Business Case: Netflix - Data Exploration and Visualisation

Business Problem

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

Basic Analysis :

- Importing Libraries
- Loading Dataset

```
## Importing DataSet
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv" -O netflix.csv
netflix_data = pd.read_csv('netflix.csv')

# https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
```

```
--2024-01-05 14:25:56--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 18.67.66.31, 18.67.66.49, 18.67.66.77, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|18.67.66.31|: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 0.02s

2024-01-05 14:25:56 (188 MB/s) - 'netflix.csv' saved [3399671/3399671]
```

Importing Libraries

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

- Checking data first and last five values

```
[218] netflix_data.head(5)
```

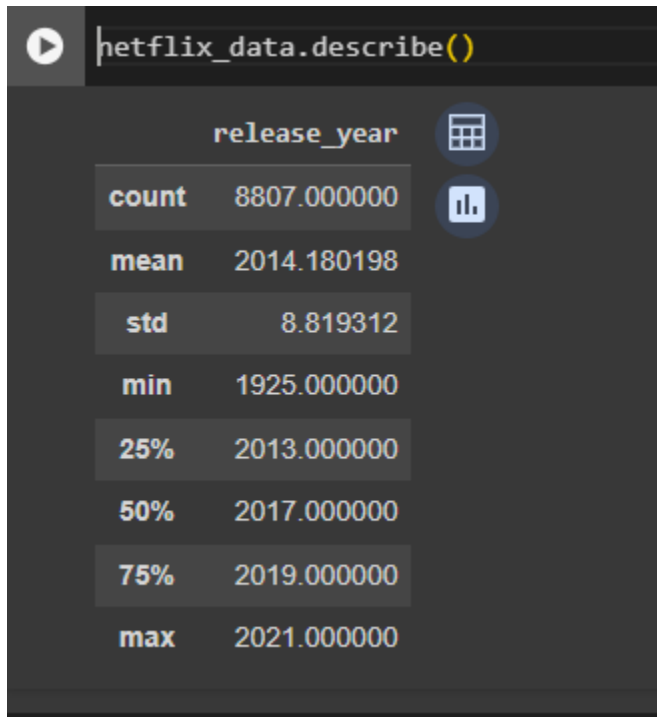|	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	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...
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...

Business Case: Netflix - Data Exploration and Visualisation

- 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

```
[88] netflix_data.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
```

```
netflix_data.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

Data Cleaning is required for identifying incorrect, incomplete, inaccurate, irrelevant, or missing pieces of data and then modifying, replacing, or deleting them as needed.

```
[95] netflix_data.isnull().any()

      show_id        False
      type           False
      title          False
      director        True
      cast            True
      country         True
      date_added      True
      release_year   False
      rating          True
      duration        True
      listed_in      False
      description    False
      dtype: bool
```

Total number of null values in each category

```
[90] netflix_data.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
```

Total number of null values preset inside dataframe

```
[96] netflix_data.isnull().sum().sum()

    4307
```

1. Un-nesting the columns

a. Un-nest the columns those have cells with multiple comma separated values by

creating multiple rows and Non-graphical analysis

Business Case: Netflix - Data Exploration and Visualisation

Unnesting title and Cast

```
[118] cast_df = netflix_data.loc[:,['title','cast']]
      cast_df.head()

      cast_df['cast'] = cast_df['cast'].str.split(',')
      cast_df.head()

      cast_df_unnest = cast_df.explode('cast')
      cast_df_unnest.head()
```

	title	cast
0	Dick Johnson Is Dead	NaN
1	Blood & Water	Ama Qamata
1	Blood & Water	Khosi Ngema
1	Blood & Water	Gail Mabalane
1	Blood & Water	Thabang Molaba

```
[215] cast_df_unnest['cast'].value_counts()

      Anupam Kher              39
      Rupa Bhimani             31
      Takahiro Sakurai         30
      Julie Tejwani            28
      Om Puri                  27
                               ..
      Vedika                    1
      Tedros Teclebrhan         1
      Maryam Zaree              1
      Melanie Straub            1
      Chittaranjan Tripathy     1
      Name: cast, Length: 39296, dtype: int64
```

Unnesting title and director

```
[196] dir_df = netflix_data.loc[:,['title','director']]
      dir_df.head()
```

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	NaN
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	NaN
4	Kota Factory	NaN

```
[▶] dir_df['director'] = dir_df['director'].str.split(',')
    dir_df.head()
```

	title	director
0	Dick Johnson Is Dead	[Kirsten Johnson]
1	Blood & Water	NaN
2	Ganglands	[Julien Leclercq]
3	Jailbirds New Orleans	NaN
4	Kota Factory	NaN

```
[198] dir_df_unnest = dir_df.explode('director')
      dir_df_unnest.head()
```

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	NaN
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	NaN
4	Kota Factory	NaN

```
[214] dir_df_unnest['director'].value_counts()
```

```
Rajiv Chilaka        22
Raúl Campos          18
 Jan Suter           18
Marcus Raboy         16
Suhas Kadav          16
                     ..
Will Eisenberg        1
Marina Seresesky      1
Kenny Leon            1
James Dearden         1
Mozez Singh           1
Name: director, Length: 5120, dtype: int64
```

Double-click (or enter) to edit

⌄ Unnesting title and country

```
[199] country_df = netflix_data.loc[:,['title','country']]
      country_df.head()

      country_df['country'] = country_df['country'].str.split(',')
      country_df.head()

      country_df_unnest = country_df.explode('country')
      country_df_unnest.head()
```

	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	NaN
3	Jailbirds New Orleans	NaN
4	Kota Factory	India

```
[213] country_df_unnest['country'].value_counts()
```

```
United States       3211
India               1008
United Kingdom       628
 United States       479
Canada               271
                     ...
 Ecuador               1
Iran                   1
Cyprus                 1
 Mongolia              1
 Montenegro            1
Name: country, Length: 197, dtype: int64
```

⌄ Unnesting title and Listed_In(genre)

```
[200] genre_df = netflix_data.loc[:,['title','listed_in']]
      genre_df.head()

      genre_df['listed_in'] = genre_df['listed_in'].str.split(',')
      genre_df.head()

      genre_df_unnest = genre_df.explode('listed_in')
      genre_df_unnest.head()
```

	title	listed_in
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows
1	Blood & Water	TV Dramas
1	Blood & Water	TV Mysteries
2	Ganglands	Crime TV Shows

```
[216] genre_df_unnest['listed_in'].value_counts()

       International Movies      2624
Dramas                          1600
Comedies                        1210
Action & Adventure               859
Documentaries                    829
                                 ...
Romantic Movies                    3
Spanish-Language TV Shows          2
LGBTQ Movies                       1
TV Sci-Fi & Fantasy                1
Sports Movies                      1
Name: listed_in, Length: 73, dtype: int64
```

2. Handling null values

Can use mean, mode, or use predictive modelling. In this case study, we will discuss the use of the fillna function from Pandas for this imputation. Drop rows containing missing values. Can use the dropna function from Pandas

```
Values before fixing null values

[151]

    value_cnt = netflix_final_df['title'].count()
    print(f"\nTotal Value counts for title : {value_cnt}")

    value_cnt = netflix_final_df['cast'].count()
    print(f"\nTotal Value counts for cast : {value_cnt}")

    value_cnt = netflix_final_df['director'].count()
    print(f"\nTotal Value counts for director : {value_cnt}")

    value_cnt = netflix_final_df['country'].count()
    print(f"\nTotal Value counts for title : {value_cnt}")

    value_cnt = netflix_final_df['listed_in'].count()
    print(f"\nTotal Value counts for listed_in : {value_cnt}")

    value_cnt = netflix_final_df['type'].count()
    print(f"\nTotal Value counts for type : {value_cnt}")

    value_cnt = netflix_final_df['date_added'].count()
    print(f"\nTotal Value counts for date_added : {value_cnt}")

    value_cnt = netflix_final_df['release_year'].count()
    print(f"\nTotal Value counts for release_year : {value_cnt}")

    value_cnt = netflix_final_df['rating'].count()
    print(f"\nTotal Value counts for rating : {value_cnt}")
```

- The counts of each categorical variable
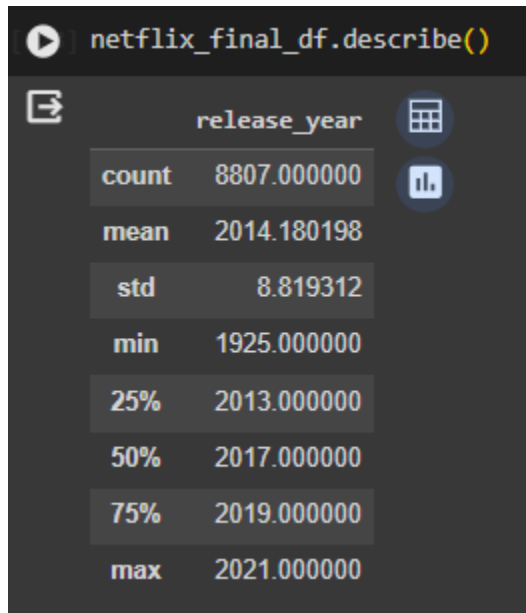
```
Total Value counts for title : 8807

Total Value counts for cast : 7982

Total Value counts for director : 6173

Total Value counts for title : 7976

Total Value counts for listed_in : 8807

Total Value counts for type : 8807

Total Value counts for date_added : 8797

Total Value counts for release_year : 8807

Total Value counts for rating : 8803
```

For missing value we are filling it by some pre-defined data which will make out task easier for giving some insights.

```
[191]
netflix_final_df.director.fillna("No Director", inplace=True)
netflix_final_df.cast.fillna("No Cast", inplace=True)
netflix_final_df.country.fillna("Country Unavailable", inplace=True)
netflix_final_df.dropna(subset=["date_added", "rating"], inplace=True)
netflix_final_df.head(3)
```

	title	cast	director	country	listed_in	show_id	type	date_added	release_year	rating	duration
0	Dick Johnson Is Dead	No Cast	[Kirsten Johnson]	[United States]	[Documentaries]	s1	Movie	September 25, 2021	2020	PG-13	90 min
1	Blood & Water	[Ama Qamata, Khosi Ngema, Gail Mabalane, Th...	No Director	[South Africa]	[International TV Shows, TV Dramas, TV Myste...	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
2	Ganglands	[Sami Bouajila, Tracy Gotoas, Samuel Jouy, ...	[Julien Leclercq]	Country Unavailable	[Crime TV Shows, International TV Shows, TV ...	s3	TV Show	September 24, 2021	2021	TV-MA	1 Season
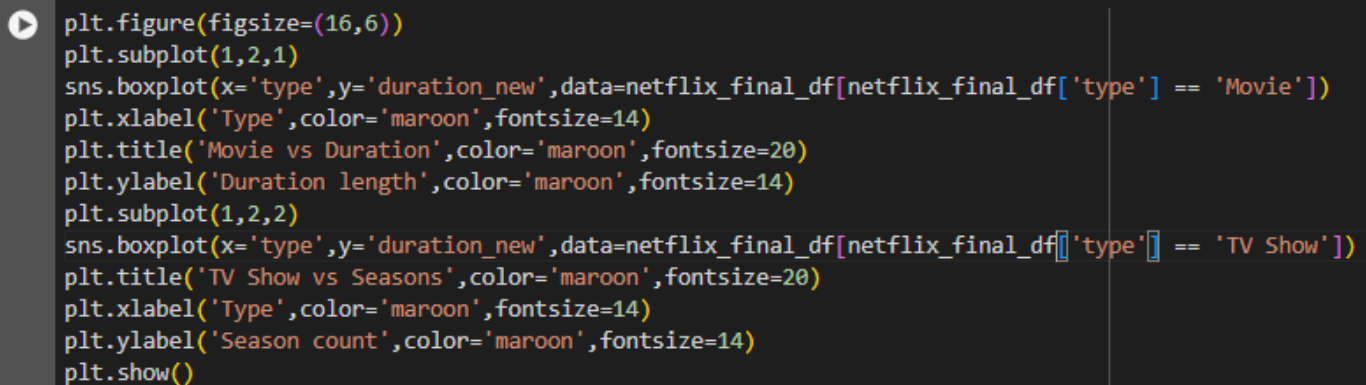
```
netflix_final_df.describe()
```

|	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

2. Comparison of tv shows vs. movies.

Analysis entire Netflix dataset consisting of both movies and shows. We can compare the total number of movies and shows in this dataset and get some insights of popularity .

```python
plt.figure(figsize=(16,6))
plt.subplot(1,2,1)
sns.boxplot(x='type',y='duration_new',data=netflix_final_df[netflix_final_df['type'] == 'Movie'])
plt.xlabel('Type',color='maroon',fontsize=14)
plt.title('Movie vs Duration',color='maroon',fontsize=20)
plt.ylabel('Duration length',color='maroon',fontsize=14)
plt.subplot(1,2,2)
sns.boxplot(x='type',y='duration_new',data=netflix_final_df[netflix_final_df['type'] == 'TV Show'])
plt.title('TV Show vs Seasons',color='maroon',fontsize=20)
plt.xlabel('Type',color='maroon',fontsize=14)
plt.ylabel('Season count',color='maroon',fontsize=14)
plt.show()
```
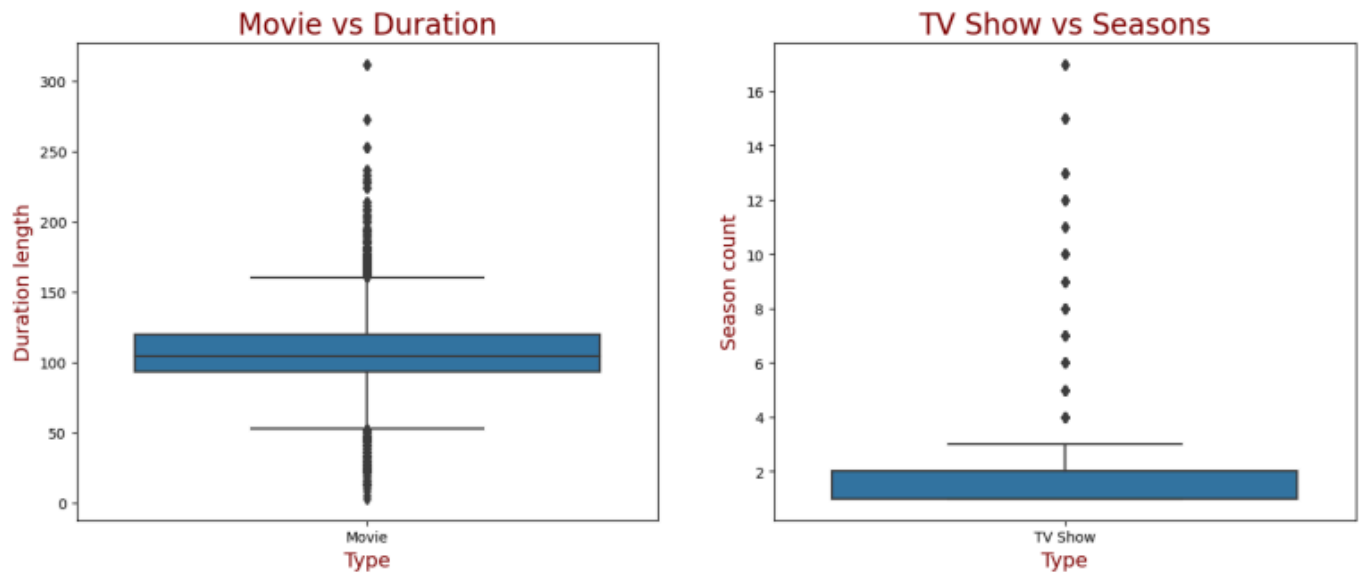
Analyzing the movie box plot, we can see that most movies fall within a reasonable duration range, with few outliers exceedingly approximately 120 minutes. This suggests that most movies on Netflix are designed to fit within a standard viewing time. For TV shows, the box plot reveals that most shows have one to three seasons, with very few outliers having longer durations. This aligns with the earlier trends, indicating that Netflix focuses on shorter series formats.

```
sns.countplot(x=netflix_final_df['rating'],order = netflix_final_df['rating'].value_counts().index)
plt.xticks(rotation=90,fontsize=10)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16]),
 [Text(0, 0, 'TV-MA'),
  Text(1, 0, 'TV-14'),
  Text(2, 0, 'TV-PG'),
  Text(3, 0, 'R'),
  Text(4, 0, 'PG-13'),
  Text(5, 0, 'TV-Y7'),
  Text(6, 0, 'TV-Y'),
  Text(7, 0, 'PG'),
  Text(8, 0, 'TV-G'),
  Text(9, 0, 'NR'),
  Text(10, 0, 'G'),
  Text(11, 0, 'TV-Y7-FV'),
  Text(12, 0, 'NC-17'),
  Text(13, 0, 'UR'),
  Text(14, 0, '74 min'),
  Text(15, 0, '84 min'),
  Text(16, 0, '66 min')])
```
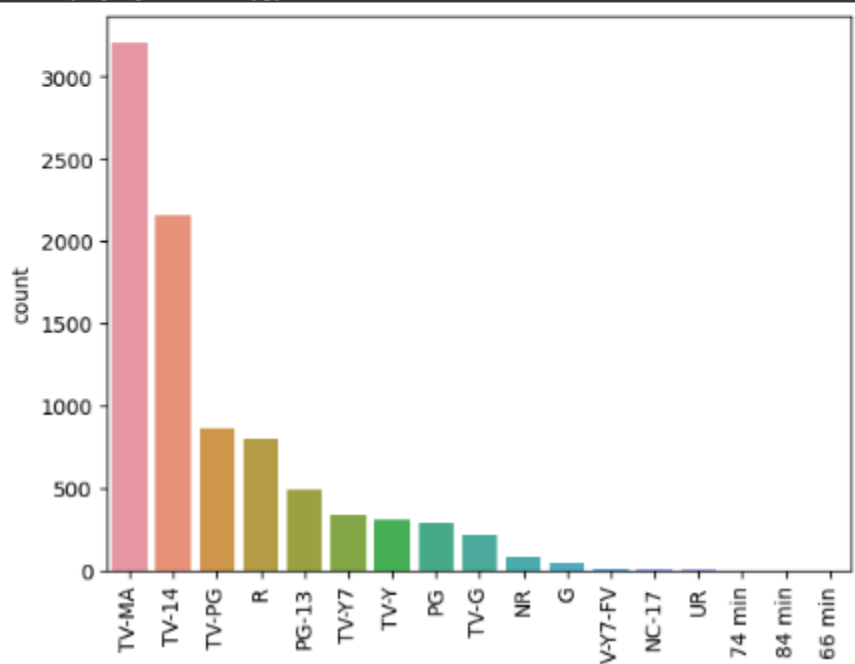

Trend in rated Movies/TV shows over the years

```
plt.figure(figsize=(12,6))
sns.lineplot(x='release_year',y='TotalCount',hue='rating',data=netflix_final_df[(netflix_final_df['release_year']>2000)])
plt.title('Trend in Rated Movies/TV Shows over Years',color='maroon',fontsize=20)
plt.xlabel('Year Released',color='maroon',fontsize=14)
plt.ylabel('Total Count',color='maroon',fontsize=14)
plt.show()
```
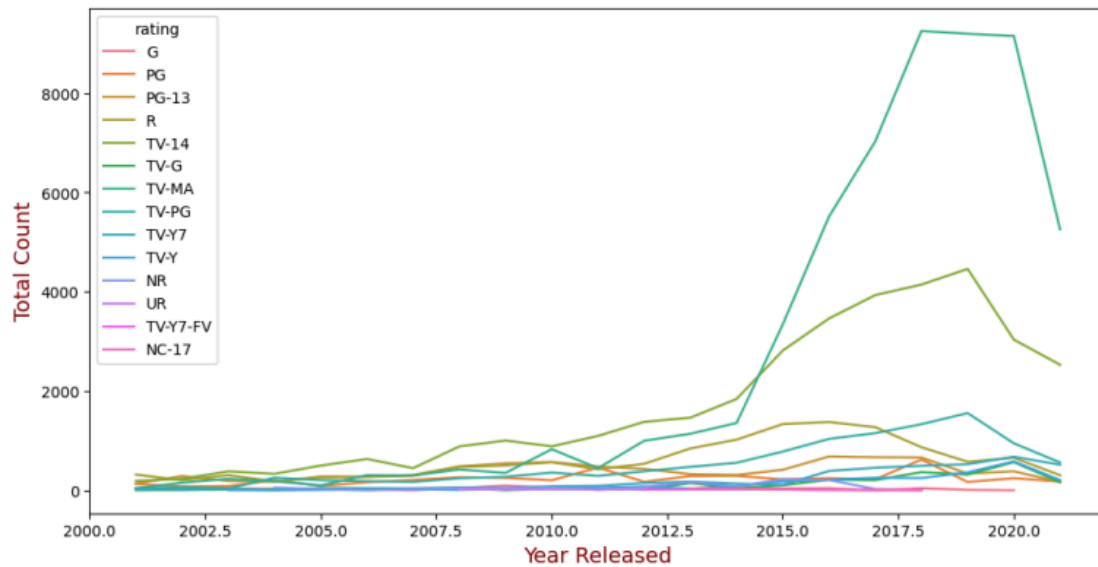
```
[233] plt.figure(figsize=(6,3))
     plt.title("Percentation of Netflix Titles that are either Movies or TV Shows")
     g=plt.pie(netflix_final_df.type.value_counts(),explode=(0.025,0.025),
     labels=netflix_final_df.type.value_counts().index, colors=['red','pink'],autopct='%1.1f%%',startangle=180)
     plt.show()
```
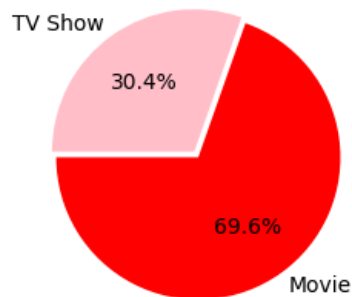

We can conclude that there are more movie titles (69.7%) that TV shows titles (30.3%).

we will explore the amount of content Netflix has added throughout the previous years. Since we are interested in when Netflix added the title onto their platform, we will add a "year_added" column to show the date from the "date_added" columns

```
[234] netflix_final_df["year_added"] = pd.to_date0me(netflix_final_df.date_added).dt.year
      netflix_movies_df["year_added"] = pd.to_date0me(netflix_movies_df.date_added).dt.year
      netflix_shows_df["year_added"] = pd.to_date0me(netflix_shows_df.date_added).dt.year
      netflix_year_df = netflix_final_df.year_added.value_counts().to_frame().reset_index().rename(columns={"index": "year","year_added":"count"})
      netflix_year_df = netflix_year_df[netflix_year_df.year != 2020]
      print(netflix_year_df)
```

	year	count
0	2019	2016
2	2018	1648
3	2021	1498
4	2017	1185
5	2016	426
6	2015	82
7	2014	24
8	2011	13
9	2013	11
10	2012	3
11	2009	2
12	2008	2
13	2010	1

- Find the number of movies produced in each country and pick the top 10
  countries. Exploring the countries contribution with the most content of Netflix.

```
import plotly.graph_objects as go
from plotly.offline import init_notebook_mode, iplot
filtered_countries = netflix_final_df.set_index('title').country.str.split(', ',
expand=True).stack().reset_index(level=1, drop=True);
filtered_countries = filtered_countries[filtered_countries != 'Country Unavailable']
iplot([go.Choropleth(
locationmode='country names',
locations=filtered_countries,
z=filtered_countries.value_counts()
)])
```

Top Directors on Netflix

```python
from wordcloud import WordCloud, ImageColorGenerator
text = " ".join(str(each) for each in netflix_final_df.director)
# Create and generate a word cloud image:
wordcloud = WordCloud(max_words=200, background_color="gray").generate(text)
plt.figure(figsize=(10,6))
plt.figure(figsize=(15,10))
# Display the generated image:
plt.imshow(wordcloud, interpolation='Bilinear')
plt.title('Most Popular Directors',fontsize = 30)
plt.axis("off")
plt.show()
```
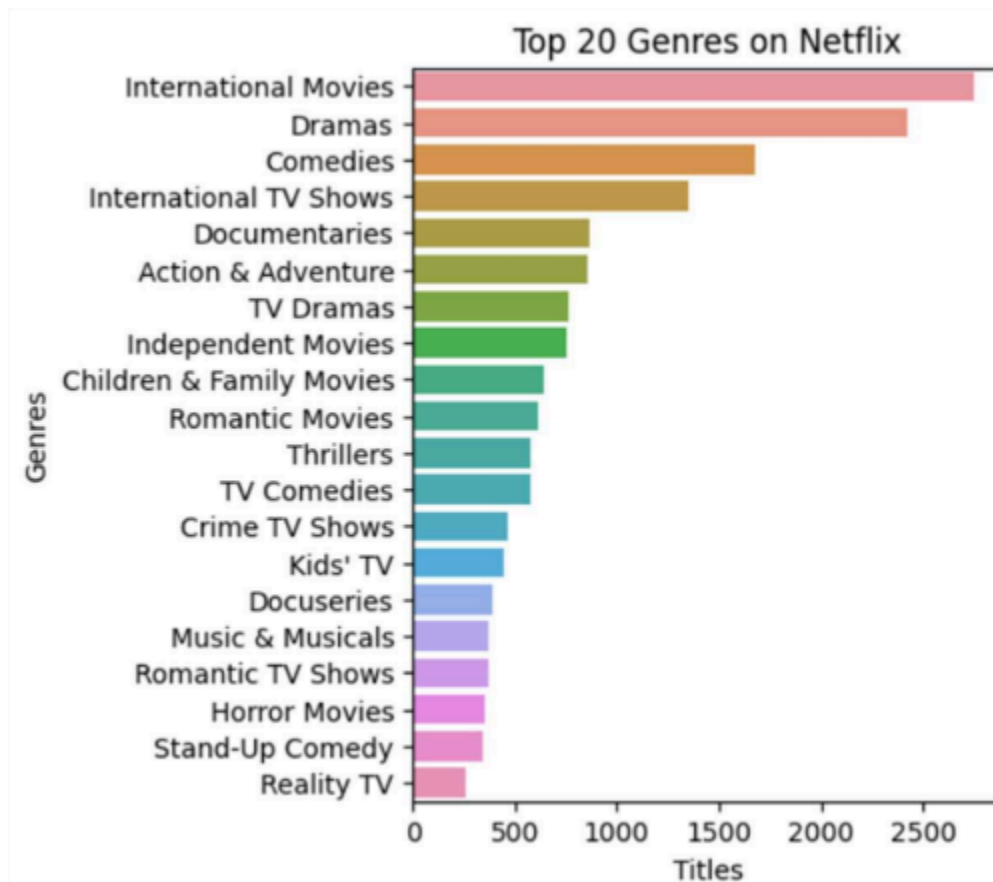
`<Figure size 1000x600 with 0 Axes>`
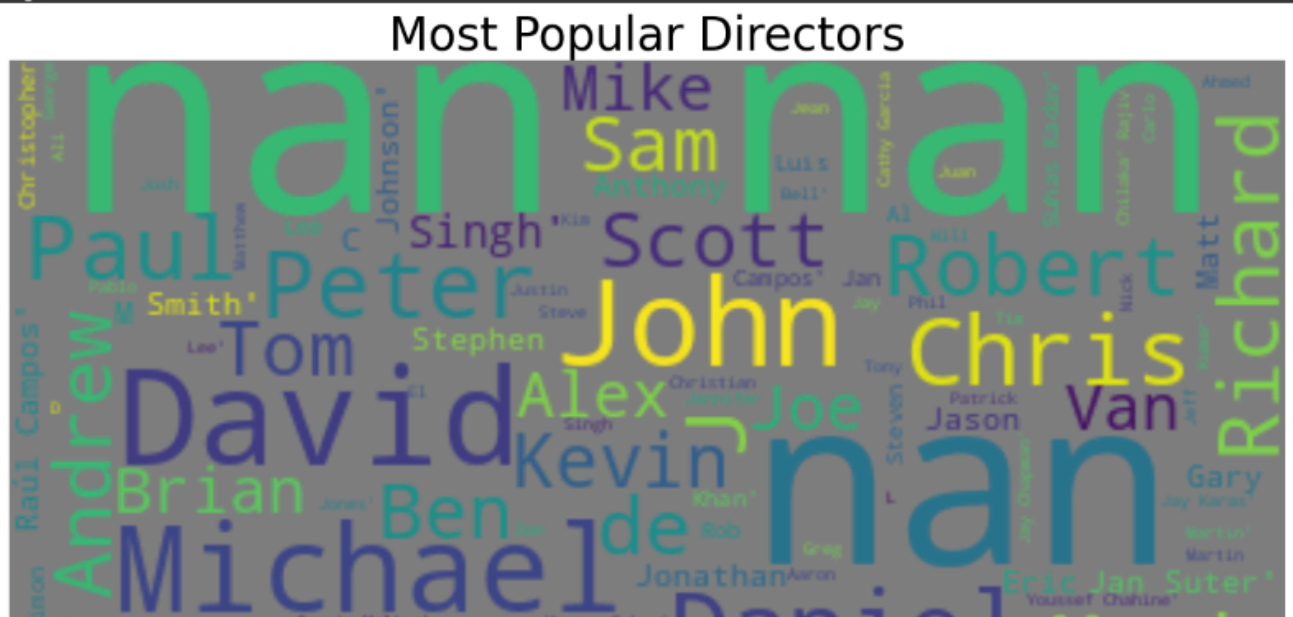
Top 20 Genres on Netflix: Count Plot

```python
filtered_genres = netflix_final_df.set_index('title').listed_in.str.split(', ',expand=True).stack().reset_index(level=1, drop=True);
plt.figure(figsize=(4,5))
g = sns.countplot(y = filtered_genres,
order=filtered_genres.value_counts().index[:20])
plt.title('Top 20 Genres on Netflix')
plt.xlabel('Titles')
plt.ylabel('Genres')
plt.show()
```

Business Case: Netflix - Data Exploration and Visualisation

- Here we are trying to find top 5 director with respect to their number of movies added to Netflix to check how and when Netflix started approaching the directors and started making content through them.

```python
top5_director = netflix_final_df['director'].value_counts().index[:5]
df1=netflix_final_df.loc[(netflix_final_df["director"].isin(top5_director))]
df2=df1.groupby(['director','date_added_year','type']).size().unstack(fill_value=0).reset_index()
Text(0.5, 1.0, 'TV Show per year')
plt.figure(figsize=(12, 8))
plt.subplot(1,2,1)
sns.barplot(x='date_added_year',y='Movie',data=df2,hue='director')
plt.xlabel('Year added')
plt.title('Movies per year')
plt.legend(bbox_to_anchor=(0, 1), loc='upper left', borderaxespad=0.,ncol=2)
plt.subplot(1,2,2)
sns.barplot(x='date_added_year',y='TV Show',data=df2,hue='director')
plt.xlabel('Year added')
plt.legend(bbox_to_anchor=(0, 1), loc='upper left', borderaxespad=0.,ncol=2)
plt.title('TV Show per year')
```
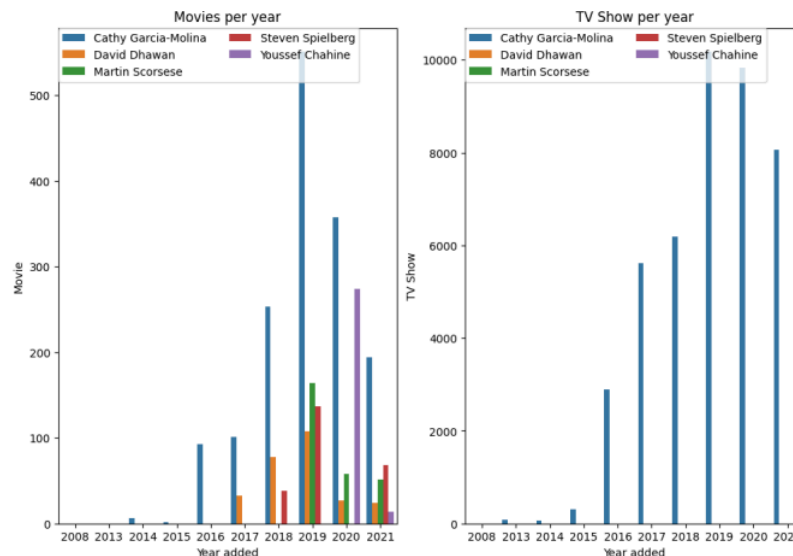

From the above graph we could clearly see that director Cathy Garcia-Molina dominates other directors in both directing Movie and TV Show. in 2019 she delivered more than 500 movies unbeaten by any other directors. Martin Scorsese came to picture in 2019 where most of his pictures were added to streaming platform

- Top Actors in Top Genre and their count

```
top3_genre = netflix_final_df['listed_in'].value_counts().index[:3]
df3=netflix_final_df.loc[(netflix_final_df['listed_in'].isin(top3_genre))]
df4=df3.groupby(['cast','listed_in']).size().unstack(fill_value=0).reset_index()
top5_actors_comedy=df4.sort_values('Comedies',ascending=False).head(5)['cast']
top5_actors_dramas=df4.sort_values('Dramas',ascending=False).head(5)['cast']
top5_actors_internationalmovies=df4.sort_values('International Movies',ascending=False).head(5)['cast']
top5_data=df4[df4['cast'].isin(top5_actors_comedy) | df4['cast'].isin(top5_actors_dramas) | df4['cast'].isin(top5_actors_internationalmovies)]
plt.figure(figsize=(20,6))
plt.subplot(1,3,1)
sns.barplot(x='cast',y='Comedies',data=top5_data)
plt.xlabel('Cast')
plt.title('Top Casts in Comedies')
plt.xticks(rotation=80)
plt.subplot(1,3,2)
sns.barplot(x='cast',y='Dramas',data=top5_data)
plt.xlabel('Cast')
plt.title('Top Casts in Dramas')
plt.xticks(rotation=80)
plt.subplot(1,3,3)
sns.barplot(x='cast',y='International Movies',data=top5_data)
plt.xlabel('Cast')
plt.title('Top Casts in International Movies')
plt.xticks(rotation=80)
plt.show()
```

- Top 4 Genre content produced over the Years

```
df5 = netflix_final_df.groupby(['release_year', 'listed_in']).size().reset_index(name='Totalcount')
top4_listed_in = df5['listed_in'].value_counts().index[:4]
top4_data = df5.loc[(df5["listed_in"].isin(top4_listed_in))]
plt.figure(figsize=(18, 6))
plt.xticks(rotation=50)
sns.lineplot(data=top4_data[top4_data['release_year']>2001], x='release_year',y='Totalcount', hue='listed_in')
plt.title('Trend of Genre produced per year',color='maroon',fontsize=20)
plt.xlabel('Year Released',color='maroon',fontsize=14)
plt.ylabel('Total Count',color='maroon',fontsize=14)
plt.legend(bbox_to_anchor=(0, 1), loc='upper left', fontsize='large',ncol=4)
plt.show()
```
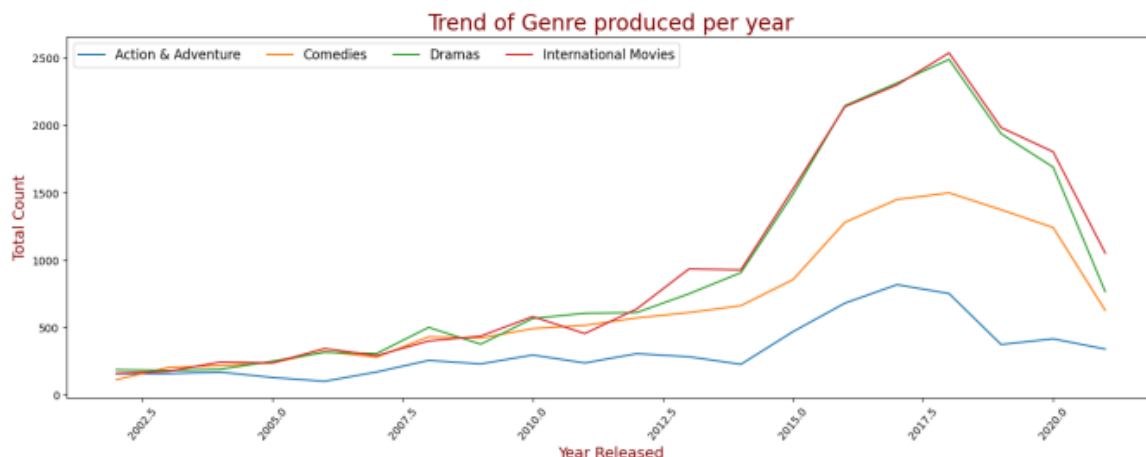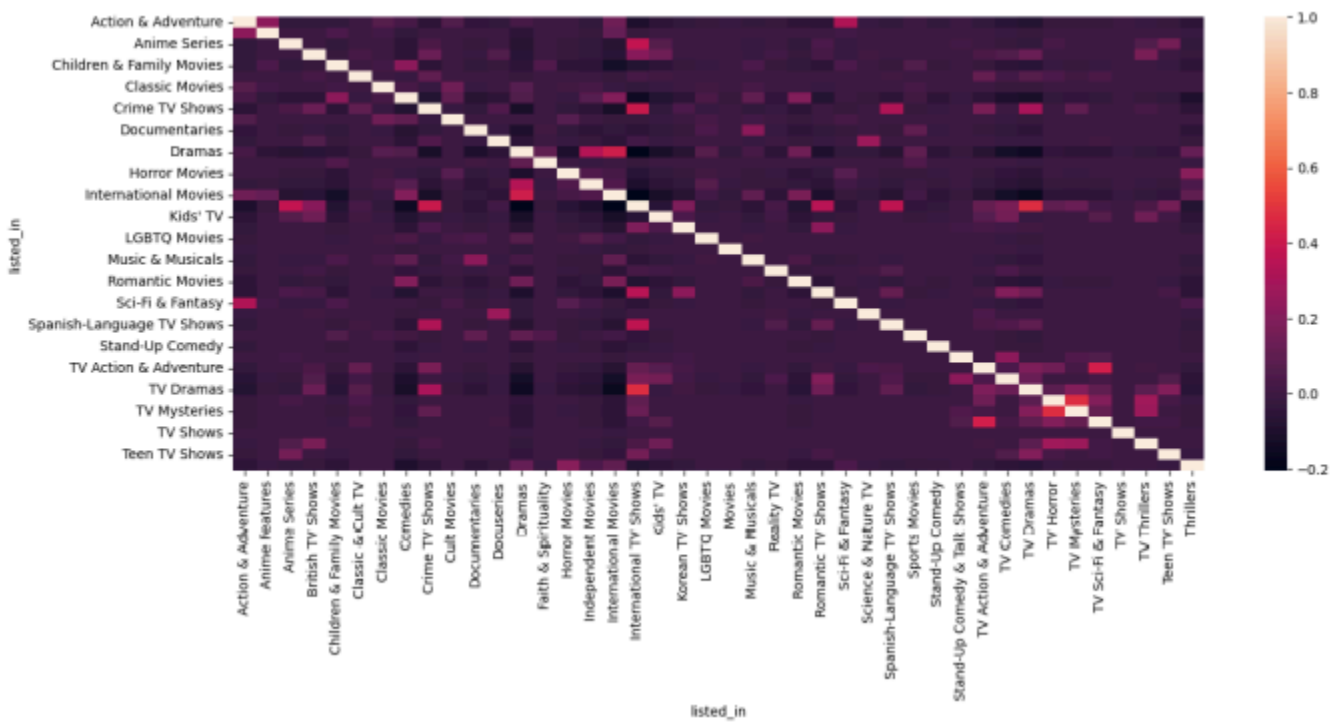
- Heatmap for all Genre columns

```
df6=netflix_final_df.groupby(['show_id','listed_in']).size().unstack(fill_value=0)
plt.figure(figsize=(16,6))
sns.heatmap(df6.corr())
plt.show()
```

Business Case: Netflix - Data Exploration and Visualisation

Business Insights: With the help of this article, we have been able to learn about

1. Quantity: Our analysis revealed that Netflix had added more movies than TV shows, aligning with the expectation that movies dominate their content library.
2. Content Addition: July emerged as the month when Netflix adds the most content, closely followed by December, indicating a strategic approach to content release.
3. Genre Correlation: Strong positive associations were observed between various genres, such as TV dramas and international TV shows, romantic and international TV shows, and independent movies and dramas. These correlations provide insights into viewer preferences and content interconnections.
4. Movie Lengths: The analysis of movie durations indicated a peak around the 1960s, followed by a stabilization around 100 minutes, highlighting a trend in movie lengths over time.
5. TV Show Episodes: Most TV shows on Netflix have one season, suggesting a preference for shorter series among viewers.
6. Common Themes: Words like love, life, family, and adventure were frequently found in titles and descriptions, capturing recurring themes in Netflix content.
7. Rating Distribution: The distribution of ratings over the years offers insights into the evolving content landscape and audience reception.
8. Data-Driven Insights: Our data analysis journey showcased the power of data in unravelling the mysteries of Netflix's content landscape, providing valuable insights for viewers and content creators.
9. Continued Relevance: As the streaming industry evolves, understanding these patterns and trends becomes increasingly essential for navigating the dynamic landscape of Netflix and its vast library.
10. Happy Streaming: We hope this blog has been an enlightening and entertaining journey into the world of Netflix, and we encourage you to explore the captivating stories within its ever-changing content offerings. Let the data guide your streaming adventures!

RECOMMENDATIONS

- Netflix has to focus on TV Shows also because there are people who will like to see tv shows rather than movies and by releasing at the analyzed time could help viewership revenue by 18-20%
- By approaching the top director, we can plan some more movies/tv shows in order to increase the popularity

Business Case: Netflix - Data Exploration and Visualisation

- Not only reaching top director we can also see the director with less no of movies and having high rating as there may be some financial issues or anything so in order to get good content Netflix can reach to them and Netflix can produce the movie and give the director a chance.
- We have seen most no of international movies genre so need to give priority to other genre like horror comedy..etc
- In TV Shows we may focus on thriller genre which will be helpful for having more no of seasons
- Most of the movies released in OTT is in a year 2019 so we need to go on increasing this value in order to attract people by showing that
- getting subscription is useful as Netflix is releasing more movies per year
- Mainly the release in OTT should focus on the festival holidays, year end and weekends which is to be mainly focused
- Some movies can be released directly into OTT which has some positive talk which may help in improving subscriptions
- Should focus on a actor who has immense following and make use of it by doing a TV Shows or web series
- Advertisement in the country which has very less movies released should be increased and attract people of that country by making their native TV Shows