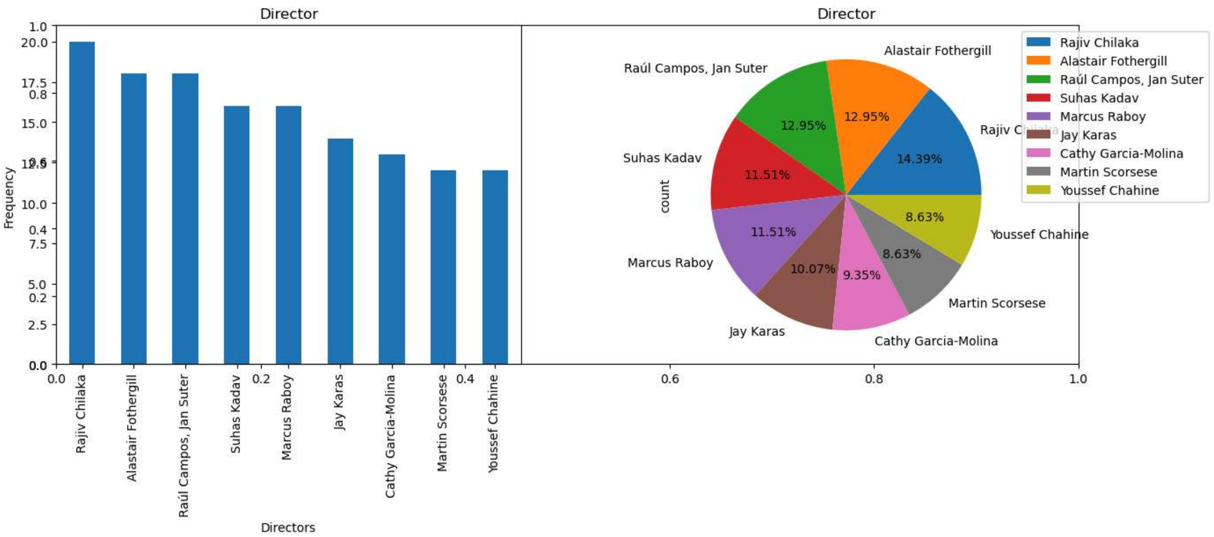
- TOP 10 DIRECTORS TO PRODUCDE MOVIES

In [111…

```
print('Top 10 directors to release movies on Netflix:\n',df['director'].value_count

# Top 10 directors whose movies are added // Excludd the movies and TV Shows whose
plt.subplots(figsize=(15,5))
plt.subplot(121)
df['director'].value_counts()[1:10].plot(kind='bar')
plt.xlabel('Directors')
plt.ylabel('Frequency')
plt.title('Director')
plt.subplot(122)
df['director'].value_counts()[1:10].plot(kind='pie',autopct='%.2f%%') # plotting th
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.title('Director')
plt.show()
```

```
Top 10 directors to release movies on Netflix:
 director
Rajiv Chilaka             20
Alastair Fothergill       18
Raúl Campos, Jan Suter    18
Suhas Kadav               16
Marcus Raboy              16
Jay Karas                 14
Cathy Garcia-Molina       13
Martin Scorsese           12
Youssef Chahine           12
Name: count, dtype: int64
```



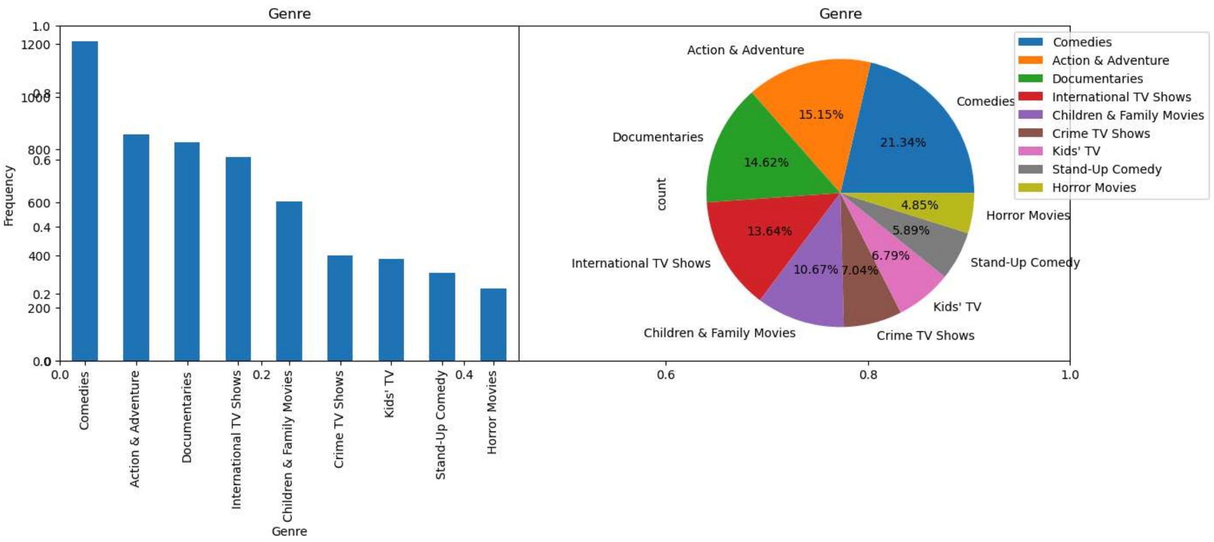
- PLOTTING THE TOP 10 GENRE WITH THEIR FREQUENCY

In [111…

```
df['genre']=df['listed_in'].str.split(',',expand=True)[0]
print('Top 10 genre: \n',df['genre'].value_counts()[0:10])

plt.subplots(figsize=(15,5))
plt.subplot(121)
df['genre'].value_counts()[1:10].plot(kind='bar')
plt.xlabel('Genre')
plt.ylabel('Frequency')
plt.title('Genre')
plt.subplot(122)
df['genre'].value_counts()[1:10].plot(kind='pie',autopct='%.2f%%') # plotting the t
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.title('Genre')
plt.show()
```

```
Top 10 genre: 
 genre
Dramas                      1599
Comedies                    1210
Action & Adventure           859
Documentaries                829
International TV Shows       773
Children & Family Movies     605
Crime TV Shows               399
Kids' TV                     385
Stand-Up Comedy              334
Horror Movies                275
Name: count, dtype: int64
```



- converting the date_added column to the datetime datatype

In [108…

```
df['date_added']=pd.to_datetime(df['date_added'])
df['added_year']=df['date_added'].dt.year # Extracting the year
df['added_month']=df['date_added'].dt.month # Extracting the month
df['added_month_name']=df['date_added'].dt.strftime('%b')
df['day_name']=df['date_added'].dt.day_name()  # Extracting the day of the week
df['day_is_weekened']=np.where(df['day_name'].isin(['Sunday','Saturday']),1,0)  # c
df.sample(5)
```

Out[108…

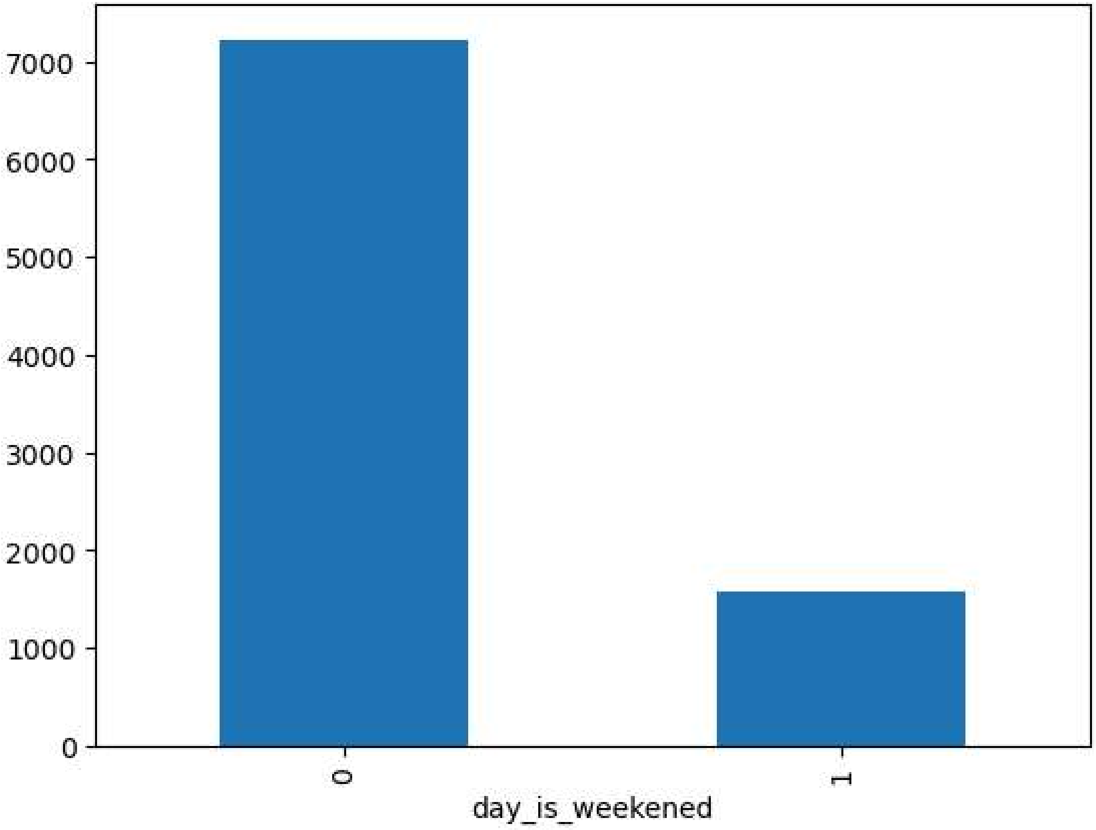
	show_id	type	title	director	country	date_added	release_year	rating	dura
7257	s2576	TV Show	WWII in HD	Not Given	United States	2020-05-02	2009	TV-14	1 Sea
4157	s5831	Movie	Rebirth	Karl Mueller	United States	2016-07-15	2016	TV-MA	101
8196	s5539	TV Show	The Get Down	Not Given	United States	2017-04-07	2017	TV-MA	Sea
6161	s8352	Movie	The Humanity Bureau	Rob W. King	Canada	2018-12-18	2017	R	94
7620	s3657	TV Show	Rookie Historian Goo Hae-Ryung	Not Given	South Korea	2019-07-18	2019	TV-14	1 Sea

- checking whether the relase date is a weekened or not?

In [108…

```
df['day_is_weekened'].value_counts().plot(kind='bar') # 1 means weekened whether 0
```

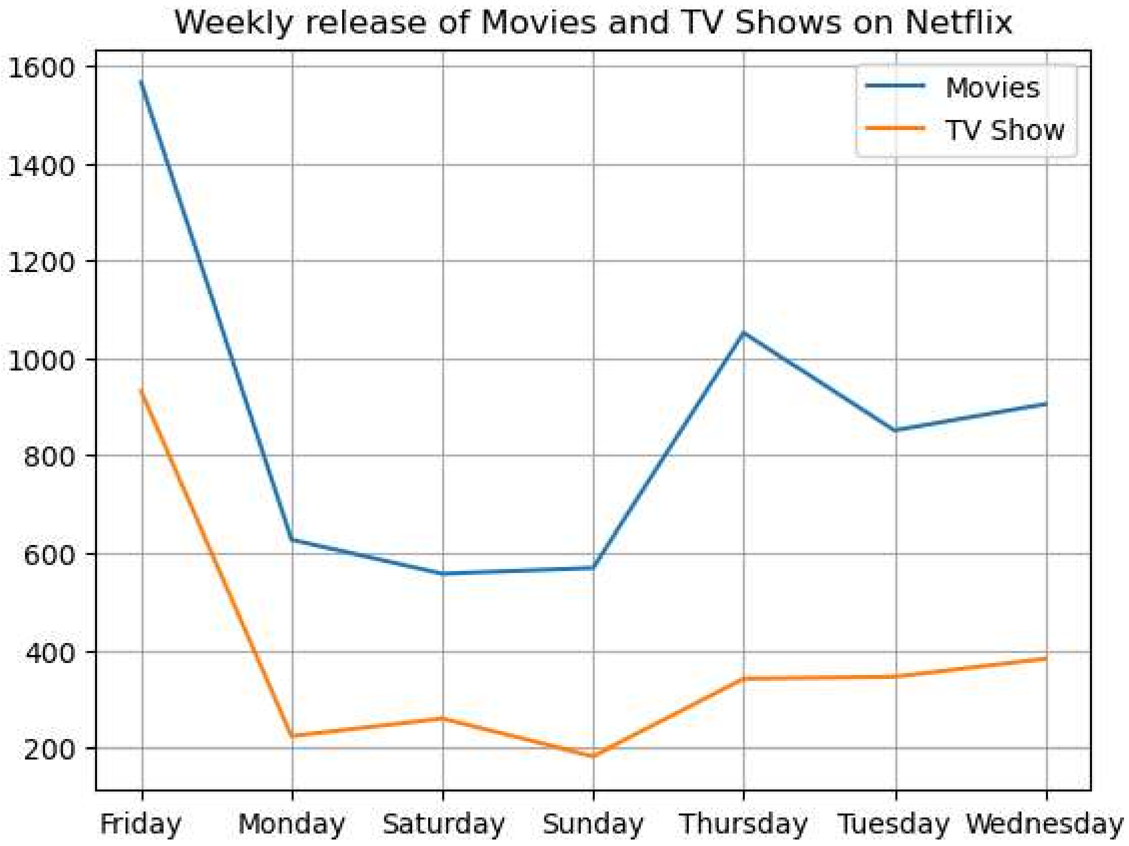
```
Out[108…  <Axes: xlabel='day_is_weekened'>
```



```
In [108…
#Weekly release of the Movies and TV Shows on Netflix
day_wise_movie_release=df[df['type']=='Movie']['day_name'].value_counts().sort_inde
day_wise_shows_release=df[df['type']=='TV Show']['day_name'].value_counts().sort_in
day_wise_movie_release,day_wise_shows_release

plt.plot(day_wise_movie_release.index,day_wise_movie_release.values,label='Movies')
plt.plot(day_wise_shows_release.index,day_wise_shows_release.values,label='TV Show'
plt.grid(True)
plt.legend()
plt.title('Weekly release of Movies and TV Shows on Netflix')
plt.show()

# The above observation shows that maximim movies and shows are releasd in the frid
```



- PLOTTING THE MOVIES AND TVSHOWS RELEASE WITH THE MONTHS

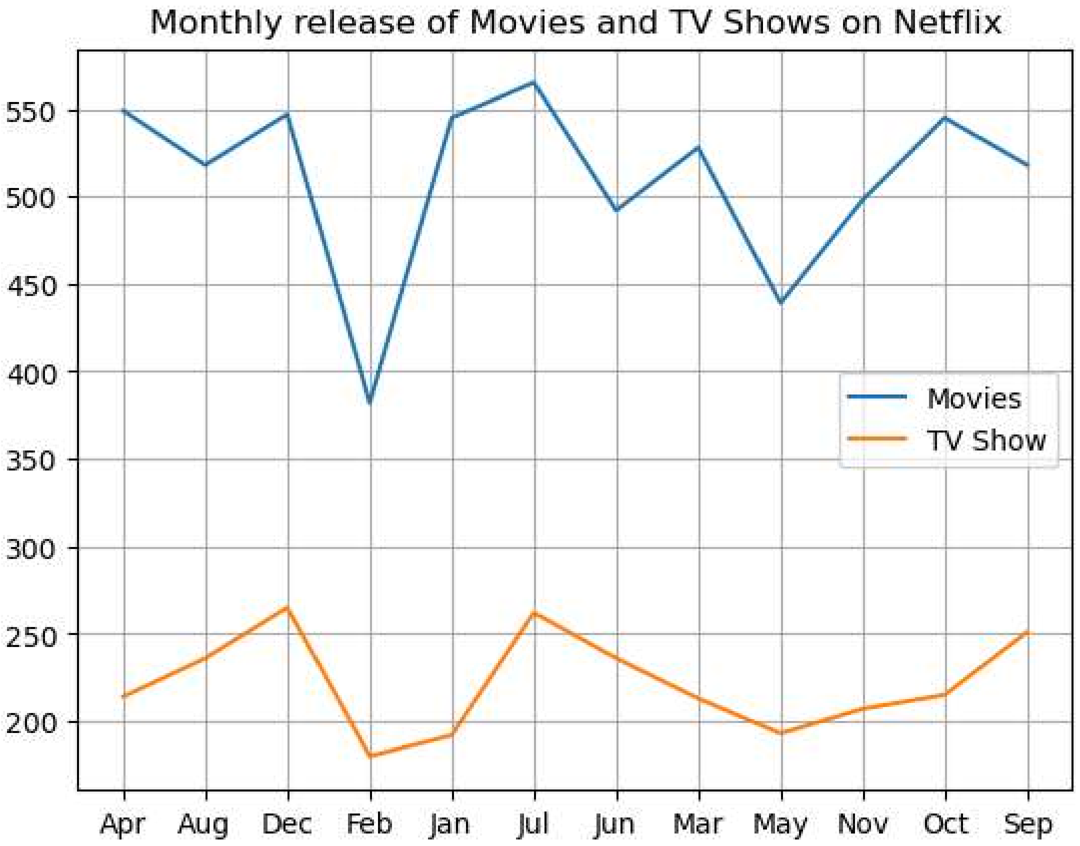
In [108…

```
# Monthly release of the movies and shows on the Netflix
monthly_movies_release=df[df['type']=='Movie']['added_month_name'].value_counts().s
monthly_shows_release=df[df['type']=='TV Show']['added_month_name'].value_counts().
monthly_movies_release,monthly_shows_release


plt.plot(monthly_movies_release.index,monthly_movies_release.values,label='Movies')
plt.plot(monthly_shows_release.index,monthly_shows_release.values,label='TV Show')
plt.grid(True)
plt.title('Monthly release of Movies and TV Shows on Netflix')
plt.legend()
```

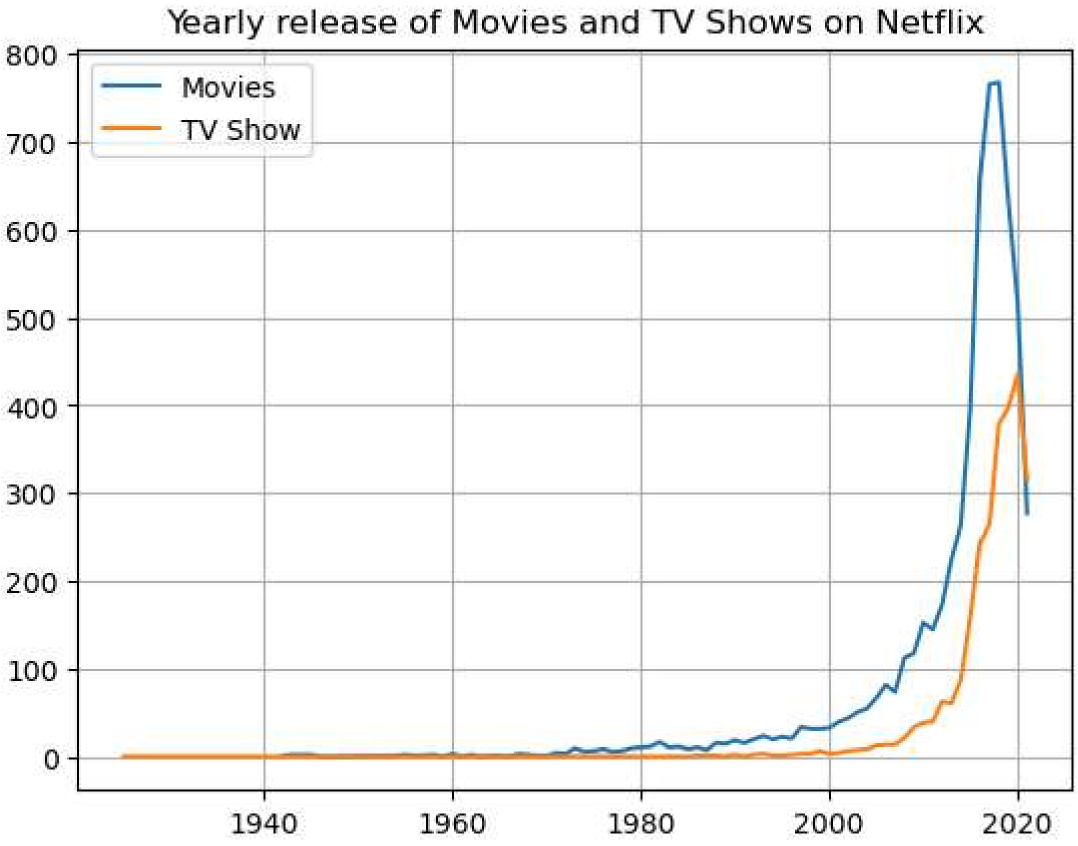
Out[108…

```
<matplotlib.legend.Legend at 0x280e6dd8c80>
```



In [109…

```
# Yearly release of the movies and shows on Netflix
yearly_movies_release=df[df['type']=='Movie']['release_year'].value_counts().sort_i
yearly_shows_release=df[df['type']=='TV Show']['release_year'].value_counts().sort_
yearly_movies_release,yearly_shows_release
plt.plot(yearly_movies_release.index,yearly_movies_release.values,label='Movies')
plt.plot(yearly_shows_release.index,yearly_shows_release.values,label='TV Show')
plt.grid(True)
plt.legend()
plt.title('Yearly release of Movies and TV Shows on Netflix')
plt.show()
```



- FINAL DATA WILL LOOK AS FOLLOWS:

In [109…

```
df.sample(5)
```

Out[109…

	show_id	type	title	director	country	date_added	release_year	rating	c
6372	s8602	Movie	Tokyo Idols	Kyoko Miyake	United Kingdom	2017-10-01	2017	TV-14	
7859	s4354	TV Show	Death by Magic	Not Given	United States	2018-11-30	2018	TV-PG	1
6241	s8449	Movie	The Peacemaker	Mimi Leder	United States	2020-01-01	1997	R	
1260	s1347	Movie	All My Friends Are Dead	Jan Belcl	Poland	2021-02-03	2020	TV-MA	
2655	s3488	Movie	The Grandmaster	Wong Kar Wai	Hong Kong	2019-09-26	2013	PG-13	

```
In [ ]:
```