Business Case: Netflix - Data Exploration and Visualization

Importing Data

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

```
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv"
```

```
--2024-04-14 07:59:45--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 18.238.92.21, 18.238.92.162, 18.238.92.63, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|18.238.92.21|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3399671 (3.2M) [text/plain]
Saving to: 'netflix.csv'

netflix.csv         100%[===================>]   3.24M  --.-KB/s    in 0.07s

2024-04-14 07:59:45 (46.2 MB/s) - 'netflix.csv' saved [3399671/3399671]
```

```
df = pd.read_csv('netflix.csv')
df
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descripti
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her fatl nears the e of his li filmn
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossi paths a party, a Ca Town
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect family fron powerful dr lo
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feu flirtations a toilet talk down am
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city coachi centers kno to train
...	...	...	...	...	...	...	...	...	...	...	...	
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A politi cartoonist crime repor and

Next steps: Generate code with df View recommended plots

```
df.shape
```

```
(8807, 12)
```

Dataset is having 8807 rows of data with 12 attributes.

Analysing basic metrics

```
basic_metrics1 = df.describe()
basic_metrics1
```

	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

Next steps: Generate code with basic_metrics1 View recommended plots

25% of the tolal data belongs to year 2019-2021

25% of the tolal data belongs to year 1925-2013

```
basic_matrix2 = df[['type', 'country', 'rating']].describe()
basic_matrix2
```

	type	country	rating
count	8807	8807	8803
unique	2	749	17
top	Movie	United States	TV-MA
freq	6131	2818	3207

Next steps: Generate code with basic_matrix2 View recommended plots

Observations:

- The "United States" is the most common country, appearing 3649 times.
- The most frequent rating is "TV-MA," occurring 3207 times.
- There are 17 unique ratings.

find the datatype, name, total entries in each column

```
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
```

We can see that type of rating and date_added columns is "object" which should be categorical and datetime.

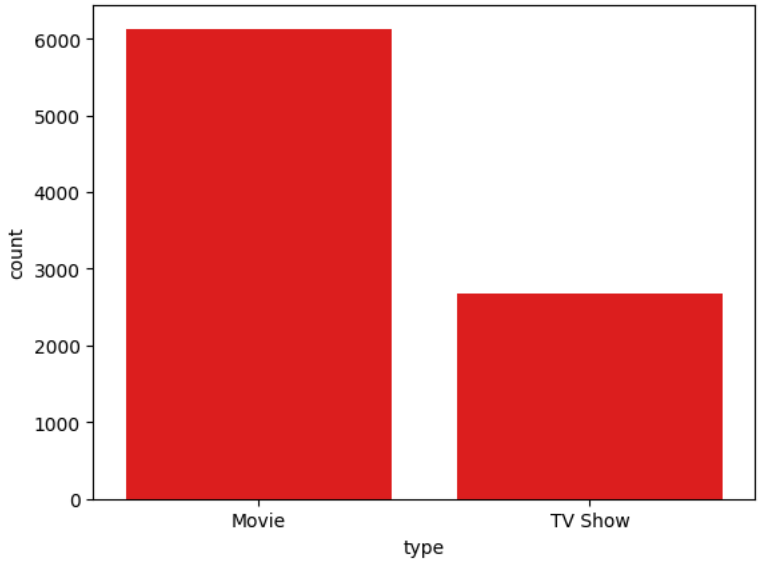
More no. of missing values in cast and director columns.

tv shows & movies.

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

```
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

```
sns.countplot(x='type', data=df, color='red')
plt.show()
```

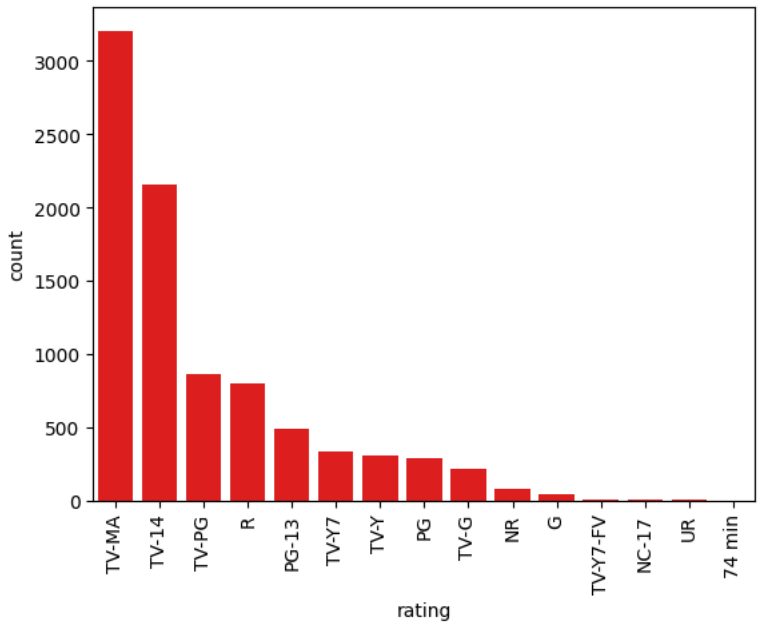


Start coding or generate with AI.

```
rating_count=df['rating'].value_counts().head(10) #checking the count of each category.
rating_count
```

```
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
Name: count, dtype: int64
```

```
countplot=sns.countplot(x='rating', data=df, order=df['rating'].value_counts().index[0:15],color='red')
plt.xticks(rotation=90)
plt.show()
```



Here "TV-MA" has highest count which is stands For Mature Audiences.

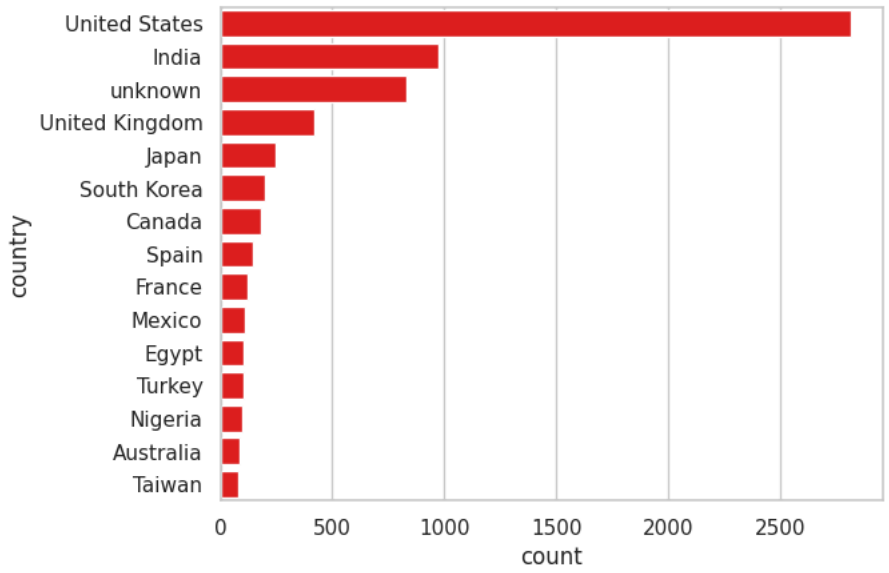
Second highest count is " TV-14"

Third highest count is "TV-PG"

```
country_count=df['country'].value_counts().head(10) #top 10 countries
country_count
```

```
country
United States     2818
India              972
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Egypt              106
Name: count, dtype: int64
```

```
sns.countplot(y='country', data=df, order=df['country'].value_counts().index[0:15],color='red')
plt.show()
```



```
Start coding or generate with AI.
```

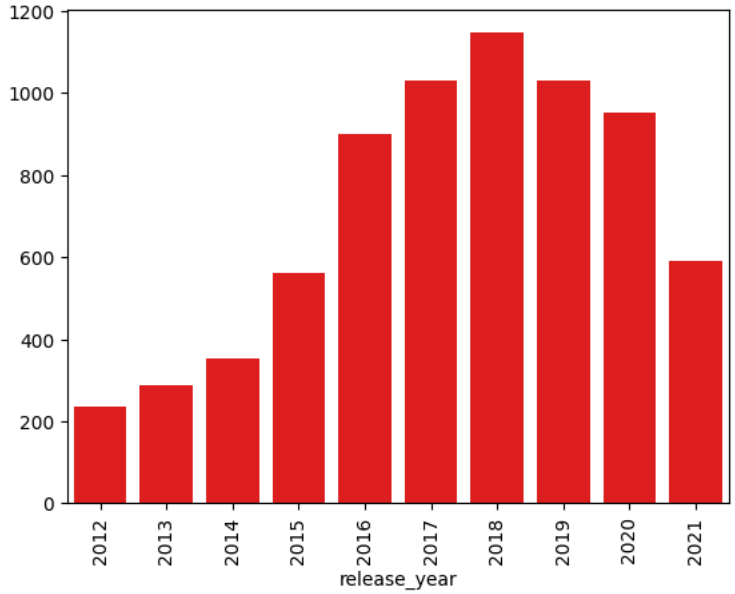
United States is the leading country in producing content.

year wise count

```
release_year_count=df['release_year'].value_counts().head(10) #top 10 years
release_year_count
```

```
release_year
2018    1147
2017    1032
2019    1030
2020     953
2016     902
2021     592
2015     560
2014     352
2013     288
2012     237
Name: count, dtype: int64
```

```
barplot=sns.barplot(x=release_year_count.index, y=release_year_count.values,color='red')
plt.xticks(rotation=90)
plt.show()
```

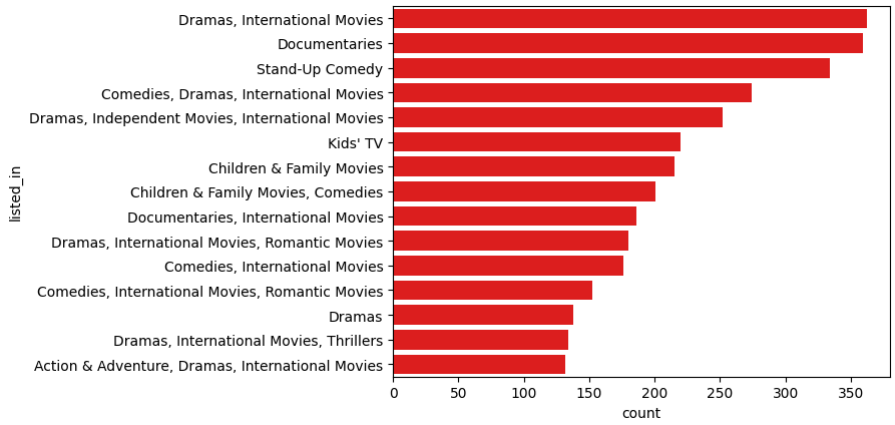


Most of the content relesed in year 2018, 2017, and 2019.

```
genre=df['listed_in'].value_counts().head(10)
genre
```

```
listed_in
Dramas, International Movies                        362
Documentaries                                       359
Stand-Up Comedy                                     334
Comedies, Dramas, International Movies              274
Dramas, Independent Movies, International Movies    252
Kids' TV                                            220
Children & Family Movies                            215
Children & Family Movies, Comedies                  201
Documentaries, International Movies                 186
Dramas, International Movies, Romantic Movies       180
Name: count, dtype: int64
```

```
countplot=sns.countplot(y='listed_in', data=df, order=df['listed_in'].value_counts().index[0:15],color='red')
plt.show()
```



Null values/Missing values

```
null_values = df.isnull().sum() #checking count of null values per column.
null_values
```

```
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
```

- Lot of missing data in director, cast and country columns as compared to others.

```
df['director'].fillna('no director',inplace=True) ##Fillling up the missing values
df['country'].fillna('unknown',inplace=True)
df['cast'].fillna('no cast',inplace=True)
```

```
df['country'].value_counts() #checking unique values in country columns.
```

```
country
United States                                2818
India                                         972
unknown                                       831
United Kingdom                                419
Japan                                         245
                                             ...
Romania, Bulgaria, Hungary                      1
Uruguay, Guatemala                              1
France, Senegal, Belgium                        1
Mexico, United States, Spain, Colombia          1
United Arab Emirates, Jordan                    1
Name: count, Length: 749, dtype: int64
```

```
df['cast'].value_counts().head(10) #checking unique values in cast columns.
```

```
cast
no cast
825
David Attenborough
19
Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mousam, Swapnil
```

```
14
Samuel West
10
Jeff Dunham
7
Craig Sechler
6
David Spade, London Hughes, Fortune Feimster
6
Kevin Hart
6
Michela Luci, Jamie Watson, Eric Peterson, Anna Claire Bartlam, Nicolas Aqui, Cory Doran, Julie Lemieux, Derek McGrath
6
Iliza Shlesinger
5
Name: count, dtype: int64
```

top 10 actor movies based on number of title

```
cast_show = df[df['cast'] != 'no cast'].set_index('title')['cast'].str.split(', ', expand=True).stack()
cast_show.value_counts().head(10)
```

```
Anupam Kher         43
Shah Rukh Khan      35
Julie Tejwani       33
Naseeruddin Shah    32
Takahiro Sakurai    32
Rupa Bhimani        31
Akshay Kumar        30
Om Puri             30
Yuki Kaji           29
Paresh Rawal        28
Name: count, dtype: int64
```

finding the very first and last year a director released a movie.

```
df.groupby(['director'])['release_year'].aggregate(['min','max'])
```

director	min	max
A. L. Vijay	2016	2019
A. Raajdheep	2020	2020
A. Salaam	1975	1975
A.R. Murugadoss	2017	2018
Aadish Keluskar	2018	2018
...	...	...
Çagan Irmak	2005	2005
Ísold Uggadóttir	2018	2018
Óskar Thór Axelsson	2017	2017
Ömer Faruk Sorak	2004	2011
Şenol Sönmez	2015	2019

4529 rows × 2 columns

```
df['date_added'].value_counts()
```

```
date_added
January 1, 2020      109
November 1, 2019      89
March 1, 2018         75
December 31, 2019     74
October 1, 2018       71
                    ...
December 4, 2016       1
November 21, 2016      1
November 19, 2016      1
November 17, 2016      1
```

```
January 11, 2020       1
Name: count, Length: 1767, dtype: int64
```

```
df['date_added'] = pd.to_datetime(df['date_added'],format='mixed')
df['date_added'].head()
```

```
0   2021-09-25
1   2021-09-24
2   2021-09-24
3   2021-09-24
4   2021-09-24
Name: date_added, dtype: datetime64[ns]
```

Comparison of tv shows vs. movies.

```
# Find the number of movies produced in each country and pick the top 10 countries.
filtered_df = df[(df['country'] != 'unknown') & (df['type'] == 'Movie')]
unique_titles_count_by_country = filtered_df.groupby('country')['title'].nunique().reset_index(name='unique_titles_count').head(

print(unique_titles_count_by_country)
```

```
                                        country  unique_titles_count
0                                , France, Algeria                    1
1                                        Argentina                   38
2  Argentina, Brazil, France, Poland, Germany, De...                    1
3                                  Argentina, Chile                    2
4                            Argentina, Chile, Peru                    1
5                                 Argentina, France                    1
6   Argentina, France, United States, Germany, Qatar                    1
7                                  Argentina, Italy                    1
8                                  Argentina, Spain                    7
9                           Argentina, United States                    1
```

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

filtered_df = df[(df['country'] != 'unknown') & (df['type'] == 'TV Show')]
unique_titles_count_by_country = filtered_df.groupby('country')['title'].nunique().reset_index(name='unique_titles_count').head(

print(unique_titles_count_by_country)
```

```
                               country  unique_titles_count
0                          , South Korea                    1
1                              Argentina                   18
2                        Argentina, Spain                    1
3       Argentina, United States, Mexico                    1
4                              Australia                   48
5                      Australia, Canada                    1
6                     Australia, Germany                    1
7                  Australia, New Zealand                    1
8  Australia, New Zealand, United States                    1
9               Australia, United Kingdom                    1
```

What is the best time to launch a TV show?

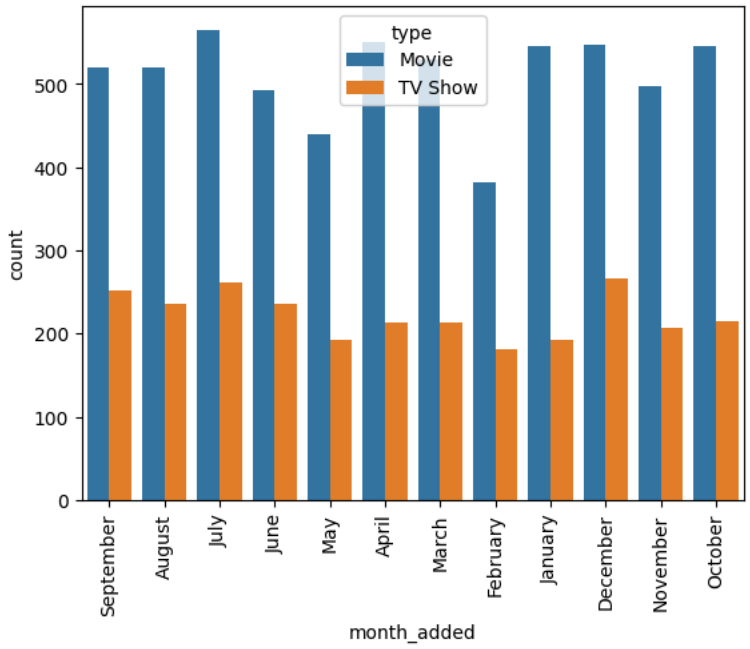
```
# Find which is the best week to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies
df['week_number'] = df['date_added'].dt.isocalendar().week

tv_shows_weekly_counts = df[df['type'] == 'TV Show'].groupby('week_number').size()

movies_weekly_counts = df[df['type'] == 'Movie'].groupby('week_number').size()

best_tv_show_week = tv_shows_weekly_counts.idxmax()

best_movie_week = movies_weekly_counts.idxmax()
print("Best week to release TV shows:", best_tv_show_week)
print("Best week to release movies:", best_movie_week)
```

```
Best week to release TV shows: 27
Best week to release movies: 1
```

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

df['month_added'] = df['date_added'].dt.month_name()

monthly_counts = df.groupby(['month_added', 'type']).size().reset_index(name='count')

tv_show_monthly_counts = monthly_counts[monthly_counts['type'] == 'TV Show']
movie_monthly_counts = monthly_counts[monthly_counts['type'] == 'Movie']

best_tv_show_month = tv_show_monthly_counts.loc[tv_show_monthly_counts['count'].idxmax()]

best_movie_month = movie_monthly_counts.loc[movie_monthly_counts['count'].idxmax()]

print("Best month to release TV shows:", best_tv_show_month['month_added'])
print("Best month to release movies:", best_movie_month['month_added'])
```

```
Best month to release TV shows: December
Best month to release movies: July
```

```
sns.countplot(x='month_added',data=df,hue='type')
plt.xticks(rotation=90)
plt.show()
```



Start coding or generate with AI.

```
df.head(2)
```

	show_id	type	title	director	cast	country	date_added	release_year	r
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	no cast	United States	2021-09-25	2020	
1	s2	TV Show	Blood & Water	no director	Ama Qamata, Khosi Ngema, Gail	South Africa	2021-09-24	2021	

Next steps: Generate code with df View recommended plots

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

```
df.groupby('cast')['title'].nunique().sort_values(ascending=False).head(10)
```

```
cast
no cast
825
David Attenborough
19
Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mousam, Swapnil
14
Samuel West
10
Jeff Dunham
7
David Spade, London Hughes, Fortune Feimster
6
Michela Luci, Jamie Watson, Eric Peterson, Anna Claire Bartlam, Nicolas Aqui, Cory Doran, Julie Lemieux, Derek McGrath
6
Kevin Hart
6
Craig Sechler
6
Iliza Shlesinger
5
Name: title, dtype: int64
```

```
# Identify the top 10 directors who have appeared in most movies or TV shows.
df.groupby('director')['title'].nunique().sort_values(ascending=False).head(10)
```

```
director
no director               2634
Rajiv Chilaka               19
Raúl Campos, Jan Suter      18
Marcus Raboy                16
Suhas Kadav                 16
Jay Karas                   14
Cathy Garcia-Molina         13
Martin Scorsese             12
Jay Chapman                 12
Youssef Chahine             12
Name: title, dtype: int64
```

```
# Which genre movies are more popular or produced more

genres = df['listed_in'].str.split(', ').explode()
genre_counts = genres.value_counts().head(10)
genre_counts
```

```
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
Name: count, dtype: int64
```

Double-click (or enter) to edit

so we can see "International Movies" is most popular papular

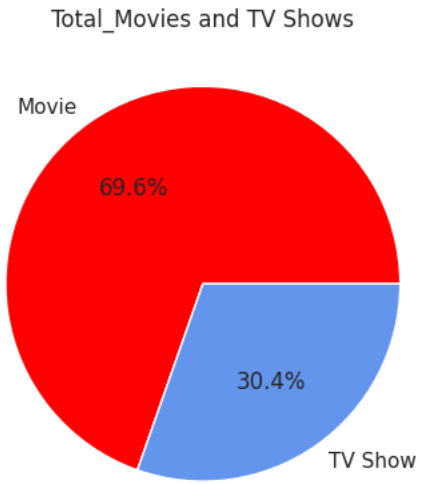
pie chart

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

```
type
Movie      6131
```

```
TV Show    2676
Name: count, dtype: int64
```

```
types=df['type'].value_counts()
plt.pie(types,labels=types.index,autopct='%1.1f%%',colors = ['red' , 'cornflowerblue'])
plt.title('Total_Movies and TV Shows')
plt.show()
```



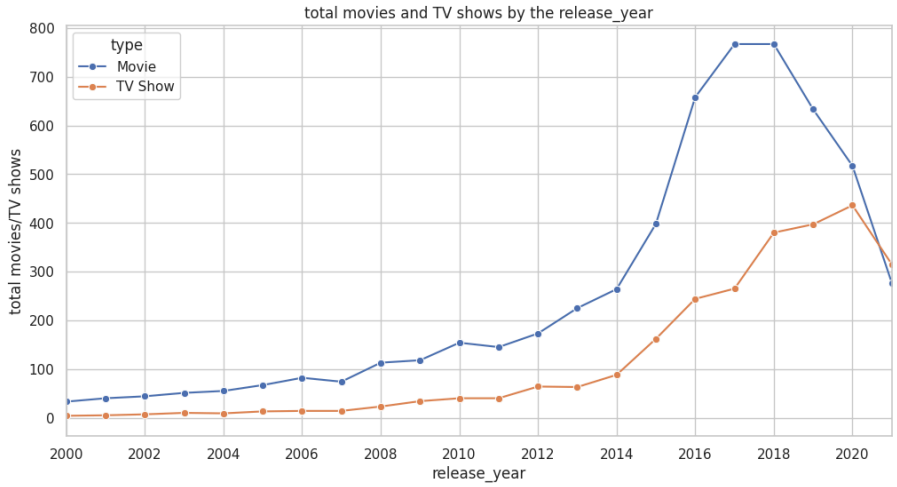

```
# How has the number of movies released per year changed over the last 20-30 years?

data= df.groupby(['type' , 'release_year'])['show_id'].count().reset_index()
data.rename({'show_id' : 'total movies/TV shows'}, axis = 1 , inplace = True)
data
```

	type	release_year	total movies/TV shows
0	Movie	1942	2
1	Movie	1943	3
2	Movie	1944	3
3	Movie	1945	3
4	Movie	1946	1
...	...	...	...
114	TV Show	2017	265
115	TV Show	2018	380
116	TV Show	2019	397
117	TV Show	2020	436
118	TV Show	2021	315

119 rows × 3 columns

Next steps: Generate code with data | View recommended plots

```
plt.figure(figsize = (12,6))
sns.lineplot(data = d , x = 'release_year' , y = 'total movies/TV shows' , hue = 'type' , marker = 'o'  , ms = 6 )
plt.xlabel('release_year' , fontsize = 12)
plt.ylabel('total movies/TV shows' , fontsize = 12)
plt.title('total movies and TV shows by the release_year' , fontsize = 12)
plt.xlim( left = 2000 , right = 2021)
plt.xticks(np.arange(2000 , 2021 , 2))
plt.show()
```



Double-click (or enter) to edit

highest number of movie and TV show releases year is 2018.

after 2018 there is a dip for movie and for tv show increasing.

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Business Insights

majority of content which is released after the year 2000.

TV-MA - Content intended for mature audiences aged 17 and above.

TV-14 - Content suitable for viewers aged 14 and above.

TV-PG - Parental guidance suggested (similar ratings - PG-13 , PG)

R - Restricted Content, that may not be suitable for viewers under age 17

Most popular genres on Netflix are International Movies and TV Shows , Dramas , Comedies, Action & Adventure, Children & Family Movies, Thrillers.

Recommendations

maximum countries need some more genres which are highly popular in the region. eg. Indian Mythological content is highly popular.

Netflix can produce higher number of content in the perticular rating as per demographic of the country