NETFLIX Case Study

Problem Statement: 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

Importing important libraries

In [297…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Importing DataSet

In [310…

```
netflix_df=pd.read_csv(r'https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv')
```

Getting familiar with Data

In [311…

```
netflix_df.head()
```

Out[311…

	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	September 25, 2021	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	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
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 his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	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	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

In [312…

```
netflix_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
```

In [313…

```
netflix_df.shape
```

Out[313…
```
(8807, 12)
```

In [314…
```
netflix_df.describe()
```

Out[314…

	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

In [315…
```
netflix_df.describe(include='object')
```

Out[315…

	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

Checking valuecounts and unique attributes

In [316…
```
cols=['type', 'title', 'director', 'cast', 'country','release_year','rating','listed_in']
```

In [283…
```
for i in cols:
    print(netflix_df[i].value_counts(ascending=False).head(3))
    print(netflix_df[i].nunique())
```

In [317…
```
raw_df=netflix_df.copy(deep=True)
```

Unnesting for prepocessing the data

In [373…
```
netflix_df['director']= netflix_df['director'].str.split(',')
netflix_df['cast']= netflix_df['cast'].str.split(',')
netflix_df['country']= netflix_df['country'].str.split(',')
```

In [374…
```
netflix_df=netflix_df.explode('director')
netflix_df=netflix_df.explode('cast')
netflix_df=netflix_df.explode('country')
```

Removing blank spaces after the explode function

In [375…
```
netflix_df['country']=netflix_df['country'].str.strip()
netflix_df['cast']=netflix_df['cast'].str.strip()
netflix_df['director']=netflix_df['director'].str.strip()
```

In [379…
```
netflix_df.drop(columns=['level_0','index'],axis=1,inplace=True)
```

Adding a month coulmn in the dataset

In [363…
```
netflix_df['month']=pd.to_datetime(netflix_df['date_added']).dt.month
```

Graphical Analysis(Uni and Bivariate)

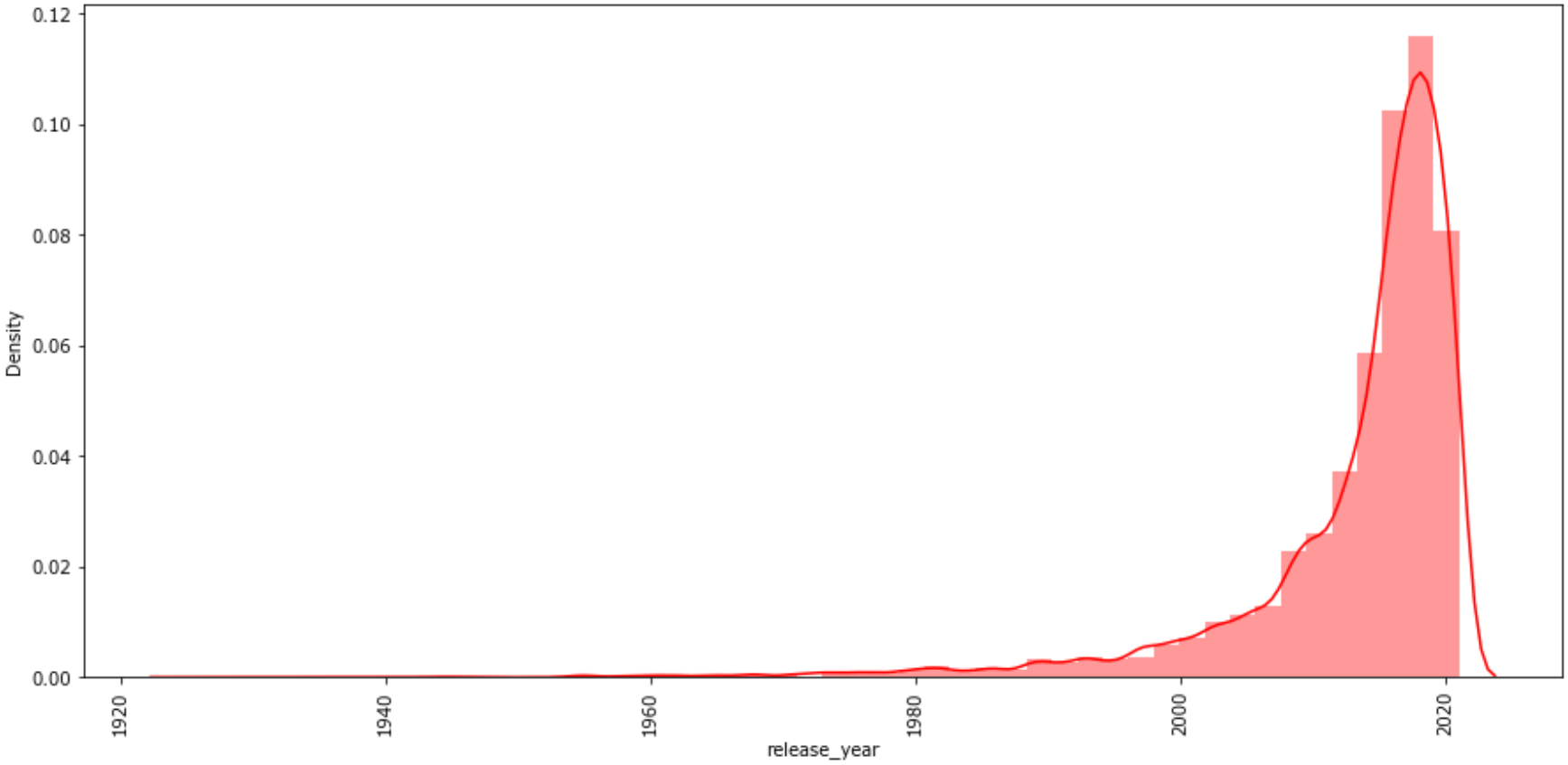
Univariate analysis(Continous variable)

-- we will be working with release year for this to check the distribution of data through the years

In [291…
```
plt.figure(figsize=(15,7))
plt.xticks(rotation=90)
```

```
# x=netflix_df[netflix_df['release_year']>1970]
sns.distplot(netflix_df['release_year'],color='red')

plt.show()
```

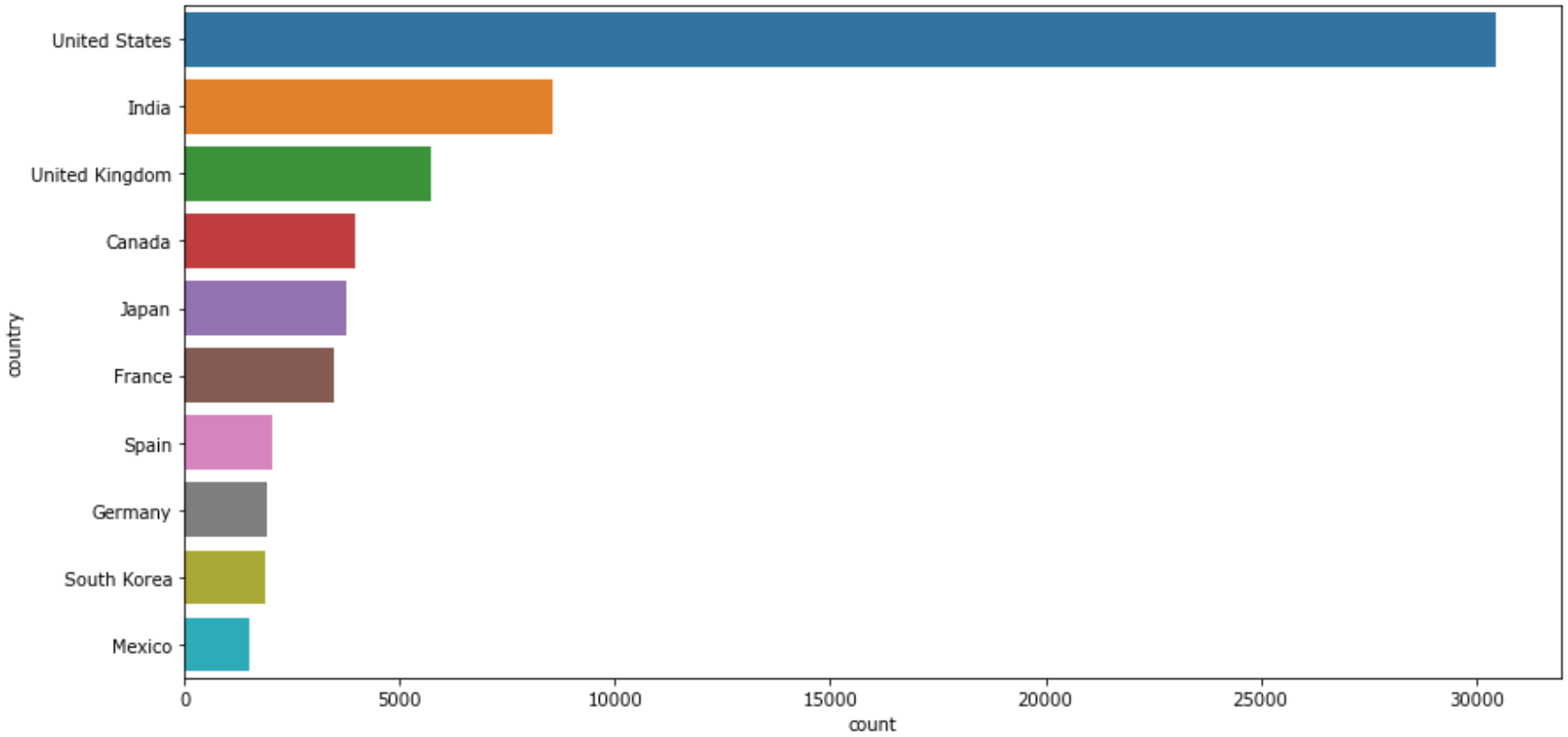


Univariate Analysis(Categorical variables)

-- we will be working with country and rating columns for this analysis

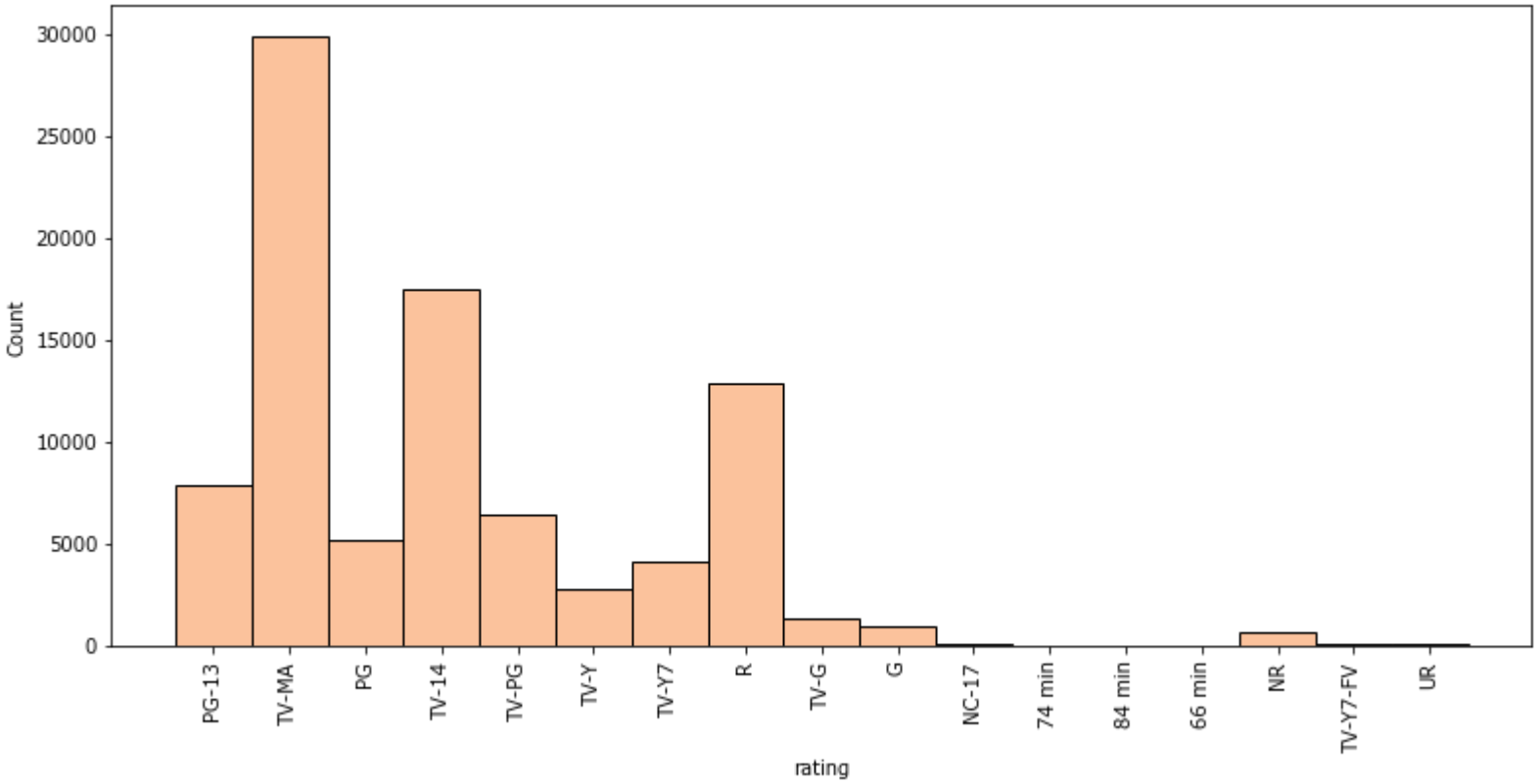
In [292…

```
plt.figure(figsize=(14,7))
sns.countplot(y=netflix_df['country'],order=netflix_df['country'].value_counts().head(10).index)
plt.show()
```



In [293…

```
plt.figure(figsize=(13,6))
sns.histplot(netflix_df['rating'],color='#FAAE7B')
plt.xticks(rotation=90)
plt.show()
```

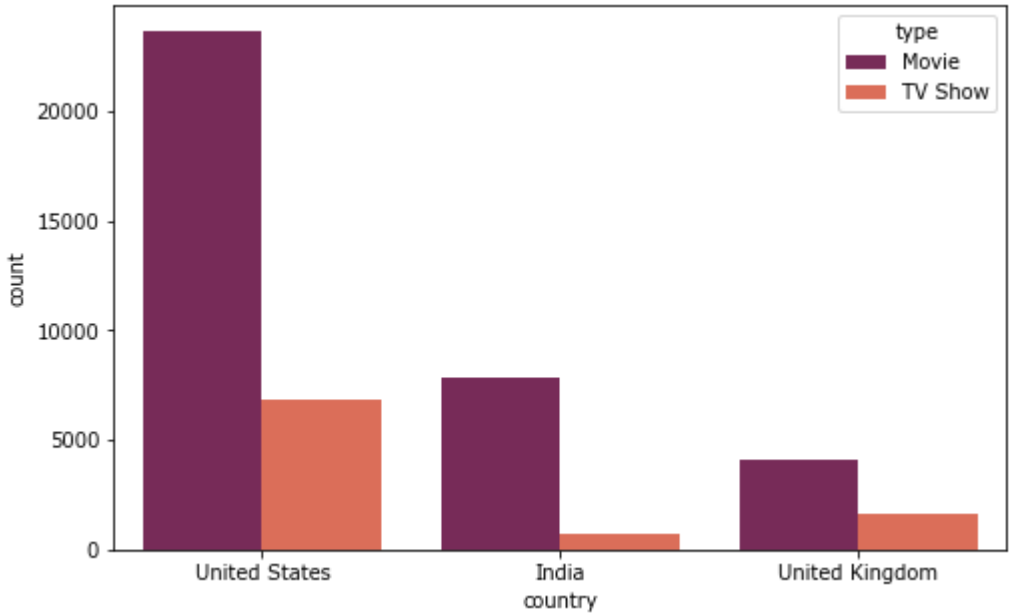


Bivariate Analysis

-- Mostly we have categorical variables, will try to analyse cat-cat below.

In [294...
```
top3_country=netflix_df['country'].value_counts().index[:3]
data_top3=netflix_df.loc[netflix_df['country'].isin(top3_country)]
```

In [295...
```
plt.figure(figsize=(8,5))
plt.rcParams['font.family'] = "Verdana"
sns.countplot(x="country", data=data_top3, hue="type",palette='rocket')
# pd.crosstab(data_top3['country'], data_top3['type']).plot(kind='bar', stacked=True)
plt.show()
```



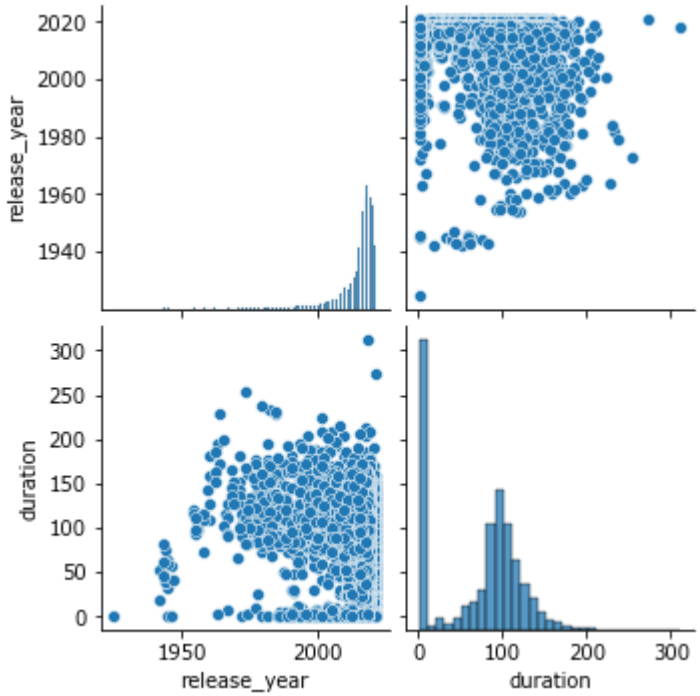
Pairplots and heatmaps wont be useful as we have mostly categorical data, one way we can do it is by converting them into discrete variables(one_hot_ecoding or get dummies)

In here I will be using duration along with release year to perform pairplot

In [319…
```
raw_df['duration']=raw_df['duration'].str.split(' ')
raw_df['duration']=raw_df['duration'].str[0].astype('float')
```

In [320…
```
sns.pairplot(raw_df)
```

Out[320… `<seaborn.axisgrid.PairGrid at 0x22dec031730>`



Checking & Imputing missing values(NaN) and we can look at outliers using IQR range

--- for outlier detection I will show an example of 1 variable from the dataframe

In [7]:
```
netflix_df.isna().sum().sort_values(ascending=False)
```

Out[7]:
```
director        2634
country          831
cast             825
date_added        10
rating             4
duration           3
show_id            0
type               0
title              0
release_year       0
listed_in          0
description        0
dtype: int64
```

In [9]:

```
# Imputation
netflix_df['director'].fillna('Unknown', inplace=True)
netflix_df['country'].fillna('Unknown', inplace=True)
netflix_df['cast'].fillna('Unknown', inplace=True)
netflix_df['date_added'].fillna('Unknown', inplace=True)
netflix_df['duration'].fillna('Unknown', inplace=True)
netflix_df['rating'].fillna('Unknown', inplace=True)
```

In [328…

```
# Outlier Detection
test_df=raw_df[raw_df['type']=='Movie']
Q1=np.percentile(test_df['duration'],25)
Q3=np.percentile(test_df['duration'],75)
IQR=Q3-Q1
lower_limit=Q1-1.5*(IQR)
upper_limit=Q3+1.5*(IQR)
```

In [333…

```
count=0
for i in test_df['duration']:
    if i < lower_limit and i > upper_limit:
        print(i)
        count+=1
    else:
        continue
print(count)
''' ================================================================
##### From above case we can see there are no outliers in the duration column
================================='''
```

```
0
```

Findings from Above Analysis

In [ ]:

```
"""
1. Most of the attributes in the dataset are categorical, so prprocessing is needed if you want to work with
   certain coulmns for example, duration which can distributed into
   different categories eg. TV Shows and Movies and the take the time from them and analyse it for discriptive analysis.
2. For Release year variable, we saw that there is more surge of demand in the later 2000 as compared to earlier years,
   which is self explanatory as netflix started booming
   in the late 2000s, we saw that United States, India and United Kingdom are the countries which are mostly engaging
   with Netflix.
3. With respect to plots, we saw that as we have more Categorical variables, we have build majorly countplots,
```

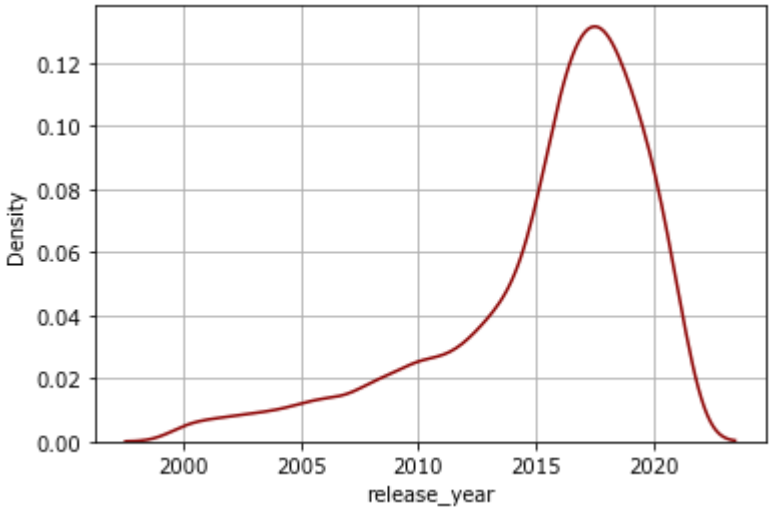
```
   histplots,distplot(release_year) and dodged countplots for
   bivariate analysis, also we tried doing pair plots by changing the duration column to continous variable.
"""
```

Exploring Few other insights which could be relevant to infer and evaluate

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

In [340…

```
x=raw_df[(raw_df['release_year']>=2000 ) & (raw_df['release_year']<=2022) & (raw_df['type']=='Movie')]
sns.kdeplot(x['release_year'],color='darkred')
plt.grid()
plt.show()
```

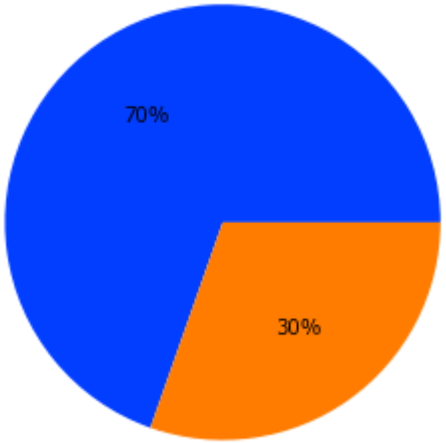


-- We see a upward trend from 2005 afterwards i.e. more movies we produced as the time has passed over the last 2 decades

1. Comparison of tv shows vs. movies.

In [361…

```
plt.figure(figsize=(5,5))
palette_color = sns.color_palette('bright')
plt.pie(raw_df['type'].value_counts(),colors=palette_color, autopct='%.0f%%')
plt.show()
```



-- As we can see around 70% Movies are made over the years as compared to TV Shows, which is around 30%

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

In [365…
```
x=raw_df[raw_df['type']=='TV Show']
x['release_month']=pd.to_datetime(x['date_added']).dt.month_name()
```

In [370…
```
x['release_month'].value_counts(ascending=False)
```

Out[370…
```
December     266
July         262
September    251
August       236
June         236
October      215
April        214
March        213
November     207
May          193
January      192
February     181
Name: release_month, dtype: int64
```

--- Not much difference between the months but December might be good month to release a TV show

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

In [387…
```
# Director/Actor(s/ess) with most movies over the years
x=netflix_df[netflix_df['type']=='Movie']
print(x['director'].value_counts(ascending=False).nlargest(3))
print(x['cast'].value_counts(ascending=False).nlargest(3))
```

```
Martin Scorsese     217
Steven Spielberg    205
Raja Gosnell        154
Name: director, dtype: int64
Alfred Molina    84
Liam Neeson      82
Salma Hayek      66
Name: cast, dtype: int64
```

--- Above are top 3 actors(s/ess) and Director(s) over the Years with most number of movies

In [389…
```
# Director/Actor(s/ess) with most TV Shows over the years
x=netflix_df[netflix_df['type']=='TV Show']
print(x['director'].value_counts(ascending=False).nlargest(3))
print(x['cast'].value_counts(ascending=False).nlargest(3))
```

```
Thomas Astruc      80
Noam Murro         63
Damien Chazelle    52
Name: director, dtype: int64
David Attenborough    28
Takahiro Sakurai      26
Vincent Tong          26
Name: cast, dtype: int64
```

--- Above are top 3 actors(s/ess) and Director(s) over the Years with most number of TV Shows

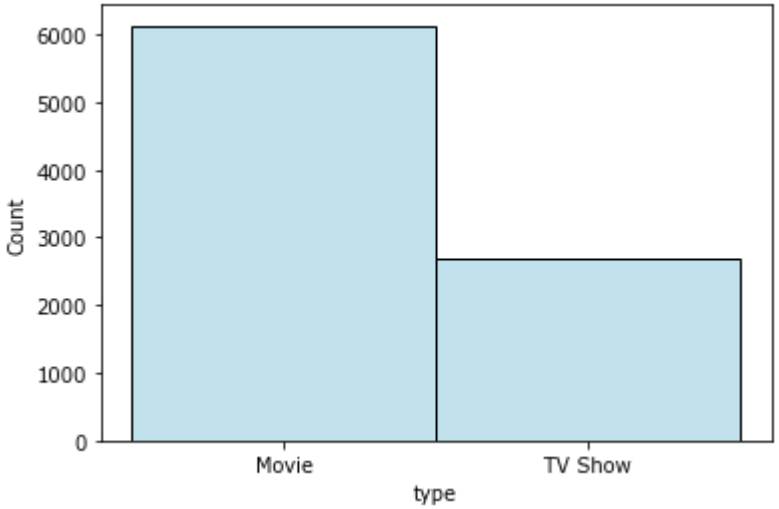
1. Does Netflix has more focus on TV Shows than movies in recent years

In [392…
```
raw_df['year']= pd.to_datetime(raw_df['date_added']).dt.year
```

In [397…
```
## For tis we will take last 5 years of data and see the difference

x=raw_df[(raw_df['year']>=2016) & (raw_df['year']<=2022)]
```

```
sns.histplot(raw_df['type'],color='lightblue')
plt.show()
```



--- Still movies are being added into Netflix platform more than TV shows in recent years

1. Understanding what content is available in different countries

In [410…

```
top3_country=netflix_df['country'].value_counts().index[:3]
data_top3=netflix_df.loc[netflix_df['country'].isin(top3_country)]
data_top3.groupby(['country','type'])['country','type'].count()
```

Out[410…

		country	type
country	**type**		
India	**Movie**	7835	7835
	TV Show	702	702
United Kingdom	**Movie**	4063	4063
	TV Show	1660	1660
United States	**Movie**	23676	23676
	TV Show	6796	6796

--- Above I have taken the top3 country data and see what sort of content they watch, we can further drill down the data into generes and see what generes are watched most
in the countries

1. Top Genres to watch

In [411…
```
netflix_df['listed_in']= netflix_df['listed_in'].str.split(',')
netflix_df=netflix_df.explode('listed_in')
netflix_df['listed_in']=netflix_df['listed_in'].str.strip()
```

In [416…
```
netflix_df['listed_in'].value_counts(ascending=False).head(5)
```

Out[416…
```
Dramas                    29806
International Movies      28243
Comedies                  20829
International TV Shows    12845
Action & Adventure        12216
Name: listed_in, dtype: int64
```

In [417…
```
netflix_df.head(1)
```

Out[417…

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	month
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	9.0

Recommendations

In [ ]:
```
'''
================================================================
1. India, UK and Cananda are good market countries to invest as they are growing in the last decade or so, so creating
   regional content would definitely help.
2. TV Shows are still less produced as compared to Movies, so we can infer that Users consume things which are of less
   duration as compared to TV shows,
   which takes more time to consume, so probably series with less duration(like short stories) might engage the audience
   more.
```

```
3. People are most intered in generes like Dramas, Comedies and International Movies/TV shows, so Netflix should be
   investing more such genre content in the future.
4. Also there is an urge of movie/TV show releases during the months of December(which is a holiday time for people
   around he world), so there is a good chance to increase customer engagement during that time period.
==============================================================
'''
```

In [ ]: