

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

```
In [ ]: df = pd.read_csv("netflix_titles.csv")
```

```
In [ ]: nf = df.copy()
```

```
In [ ]: nf.shape
```

```
Out[ ]: (8807, 12)
```

```
In [ ]: nf.isna().sum()
```

```
Out[ ]:
```

	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

```
In [ ]: nf.dropna(subset=["date_added", "rating", 'duration'], inplace=True)
```

```
In [ ]: nf.shape
```

```
Out[ ]: (8790, 12)
```

```
In [ ]: nf.nunique()
```

```
Out[ ]:
```

	0
show_id	8790
type	2
title	8790
director	4526
cast	7678
country	748
date_added	1765
release_year	74
rating	14
duration	220
listed_in	513
description	8758

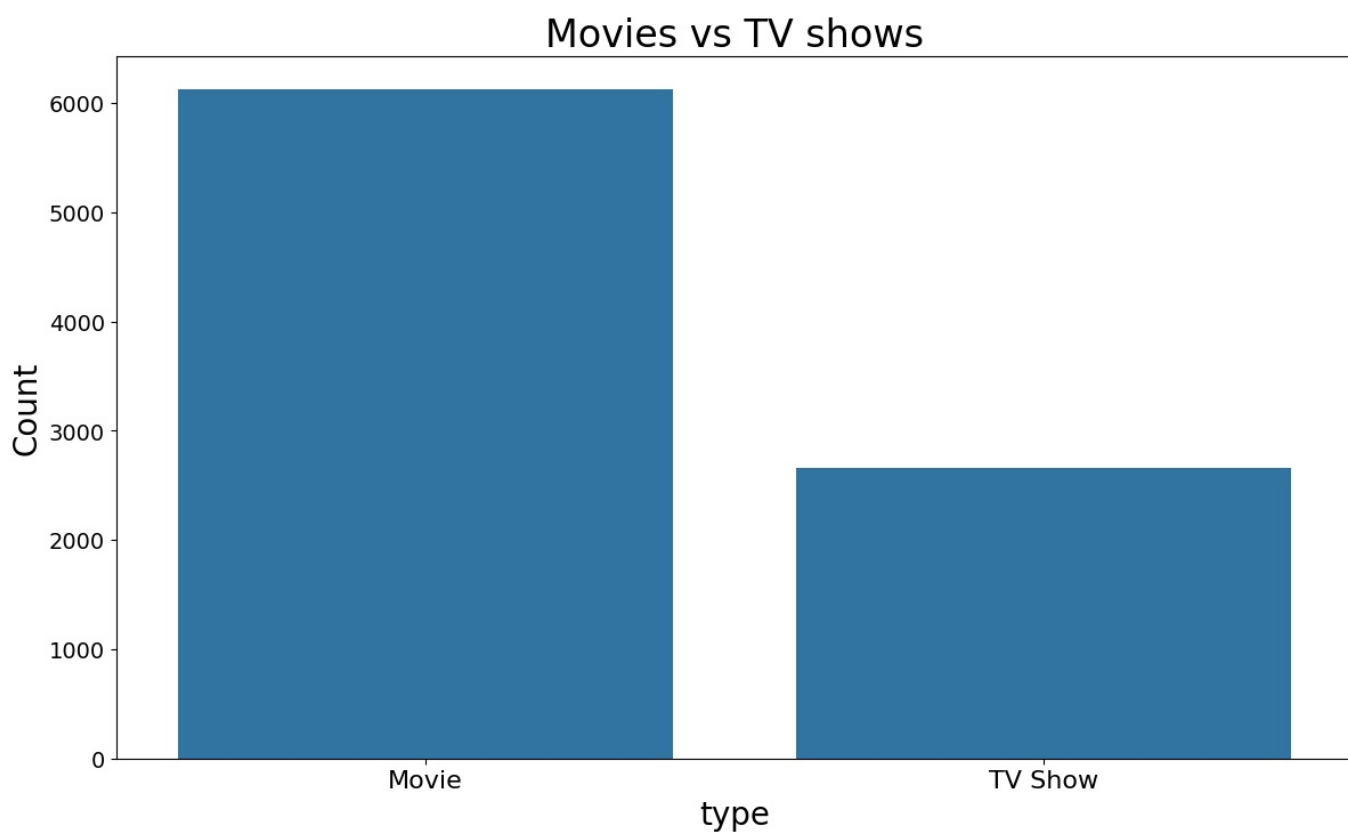
dtype: int64

```
In [ ]: nf["type"].value_counts()
```

```
Out[ ]:      count
      type
Movie    6126
TV Show   2664
```

dtype: int64

```
In [ ]: plt.figure(figsize=(14,8))
sns.countplot(x=nf['type'])
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("type", fontsize=20)
plt.ylabel("Count", fontsize=20)
plt.title("Movies vs TV shows", fontsize=24)
plt.show()
```



There is a huge difference between the number of movies and Tv shows that releases on Netflix platform.we can say that,in past Netflix mostly concentrated on the movies than Tv shows.As in todays world,there is a huge audience for Tv shows.Netflix should produce more TV shows to attract viewers.

```
In [ ]: nf.isna().sum()
```

```
Out[ ]:
```

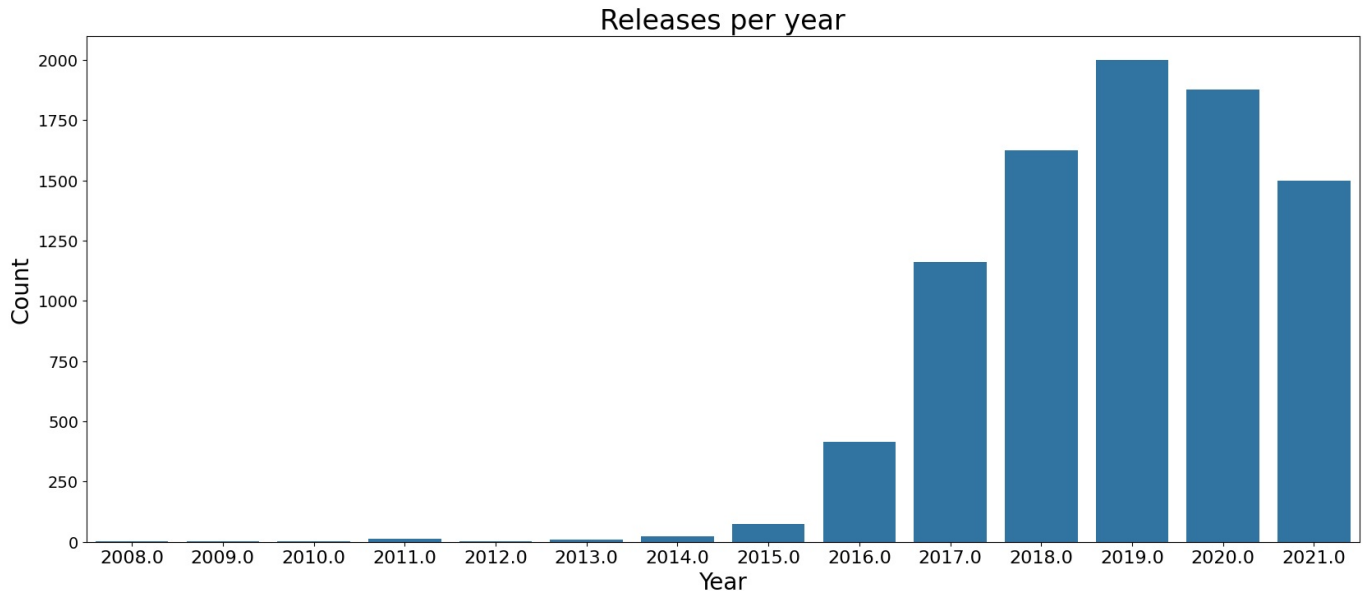
show_id	0
type	0
title	0
director	2621
cast	825
country	829
date_added	0
release_year	0
rating	0
duration	0
listed_in	0
description	0

dtype: int64

```
In [ ]: # nf["date_added"]=pd.to_datetime(nf["date_added"])
nf["date_added"] = pd.to_datetime(nf["date_added"], format='%B %d, %Y', errors='coerce')
```

```
In [ ]: nf["year"]=nf["date_added"].dt.year
nf["month"]=nf["date_added"].dt.month_name()
nf["weekday"]=nf["date_added"].dt.day_name()
```

```
In [ ]: plt.figure(figsize=(20,8))
sns.countplot(data=nf,x="year")
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Year",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.title("Releases per year",fontsize=24)
plt.show()
```

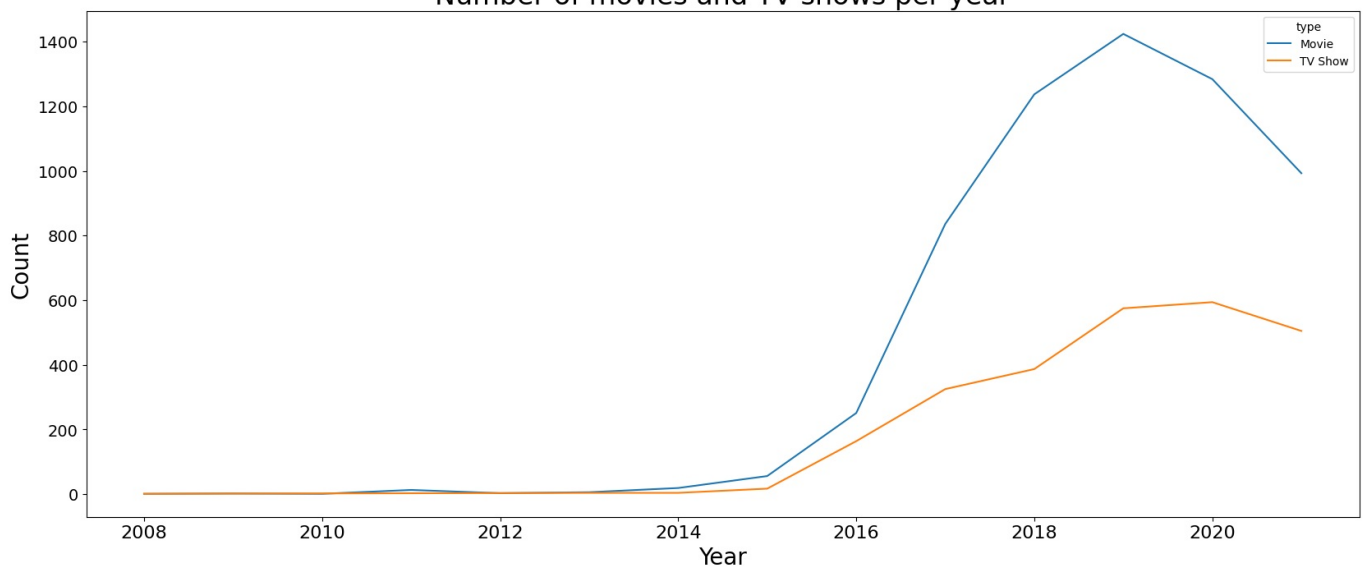


The number of releases on Netflix platform per year had increased dramatically from 2015 attaining the highest number in 2019 and later dropped slowly. To sustain its existing subscribers and attract new ones, Netflix has to release more shows without any drop further.

```
In [ ]: year_count=nf.groupby(["year","type"]).size().reset_index()
```

```
In [ ]: plt.figure(figsize=(20,8))
sns.lineplot(x=year_count["year"],y=year_count[0],hue=year_count["type"])
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Year",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.title("Number of movies and TV shows per year",fontsize=24)
plt.show()
```

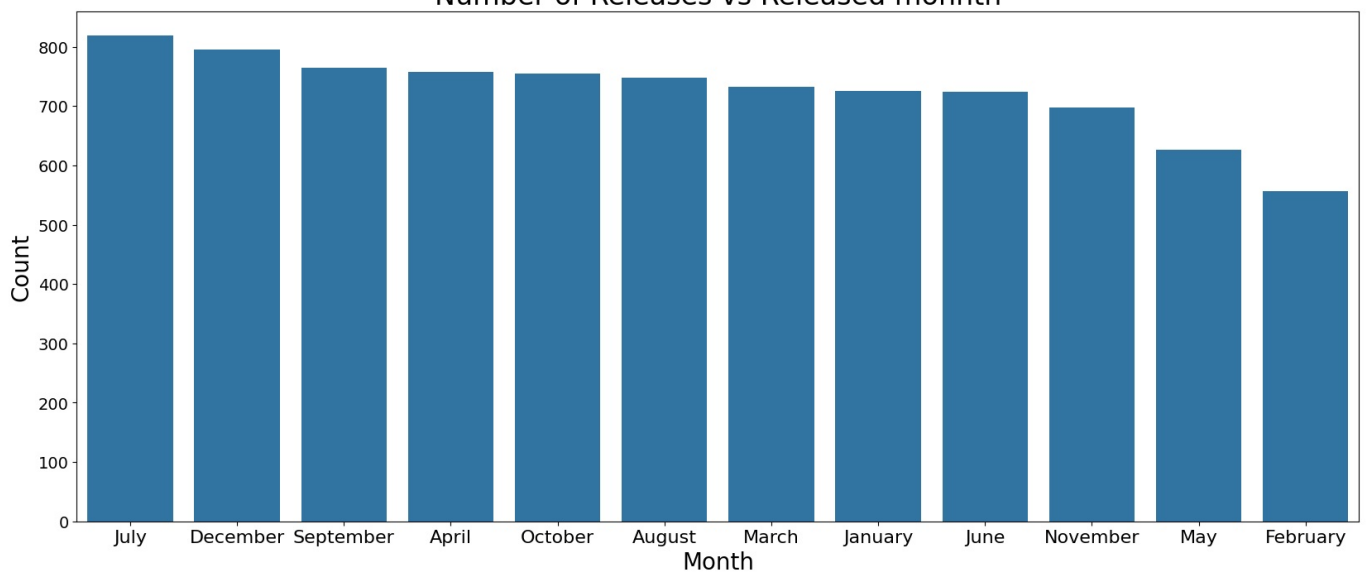
### Number of movies and TV shows per year



The growth in number of movies released per year is higher than Tv shows released per year. Even though the growth in both types started in 2015, movies count has risen far more than Tv shows. Also recently Netflix has definitely increased their focus on tv shows but still lags behind in comparison to movies.

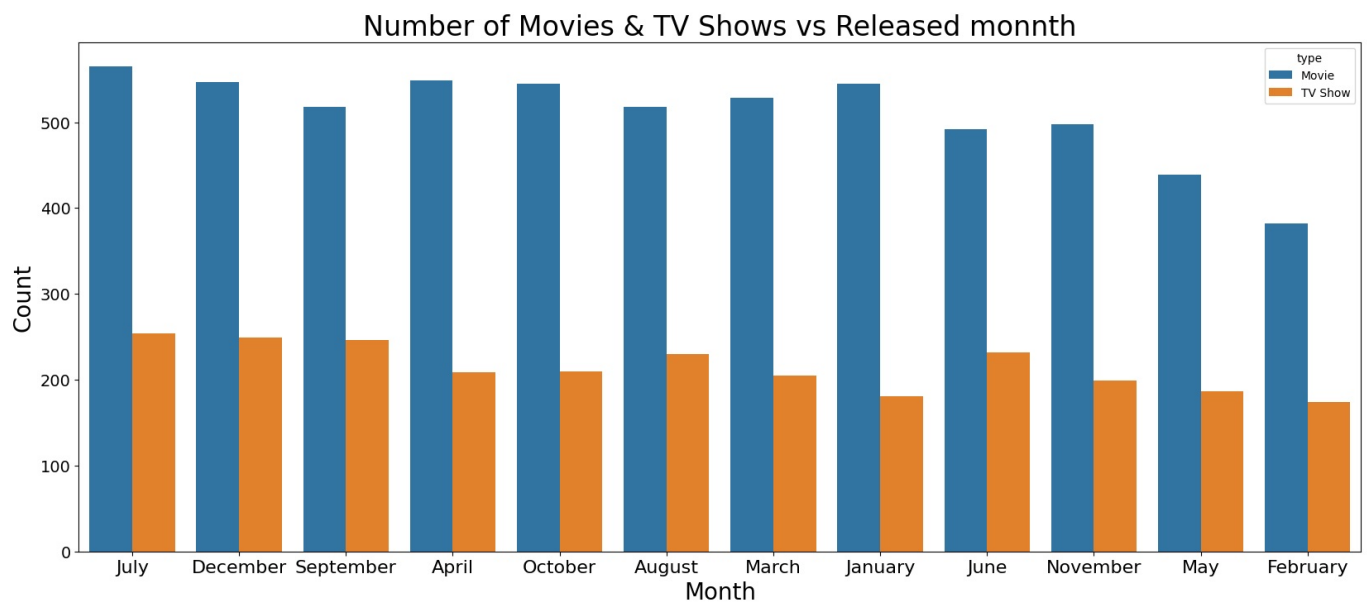
```
In [ ]: plt.figure(figsize=(20,8))
sns.countplot(data=nf,x="month",order=nf["month"].value_counts().index)
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Month",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.title("Number of Releases vs Released month",fontsize=24)
plt.show()
```

### Number of Releases vs Released month

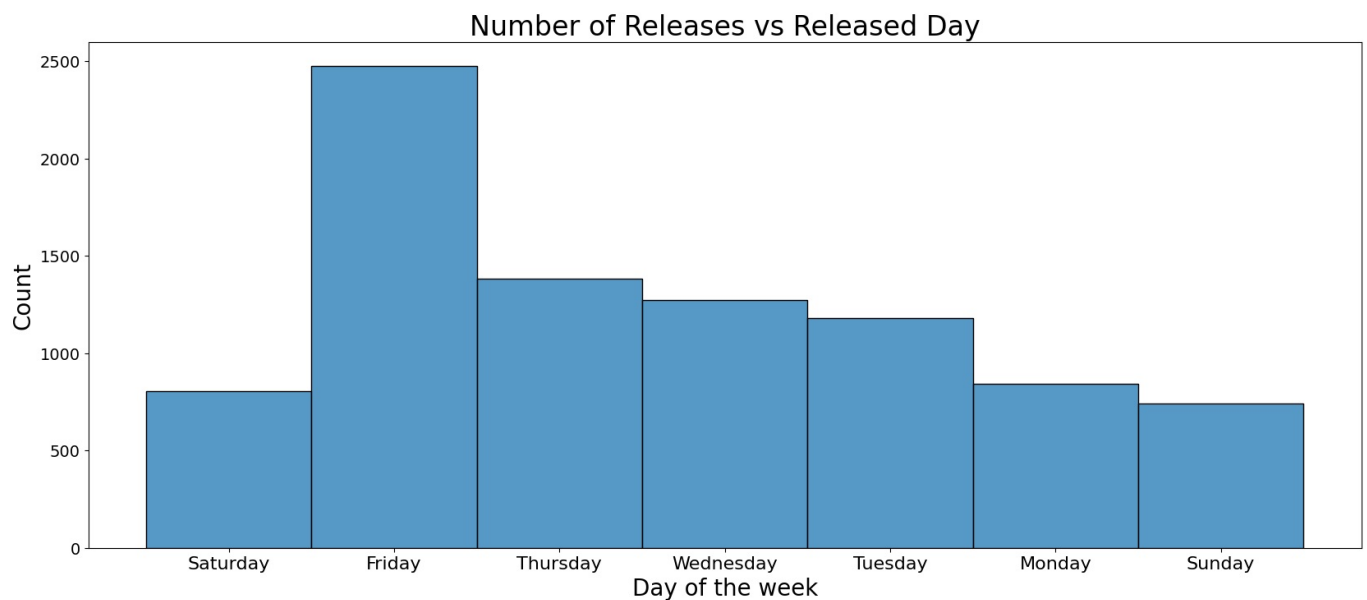


Netflix releases its content every month. According to the data, July is the month that has most number of releases and February is the month that has least. Large amount of Netflix's content is released in US & UK, July is the month where summer holidays start and December is next big holiday season in these countries. So releasing in these months would be beneficial for the viewership of their content.

```
In [ ]: plt.figure(figsize=(20,8))
sns.countplot(data=nf,x="month",order=nf["month"].value_counts().index,hue="type")
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Month",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.title("Number of Movies & TV Shows vs Released month",fontsize=24)
plt.show()
```

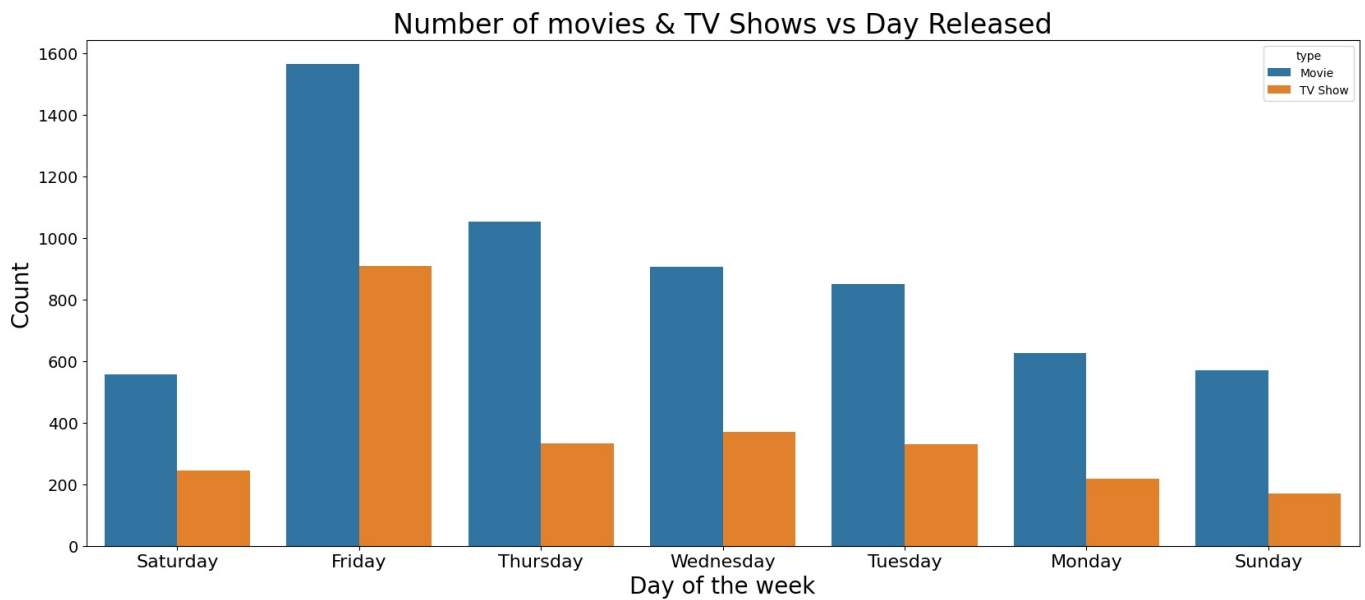


```
In [ ]: plt.figure(figsize=(20,8))
sns.histplot(data=nf,x="weekday")
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Day of the week",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.title("Number of Releases vs Released Day",fontsize=24)
plt.show()
```



People work on the weekdays and stream on the weekends. So, making the content available for them by weekends is beneficial. Netflix has already applied this and most number of releases are on Friday, so that subscribers can stream on Saturday and Sunday. Netflix released most of its content in this pattern and the least being on weekends.

```
In [ ]: plt.figure(figsize=(20,8))
sns.countplot(data=nf,x="weekday",hue="type")
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Day of the week",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.title("Number of movies & TV Shows vs Day Released",fontsize=24)
plt.show()
```



```
In [ ]: g = nf["release_year"].value_counts()
g.head()
```

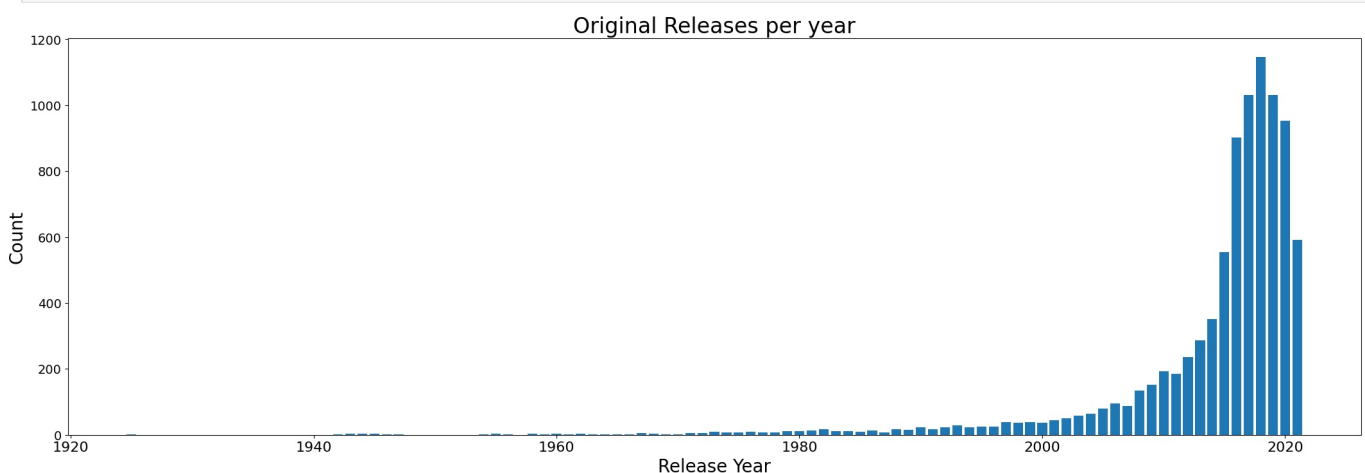
```
Out[ ]:      count
release_year
2018      1146
2017      1030
2019      1030
2020       953
2016       901
```

**dtype:** int64

```
In [ ]: g.index
```

```
Out[ ]: Index([2018, 2017, 2019, 2020, 2016, 2021, 2015, 2014, 2013, 2012, 2010, 2011,
2009, 2008, 2006, 2007, 2005, 2004, 2003, 2002, 2001, 1999, 1997, 2000,
1998, 1993, 1995, 1996, 1992, 1994, 1990, 1988, 1982, 1991, 1989, 1981,
1986, 1984, 1980, 1979, 1983, 1985, 1973, 1976, 1987, 1977, 1974, 1975,
1978, 1972, 1967, 1971, 1960, 1945, 1944, 1955, 1968, 1962, 1943, 1958,
1964, 1969, 1970, 1963, 1956, 1965, 1946, 1942, 1954, 1959, 1925, 1961,
1947, 1966],
dtype='int64', name='release_year')
```

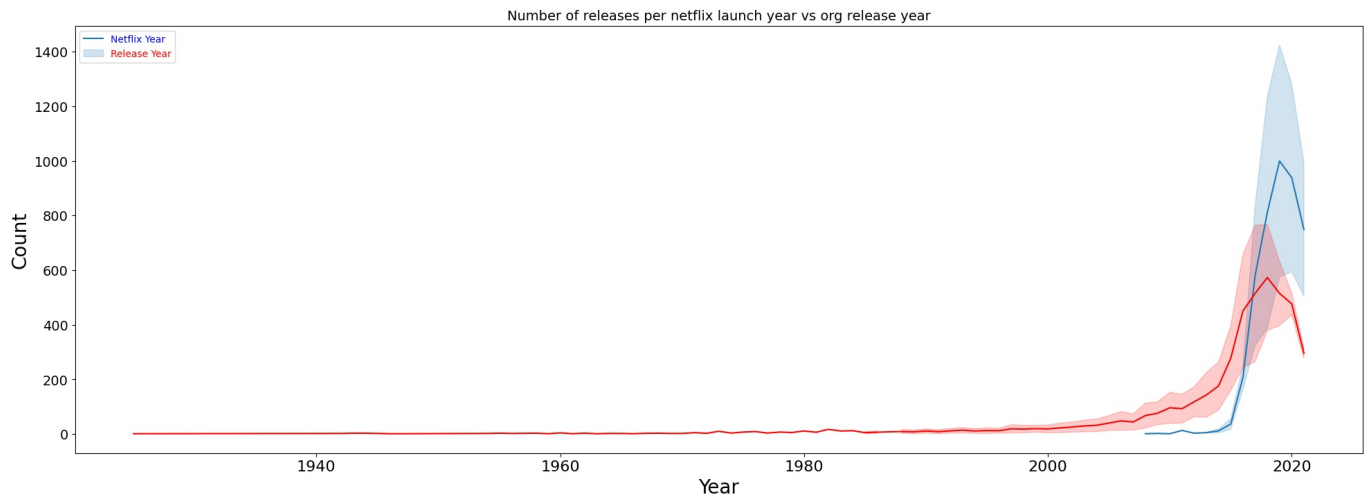
```
In [ ]: plt.figure(figsize=(26,8))
plt.bar(g.index,g)
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Release Year", fontsize=20)
plt.ylabel("Count", fontsize=20)
plt.title("Original Releases per year", fontsize=24)
plt.show()
```



We can see that, in the last 20-30 years the number of releases per year have increased dramatically. This data includes both netflix direct releases and theatrical releases.

```
In [ ]: ryear_count=nf.groupby(["release_year", "type"]).size().reset_index()
```

```
In [ ]: plt.figure(figsize=(24,8))
sns.lineplot(x=year_count["year"],y=year_count[0])
sns.lineplot(x=ryear_count["release_year"], y=ryear_count[0],color="r")
plt.legend(["Netflix Year","Release Year"], loc="upper left", labelcolor=["blue","red"])
plt.title("Number of releases per netflix launch year vs org release year", fontsize=14)
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Year", fontsize=20)
plt.ylabel("Count", fontsize=20)
plt.show()
```



According to the above graph, we can infer that Netflix is releasing old movies/tv shows which since 1940's in 2000's. Most of its content started streaming from 2005 but their original release years go back to 1940's. It should start producing newer content and direct stream them to attract more subscribers.

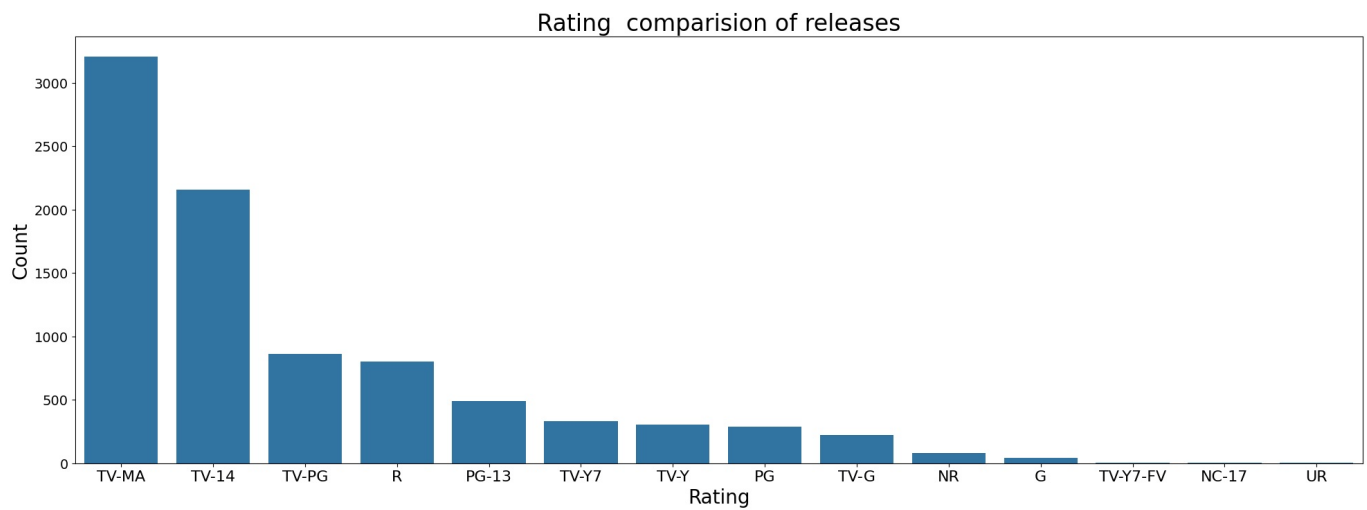
```
In [ ]: nf["rating"].value_counts()
```

```
Out[ ]:
```

rating	count
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
G	41
TV-Y7-FV	6
NC-17	3
UR	3

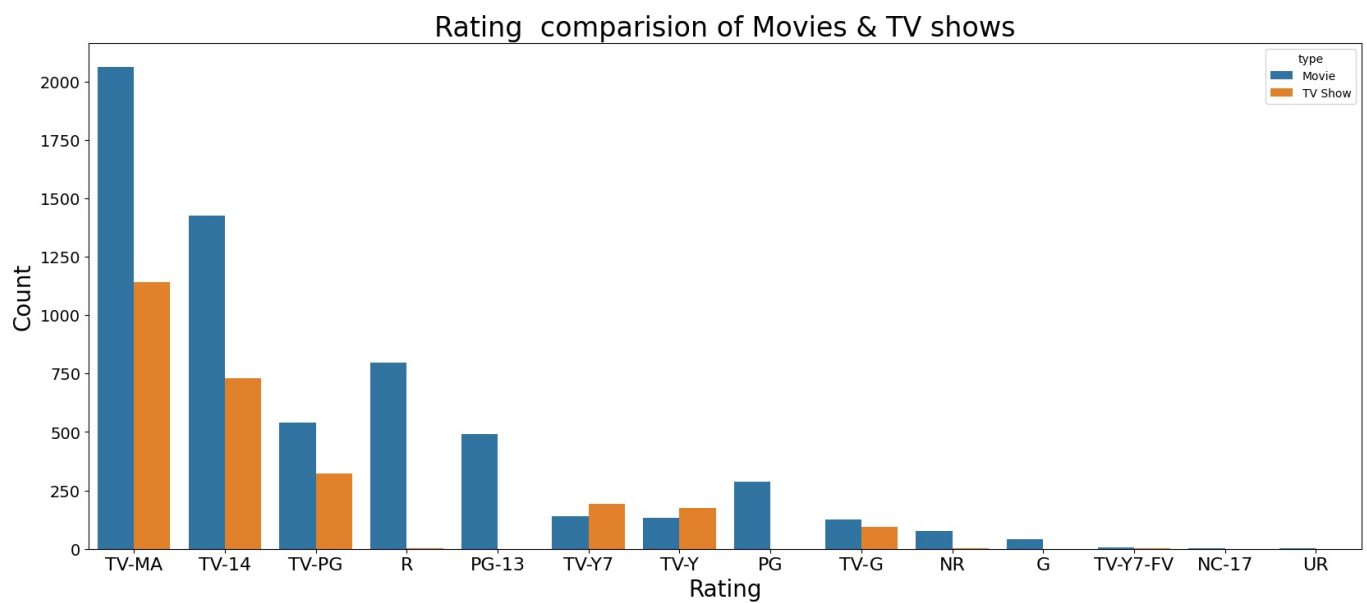
dtype: int64

```
In [ ]: plt.figure(figsize=(24,8))
sns.countplot(data=nf,x="rating",order=nf["rating"].value_counts().index)
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Rating", fontsize=20)
plt.ylabel("Count", fontsize=20)
plt.title("Rating comparison of releases", fontsize=24)
plt.show()
```



As we can see most of the content of Netflix is not suitable for children. whereas children are those audience who have free time to stream, but there isn't much content for them to watch. If Netflix produces content suited for younger audience too, it would boost the contents views.

```
In [ ]: plt.figure(figsize=(20,8))
sns.countplot(data=nf,x="rating",order=nf["rating"].value_counts().index,hue="type")
plt.xticks(fontsize=16)
plt.yticks(fontsize=14)
plt.xlabel("Rating",fontsize=20)
plt.ylabel("Count",fontsize=20)
plt.title("Rating comparison of Movies & TV shows",fontsize=24)
plt.show()
```



```
In [ ]: nf_country = nf.copy()
```

```
In [ ]: nf_country['country']=nf_country['country'].str.split(',')
```

```
In [ ]: nf_country = nf_country.explode('country')
```

```
In [ ]: nf_country['country'] = nf_country['country'].str.lstrip()
```

```
In [ ]: nf_country.shape
```

```
Out[ ]: (10833, 15)
```

```
In [ ]: nf_country.isna().sum()
```



Out[ ]:

	0
show_id	0
type	0
title	0
director	2957
cast	1006
country	829
date_added	101
release_year	0
rating	0
duration	0
listed_in	0
description	0
year	101
month	101
weekday	101

dtype: int64

In [ ]:

nf\_country.dropna(subset=['country'],inplace=True)

In [ ]:

nf\_country.isna().sum()

Out[ ]:

	0
show_id	0
type	0
title	0
director	2550
cast	852
country	0
date_added	98
release_year	0
rating	0
duration	0
listed_in	0
description	0
year	98
month	98
weekday	98

dtype: int64

In [ ]:

nf\_country.drop(index=nf\_country.loc[nf\_country['country']==""].index,inplace=True)

In [ ]:

nf\_country["country"].value\_counts()

Out [ ]:

	count
country	
United States	3680
India	1046
United Kingdom	803
Canada	445
France	391
...	...
Bermuda	1
Ecuador	1
Armenia	1
Mongolia	1
Montenegro	1

122 rows × 1 columns

dtype: int64

```
In [ ]: nf_country_grp = nf_country_grp=nf_country.groupby('country')[["title"]].agg("count")

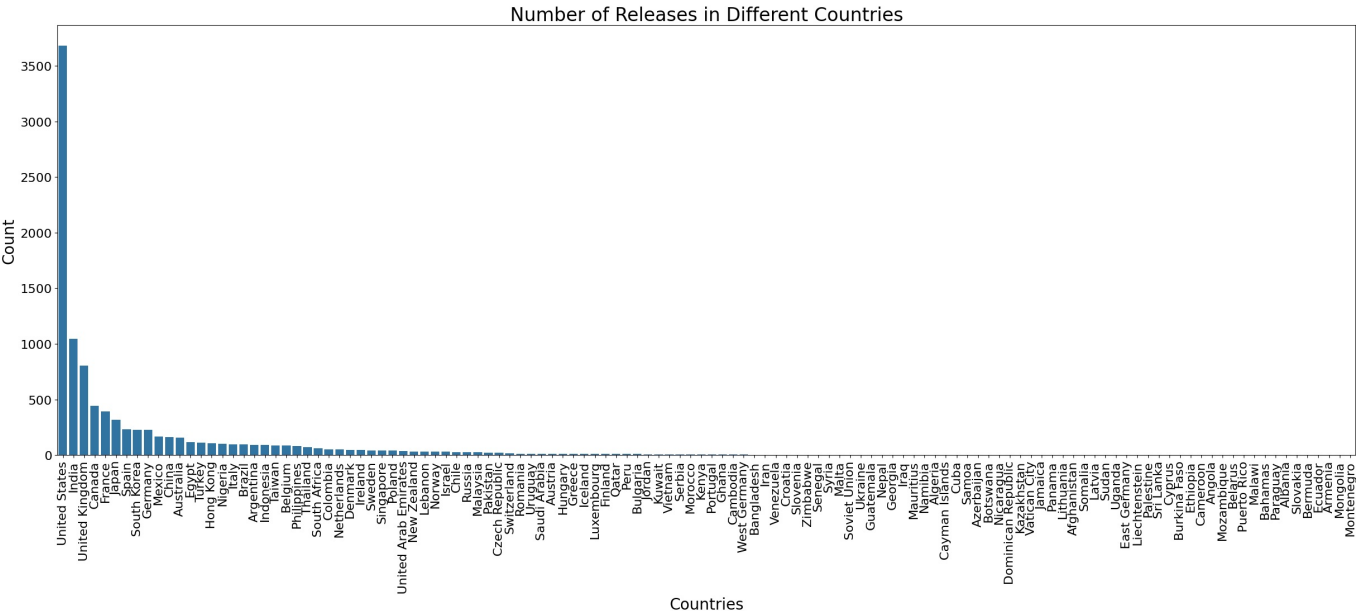
In [ ]: print(nf_country_grp)
```

country	title
Afghanistan	1
Albania	1
Algeria	2
Angola	1
Argentina	91
...	...
Vatican City	1
Venezuela	4
Vietnam	7
West Germany	5
Zimbabwe	3

[122 rows x 1 columns]

```
In [ ]: top_countries_list=nf_country_grp.index
top_countries=nf_country.loc[nf_country["country"].isin(top_countries_list)]

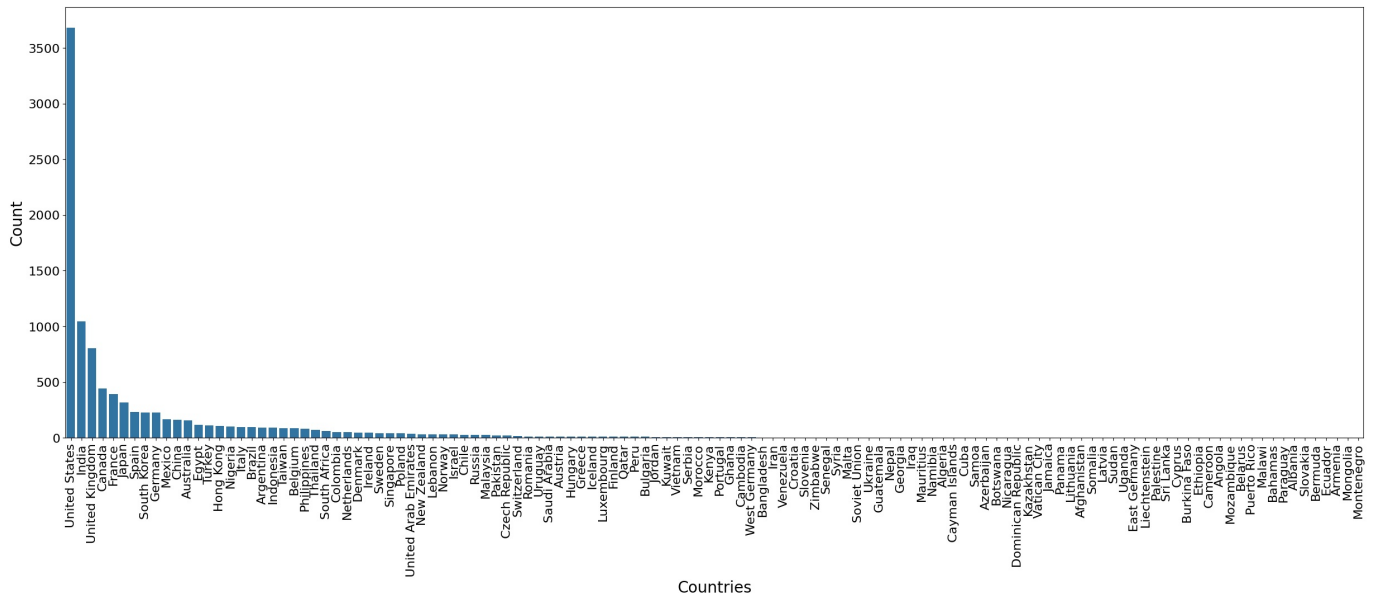
In [ ]: plt.figure(figsize=(30,10))
sns.countplot(data=top_countries,x="country",order=top_countries["country"].value_counts().index)
plt.xticks(rotation=90, fontsize=16)
plt.xlabel("Countries", fontsize=20)
plt.ylabel("Count", fontsize=20)
plt.title("Number of Releases in Different Countries", fontsize=24)
plt.yticks(fontsize=16)
plt.show()
```



Majority of the Netflix's content is available in United states and then in india.Us,Uk,India are the only countries that has more than 500 releases. All the other countries have really low number of releases. To increase their market in these countries,Netflix can try releasing the hit movies/TV shows from Us(as it has most releases)in these countries.This way Netflix can increase their content in other countries too and the contetnt already being a hit would mostly gain popularity for the streaming platform.

```
In [ ]: plt.figure(figsize=(30,10))
sns.countplot(data=top_countries,x="country",order=top_countries["country"].value_counts().index)
plt.xticks(rotation=90, fontsize=16)
plt.xlabel("Countries", fontsize=20)
plt.yticks(fontsize=16)
plt.ylabel("Count", fontsize=20)
```

```
Out[ ]: Text(0, 0.5, 'Count')
```



```
In [ ]: nf_cast=nf.copy()
nf_cast.shape
```

```
Out[ ]: (8790, 15)
```

```
In [ ]: nf_cast["cast"]=nf_cast["cast"].str.split(",")
```

```
In [ ]: nf_cast=nf_cast.explode("cast")
```

```
In [ ]: nf_cast.shape
```

```
Out[ ]: (64841, 15)
```

```
In [ ]: nf_cast.isna().sum()
```

Out[ ]:

	0
show_id	0
type	0
title	0
director	18913
cast	825
country	5050
date_added	621
release_year	0
rating	0
duration	0
listed_in	0
description	0
year	621
month	621
weekday	621

dtype: int64

```
In [ ]: nf_cast.dropna(subset=["cast", "country"], inplace=True)
```

```
In [ ]: nf_cast.isna().sum()
```

Out[ ]:

	0
show_id	0
type	0
title	0
director	16409
cast	0
country	0
date_added	605
release_year	0
rating	0
duration	0
listed_in	0
description	0
year	605
month	605
weekday	605

dtype: int64

```
cast_tv=nf_cast.loc[nf_cast["type"]=="TV Show"]
```

```
In [ ]: cast_tv=nf_cast.loc[nf_cast["type"]=="TV Show"]
cast_tv.shape
```

Out[ ]: (17303, 15)

```
In [ ]: cast_movies=nf_cast.loc[nf_cast["type"]=="Movie"]
cast_movies.shape
```

Out[ ]: (41817, 15)

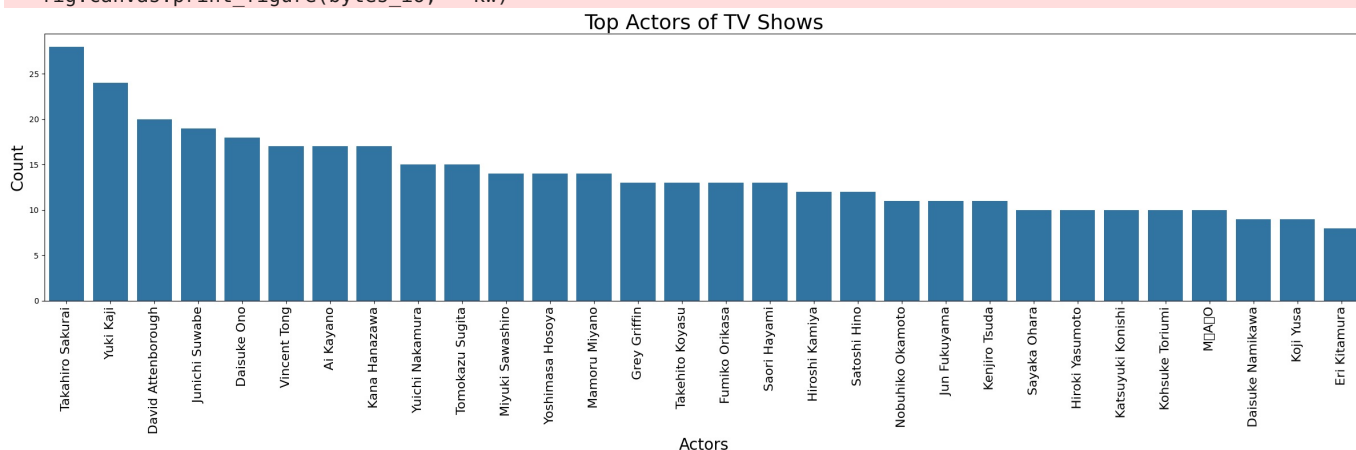
```
In [ ]: top_cast_tv = cast_tv.groupby("cast")[["title"]].agg("count").nlargest(30, "title")
top_cast_all = nf_cast.groupby("cast")[["title"]].agg("count").nlargest(30, "title")
top_cast_movies = cast_movies.groupby("cast")[["title"]].agg("count").nlargest(30, "title")
```

```
In [ ]: top_actors_all = nf_cast[nf_cast["cast"].isin(top_cast_all.index)]
top_cast_tv = nf_cast[nf_cast["cast"].isin(top_cast_tv.index)]
```

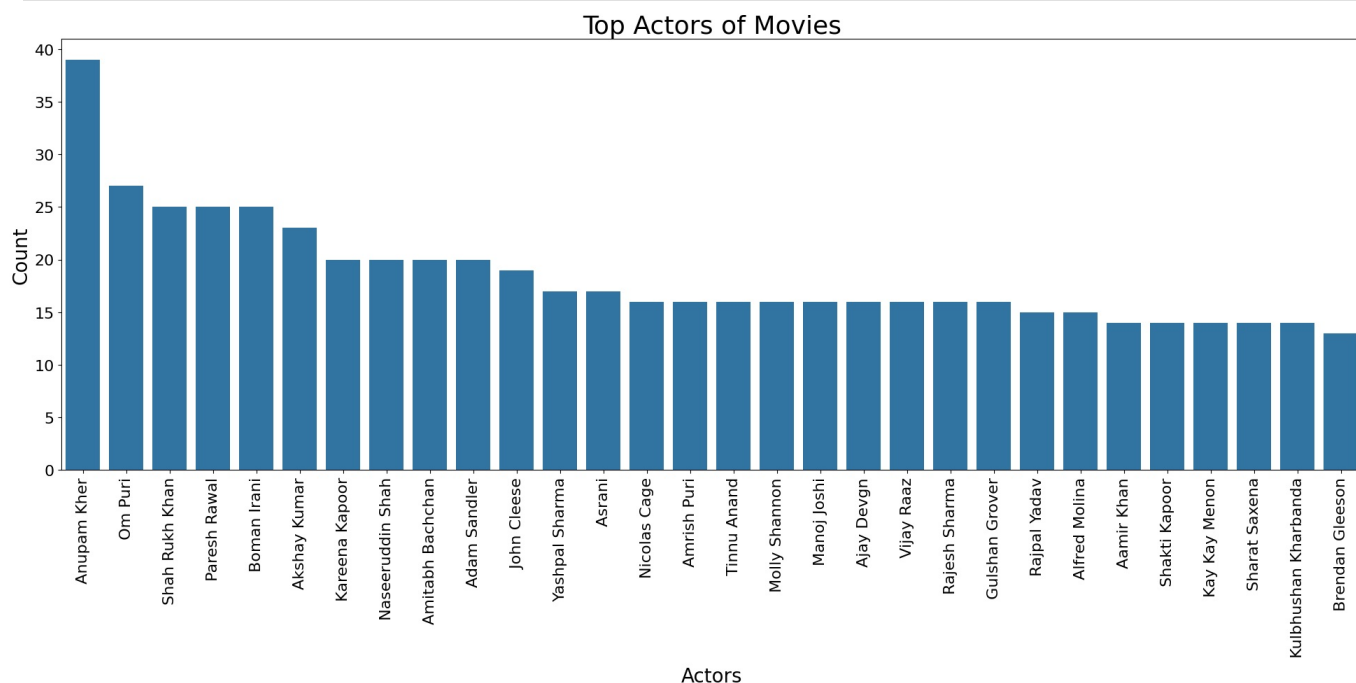
```
top_actors_movies = nf_cast[nf_cast["cast"].isin(top_cast_movies.index)]
```

```
In [ ]: plt.figure(figsize=(24,8))
sns.countplot(data=top_cast_tv,x="cast",order=top_cast_tv['cast'].value_counts().index)
plt.xticks(rotation=90, fontsize=16)
plt.xlabel("Actors", fontsize=20)
plt.ylabel("Count", fontsize=20)
plt.title("Top Actors of TV Shows", fontsize=26)
plt.tight_layout()
plt.show()
```

```
<ipython-input-77-df1cf27f7368>:7: UserWarning: Glyph 12539 (\N{KATAKANA MIDDLE DOT}) missing from current font.
plt.tight_layout()
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 12539 (\N{KATAKANA MI
DDLE DOT}) missing from current font.
fig.canvas.print_figure(bytes_io, **kw)
```



```
In [ ]: plt.figure(figsize=(24,8))
sns.countplot(data=top_actors_movies,x="cast",order=top_actors_movies['cast'].value_counts().index)
plt.xticks(rotation=90, fontsize=16)
plt.xlabel("Actors", fontsize=20)
plt.yticks(fontsize=16)
plt.ylabel("Count", fontsize=20)
plt.title("Top Actors of Movies", fontsize=26)
plt.show()
```



Actors with most number of releases are mostly from movies. There are very few actors that are in TV shows in the above graph. Netflix can try producing Tv shows with these top actors, these top actors popularity might increase the views of the Tv shows

```
In [ ]: nf_genre=nf.copy()
```

```
In [ ]: nf_genre.shape
```

```
Out[ ]: (8790, 15)
```

```
In [ ]: nf_genre["listed_in"]=nf_genre["listed_in"].str.split(",")
```

In [ ]:

nf\_genre.head()

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	[Documentaries]	As he father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	[International TV Shows, TV Dramas, TV Myste...	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	[Crime TV Shows, International TV Shows, TV ...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	1 Season	[Docuseries, Reality TV]	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	[International TV Shows, Romantic TV Shows, ...	In a city of coaching centers, known to train I...

In [ ]:

nf\_genre=nf\_genre.explode("listed\_in")

In [ ]:

nf\_genre.head()

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows	To protect his family from a powerful drug lor...

In [ ]:

nf\_genre.shape

Out[ ]:

(19294,15)

```
In [ ]: nf_genre["listed_in"].value_counts()
```

```
Out[ ]:
```

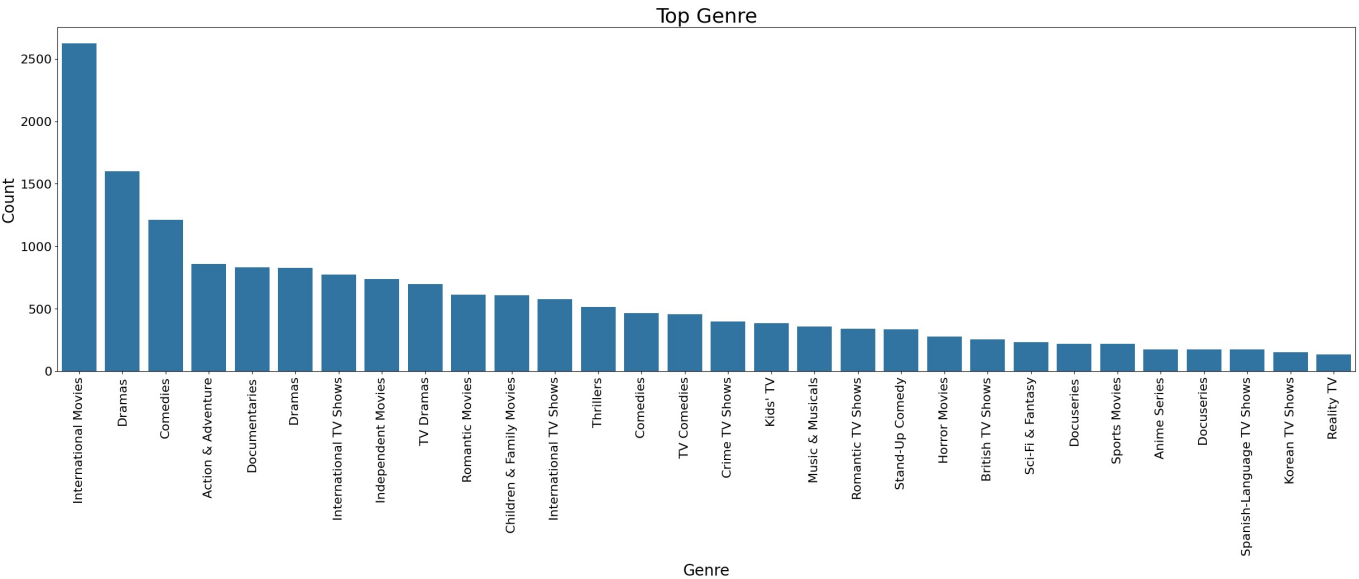
	count
listed_in	
International Movies	2624
Dramas	1599
Comedies	1210
Action & Adventure	859
Documentaries	829
...	...
Romantic Movies	3
Spanish-Language TV Shows	2
LGBTQ Movies	1
TV Sci-Fi & Fantasy	1
Sports Movies	1

73 rows × 1 columns

dtype: int64

```
In [ ]: top_genre_list=nf_genre.groupby(["listed_in"]).size().nlargest(30).index
top_genre=nf_genre.loc[nf_genre["listed_in"].isin(top_genre_list)]
```

```
In [ ]: plt.figure(figsize=(30,8))
sns.countplot(x=top_genre["listed_in"],order=top_genre["listed_in"].value_counts().index)
plt.xticks(rotation=90, fontsize=16)
plt.xlabel("Genre", fontsize=20)
plt.yticks(fontsize=16)
plt.ylabel("Count", fontsize=20)
plt.title("Top Genre", fontsize=26)
plt.show()
```



International Movies takes top among the Genre.

```
In [ ]: nf["director"].value_counts()
```

Out[ ]:

director		count
Rajiv Chilaka		19
Raúl Campos, Jan Suter		18
Marcus Raboy		16
Suhas Kadav		16
Jay Karas		14
...		...
Raymie Muzquiz, Stu Livingston		1
Joe Menendez		1
Eric Bross		1
Will Eisenberg		1
Mozez Singh		1

4526 rows × 1 columns

dtype: int64

In [ ]:

nf\_dir=nf.copy()

In [ ]:

nf\_dir.shape

Out[ ]:

(8790, 15)

In [ ]:

nf\_dir.isna().sum()

Out[ ]:

	0
show_id	0
type	0
title	0
director	2621
cast	825
country	829
date_added	88
release_year	0
rating	0
duration	0
listed_in	0
description	0
year	88
month	88
weekday	88

dtype: int64

In [ ]:

nf\_dir.dropna(subset="director",inplace=True)

In [ ]:

nf\_dir.isna().sum()



Out[ ]:

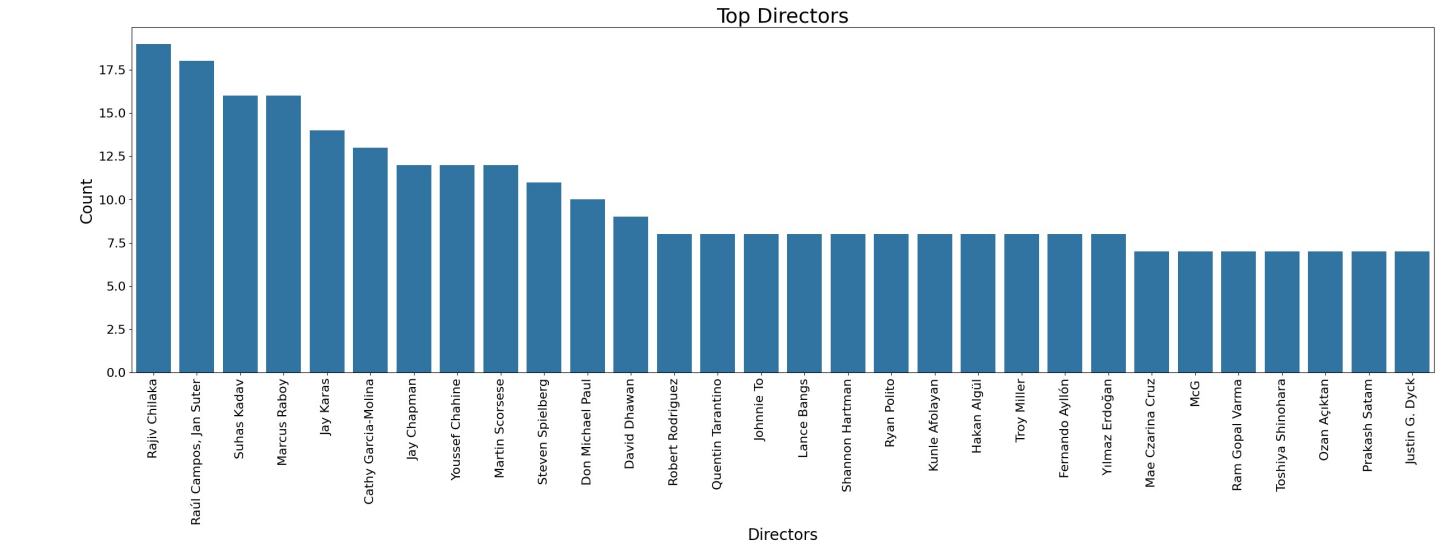
show_id	0
type	0
title	0
director	2621
cast	825
country	829
date_added	88
release_year	0
rating	0
duration	0
listed_in	0
description	0
year	88
month	88
weekday	88

dtype: int64

```
In [ ]: top_dir_list=nf_dir["director"].value_counts().nlargest(30).index

In [ ]: top_dir=nf_dir.loc[nf["director"].isin(top_dir_list)]

In [ ]: plt.figure(figsize=(30,8))
sns.countplot(x=top_dir["director"],order=top_dir["director"].value_counts().index)
plt.xticks(rotation=90, fontsize=16)
plt.xlabel("Directors", fontsize=20)
plt.yticks(fontsize=16)
plt.ylabel("Count", fontsize=20)
plt.title("Top Directors", fontsize=26)
plt.show()
```



These are the Directors with most number of releases on Netflix platform.