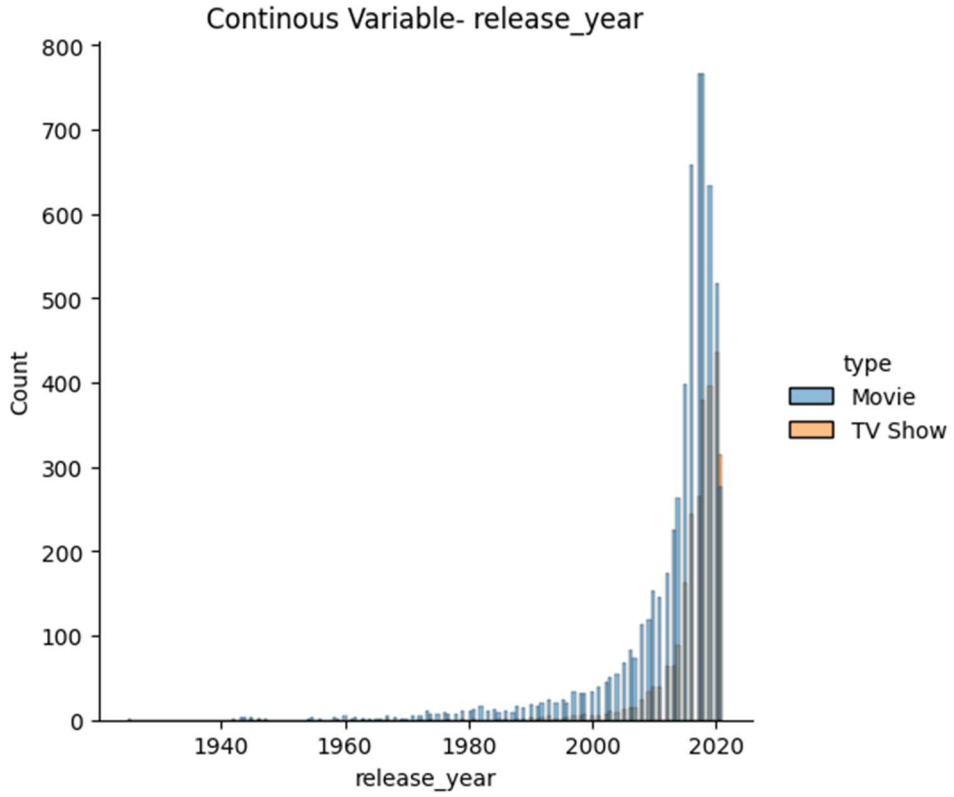
Netflix - Data Exploration and Visualization

Start by exploring a few questions: What type of content is available in different countries?

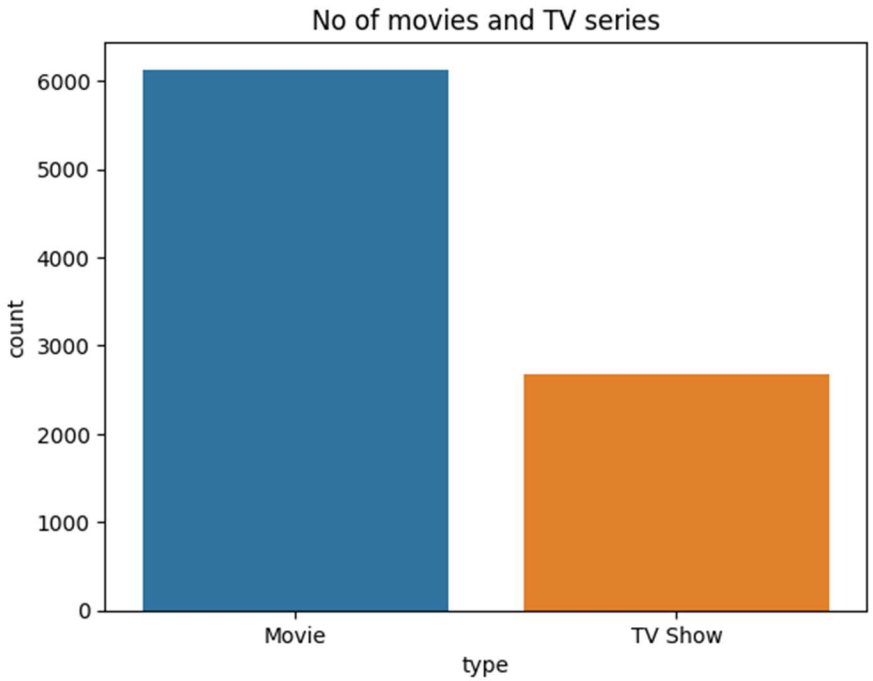
1. How has the number of movies released per year changed over the last 20-30 years?

```
sns.displot(x='release_year',data=df,hue='type')
plt.title("Continous Variable- release_year")
plt.show()
```



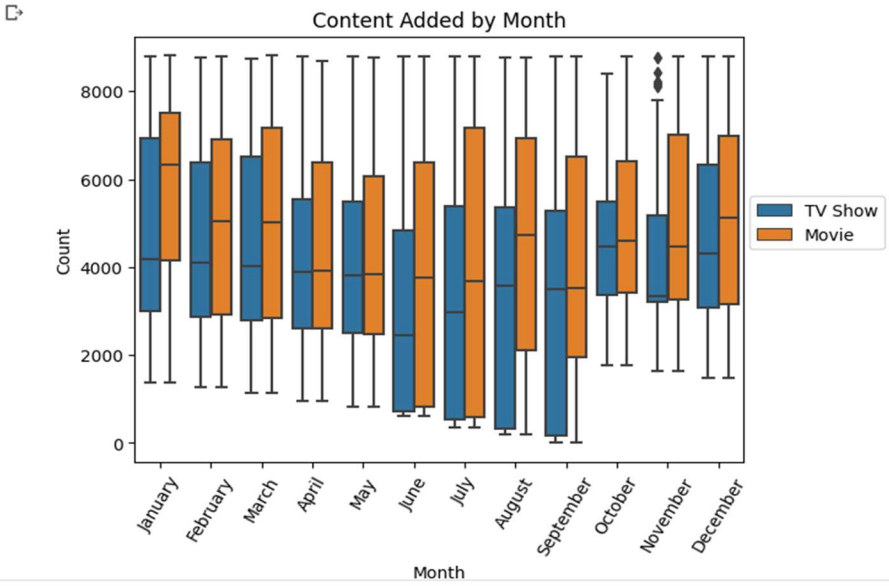
2.Comparison of tv shows vs. movies.

```
sns.countplot(x = "type" , data = df_datetime) #countplot to count the no of movies and tv shows ava
plt.title("No of movies and TV series")
plt.show()
```



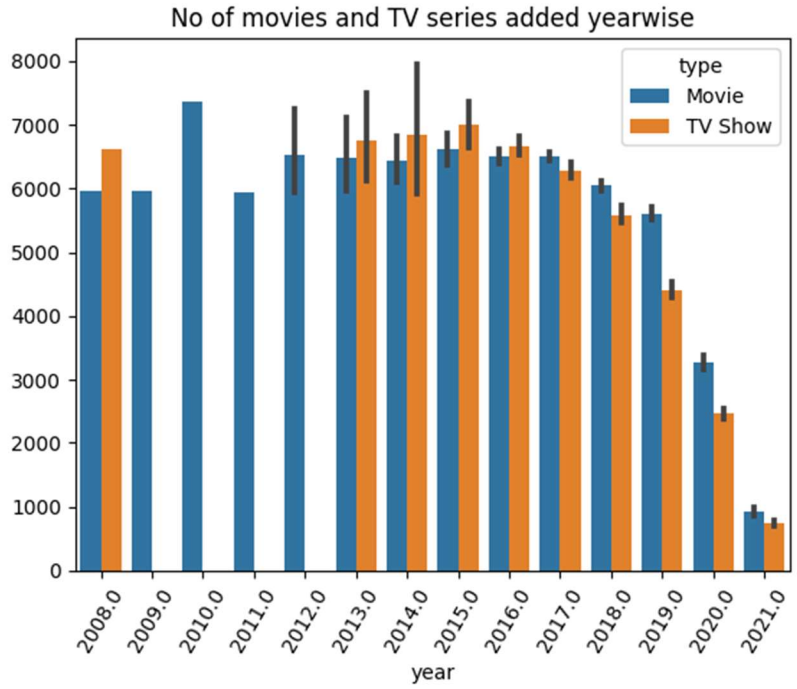
3.What is the best time to launch a TV show?

```
sns.boxplot(y=df_datetime_month['month_name'].index,x=df_datetime_month['month_name'],data=df_datetime_month,hue='type')
plt.legend(loc=(1.01,0.5))
plt.xticks(rotation=60)
plt.xlabel('Month')
plt.ylabel('Count')
plt.title('Content Added by Month')
plt.show()
```



5.Does Netflix has more focus on TV Shows than movies in recent years

```
sns.barplot(x=df_datetime['year'],y=df_datetime['year'].index, data=df_datetime,hue=df_datetime['type'])
plt.xticks(rotation=60)
plt.title("No of movies and TV series added yearwise")
plt.show()
```



Conclusion: There is deep decline in TV shows production due to pandemic .

6.Understanding what content is available in different countries

```
data1=df.groupby(['listed_in'])['country'].count()
data1.head()
```

```
listed_in
Action & Adventure                                          127
Action & Adventure, Anime Features                            0
Action & Adventure, Anime Features, Children & Family Movies      4
Action & Adventure, Anime Features, Classic Movies            1
Action & Adventure, Anime Features, Horror Movies             1
Name: country, dtype: int64
```

1. Defining Problem Statement and Analyzing basic metrics

Analyze the data and generate insights that could help Netflix in deciding which type of shows/movies to produce and how they can grow the business in different countries.

```
df.info() # To get the datatypes of the attributes.
```

```
<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
```

```
df.shape
```

```
(8807, 12)
```

Shape of a dataframe is 8807

```
df.columns
```

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

df.colomns—Gives the columns names.

df.info()→provides how many null values and datatypes.

Column Duration has 2 values 1. Minutes 2. Seasons

Few of the Direcot,actor and country data is missing.

Date_added column is defined as categorical object which should be datetime object.

Few of the values wrongly entered for the column's duration and rating.

2. **Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary**.

Shape

```
import numpy as np
import pandas as pd

df=pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv")
df.shape
```

```
(8807, 12)
```

Netflix dataset has 8807 rows and 12 attributes.

Datatypes

```
import numpy as np
import pandas as pd

df=pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv")
df.shape
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
```

Conclusion

We have observed 2 datatypes 'object' and 'int' in the dataset.

The column "date_added" Dtype at present is object and this should needs to be categorized as Datetime.

Observed more missing values in director, cast and country when compared to other attributes.

Statistical Summary

```
import numpy as np
import pandas as pd

df=pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv")
df.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

Conclusion:

25% of total data belongs to the year 1925-2012.

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

Insight:- Netflix should add more latest movies to attract more no of customers.

```
df.describe(include=object)
```

	show_id	type	title	director	cast	country	date_added	rating	duration	listed_in	description
count	8807	8807	8807	6173	7982	7976	8797	8803	8804	8807	8807
unique	8807	2	8807	4528	7692	748	1767	17	220	514	8775
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	TV-MA	1 Season	Dramas, International Movies	Paranormal activity at a lush, abandoned prope...
freq	1	6131	1	19	19	2818	109	3207	1793	362	4

Conclusion:

Show_id and Title are the unique factors.

"Type" and "rating" column needs to be changed to categorical data.

"United States" is having the maximum content available.

Missing Value Detection

```
df.isnull().sum()
```

```
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
```

Conclusion:

Lots of missing data found in director, cast and country columns as compared to others.

Percentage contribution of missing values:

```
for col in df:
  null_count=df[col].isnull().sum()/len(df)*100
  print(col,":",null_count)
```

```
show_id : 0.0
type : 0.0
title : 0.0
director : 29.908027705234474
cast : 9.367548540933349
country : 9.435676166685592
date_added : 0.11354604292040424
release_year : 0.0
rating : 0.04541841716816169
duration : 0.034063812876121265
listed_in : 0.0
description : 0.0
```

Conclusion:

Director share is 29% of null values followed by country and cast and we cannot drop this data which may cause inconsistency. So we need to do fill the NULL values for director, cast and country with No data.

Handling Missing Values

Replace blank countries with the most common country.

Replace director name and cast with no data .

```
df['country']=df['country'].fillna(df['country'].mode()[0])
df['cast'].replace(np.nan,"No Data",inplace = True)
df['director'].replace(np.nan,"No Data",inplace=True)

df.isnull().sum()
```

```
show_id          0
type             0
title            0
director         0
cast             0
country          0
date_added      10
release_year     0
rating           4
duration         3
listed_in        0
description      0
dtype: int64
```

Drop invalid and duplicate data:

```
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
df.isnull().sum()
```

```
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
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8790 entries, 0 to 8806
Data columns (total 12 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   cast          8790 non-null   object
 5   country       8790 non-null   object
 6   date_added    8790 non-null   object
 7   release_year  8790 non-null   int64 
 8   rating        8790 non-null   object
 9   duration      8790 non-null   object
 10  listed_in     8790 non-null   object
 11  description   8790 non-null   object
dtypes: int64(1), object(11)
memory usage: 892.7+ KB
```

Now the dataset is cleaned with missing values ,invalid and duplicates.

Category Change:

Date_added dtype should not be a object(string) . We need to convert object dtype to datetime dtype.

```
df['date_added']=pd.to_datetime(df['date_added'])
df['month']=df['date_added'].dt.month
df['month_name']=df['date_added'].dt.month_name()
df['year']=df['date_added'].dt.year
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	month	month_name	year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No Data	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	9	September	2021
1	s2	TV Show	Blood & Water	No Data	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	9	September	2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	9	September	2021
3	s4	TV Show	Jailbirds New Orleans	No Data	No Data	United States	2021-09-24	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...	9	September	2021
4	s5	TV Show	Kota Factory	No Data	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...	9	September	2021

Additional Data Cleaning

```
df[df['duration'].isnull()]
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	74 min	NaN	Movies	Louis C.K. muses on religion, eternal love, gi...
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	84 min	NaN	Movies	Emmy-winning comedy writer Louis C.K. brings h...
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	66 min	NaN	Movies	The comic puts his trademark hilarious/thought...

We observed duration values as NaN for the director “Louis C K “ and the values in ratings should in duration. This may be due to input error .Now we need to copy the rating values to duration and make a rating column as No data for the director Louis C K.

Check to make sure there is no other content with the same director to avoid accidental overwriting.

```
df[df['director']=='Louis C.K.'].head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	74 min	NaN	Movies	Louis C.K. muses on religion, eternal love, gi...
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	84 min	NaN	Movies	Emmy-winning comedy writer Louis C.K. brings h...
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	66 min	NaN	Movies	The comic puts his trademark hilarious/thought...

Since there is no extra data , we can start overwriting the duration .

```
df.loc[df['director']=='Louis C.K.' , 'duration']=df['rating']
df.loc[df['director']=='Louis C.K.' , 'rating']='No Data'
df[df['director']=='Louis C.K.'].head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	No Data	74 min	Movies	Louis C.K. muses on religion, eternal love, gi...
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	No Data	84 min	Movies	Emmy-winning comedy writer Louis C.K. brings h...
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	No Data	66 min	Movies	The comic puts his trademark hilarious/thought...

Now we have overwritten the ratings and duration and requirement is fulfilled for director Louis C.K.

3. Non-Graphical Analysis: Value counts and unique attributes

```
## Top 2 directors of top 5 Countries
director = df["director"].apply(lambda x : str(x).split(", ")).tolist()  #exploding the nested data in directors column.
df_director = pd.DataFrame(director, index = df["title"])
df_director= df_director.stack()
df_director = df_director.reset_index()
df_director.drop(columns ="level_1" , inplace = True) #droping the columns
df_director.columns = ["title" , "director"] #renaming the columns
df_fav_director = df.merge(df_director , on = "title" ) #merging of the dataframes
df_fav_director.head(4)
```

```
[64] director_countrywise= df_fav_director.merge(df_country , on = "title")
     director_countrywise= director_countrywise.drop(columns = ["director_x" , "country_x" ])
     director_countrywise.rename(columns = {"director_y": "director" , "country_y" : "country"}, inplace = True)
     director_countrywise = director_countrywise.loc[director_countrywise["director"] != "Unknown"]
     director_countrywise.reset_index(inplace= True)
     director_countrywise.head()
```

```
[65] country = director_countrywise['country'].value_counts()[:6].index.tolist()
     print(' Top 2 Directors of Top 5 Countries')
     print('\n')
     for val in country:
       if val != 'Unknown':
         print(f'**{val}**')
         print(director_countrywise.loc[director_countrywise['country']==val, 'director'].value_counts()[:2])
         print('\n')
```

```
 Top 2 Directors of Top 5 Countries


**United States**
Jay Karas       15
Marcus Raboy    15
Name: director, dtype: int64


**India**
Anurag Kashyap    9
David Dhawan      9
Name: director, dtype: int64


**United Kingdom**
Alastair Fothergill    4
Edward Cotterill       4
Name: director, dtype: int64


**Canada**
Justin G. Dyck      8
Mike Clattenburg    5
Name: director, dtype: int64


**France**
Thierry Donard     5
Youssef Chahine    4
Name: director, dtype: int64
```

Conclusion :

- Anurag Kashyap and David Dhawan are the most famous directors for India.
- Jay Karas and Marcus Raboyare, the most famous directors in United States.

Top 2 directors

Conclusion:

Rajiv Chilaka is the most famous director among all followed by Jan Suter.

```
#Top 2 actors among top 5 Countries
cast = df["cast"].apply(lambda x : str(x).split(", ")).tolist()
df_cast = pd.DataFrame(cast,  index = df["title"])
df_cast = df_cast.stack()
df_cast = df_cast.reset_index()
df_cast.drop(columns = "level_1" , inplace = True)
df_cast.columns = ["title" , "cast"]
df_fav_cast = df.merge(df_cast , on = "title" )
```

```
cast_countrywise= df_fav_cast.merge(df_country , on = "title")
cast_countrywise= cast_countrywise.drop(columns = ["cast_x" , "country_x"])
cast_countrywise = cast_countrywise.rename(columns = {"cast_y" : "cast" , "country_y" : "country"})
cast_countrywise = cast_countrywise.loc[cast_countrywise["cast"] != "Unknown"].reset_index() #making new dataframe by dropping all rows whose cast is unknown and then resetting the index..00
cast_countrywise.head()
```

```
country_actor = cast_countrywise['country'].value_counts()[:6].index.tolist()
print(' Top 2 Actors of Top 5 Countries')
print('\n')
for val in country:
  if val != 'Unknown':
    print(f'--{val}--')
    print(cast_countrywise.loc[cast_countrywise['country']==val, 'cast'].value_counts()[:2])
    print('\n')
```

```
 Top 2 Actors of Top 5 Countries


--United States--
Tara Strong          22
Samuel L. Jackson    22
Name: cast, dtype: int64


--India--
Anupam Kher       40
Shah Rukh Khan    34
Name: cast, dtype: int64
```


Top 2 Actors of Top 5 Countries

--United States--
Tara Strong   22
Samuel L. Jackson   22
Name: cast, dtype: int64

--India--
Anupam Kher   40
Shah Rukh Khan   34
Name: cast, dtype: int64


```
--United Kingdom--
David Attenborough    17
John Cleese           16
Name: cast, dtype: int64


--Canada--
John Paul Tremblay    14
Robb Wells            14
Name: cast, dtype: int64


--France--
Wille Lindberg    5
Benoît Magimel    5
Name: cast, dtype: int64
```

Conclusion :-

- These are the top two cast of these countries.
- Netflix has added more content for India in which cast are- Anupam Kher or Shah Rukh Khan.

```
cast_countrywise["cast"].value_counts().head(5) #value_counts of the cast columns to get the most famous actors
```

```
Anupam Kher          46
David Attenborough   45
Vincent Tong         42
John Cleese          40
Tara Strong          39
Name: cast, dtype: int64
```

Conclusion:

Anupam Kher is the topmost actor among all followed by David Attenborough.

What type of shows that have been watched on Netflix

Column 'type' Analysis :

```
df['type'].value_counts()#To get the what type of shows available on Netflix
```

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

Conclusion:

Two types of shows being produced on NetScaler i.e., Movies and TV show. Movies content has higher number when compared to TV Show.

Country Column Analysis:

```
df['country'].value_counts().head(10) # To get the Netflix content provided for top 10 countries
```

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

Conclusion:

We have gathered the top 10 countries where Netflix provided content(both movies & TV Shows). United States tops for Netflix content followed by India and United Kingdom.

Check the type of content based on Country:

Netflix Movie content produced to Countries:

```
df_mov=df[df['type']=='Movie']
df_tv=df[df['type']=='TV Show']
df_mov['country'].value_counts().head(10)
```

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

Conclusion:

The highest Netflix movie content country is United States followed by India.

Netflix TV shows content produced to Countries:

```
df_tv['country'].value_counts().head(10)
```

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

The highest Netflix TV shows content country is United States followed by United Kingdom.

Content Ratings:

```
df['rating'].value_counts().head(10)
```

```
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: rating, dtype: int64
```

Conclusion:

TV-MA certified content has been most produced by Netflix followed by TV-14

Release year:

```
df['release_year'].value_counts().head(20)
```

```
2018    1147
2017    1032
2019    1030
2020     953
2016     902
2021     592
2015     560
2014     352
2013     288
2012     237
2010     194
2011     185
2009     152
2008     136
2006      96
2007      88
2005      80
2004      64
2003      61
2002      51
Name: release_year, dtype: int64
```

Conclusion:

Above data reveals that year 2018 has highest number of Netflix content releases done followed by Year 2017 and 2019.

Popular Genres Analysis:

```
df['listed_in'].value_counts().head()
```

```
Dramas, International Movies                          362
Documentaries                                         359
Stand-Up Comedy                                       334
Comedies, Dramas, International Movies                274
Dramas, Independent Movies, International Movies      252
Name: listed_in, dtype: int64
```

Conclusion:

The most viewed Netflix content is Dramas ,International Movies followed by Documentaries and Stand-Up comedy.

Summary:

Netflix has more number of movies than TV Shows

Most number of movies produced by United States followed by India which is the 2nd most number of movies in Netflix.

Most of the content Movies & TV Shows are for Mature Audience.

2018 is the year where Netflix released a lot of content when compared to previous Years.

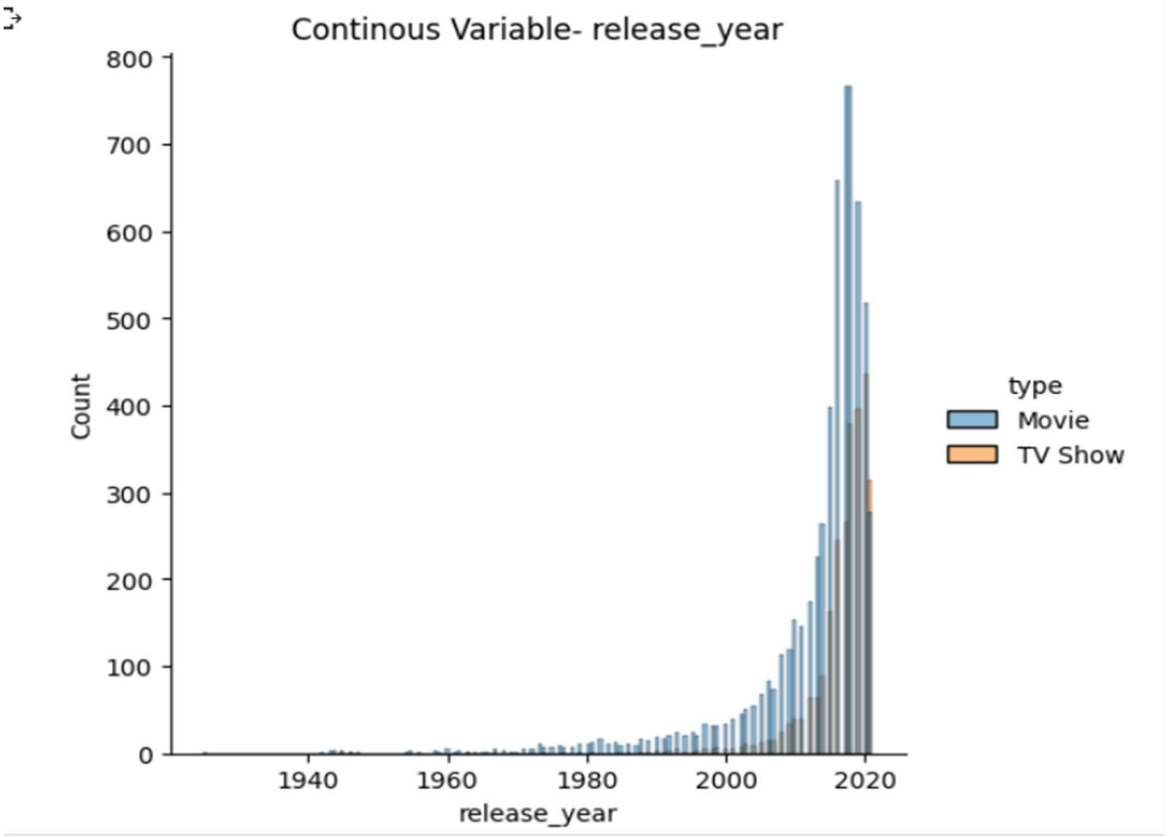
International Movies and movies are the most liked and most popular Genres on Netflix.

4. Visual Analysis - Univariate, Bivariate after pre-processing of the data

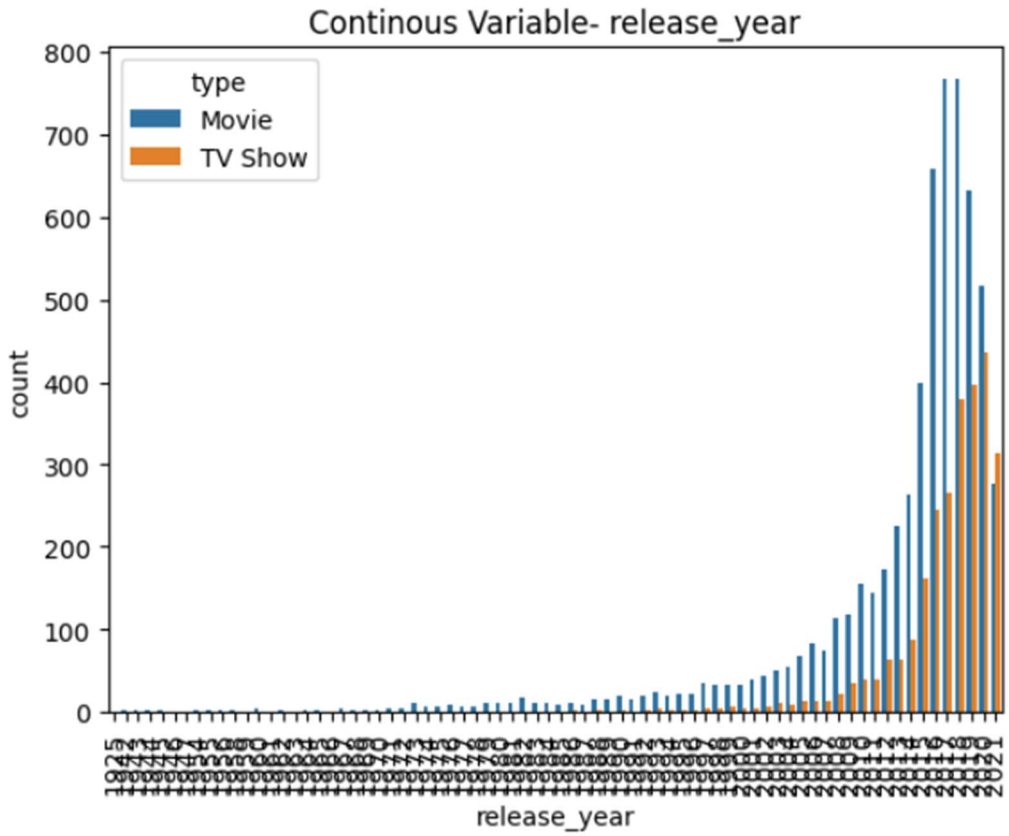
Univariate:

For continuous variable(s):

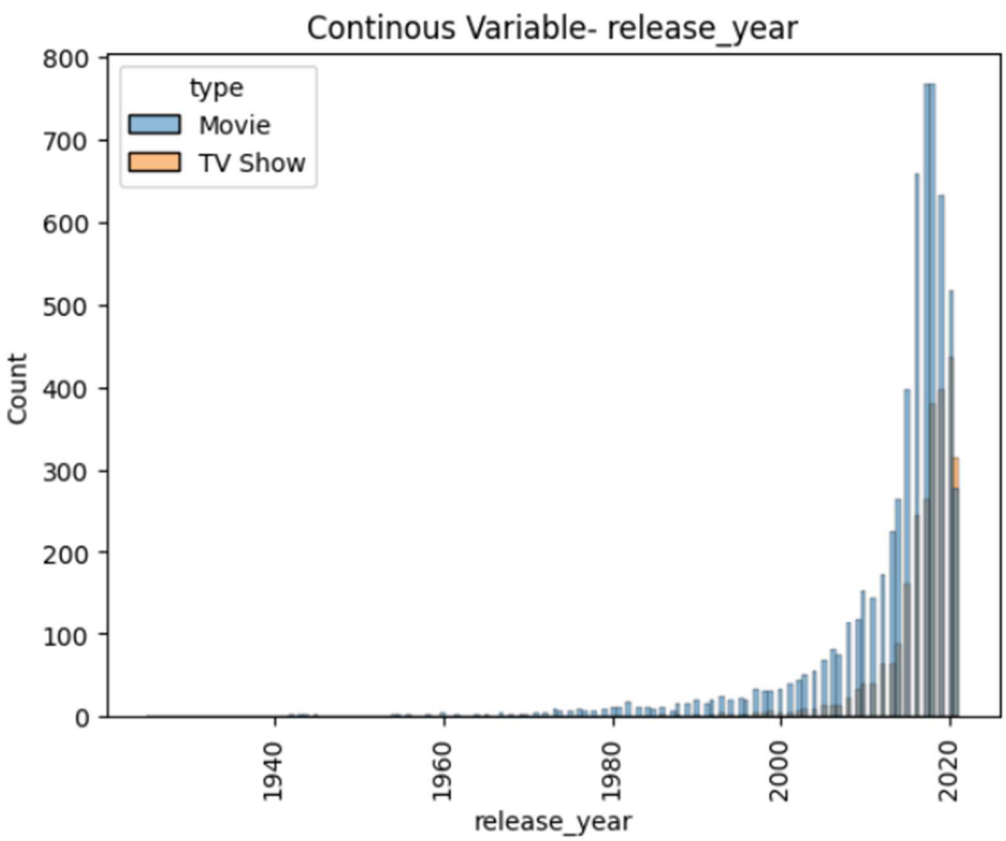
```
sns.displot(x='release_year',data=df,hue='type')
plt.title("Continous Variable- release_year")
plt.show()
```



```
sns.countplot(x='release_year',data=df,hue='type')
plt.title("Continous Variable- release_year")
plt.xticks(rotation=90)
plt.show()
```



```
sns.histplot(x='release_year',data=df,hue='type')
plt.title("Continous Variable- release_year")
plt.xticks(rotation=90)
plt.show()
```



Conclusion:

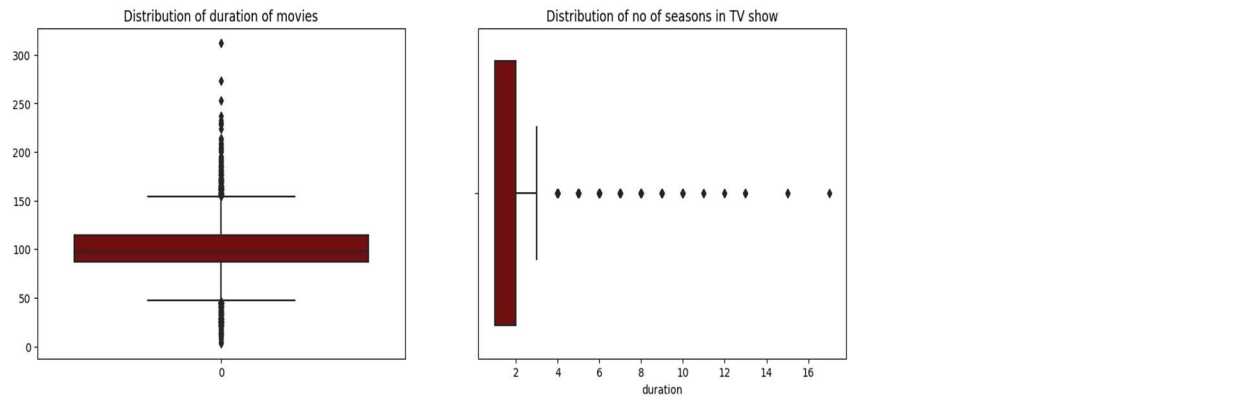
From the above visualization, we can observe that there is a significant increase of Netflix content production from year 2018-2021.

Among movies and TV Shows, Movies has a seen a significant growth.

4.2 For categorical variable(s):

Duration:

```
plt.figure(figsize=(15,5))
duration_df = df.loc[df["duration"].str.contains("min")== True]["duration"].apply(lambda x: x.split()[0]).astype(int)  # splting the movies duration as its type is string , extracting the numeri value and converting it into int type
plt.subplot(1,2,1) #subplots to make the data look easy for comparison.
sns.boxplot(duration_df , color = "maroon")
plt.title("Distribution of duration of movies")
duration_seson_df = df.loc[df["duration"].str.contains("Season")== True]["duration"].apply(lambda x: x.split()[0]).astype(int)
df2=duration_seson_df.reset_index()
plt.subplot(1,2,2)
sns.boxplot(x=df2['duration'] , color = "maroon")
plt.title("Distribution of no of seasons in TV show")
plt.show()
```



Conclusion:

Average duration of movies is 100 mins.

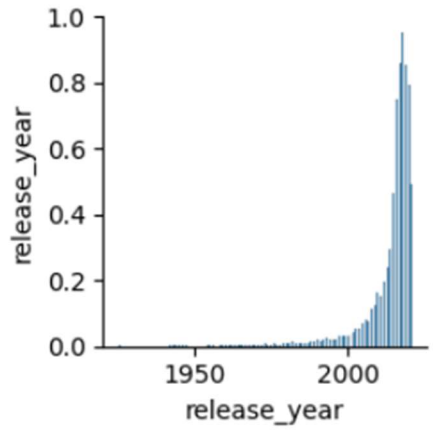
TV shows mostly are having 1 or 2 Seasons.

There are lot of outliers present in movies as compared to TV shows.

4.3 For correlation: Heatmaps, Pairplots

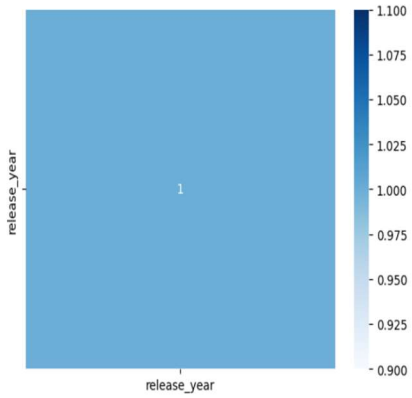
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7c0bf095ae30>
```




```
corr=df.corr()
sns.heatmap(corr,cmap='Blues',annot=True)
```

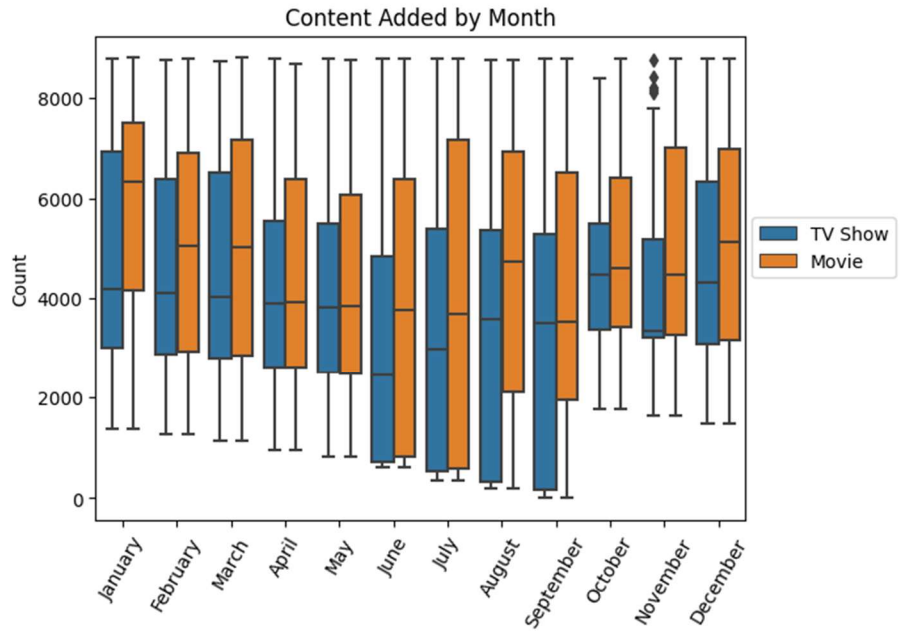
```
<ipython-input-140-48874d0733a6>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the
  corr=df.corr()
<Axes: >
```



Bivariate

Analysis of number of contents added on to Netflix over the period

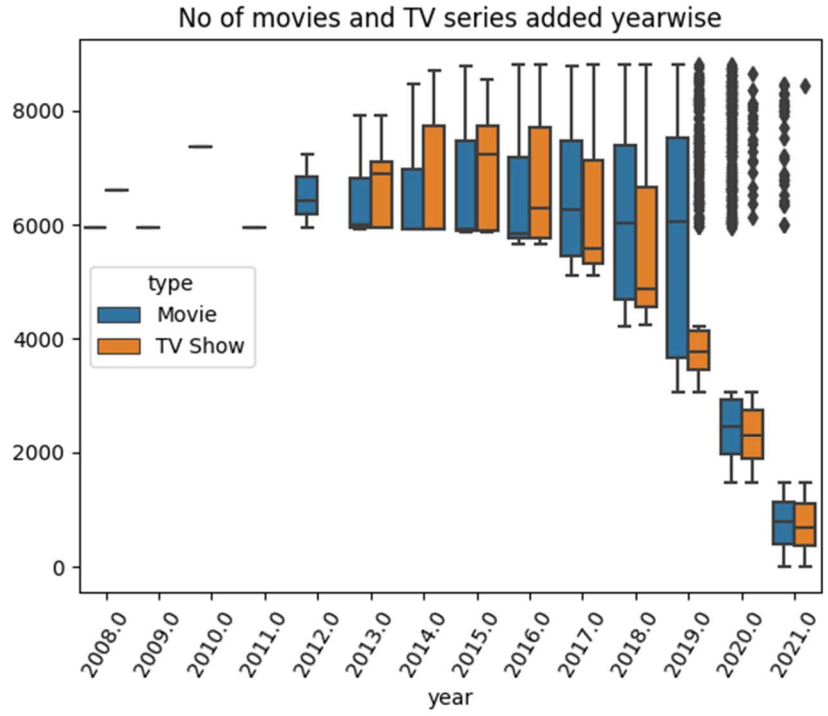
```
df["date_added"] = pd.to_datetime(df["date_added"])
df_datetime=df.copy()
df_datetime['year']=df['date_added'].dt.year
df_datetime['month']=df['date_added'].dt.month
df_datetime['day']=df['date_added'].dt.day_name()
df_datetime_month = df_datetime.sort_values(by ="month")
df_datetime_month['month_name']=df['date_added'].dt.month_name()
sns.boxplot(y=df_datetime_month['month_name'].index,x=df_datetime_month['month_name'],data=df_datetime_month,hue='type')
plt.legend(loc=(1.01,0.5))
plt.xticks(rotation=60)
plt.xlabel('Month')
plt.ylabel('Count')
plt.title('Content Added by Month')
plt.show()
```



Conclusion :-*

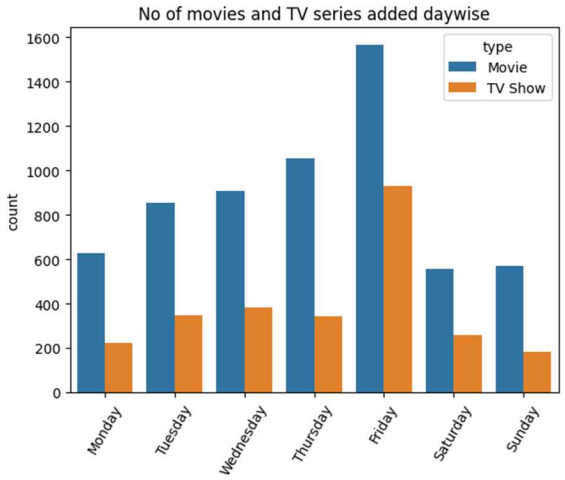
- July, August and September are the months when most content was added because no of TV shows during these 3 months are maximum among all.
- June, July and August are the months where most content added.
- No of movies added per month is greater than no of TV shows added per month.

```
sns.boxplot(x=df_datetime['year'],y=df_datetime['year'].index, data=df_datetime,hue=df_datetime['type'])
plt.xticks(rotation=60)
plt.title("No of movies and TV series added yearwise")
plt.show()
```



- Conclusion:
- More no of movies were added during 2019 followed by 2018 and 2017.
- More no of TV shows added in 2018.
- We can see drop of count of movies from 2019 onwards due to pandemic.

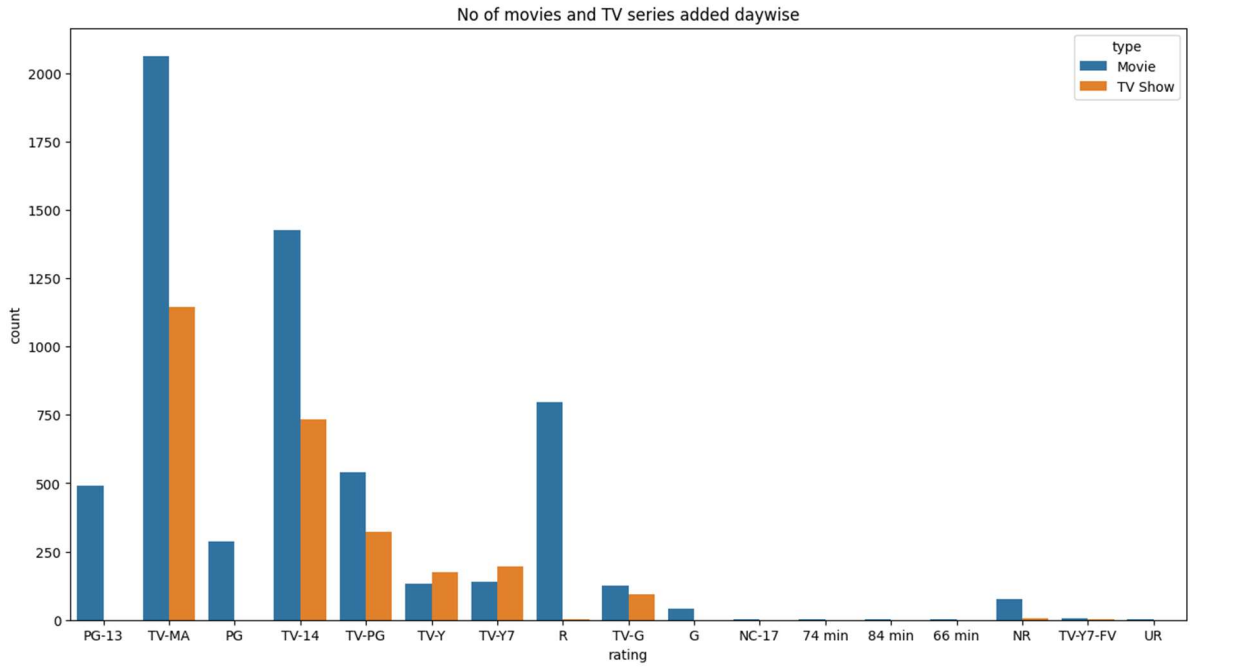
```
sns.countplot(x=df_datetime['day'], data=df_datetime,hue=df_datetime['type'],order=["Monday" , "Tuesday" , "Wednesday", "Thursday", "Friday","Saturday","Sunday"])
plt.xticks(rotation=60)
plt.title("No of movies and TV series added daywise")
plt.show()
```



Conclusion:

Most of the content is added on Netflix on "Friday" followed by Thursday as weekend approaches after these days.

```
plt.figure(figsize=(15,8))
sns.countplot(x=df_datetime['rating'],hue=df_datetime['type'])
plt.title("No of movies and TV series added daywise")
plt.show()
```

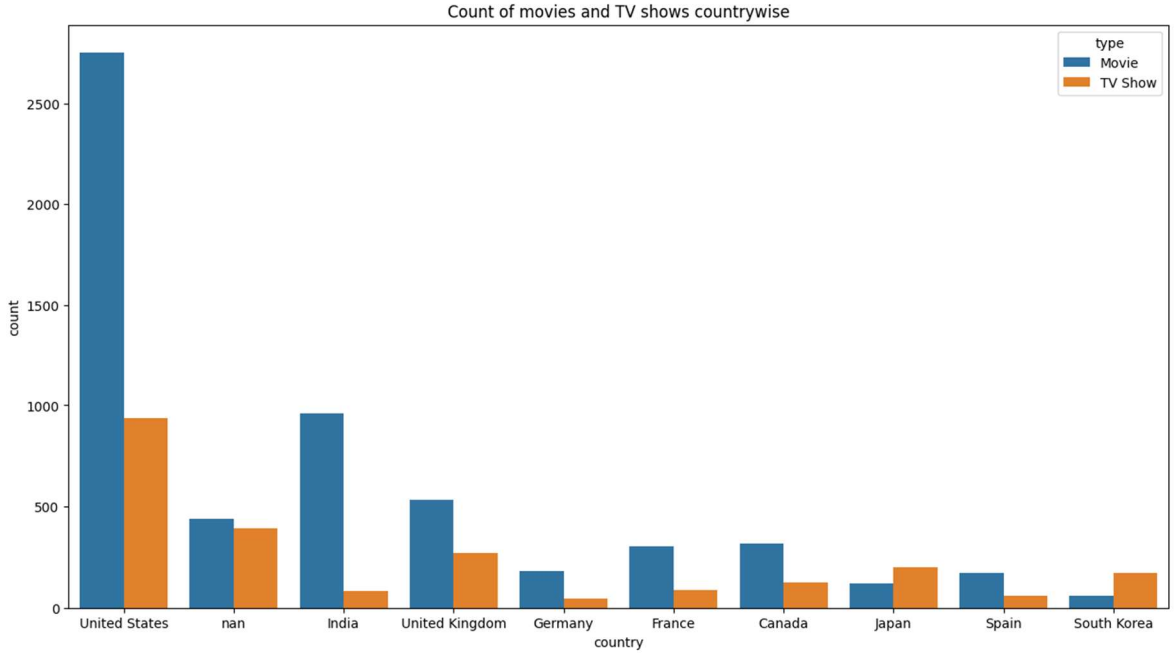


Conclusion:

Mostly TV shows and movies are belongs to TV-MA & TV-14 rating.

Mostly content available on Netflix is for adults and teenagers.

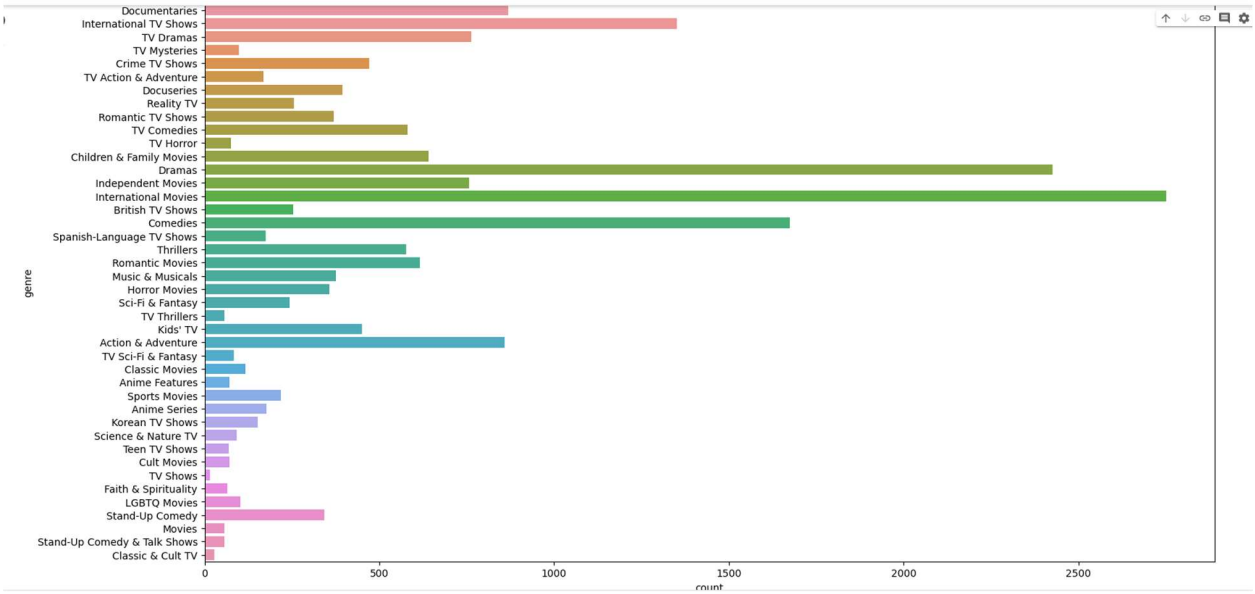
```
Country_wise_trend = df.merge(df_country , on = "title") #making new dataframe by merfing df_country and original dataframe.
Country_wise_trend.drop(columns = "country_x" , inplace = True)
Country_wise_trend.rename(columns = {"country_y" : "country"}, inplace = True)
Country_wise_trend = Country_wise_trend.loc[Country_wise_trend["country"] != "Unknown"]
top10_country = Country_wise_trend["country"].value_counts().head(10).reset_index()
top10_country.rename(columns = {"index" :"country" , "country" : "count"}, inplace = True)
Country_wise_trend = Country_wise_trend.merge(top10_country, how = "inner" , on = "country")
plt.figure(figsize = (15,8))
sns.countplot(x ="country" , data =Country_wise_trend , hue = "type" )
plt.title("Count of movies and TV shows countrywise")
plt.show()
```



Conclusion:

- Netflix should target to add more movies in Unites states and India as compare to TV Series.
- Netflix should target to add more TV shows in Japan and South Korea.

```
plt.figure(figsize = (18,10))
sns.countplot(y = "genre" , data =df_genre )
plt.title("Ditribution of conent Rating_wise")
plt.show()
```



Conclusion:

Most appearing category in Netflix movies and TV shows are:-

- International Movies
- Dramas
- Comedies
- International TV show

5. Missing Value & Outlier check (Treatment optional)

Missing Value Detection

```
df.isnull().sum()
```

```
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
```

Conclusion:

Lots of missing data found in director, cast and country columns as compared to others.

Percentage contribution of missing values:

```
for col in df:
  null_count=df[col].isnull().sum()/len(df)*100
  print(col,":",null_count)
```

```
show_id : 0.0
type : 0.0
title : 0.0
director : 29.908027705234474
cast : 9.367548540933349
country : 9.435676166685592
date_added : 0.11354604292040424
release_year : 0.0
rating : 0.04541841716816169
duration : 0.034063812876121265
listed_in : 0.0
description : 0.0
```

Conclusion:

Director share is 29% of null values followed by country and cast and we cannot drop this data which may cause inconsistency. So we need to do fill the NULL values for director, cast and country with No data.

Handling Missing Values

Replace blank countries with the most common country.

Replace director name and cast with no data .

```
df['country']=df['country'].fillna(df['country'].mode()[0])
df['cast'].replace(np.nan,"No Data",inplace = True)
df['director'].replace(np.nan,"No Data",inplace=True)

df.isnull().sum()
```

```
show_id          0
type             0
title            0
director         0
cast             0
country          0
date_added      10
release_year     0
rating           4
duration         3
listed_in        0
description      0
dtype: int64
```

Drop invalid and duplicate data:

```
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
df.isnull().sum()
```

```
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
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8790 entries, 0 to 8806
Data columns (total 12 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   cast          8790 non-null   object
 5   country       8790 non-null   object
 6   date_added    8790 non-null   object
 7   release_year  8790 non-null   int64 
 8   rating        8790 non-null   object
 9   duration      8790 non-null   object
 10  listed_in     8790 non-null   object
 11  description   8790 non-null   object
dtypes: int64(1), object(11)
memory usage: 892.7+ KB
```

Now the dataset is cleaned with missing values ,invalid and duplicates.

6. Insights based on Non-Graphical and Visual Analysis

6.1 Comments on the range of attributes

```
df.min()
```

```
<ipython-input-125-c3612c624a3f>:1: FutureWarning: The default value of numeric_only in DataFrame.min is deprecated
  df.min()
show_id                                                    s1
type                                                    Movie
title                                                  #Alive
date_added                                2008-01-01 00:00:00
release_year                                             1925
listed_in                                  Action & Adventure
description     "Bridgerton" cast members share behind-the-sce...
dtype: object
```

```
df.max()
```

```
<ipython-input-126-4c1ddf8920ff>:1: FutureWarning: The default value of numeric_only in DataFrame.max
  df.max()
show_id                                                 s999
type                                                 TV Show
title                                  최강전사 미니특공대 : 영웅의 탄생
date_added                             2021-09-25 00:00:00
release_year                                          2021
listed_in                                         Thrillers
description     “Last Chance U” hits the hardwood in East Los ...
dtype: object
```

Min() and Max() function selects the min and max value respectively from each attribute which is of any dtype.

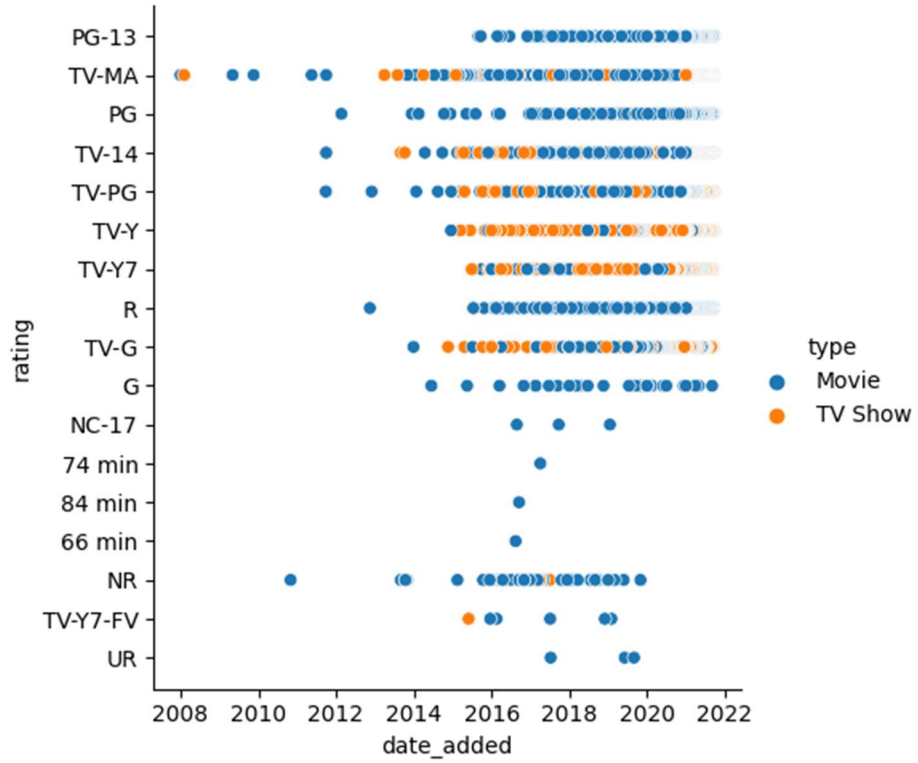
6.2 Comments on the distribution of the variables and relationship between them

Every attribute is correlated and analyzing will be easy to find t.

For example, to find high rating in a year for movie or TV shows , we can establish relationship using replots.

```
sns.relplot(x=df['date_added'],y=df['rating'],hue=df['type'])
```

```
<seaborn.axisgrid.FacetGrid at 0x7ae68d7579a0>
```



6.3 Comments for each univariate and bivariate plot

- Netflix added more movies as compare to TV shows.
- Content for United States on Netflix is maximum as compare to other countries.
- Netflix content is mostly available for adults only.
- Most popular genres in recent years are international movies, Dramas, Comedies, International TV Shows and Action & Adventure.
- In 2021 , there is significant amount of drop in content added due to COVID pandemic.
- Most movies are 100 min duration.
- Netflix should add more movies for United States and India falling in category of international movies and comedies.
- Netflix should add more movies for United States and India having rating of TV-MA & TV-14.
- Top three countries where movies added are United States, India & United Kingdom.
- Netflix should add TV Show on Friday than any other weekday.
- As per 2021 data, count of TV shows are more than movies , this means people wants more web-series as they have for leisure time may be due to work from home scenario.

7. Business Insights

- Netflix added more movies as compare to TV shows.
- Content for United States on Netflix is maximum as compare to other countries.
- Netflix content is mostly available for adults only.
- Most popular genres in recent years are international movies, Dramas, Comedies, International TV Shows and Action & Adventure.
- In 2021 , there is significant amount of drop in content added due to COVID pandemic.
- Most of viewers of Netflix is from United States followed by India & United Kingdom

Movies:-

- In United States , India and United Kingdom movies are more popular as compare to other countries.
- Almost same no. of movies are added on Netflix every month.
- Mostly movies are of "100 min" duration.
- Top people casted in Movies are from India.
- "Rajiv Chilakaa" is the most famous director among all.

TV Shows :-

- TV Shows mostly are having season 1 and season 2 respectively.
- For Japan and South Korea, Netflix should focus more on TV shows as compare to movies.

8. Recommendations

Movies :-

- Preferred movies duration is between 90-100 minutes.

- Netflix should add more movies for United States and India falling in category of international movies and comedies.
- Netflix should add more movies for United States and India having rating of TV-MA & TV-14.
- Top three countries where movies added are United States, India & United Kingdom.
- Netflix should add TV Show on Friday than any other weekday.

TV Show:-

- Preferred movies duration is 1-2 seasons.
- Netflix should focus on countries like Japan, South Korea and France in TV shows , as they prefer TV shows over movies.
- Netflix should add TV Show on Friday than other weekday.
- As per 2021 data, count of TV shows are more than movies , this means people wants more web-series as they have for leisure time may be due to work from home scenario.