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

#Loading the dataframe into a variable
df = pd.read_csv('/content/drive/MyDrive/SCALER/BUSINESS
CASES/Netflix_Dataset.csv')
```

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. We are going to discuss on few of the metrics like Content Voulme, Content Growth & Content Performance.

INITIAL OBSERVATION ON THE DATASET

```
#This tells the number of rows & columns present in the dataframe.
Totally there are 8807 rows & 12 columns.
df.shape
```

```
(8807, 12)
```

```
# This gives the summary of a DataFrame's structure with the type of
the data ecach column has.
# We can see there are a total of 8807 entries for movies & tv shows
combined in the dataframe. There are missing values for few columns
which we need to deal with.
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
```

DATA CLEANING

```
#checking if there are any duplicate titles by converting them into
lower case
df['title'] = df['title'].str.lower()
df = df.drop_duplicates(subset=['type','title'])

#There are few columns with missing values:
# director, cast & country - we are going to fill the missing values
as'Non Available' as this seems to be the better option instead of
filling random values.
#date_added - 10 rows have missing values. We are going fill it with
the corresponding release year value

df['director'].fillna('Not-Available', inplace = True)
df['cast'].fillna('Not-Available', inplace = True)
df['country'].fillna('Not-Available', inplace = True)
df['date_added'].fillna(df['release_year'],inplace = True)
df['date_added'] = df['date_added'].astype('str')

'''duration - row number(5541,5794 & 5813) has mssing values. Upon
checking the data we can see that the rating column has the duration
value present.
So we are going to assign those values to duration column. Then for
rating column for these 3 assigning value as 'Not Available'. '''

df['duration'].fillna(df['rating'],inplace = True)

df['rating'].iloc[[5541,5794,5813]] = np.nan
df['rating'].fillna('Not-Available', inplace = True)

<ipython-input-533-1bfc33ffba18>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df['rating'].iloc[[5541,5794,5813]] = np.nan

#Converting date_added column to datetime format
df['date_added'] = pd.to_datetime(df['date_added'])

#Separating the duration of movies & tv shows
def separate_duration(df):
  if df['type'].lower() == 'movie':
```

```
    df['length_in_minutes'] = int(df['duration'].split(' ')[0])
  if df['type'].lower() == 'tv show':
    df['season'] = int(df['duration'].split(' ')[0])
  return df


df = df.apply(separate_duration,axis=1)
df.drop('duration',axis=1,inplace=True)

#Getting the year from date_added column for analysis.
#Ordering the columns in the dataframe.
df['added_year'] = df['date_added'].dt.year
df['added_month'] = df['date_added'].dt.month_name()
df = df[['show_id','type','title',    'director',     'cast',
     'country', 'release_year', 'rating',  'length_in_minutes',
'season','listed_in','description','date_added',
'added_year','added_month']]
```

Exploratory Data Analysis

Descriptive Analysis

```
#How many movies & TV- shows are present on Netflix
df['type'].value_counts()
```

```
Movie      6129
TV Show    2675
Name: type, dtype: int64
```

```
#Year-wise movie & tv-shows addition to netflix
df.groupby(['type','added_year']).added_year.count()
```

```
type     added_year
Movie    2008           1
         2009           2
         2010           1
         2011          13
         2012           3
         2013           6
         2014          19
         2015          56
         2016         253
         2017         839
         2018        1235
         2019        1424
         2020        1284
         2021         993
TV Show  2003           2
```

```
          2008              2
          2010              1
          2012              1
          2013              6
          2014              5
          2015             28
          2016            177
          2017            349
          2018            413
          2019            591
          2020            595
          2021            505
Name: added_year, dtype: int64
```

```
#rating wise distribution of movies & tv shows
df.groupby(['type','rating']).rating.count()
```

```
type     rating
Movie    G                  41
         NC-17               3
         NR                 74
         Not-Available       5
         PG                287
         PG-13             490
         R                 797
         TV-14            1427
         TV-G              126
         TV-MA            2061
         TV-PG             540
         TV-Y              131
         TV-Y7             139
         TV-Y7-FV            5
         UR                  3
TV Show  NR                  5
         Not-Available       2
         R                   2
         TV-14             733
         TV-G               94
         TV-MA            1144
         TV-PG             323
         TV-Y              176
         TV-Y7             195
         TV-Y7-FV            1
Name: rating, dtype: int64
```

```
#creating dataframes for movies & tv shows
movie = df[df['type']=='Movie'].drop('season',axis=1)
tv = df[df['type']=='TV Show'].drop('length_in_minutes',axis=1)
```

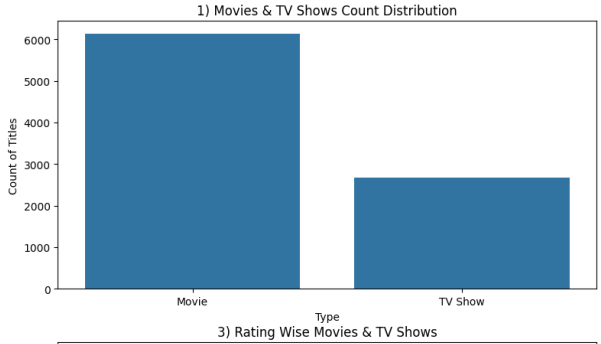
```
#Finding average duration of movies over the years
tv['rating'].value_counts()
```

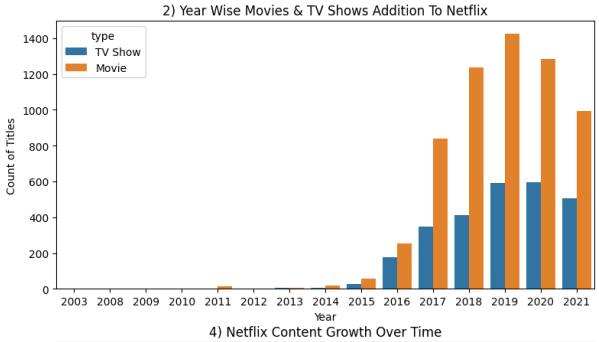
```
TV-MA             1144
TV-14              733
TV-PG              323
TV-Y7              195
TV-Y               176
TV-G                94
NR                   5
R                    2
Not-Available        2
TV-Y7-FV             1
Name: rating, dtype: int64
```

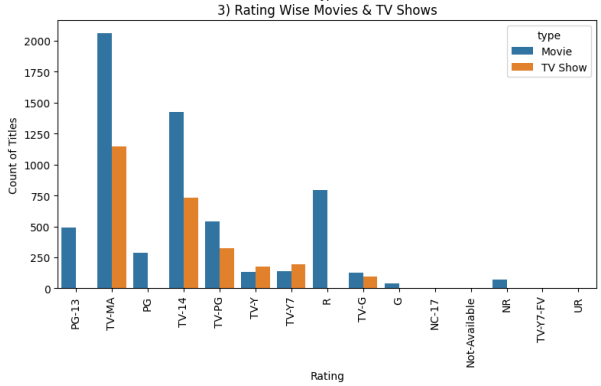
VISUAL ANALYSIS

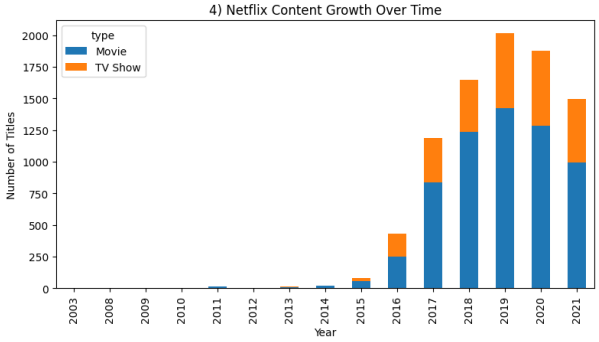
```
fig, axes = plt.subplots(2,2,figsize=(20,10))

#1) Univariate Analysis - How many movies & TV- shows are present on
Netflix
sns.countplot(x='type',data=df,ax=axes[0,0])
axes[0,0].set_xlabel('Type')
axes[0,0].set_ylabel('Count of Titles')
axes[0,0].set_title('1) Movies & TV Shows Count Distribution')

#2 Bivariate Analysis - Year-wise movie & tv-shows addition to netflix
sns.countplot(x='added_year',data=df,hue='type',ax=axes[0,1])
axes[0,1].set_xlabel('Year')
axes[0,1].set_ylabel('Count of Titles')
axes[0,1].set_title('2) Year Wise Movies & TV Shows Addition To
Netflix')

#3) Bivariate Analysis - Rating wise distribution of movies & tv shows
sns.countplot(x='rating',hue='type',ax=axes[1,0],data=df)
axes[1,0].set_xlabel('Rating')
axes[1,0].set_ylabel('Count of Titles')
axes[1,0].set_title('3) Rating Wise Movies & TV Shows')
axes[1,0].tick_params(labelrotation=90,axis='x')

#4) Univariate Analysis - Finding average duration of movies over the
years
content_growth = df.groupby(['added_year','type'])
['type'].count().unstack().fillna(0)
content_growth.plot(kind='bar',stacked=True,ax=axes[1,1])
axes[1,1].set_xlabel('Year')
axes[1,1].set_ylabel('Number of Titles')
axes[1,1].set_title('4) Netflix Content Growth Over Time')

plt.show()
```

```
#Insights for these graphs:
#1) We can see the movie count is more than the tv shows count.
#2) We can see from around the year 2016, there is an increase in the
number of movies & tv shows but we can see a slight drop in 2021.
#3) More number of movies & tv shows come under 'TV-MA' rating.
#4) From 2016, there is a significant growth in the netflix's content.
In 2021 there is a drop in the conent for some external reasons.
```





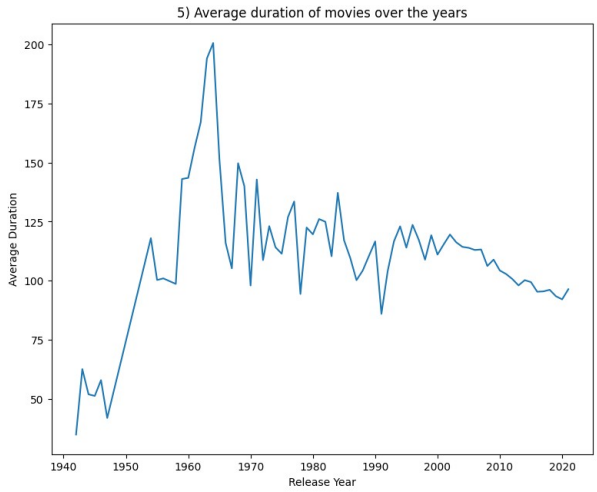


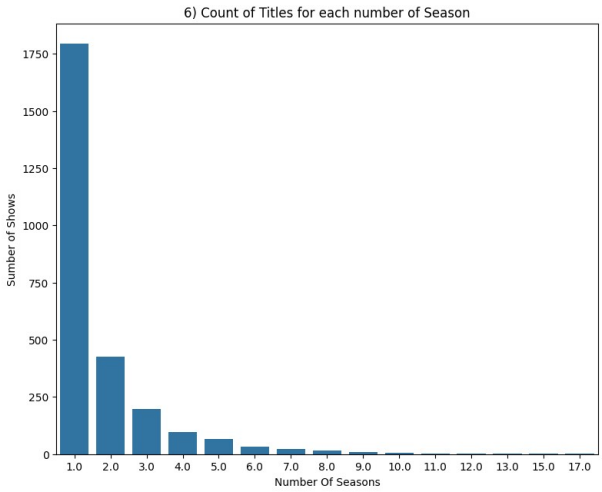


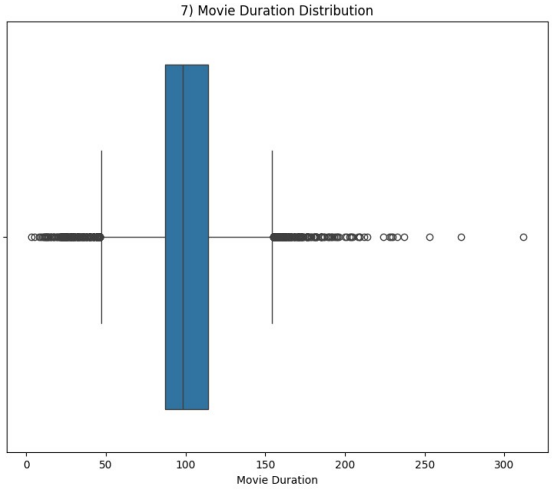

```
plt.figure(figsize=(20,8))

fig, axes = plt.subplots(2,2,figsize=(20,16))

#5) Univariate Analysis - Finding average duration of movies over the
years
sns.lineplot(x=movie.groupby('release_year').length_in_minutes.mean().
index,y=movie.groupby('release_year').length_in_minutes.mean(),ax=axes
[0,0])
axes[0,0].set_xlabel('Release Year')
axes[0,0].set_ylabel('Average Duration')
axes[0,0].set_title('5) Average duration of movies over the years')


#6) Univariate Analysis - TV Show Seasons Distribution on Netflix
sns.countplot(x='season',data=tv,ax=axes[0,1])
axes[0,1].set_xlabel('Number Of Seasons')
axes[0,1].set_ylabel('Sumber of Shows')
axes[0,1].set_title('6) Count of Titles for each number of Season')
```

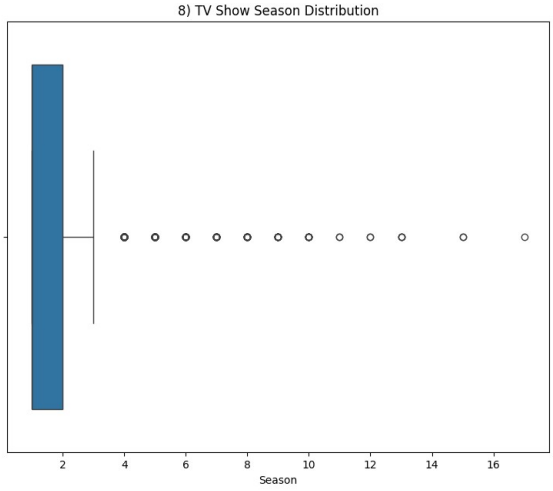
```
#7) Univariate Analysis -
sns.boxplot(x='length_in_minutes',data=movie,ax=axes[1,0])
axes[1,0].set_xlabel('Movie Duration')
axes[1,0].set_title('7) Movie Duration Distribution')

#8) Univariate Analysis - Finding average duration of movies over the
years
sns.boxplot(x='season',data=tv,ax=axes[1,1])
axes[1,1].set_xlabel('Season')
axes[1,1].set_title('8) TV Show Season Distribution')

plt.show()
#Insights for the below graphs:
#5) There is a flucation in the average duration of the movies during
the initial years where in 1940's it was around 60 minutes & in 1960 -
1970, it was more than 150 minutes.
#   In the recent years the average duration for movies is between 100
to 120 minutes.
#6) We can see a trend where more number of tv shows are produced with
only 1 season.
#7) This shows that for movies the min duration ~ 40 minutes & max
duration `~ 150 minutes. Other duration outside the boxplot are
outliers where the movies with those durations are rarely made.
#8) This shows that for tv shows the max number of seasons usually
made are 3 seasons. Very few tv shows are made with more than 3
seasons.
```

```
<Figure size 2000x800 with 0 Axes>
```









FURTHER ANALYSIS

```
#unnesting columns - director, cast, country & listed_in
def unnest(df,col):
  a = df[col].str.split(', ').to_list()
  b =
pd.DataFrame(a,index=df.index).stack().reset_index().drop('level_1',ax
is=1).set_index('level_0').rename({0:col},axis=1)
  df = df.merge(b,left_index=True,right_index=True,how='inner')
  df[col+'_x'] = df[col+'_y']
  df = df.drop(col+'_y',axis=1).rename({col+'_x':col},axis=1)
  df = df.drop_duplicates(keep='first')
  return df

df = unnest(df,'cast')
df = unnest(df,'director')
df = unnest(df,'country')
df = unnest(df,'listed_in')
```

DESCRIPTIVE ANALYSIS

```
#Finding out how many unique directors, cast & country and genre is in
the dataset

director =
df.drop_duplicates(subset=['title','director']).director.unique().size
cast = df.drop_duplicates(subset=['title','cast']).cast.unique().size
country =
df.drop_duplicates(subset=['title','country']).country.unique().size
genre =
df.drop_duplicates(subset=['title','listed_in']).listed_in.unique().si
ze

print('Count\nDirector: {}\nCast: {}\nCountry: {}\nGenre:
{}'.format(director,cast,country,genre))
```

```
Count
Director: 4994
Cast: 36438
Country: 128
Genre: 42
```

```
#Number of titles based on Genre for Movies & TV Shows
df.drop_duplicates(subset=['title','listed_in']).groupby(['type','list
ed_in']).title.count()
```

```
type     listed_in
Movie    Action & Adventure                859
         Anime Features                     71
         Children & Family Movies          641
         Classic Movies                    116
         Comedies                         1672
         Cult Movies                        70
         Documentaries                     869
         Dramas                           2426
         Faith & Spirituality               65
         Horror Movies                     357
         Independent Movies                756
         International Movies             2750
         LGBTQ Movies                      102
         Movies                             57
         Music & Musicals                  375
         Romantic Movies                   616
         Sci-Fi & Fantasy                  243
         Sports Movies                     219
         Stand-Up Comedy                   343
         Thrillers                         577
TV Show  Anime Series                      176
         British TV Shows                  253
```

```
Classic & Cult TV                   28
Crime TV Shows                     470
Docuseries                         395
International TV Shows            1350
Kids' TV                           451
Korean TV Shows                    151
Reality TV                         255
Romantic TV Shows                  370
Science & Nature TV                 92
Spanish-Language TV Shows          173
Stand-Up Comedy & Talk Shows        56
TV Action & Adventure              168
TV Comedies                        581
TV Dramas                          762
TV Horror                           75
TV Mysteries                        98
TV Sci-Fi & Fantasy                 84
TV Shows                            16
TV Thrillers                        57
Teen TV Shows                       69
Name: title, dtype: int64
```

```
# assigning dataframes to new variables after unnesting
movie_new = df[df['type']=='Movie'].drop('season',axis=1)
tv_new = df[df['type']=='TV Show'].drop('length_in_minutes',axis=1)

#Number of movies done by the actors
movie_new.drop_duplicates(subset=['title','cast']).cast.value_counts()
.drop('Not-Available')
```

```
Anupam Kher               42
Shah Rukh Khan            35
Naseeruddin Shah          32
Om Puri                   30
Akshay Kumar              30
                          ..
Sri Muktha                 1
Ravi Babu                  1
Daniel Newman              1
Michael Trotter            1
Chittaranjan Tripathy      1
Name: cast, Length: 25950, dtype: int64
```

```
#Number of movies directed by the directors
movie_new.drop_duplicates(subset=['title','director']).director.value_
counts().drop('Not-Available')
```

```
Rajiv Chilaka        22
Jan Suter            21
Raúl Campos          19
```

```
Suhas Kadav           16
Marcus Raboy          15
                      ..
Vrinda Samartha        1
Nicholaus Goossen      1
Stig Bergqvist         1
Paul Demeyer           1
Mozez Singh            1
Name: director, Length: 4777, dtype: int64
```

```
#Number of movies available in each country
movie_new.drop_duplicates(subset=['title','country']).country.value_co
unts().drop('Not-Available')
```

```
United States     2751
India              962
United Kingdom     532
Canada             319
France             303
                   ...
Ecuador              1
Armenia              1
Mongolia             1
Mozambique           1
Montenegro           1
Name: country, Length: 122, dtype: int64
```

```
# Number of tv shows done by the actors
tv_new.drop_duplicates(subset=['title','cast']).cast.value_counts().dr
op('Not-Available')
```

```
Takahiro Sakurai      25
Yuki Kaji             19
Junichi Suwabe        17
Ai Kayano             17
Daisuke Ono           17
                      ..
Morla Gorrondona       1
Dana Davis             1
Genesis Rodriguez      1
Merit Leighton         1
Hina Khawaja Bayat     1
Name: cast, Length: 14862, dtype: int64
```

```
#Number of tv shows directed by the directors
tv_new.drop_duplicates(subset=['title','director']).director.value_cou
nts().drop('Not-Available')
```

```
Alastair Fothergill      3
Ken Burns                3
Iginio Straffi           2
```

```
Gautham Vasudev Menon    2
Hsu Fu-chun              2
                        ..
Jesse Vile               1
Ellena Wood              1
Picky Talarico           1
Pedro Waddington         1
Michael Cumming          1
Name: director, Length: 299, dtype: int64
```

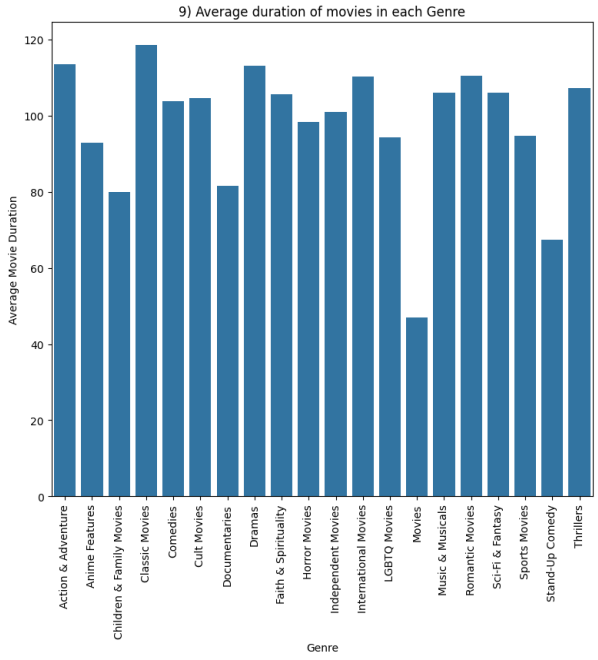
```
#Number of tv shows available in each country
tv_new.drop_duplicates(subset=['title','country']).country.value_counts().drop('Not-Available')
```

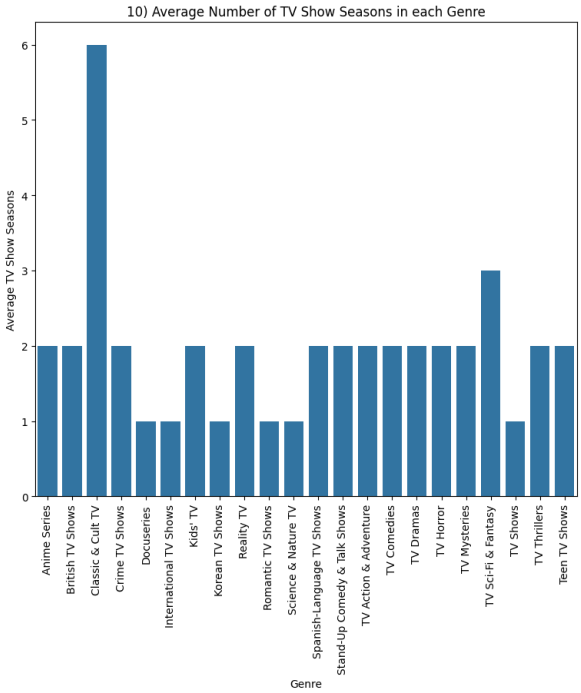
```
United States            937
United Kingdom           272
Japan                    199
South Korea              170
Canada                   126
                        ...
Malta                      1
United Arab Emirates       1
Belarus                    1
Uruguay                    1
Switzerland                1
Name: country, Length: 66, dtype: int64
```

VISUAL ANALYSIS

```
fig, axes = plt.subplots(1,2,figsize=(20,8))

#9) Univariate Analysis - Average duration of movies in each Genre
mv =
movie_new.drop_duplicates(subset=['title','listed_in']).groupby('listed_in').length_in_minutes.mean()
sns.barplot(x=mv.index,y=mv,ax=axes[0])
axes[0].set_xlabel('Genre')
axes[0].set_ylabel('Average Movie Duration')
axes[0].set_title('9) Average duration of movies in each Genre')
axes[0].tick_params(labelrotation=90,axis='x')

#10) Univariate Analysis - Average Number of TV Show Seasons in each Genre
t =
tv_new.drop_duplicates(subset=['title','listed_in']).groupby('listed_in').season.mean()
sns.barplot(x=t.index,y=round(t),ax=axes[1])
axes[1].set_xlabel('Genre')
axes[1].set_ylabel('Average TV Show Seasons')
axes[1].set_title('10) Average Number of TV Show Seasons in each
```

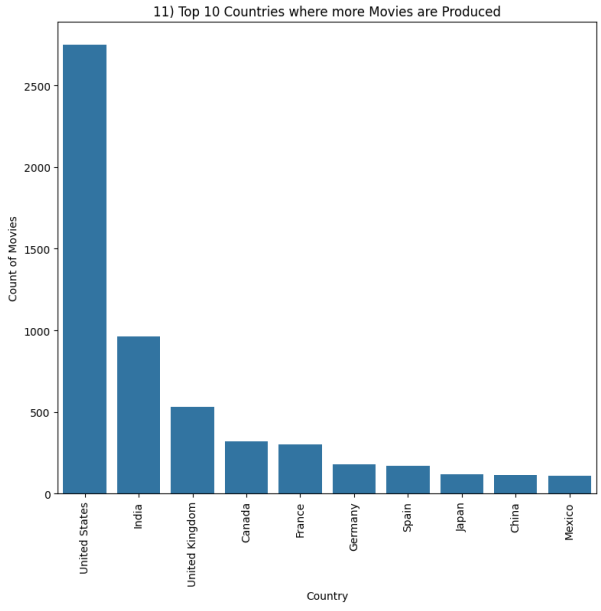
```
Genre')
axes[1].tick_params(labelrotation=90,axis='x')

plt.show()

#Insights for the below graphs:

#9) This shows the average duration of movies under each genres.
Movies in most of the Genres are between 80 to 120 minutes.
#10) This shows the average TV Show Seasons under each genres. TV
Shows in most of the Genres are made of either 1 or 2 seasons.
```

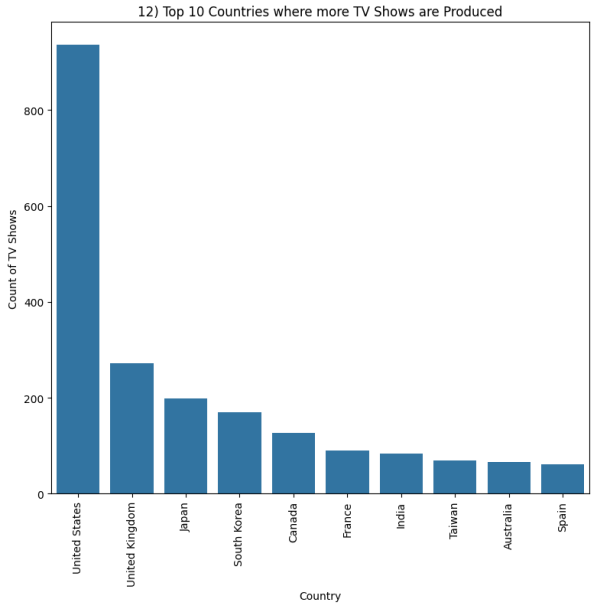





```
fig, axes = plt.subplots(1,2,figsize=(20,8))


#11) Univariate Analysis - Top 10 Countries where more Movies are
Produced
top10_cm_df = movie_new.drop_duplicates(subset=['title','country'])
top10_country = top10_cm_df['country'].value_counts().drop('Not-
Available').iloc[0:10].index
top10_country_movie_produced =
top10_cm_df[top10_cm_df['country'].isin(top10_country)]

sns.countplot(x='country',data=top10_country_movie_produced,order=top1
0_country_movie_produced['country'].value_counts(ascending=False).inde
x,ax=axes[0])
axes[0].set_xlabel('Country')
axes[0].set_ylabel('Count of Movies')
```

```
axes[0].set_title('11) Top 10 Countries where more Movies are
Produced')
axes[0].tick_params(labelrotation=90,axis='x')


#12) Univariate Analysis - Top 10 Countries where more TV Shows are
Produced
top10_ctv_df = tv_new.drop_duplicates(subset=['title','country'])
top10_country = top10_ctv_df['country'].value_counts().drop('Not-
Available').iloc[0:10].index
top10_country_tv_produced =
top10_ctv_df[top10_ctv_df['country'].isin(top10_country)]

sns.countplot(x='country',data=top10_country_tv_produced,order=top10_c
ountry_tv_produced['country'].value_counts(ascending=False).index,ax=a
xes[1])
axes[1].set_xlabel('Country')
axes[1].set_ylabel('Count of TV Shows')
axes[1].set_title('12) Top 10 Countries where more TV Shows are
Produced')
axes[1].tick_params(labelrotation=90,axis='x')

#Insights for the below graphs:
#11) More movies are produced in United States followed by India.
#12) More Tv Shows are produced in United States followed by United
Kingdom.
```





Co-Relation, PairPlot and HeatMaps

```
#Co-Relation: Measures the relationship between 2 numerical columns in the dataframe.
# We don't see much of the co-relation in this dataframe as there are lesser numerical columns.
# However, for movies dataframe we can see a slight negative co-relation between release_year and length_in_minutes meaning as the year passes the duration of the movie drops.
```

```
movie.corr()
```

```
<ipython-input-559-eb03073480ac>: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 value of numeric_only to silence this warning.
  movie.corr()
```


```
{"summary":"{\n  \"name\": \"movie\",\n  \"rows\": 3,\n  \"fields\": [\n    {\n      \"column\": \"release_year\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.6376325633629469,\n        \"min\": -0.20651967502726185,\n        \"max\": 1.0,\n        \"num_unique_values\": 3,\n        \"samples\": [\n          1.0,\n          -0.20651967502726185,\n          0.03902720086561585\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"length_in_minutes\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.6233237020661075,\n        \"min\": -0.20651967502726185,\n        \"max\": 1.0,\n        \"num_unique_values\": 3,\n        \"samples\": [\n          -0.20651967502726185,\n          1.0,\n          0.1250233650643551\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"added_year\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.5317342965266856,\n        \"min\": 0.03902720086561585,\n        \"max\": 1.0,\n        \"num_unique_values\": 3,\n        \"samples\": [\n          0.03902720086561585,\n          0.1250233650643551,\n          1.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


```
tv.corr()
```

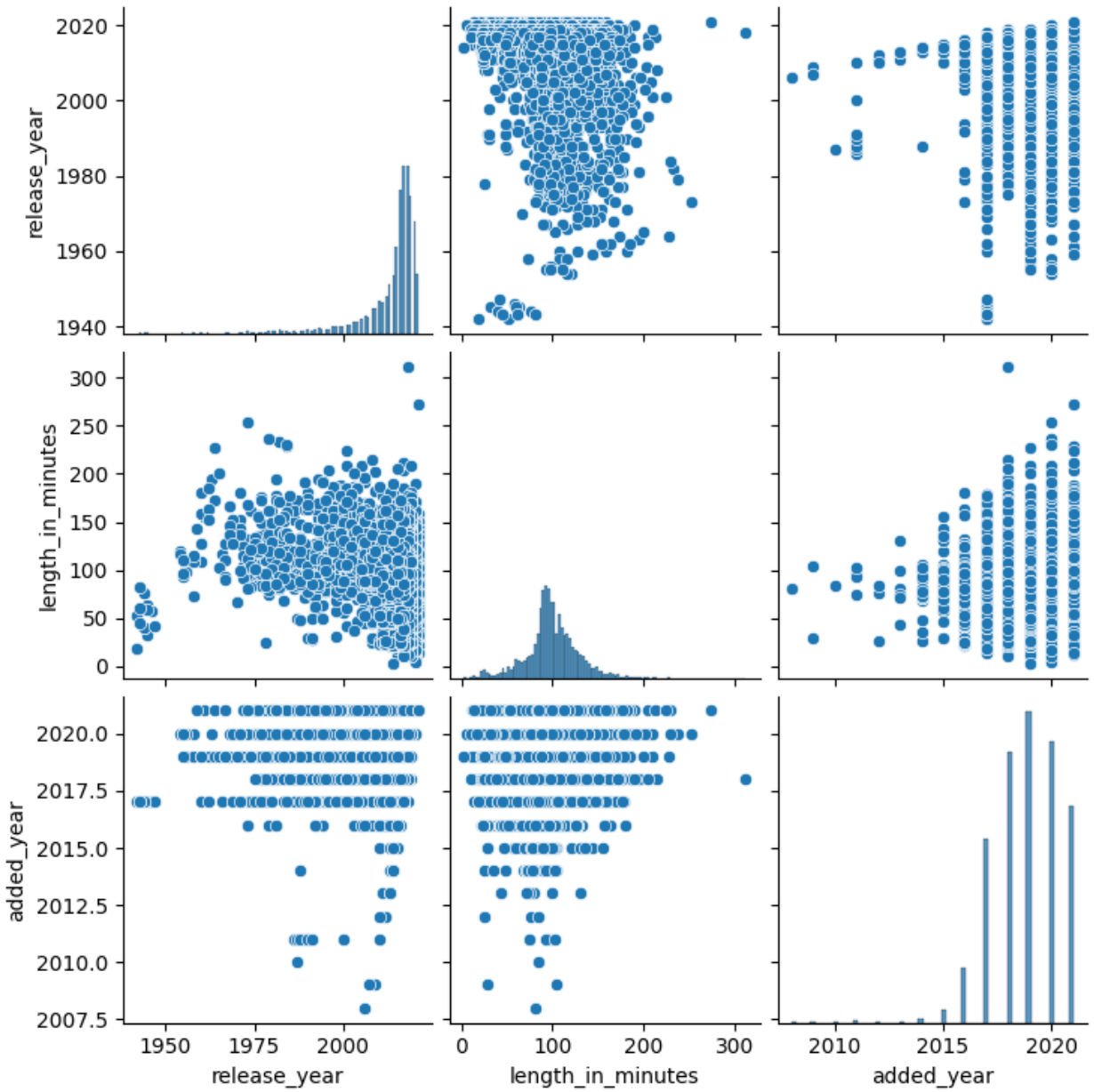
```
<ipython-input-560-e537718500ae>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  tv.corr()
```


```
{"summary":"{\n  \"name\": \"tv\",\n  \"rows\": 3,\n  \"fields\": [\n    {\n      \"column\": \"release_year\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.5463962452340321,\n
```

\"min\": -0.09027615655282478,\n   \"max\": 1.0,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   1.0,\n
-0.09027615655282478,\n   0.39067471694844724\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"season\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
0.6074931287403706,\n   \"min\": -0.09027615655282478,\n
\"max\": 1.0,\n   \"num_unique_values\": 3,\n   \"samples\":
[\n   -0.09027615655282478,\n   1.0,\n   -
0.009483995927890928\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"added_year\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.50834080334805,\n   \"min\": -
0.009483995927890928,\n   \"max\": 1.0,\n
\"num_unique_values\": 3,\n   \"samples\": [\n
0.39067471694844724,\n   -0.009483995927890928,\n   1.0\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

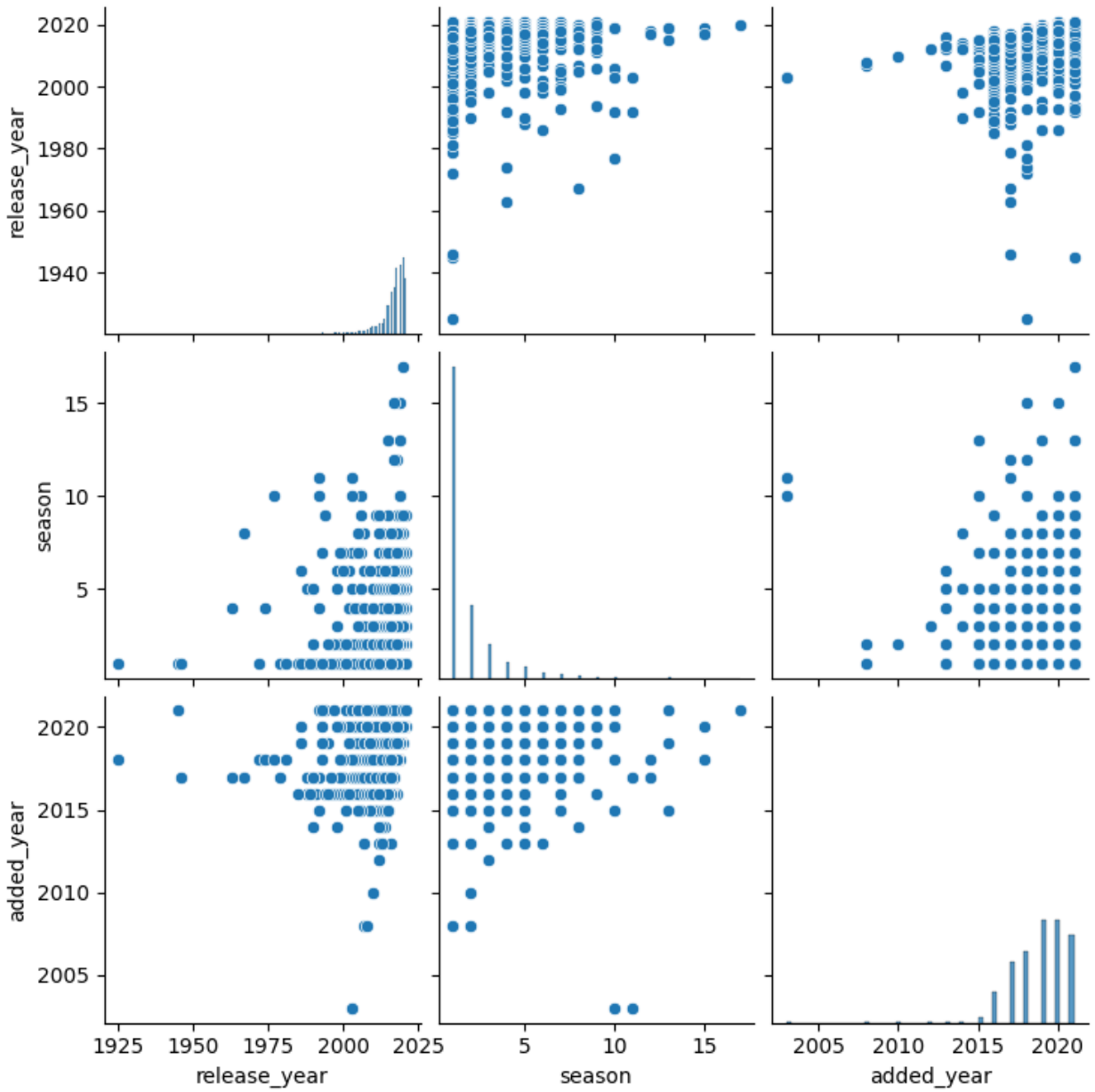
```
sns.pairplot(movie)
```

```
<seaborn.axisgrid.PairGrid at 0x786e320e0940>
```

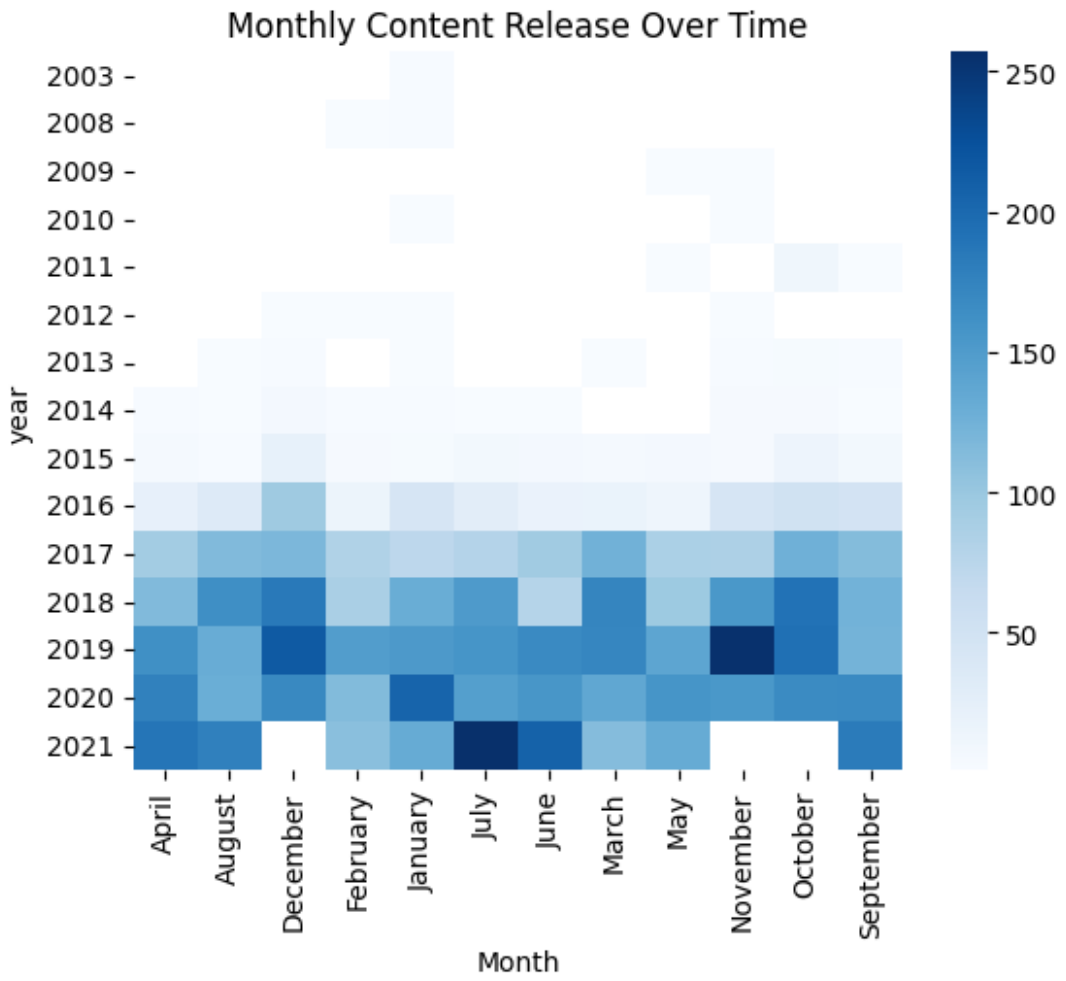



```
sns.pairplot(tv)
```

```
<seaborn.axisgrid.PairGrid at 0x786e329d6380>
```




```
#HeatMap
#This shows the volume of content released on Netflix over Month and
year
hm = df.drop_duplicates(subset=['type','title','added_year'])
hm =
hm.pivot_table(index='added_year',columns='added_month',values='title'
,aggfunc='count')
sns.heatmap(hm,cmap='Blues',cbar=True)
plt.xlabel('Month')
plt.ylabel('year')
plt.title('Monthly Content Release Over Time')
plt.show()
```



OVERALL INSIGHTS AND RECOMMENDATIONS

INSIGHTS:

At the moment in the dataset, we can see that there are more movies than tv shows. From the year 2016, there is an increase in the number of movies & tv shows but we can see a slight drop in 2021 for some reasons. More number of movies & tv shows come under 'TV-MA' rating. Movies released in the recent years have an average duration between 100 to 120 minutes and most of the tv shows produced have only 1 season. More movies are produced in United States followed by India and more Tv Shows are produced in United States followed by United Kingdom. More movies are released in the Genres - International Movies, Dramas & Comedies. More tv shows are released in the Genres - International TV Shows, TV Dramas & TV Comedies.

RECOMMENDATIONS:

Netflix should acquire and also produce more tv shows as well to attract more user base which helps Netflix in generating more revenue. As there are more tv shows with 1 season, netflix should make more limited sereis with 1 season and some shows with 2 seasons. Netflix should release mote movies which are around 2 hours in duration. Netflix should focus on releasing movies and tv shows of different genres. Netflix should also start acquiring more movies and tv shows from different countries to increase its user base and enrich its content.