Netflix is an American subscription video on-demand over-the-top streaming service. Launched on January 16, 2007, nearly a decade after Netflix, Inc. began its pioneering DVD-by-mail movie rental service, Netflix is the most-subscribed video on demand streaming media service, with 260.28 million paid memberships in more than 190 countries as of January 2024. Current stock price:
NFLX NASDAQ $562.06 -2.58 -0.46% as of 06 Feb 2024

Business Problem

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

Importing Libraries:

1. **Defining Problem Statement and Analysing basic metrics Import Libraries Importing the libraries we need.**
   import numpy as np
   import pandas as pd
   import matplotlib
   import matplotlib.pyplot as plt
   import seaborn as sns

2. **Loading The Datase**

```
netlix_df = pd.read_csv"netflix.csv"
```

Data Exploration

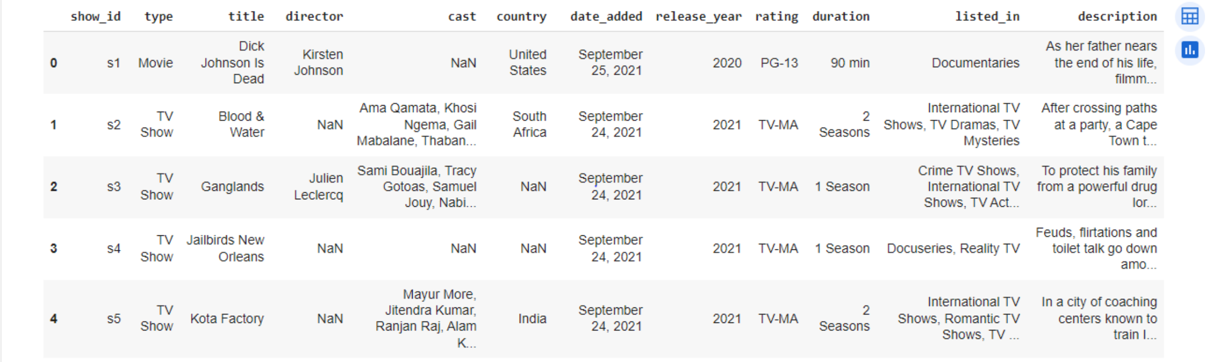
1. **Checking missing values**

```
netlix_df = pd.read_csv"netflix.csv"
missing_values = netlix_df.isnull.sum
printmissing_values
```

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

2. Top 5 data check

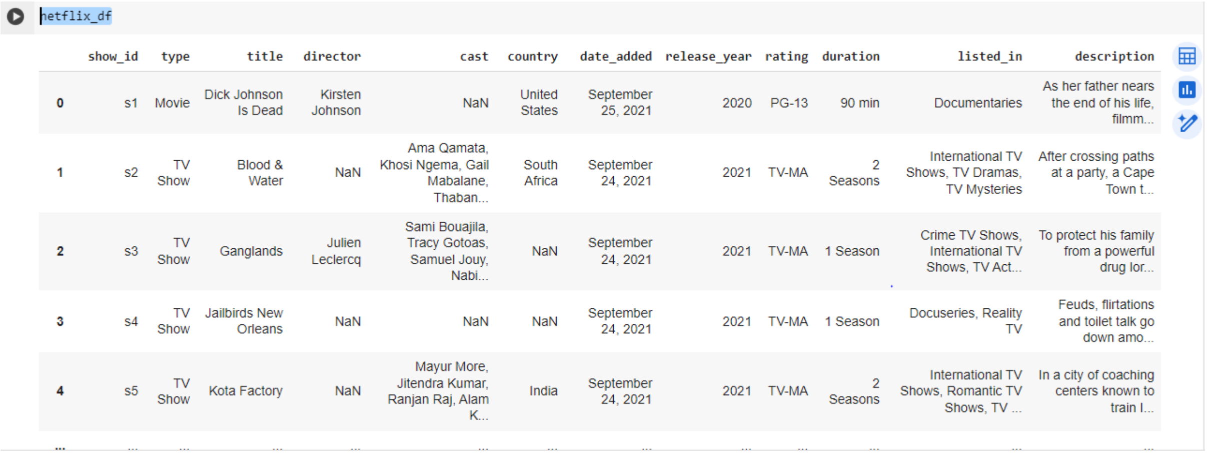
```
netflix_df.head
```



	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...

```
netflix_df
```

3. The dataset contains over 8807 titles, 12 descriptions. After a quick view of the data frames, it looks like a typical movie/TV shows data frame without ratings. We can also see that there are NaN values in some columns.



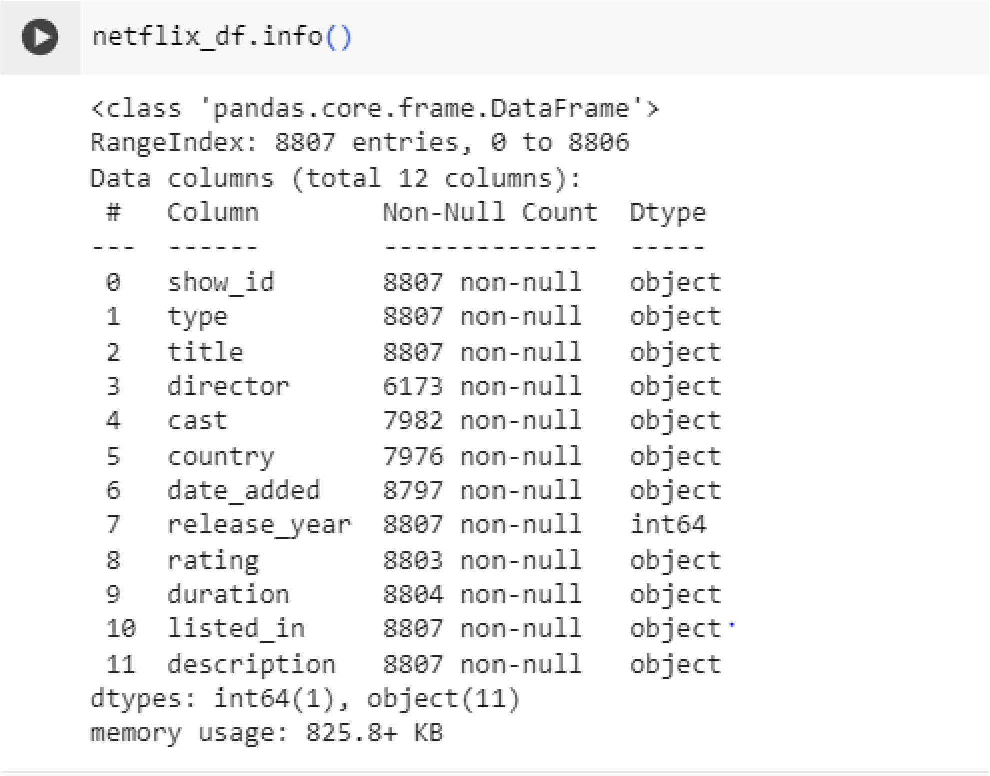
```
netflix_df
```

	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...
...	...	...	...	...	...	...	...	...	...	...	...	...

4. Summary statistics

```
printnetflix_df.describe
```

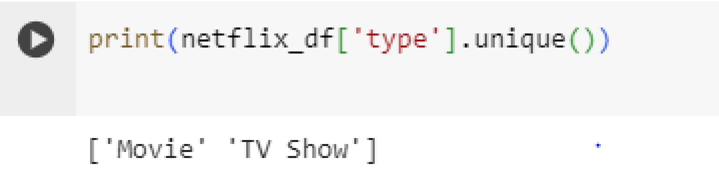
```
release_year
count    8807.000000
mean     2014.180198
std         8.819312
min      1925.000000
25%      2013.000000
50%      2017.000000
75%      2019.000000
   max      2021.000000
```



```
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
```

5.



```
print(netflix_df['type'].unique())
```

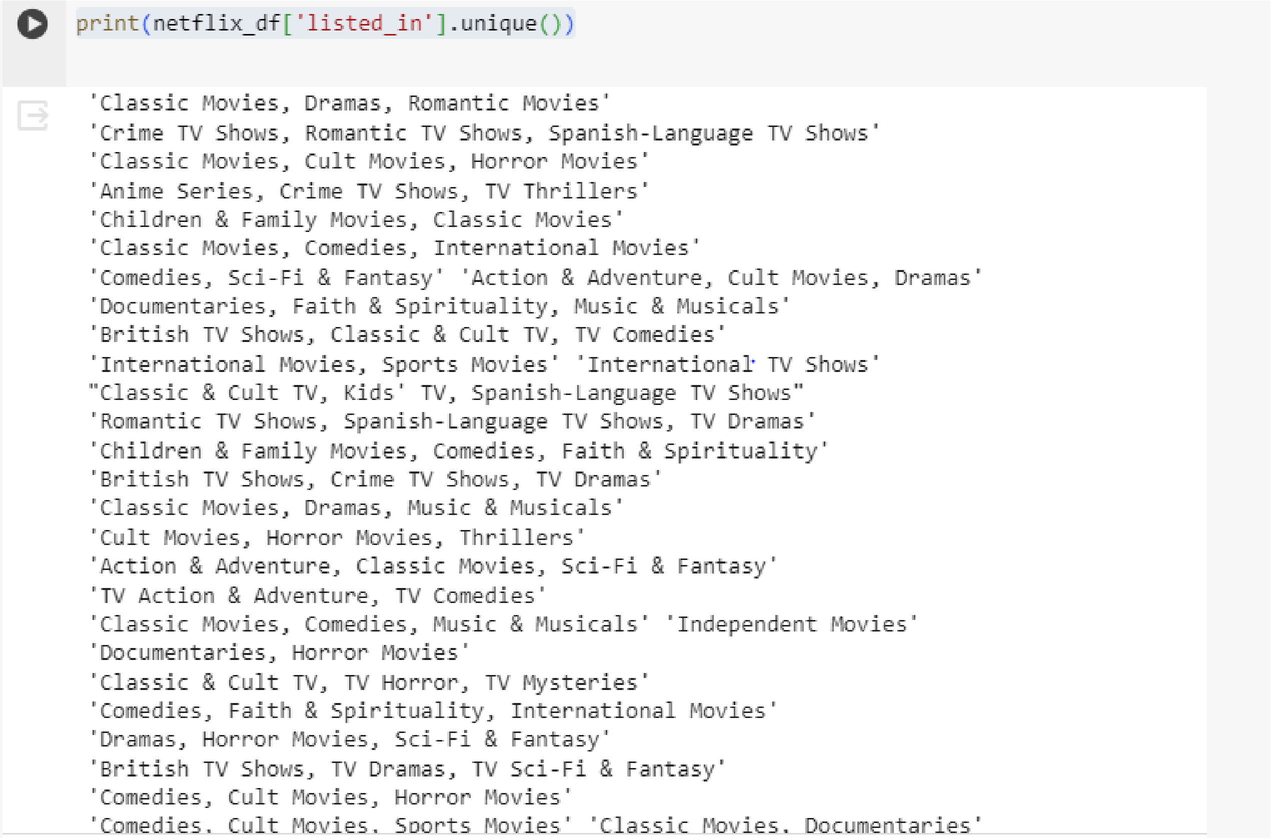
```
['Movie' 'TV Show']
```

There are 2 types in the data movie and TV shows

6. Rating: `printnetflix_df['rating'].unique`

```
['PG-13' 'TV-MA' 'PG' 'TV-14' 'TV-PG' 'TV-Y' 'TV-Y7' 'R' 'TV-G' 'G'
 'NC-17' '74 min' '84 min' '66 min' 'NR' nan 'TV-Y7-FV' 'UR']
```

7. `printnetflix_df['listed_in'].unique`

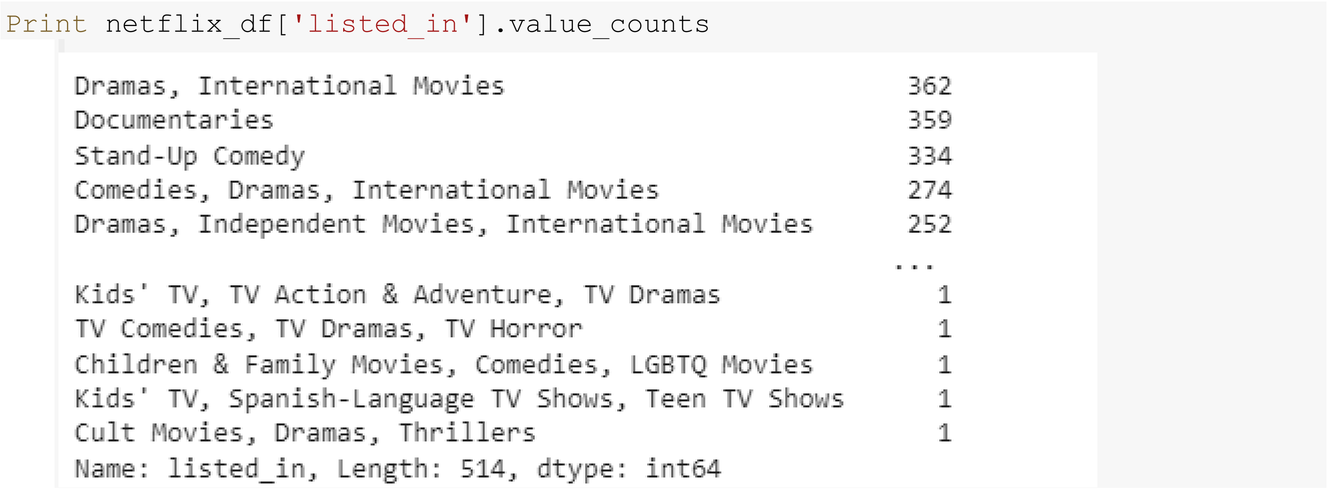


```
print(netflix_df['listed_in'].unique())
```

```
 'Classic Movies, Dramas, Romantic Movies'
 'Crime TV Shows, Romantic TV Shows, Spanish-Language TV Shows'
 'Classic Movies, Cult Movies, Horror Movies'
 'Anime Series, Crime TV Shows, TV Thrillers'
 'Children & Family Movies, Classic Movies'
 'Classic Movies, Comedies, International Movies'
 'Comedies, Sci-Fi & Fantasy' 'Action & Adventure, Cult Movies, Dramas'
 'Documentaries, Faith & Spirituality, Music & Musicals'
 'British TV Shows, Classic & Cult TV, TV Comedies'
 'International Movies, Sports Movies' 'International TV Shows'
 "Classic & Cult TV, Kids' TV, Spanish-Language TV Shows"
 'Romantic TV Shows, Spanish-Language TV Shows, TV Dramas'
 'Children & Family Movies, Comedies, Faith & Spirituality'
 'British TV Shows, Crime TV Shows, TV Dramas'
 'Classic Movies, Dramas, Music & Musicals'
 'Cult Movies, Horror Movies, Thrillers'
 'Action & Adventure, Classic Movies, Sci-Fi & Fantasy'
 'TV Action & Adventure, TV Comedies'
 'Classic Movies, Comedies, Music & Musicals' 'Independent Movies'
 'Documentaries, Horror Movies'
 'Classic & Cult TV, TV Horror, TV Mysteries'
 'Comedies, Faith & Spirituality, International Movies'
 'Dramas, Horror Movies, Sci-Fi & Fantasy'
 'British TV Shows, TV Dramas, TV Sci-Fi & Fantasy'
 'Comedies, Cult Movies, Horror Movies'
 'Comedies, Cult Movies, Sports Movies' 'Classic Movies, Documentaries'
```

Data Exploration:

Listed in is very big so let's check the counts

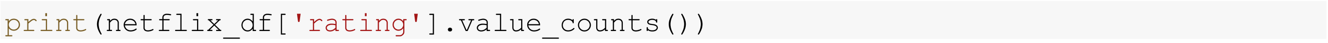


```
Print netflix_df['listed_in'].value_counts
```

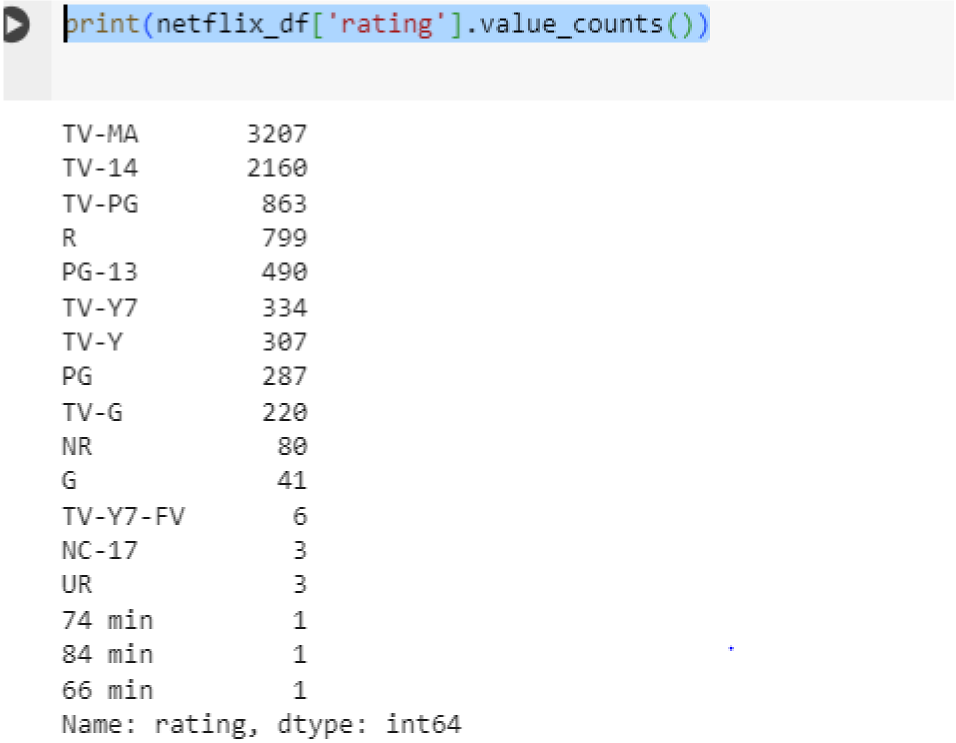
```
Dramas, International Movies                           362
Documentaries                                          359
Stand-Up Comedy                                        334
Comedies, Dramas, International Movies                 274
Dramas, Independent Movies, International Movies       252
                                                      ...
Kids' TV, TV Action & Adventure, TV Dramas               1
TV Comedies, TV Dramas, TV Horror                        1
Children & Family Movies, Comedies, LGBTQ Movies         1
Kids' TV, Spanish-Language TV Shows, Teen TV Shows       1
Cult Movies, Dramas, Thrillers                           1
Name: listed_in, Length: 514, dtype: int64
```

Drama, International movies is the maximum listed movies

Let's check ratings:



```
print(netflix_df['rating'].value_counts())
```



```
print(netflix_df['rating'].value_counts())
```

```
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
G             41
TV-Y7-FV       6
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64
```

TV-MA Mature Audience

- **TV-14** Parents Strongly Cautioned
- **TV-PG** Parental Guidance Suggested
- **R** rating
- **PG-13** Parents Strongly Cautioned - 13
- **TV-Y7** Directed to Older Children - 7
- **TV-Y** Directed to Younger Children
- **PG** Parental Guidance Suggested
- **TV-G** General Audience
- **NR** Not Rated.
- **G** General Audience
- **TV-Y7-FV** Directed to Older Children - Fantasy Violence
- **NC-17** No One 17 and Under Admitted
- **UR** Unrated:

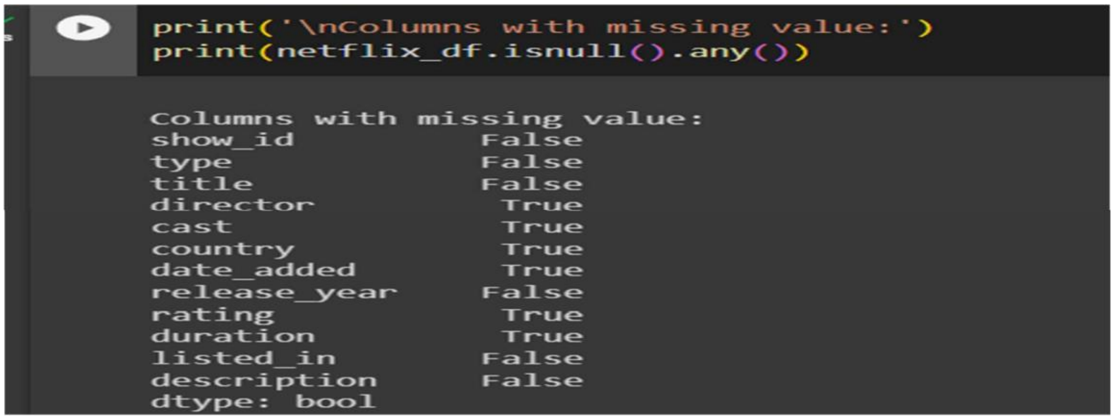
Therefore mature audience movie and TV shows has high rating.

Missing Value Detection

Data Profiling & Cleaning

Data Cleaning means the process of identifying incorrect, incomplete, inaccurate, irrelevant, or missing pieces of data and then modifying, replacing, or deleting them as needed. Data Cleansing is considered as the basic element of Data Science.

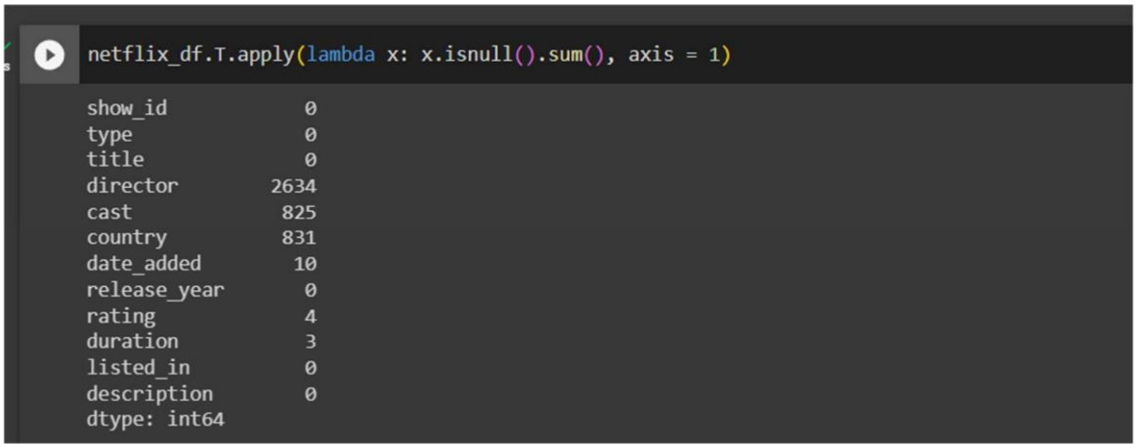
print('\nColumns with missing value:') print(netflix_df.isnull().any())



```
print('\nColumns with missing value:')
print(netflix_df.isnull().any())

Columns with missing value:
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
```

From the info, we know that there are 8807 entries and 12 columns to work with for this EDA. There are a few columns that contain null values, “director,” “cast,” “country,” “date_added,” “rating.”

netflix_df.T.apply(lambda x: x.isnull().sum(), axis = 1)



```
netflix_df.T.apply(lambda x: x.isnull().sum(), axis = 1)

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
```

```
netflix_df.isnull().sum().sum()
```

```
4307
```

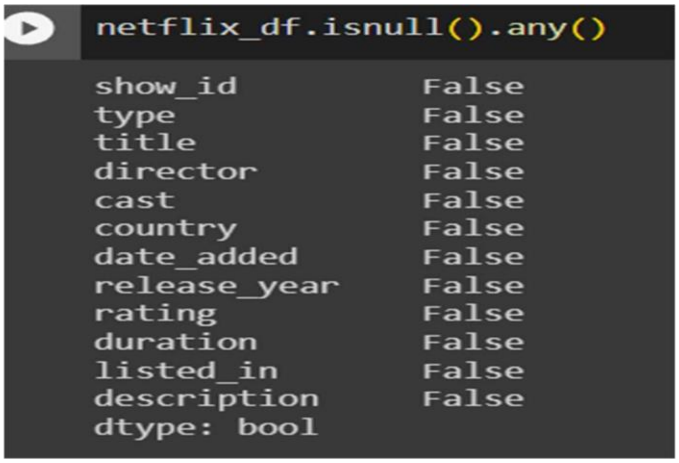
There are a total of 4307 null values across the entire dataset with 2634 missing points under "director", 825 under "cast", 831 under "country", 11 under "date_added", 4 under "rating" and 3 under “duration ”. We will have to handle all null data points before we can dive into EDA and modelling.

Imputation is a treatment method for missing value by filling it in using certain techniques.

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.

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

Check missing value

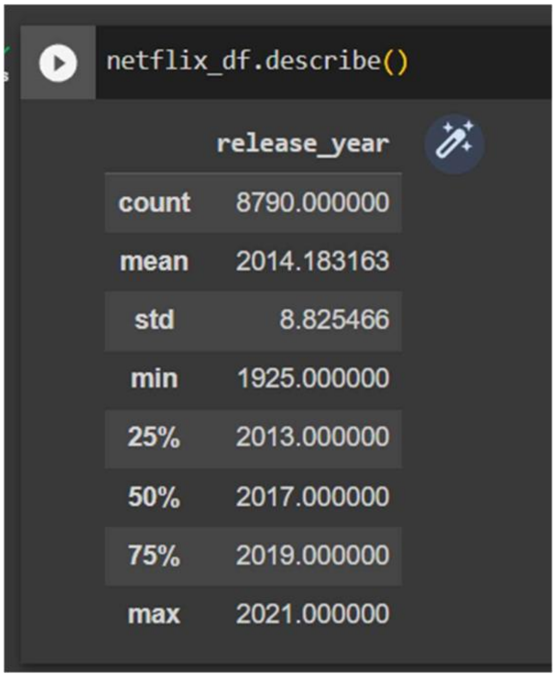


```
netflix_df.isnull().any()
```

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

For missing values, the easiest way to get rid of them would be to delete the rows with the missing data. However, this wouldn't be beneficial to our EDA since the is a loss of information. Since "director", "cast", and "country" contain the majority of null values, we chose to treat each missing value is unavailable. The other two label "date_added”,” duration” and "rating" contain an insignificant portion of the data so it drops from the dataset. Finally, we can see that there are no more missing values in the data frame

Statistical Summary After Data Cleaning:



```
netflix_df.describe()
```

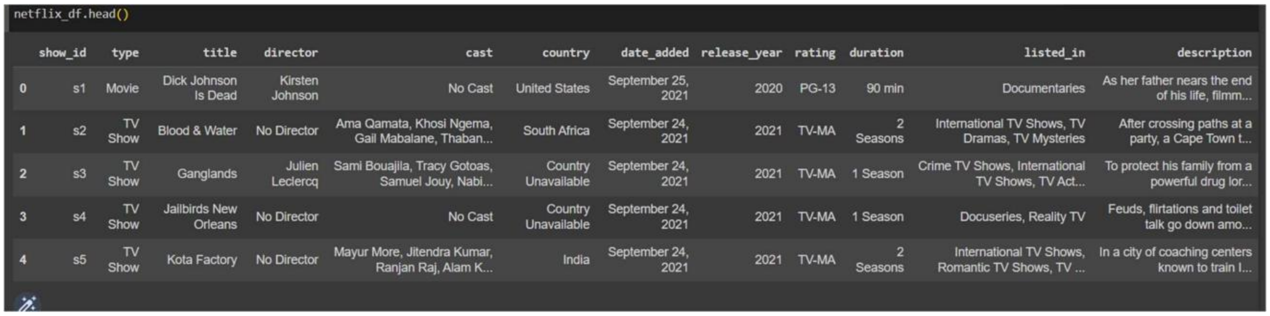
	release_year
count	8790.000000
mean	2014.183163
std	8.825466
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

3. Non-Graphical Analysis:

Non-Graphical Analysis involves calculating the summary statistics, without using pictorial or graphical representations. There are 3 main functions that Pandas library provide us, and I will be discussing about them. Those functions are:

1. `info()`
2. `isna().sum()` or `isnull().sum()`
3. `describe()`

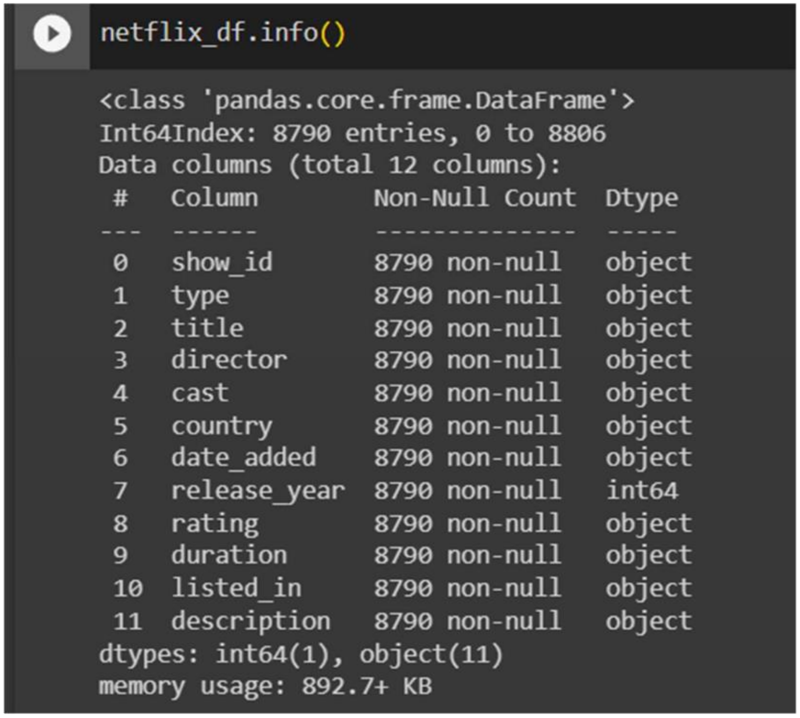
Checking the data using .head()



```
netflix_df.head()
```

	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	No Cast	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	No Director	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...	Country Unavailable	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	No Director	No Cast	Country Unavailable	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	No Director	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...

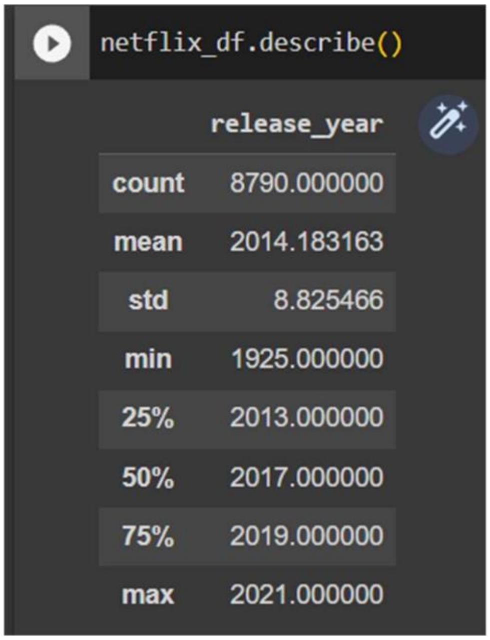
1.info() mainly indicates the number of features, non-null count, and data type of each features. Additionally, it also shows the number of features in present in each data type(s). This helps us to determine how many numerical and categorical features we have.



```
netflix_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
```

1. Read The Description Of The Data

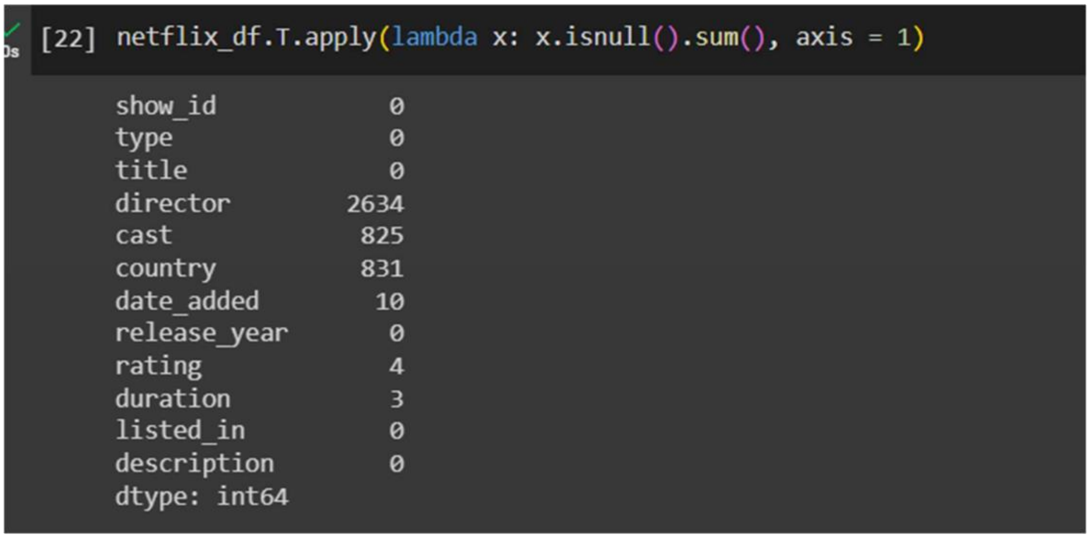


```
netflix_df.describe()
```

	release_year
count	8790.000000
mean	2014.183163
std	8.825466
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

2. isna().sum() or isnull().sum()

netflix_df.T.apply(lambda x: x.isnull().sum(), axis = 1)



```
[22] netflix_df.T.apply(lambda x: x.isnull().sum(), axis = 1)

     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
```

4: Exploratory Analysis and Visualization

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

Univariate analysis

Analysis done based only on one variable. we are not going to the math behind these concepts, for now, let's see what these are in graphs. (*please have some basic idea on these concepts if you don't get them by seeing graphs*).

A==>Pie plot:

Netflix Content By Type

Analysis entire Netflix dataset consisting of both movies and shows. Let’s compare the total number of movies and shows in this dataset to know which one is the majority.

plt.figure(figsize=(6,3))

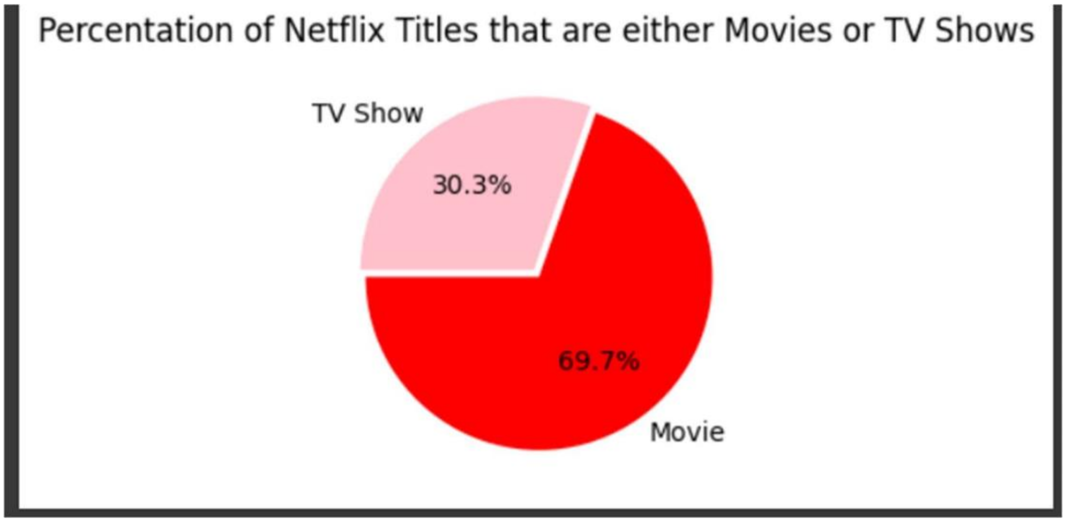
plt.title("Percentation of Netflix Titles that are either Movies or TV Shows")

g=plt.pie(netflix_df.type.value_counts(),explode=(0.025,0.025),

labels=netflix_df.type.value_counts().index, colors=['red','pink'],autopct='%1.1f%%',

startangle=180)

```
plt.show()
```

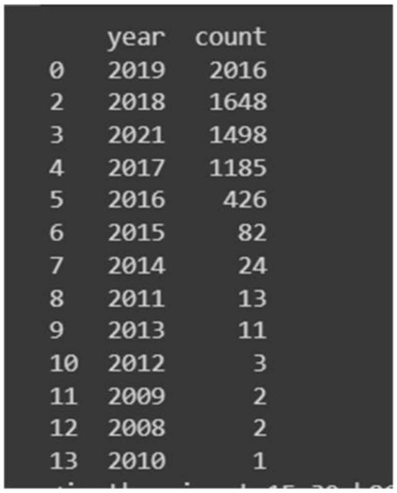


There are far more movie titles (69.7%) that TV shows titles (30.3%) in terms of title.

→ 2. Amount of Content as a Function of Time: Distplot

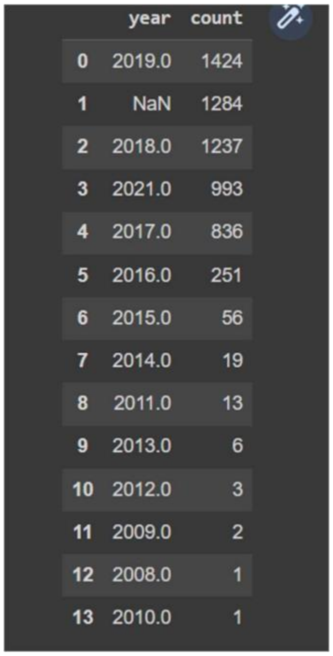
we will explore the amount of content Netflix has added throughout the previous years. Since we are interested in when Netflix added the title onto their platform, we will add a “year_added” column to show the date from the “date_added” columns.

```
netflix_df["year_added"] = pd.to_datetime(netflix_df.date_added).dt.year
netflix_movies_df["year_added"] = pd.to_datetime(netflix_movies_df.date_added).dt.year
netflix_shows_df["year_added"] = pd.to_datetime(netflix_shows_df.date_added).dt.year
netflix_year_df =
netflix_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)
```



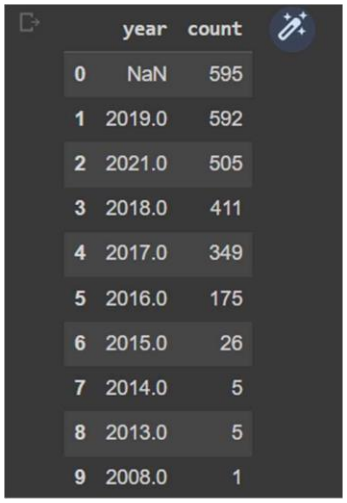
```
     year  count
0    2019   2016
2    2018   1648
3    2021   1498
4    2017   1185
5    2016    426
6    2015     82
7    2014     24
8    2011     13
9    2013     11
10   2012      3
11   2009      2
12   2008      2
13   2010      1
```

movies_year_df =
netflix_movies_df.year_added.value_counts().to_frame().reset_index().rename(columns={"index":
"year", "year_added":"count"})
movies_year_df = movies_year_df[movies_year_df != 2020]
movies_year_df



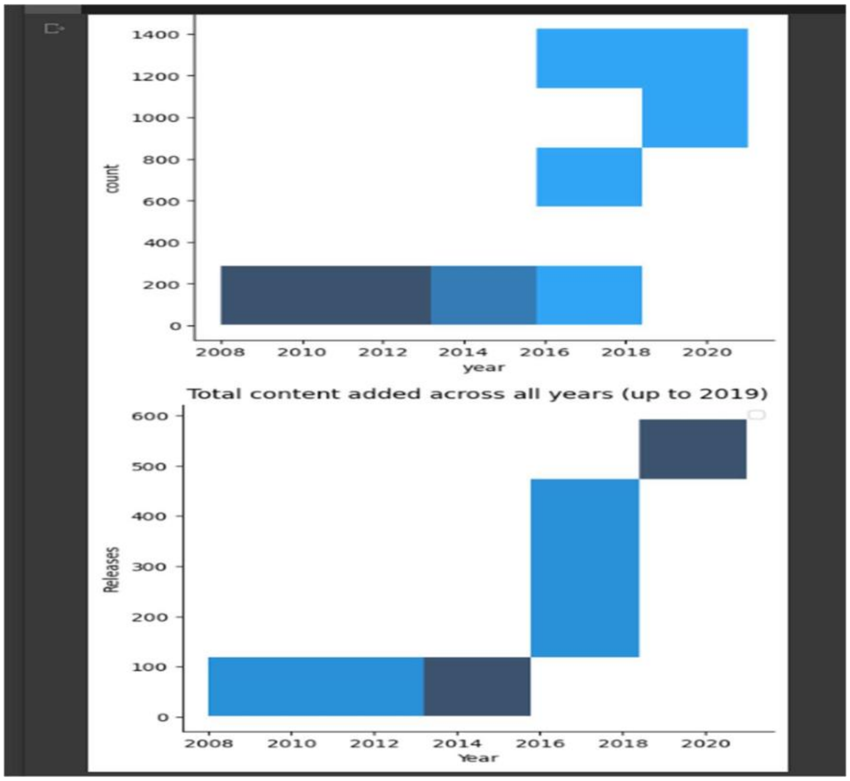
```
      year  count
0   2019.0   1424
1      NaN   1284
2   2018.0   1237
3   2021.0    993
4   2017.0    836
5   2016.0    251
6   2015.0     56
7   2014.0     19
8   2011.0     13
9   2013.0      6
10  2012.0      3
11  2009.0      2
12  2008.0      1
13  2010.0      1
```

shows_year_df =
netflix_shows_df.year_added.value_counts().to_frame().reset_index().rename(columns={"index":
"year", "year_added":"count"})
shows_year_df = shows_year_df[shows_year_df != 2020]
shows_year_df



```
     year  count
0     NaN    595
1  2019.0    592
2  2021.0    505
3  2018.0    411
4  2017.0    349
5  2016.0    175
6  2015.0     26
7  2014.0      5
8  2013.0      5
9  2008.0      1
```

```
fig, ax = plt.subplots(figsize=(7, 5))
sns.displot(data=netflix_year_df, x='year', y='count')
sns.displot(data=movies_year_df, x='year', y='count')
sns.displot (data=shows_year_df, x='year', y='count')
ax.set_xticks(np.arange(2008, 2020, 1))
plt.title("Total content added across all years (up to 2019)")
plt.legend(['Total','Movie','TV Show'])
plt.ylabel("Releases")
plt.xlabel("Year")
plt.show()
```



Based on the timeline above, we can conclude that the popular streaming platform started gaining traction after 2013. Since then, the amount of content added has been increasing significantly. The growth in the number of movies on Netflix is much higher than that on TV shows. About 1,300 new movies were added in both 2018 and 2019. Besides, we can know that Netflix has increasingly focused on movies rather than TV shows in recent years

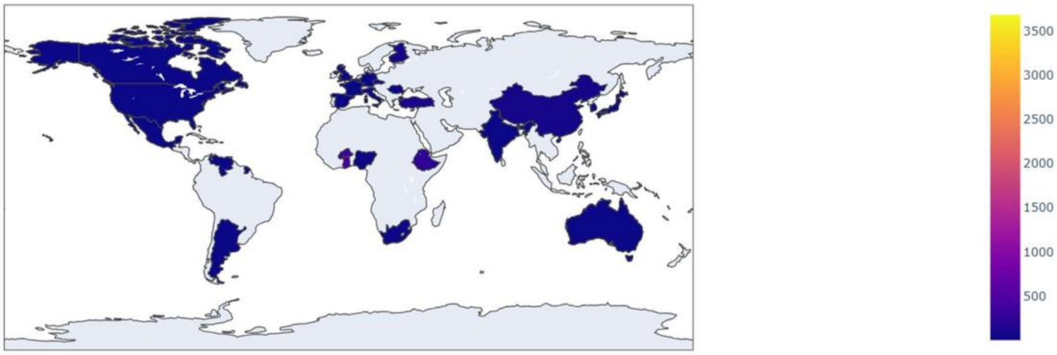
→ 3. Exploring the countries contribution with the most content of Netflix.

Next is exploring the countries by the amount of the produces content of Netflix. We need to separate all countries within a film before analysing it, then removing titles with no countries available.

```
import plotly.graph_objects as go

from plotly.offline import init_notebook_mode, iplot
```

We need to separate all countries within a film before analyzing it, then removing titles with no countries available.

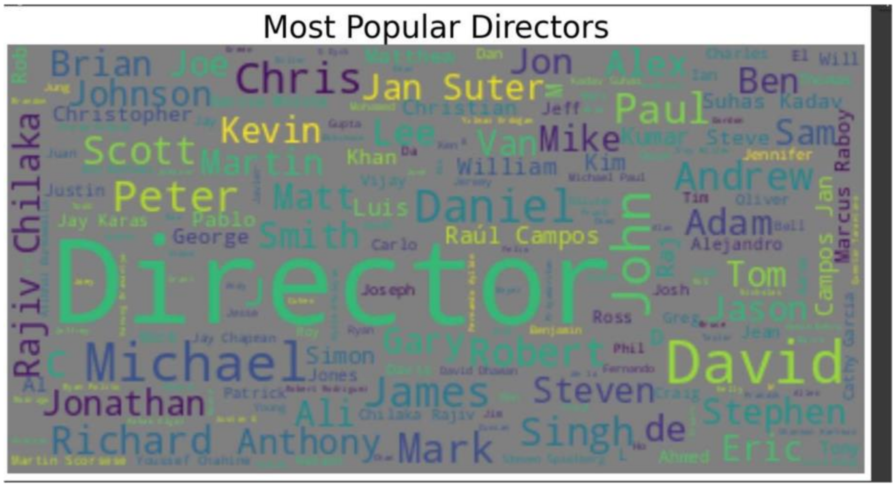
```
filtered_countries = netflix_df.set_index('title').country.str.split(', ',
expand=True).stack().reset_index(level=1, drop=True);

filtered_countries = filtered_countries[filtered_countries != 'Country Unavailable']
iplot([go.Choropleth(

locationmode='country names',
locations=filtered_countries,
z=filtered_countries.value_counts()

)])
```



→ 4. Top Directors on Netflix

To know the most popular director, we can visualize it.

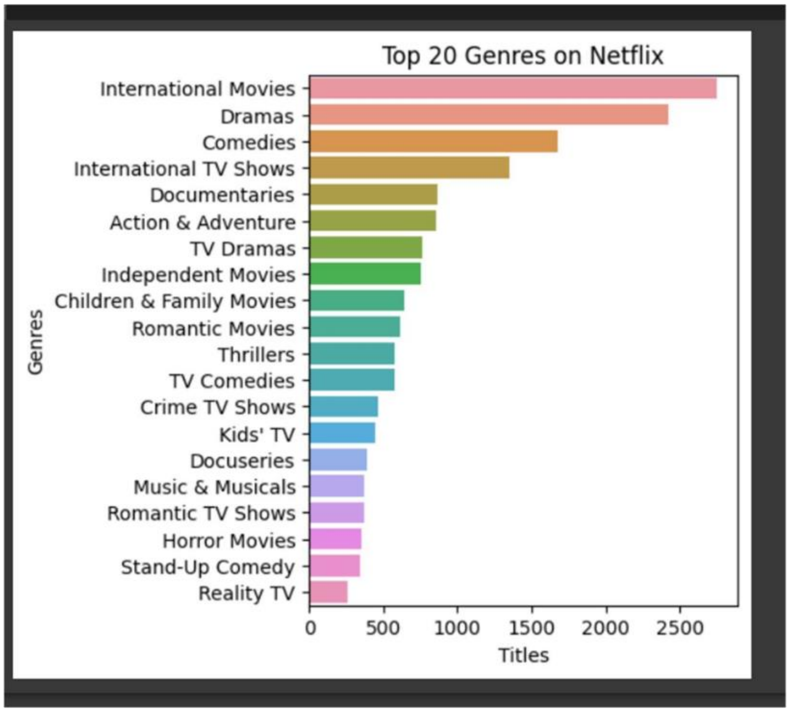
```
from wordcloud import WordCloud, ImageColorGenerator text = "
".join(str(each) for each in netflix_df.director)

# 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()
```



The most popular director on Netflix, with the most titles, is mainly international.

→ 5. Top 20 Genres on Netflix: Count Plot

```
filtered_genres = netflix_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()
```



From the graph, we know that International Movies take the first place, followed by dramas and comedies.

Bivariate Analysis:

Bi means two and variate means variable, so here there are two variables. The analysis is related to cause and the relationship between the two variables. There are three types of bivariate analysis.

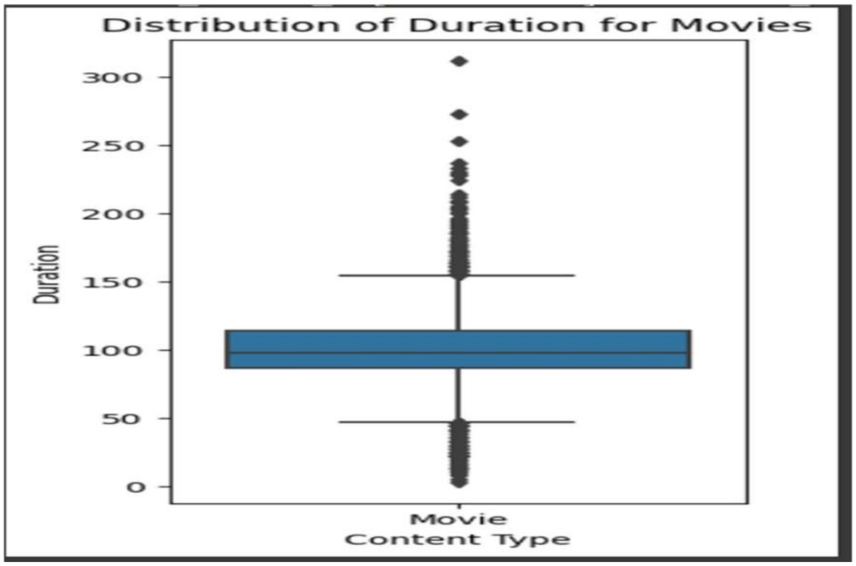
A➔ **Bivariate Analysis of two Numerical Variables (Numerical-Numerical)**

4.2 For categorical variable(s): Boxplot

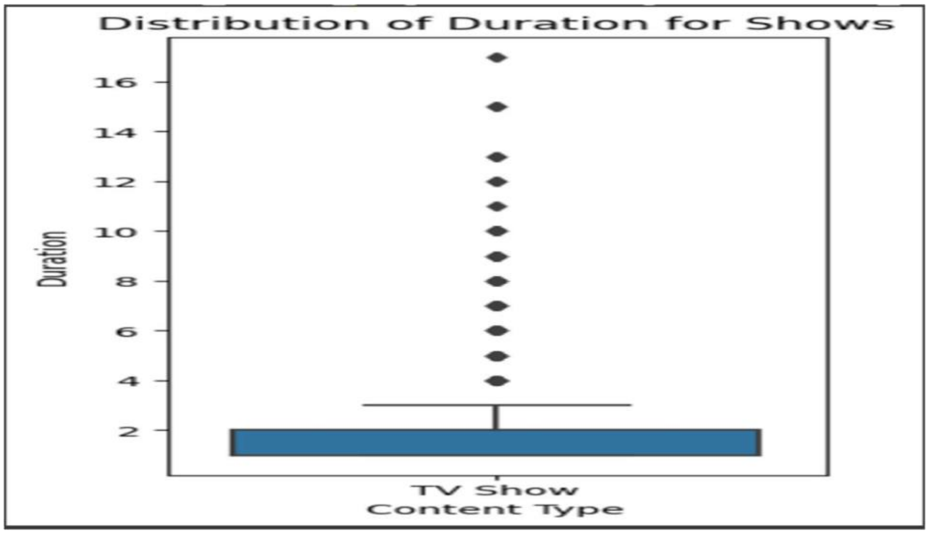
Duration Distribution for Movies and TV Shows

Analysing the duration distribution for movies and TV shows allows us to understand the typical length of content available on Netflix. We can create box plots to visualize these distributions and identify outliers or standard durations.

```
netflix_movies_df = netflix_df[netflix_df.type.str.contains("Movie")]

netflix_movies_df['duration'] = netflix_movies_df['duration'].str.extract('(\d+)',
expand=False).astype(int)


# Creating a boxplot for movie duration

plt.figure(figsize=(10, 6))

sns.boxplot(data=netflix_movies_df, x='type', y='duration')

plt.xlabel('Content Type')

plt.ylabel('Duration')

plt.title('Distribution of Duration for Movies')

plt.show()
```



```
netflix_shows_df = netflix_df[netflix_df.type.str.contains("TV Show")]

netflix_shows_df['duration'] = netflix_shows_df['duration'].str.extract('(\d+)', expand=False).astype(int)


# Creating a boxplot for movie duration

plt.figure(figsize=(3, 6))

sns.boxplot(data=netflix_shows_df, x='type', y='duration')

plt.xlabel('Content Type')

plt.ylabel('Duration')

plt.title('Distribution of Duration for Shows')

plt.show()
```



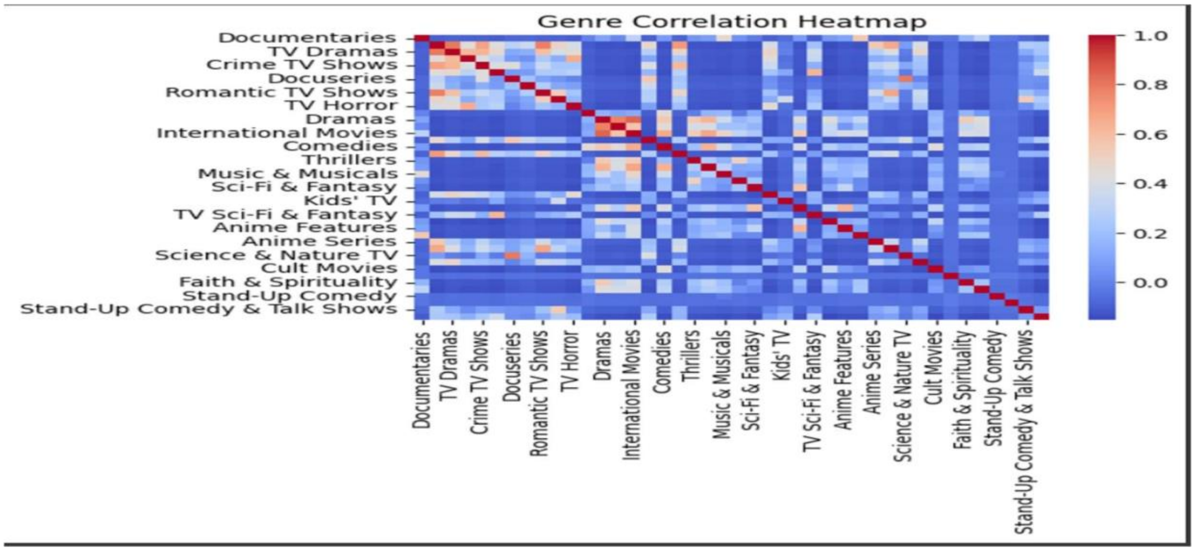
Analysing the movie box plot, we can see that most movies fall within a reasonable duration range, with few outliers exceedingly approximately 2.5 hours. This suggests that most movies on Netflix are designed to fit within a standard viewing time.

For TV shows, the box plot reveals that most shows have one to four seasons, with very few outliers having longer durations. This aligns with the earlier trends, indicating that Netflix focuses on shorter series formats.

4.3 For correlation: Heatmaps, Pairplots

Genre Correlation Heatmap:

Genres play a significant role in categorizing and organizing content on Netflix. analysing the correlation between genres can reveal interesting relationships between different types of content. We create a genre data DataFrame to investigate genre correlation and fill it with zeros. By iterating over each row in the original DataFrame, we update the genre data DataFrame based on the listed genres. We then create a correlation matrix using this genre data and visualize it as a heatmap.



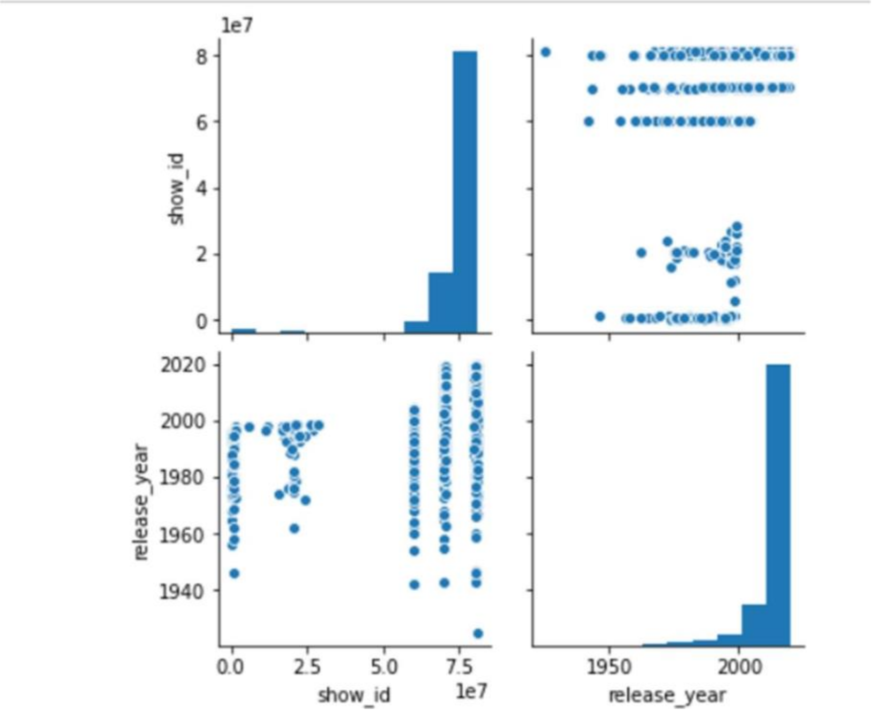
The heatmap demonstrates the correlation between different genres. By analysing the heatmap, we can identify strong positive correlations between specific genres, such as TV Dramas and International TV Shows, Romantic TV Shows, and International TV Shows.

Pairplots

A pairplot plot a pairwise relationships in a dataset.

The pairplot function creates a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column.

sns.pairplot(nf_df);



5. Missing Value & Outlier check (Treatment optional)

What is an outlier?

In a random sampling from a population, an outlier is defined as an observation that deviates abnormally from the standard data. In simple words, an outlier is used to define those data values which are far away from the general values in a dataset. An outlier can be broken down into out-of-line data.

For example, let us consider a row of data [10,15,22,330,30,45,60]. In this dataset, we can easily conclude that 330 is way off from the rest of the values in the dataset, thus 330 is an outlier. It was easy to figure out the outlier in such a small dataset, but when the dataset is huge, we need various methods to determine whether a certain value is an outlier or necessary information.

Why do we need to treat outliers?

Outliers can lead to vague or misleading predictions while using machine learning models. Specific models like linear regression, logistic regression, and support vector machines are susceptible to outliers. Outliers decrease the mathematical power of these models, and thus the output of the models becomes unreliable. However, outliers are highly subjective to the dataset. Some outliers may portray extreme changes in the data as well

Visual Detection

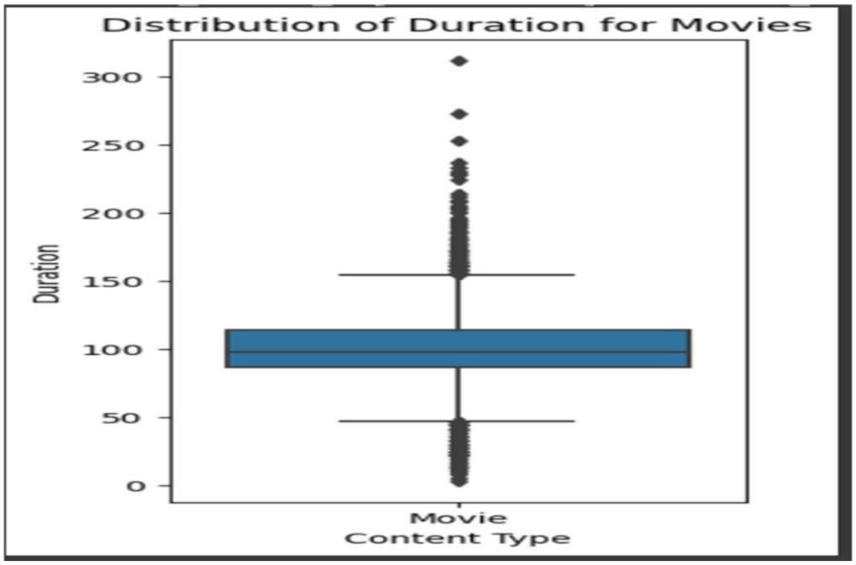
Box plots are a simple way to visualize data through quantiles and detect outliers. IQR(Interquartile Range) is the basic mathematics behind boxplots. The top and bottom whiskers can be understood as the boundaries of data, and any data lying outside it will be an outlier.

For categorical variable(s): Boxplot

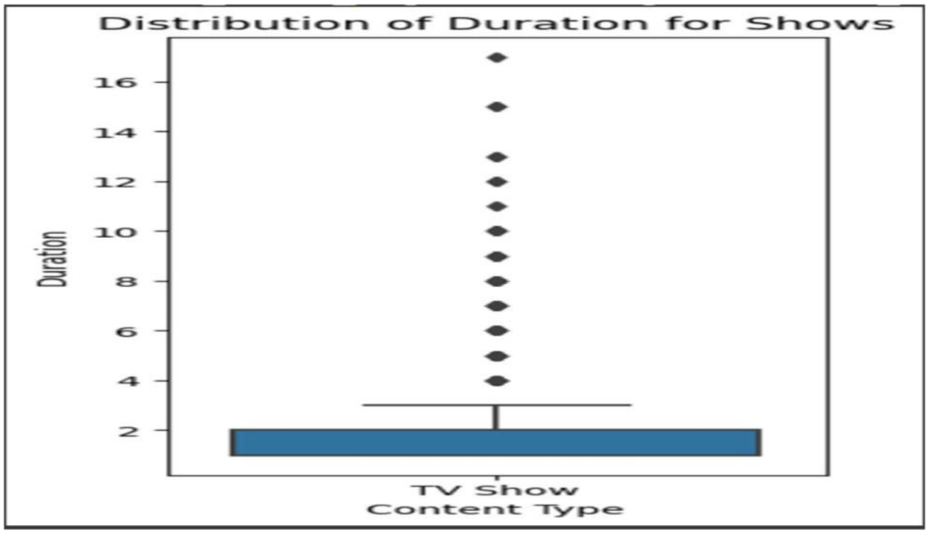
Duration Distribution for Movies and TV Shows

Analysing the duration distribution for movies and TV shows allows us to understand the typical length of content available on Netflix. We can create box plots to visualize these distributions and identify outliers or standard durations.

```
netflix_movies_df = netflix_df[netflix_df.type.str.contains("Movie")]

netflix_movies_df['duration'] = netflix_movies_df['duration'].str.extract('(\d+)',
expand=False).astype(int)


# Creating a boxplot for movie duration

plt.figure(figsize=(10, 6))

sns.boxplot(data=netflix_movies_df, x='type', y='duration')

plt.xlabel('Content Type')

plt.ylabel('Duration')
plt.title('Distribution of Duration for Movies')
plt.show()
```



```
netflix_shows_df = netflix_df[netflix_df.type.str.contains("TV Show")]
netflix_shows_df['duration'] = netflix_shows_df['duration'].str.extract('(\d+)',
expand=False).astype(int)

# Creating a boxplot for movie duration
plt.figure(figsize=(3, 6))
sns.boxplot(data=netflix_shows_df, x='type', y='duration')
plt.xlabel('Content Type')
plt.ylabel('Duration')
plt.title('Distribution of Duration for Shows')
plt.show()
```



Analysing the movie box plot, we can see that most movies fall within a reasonable duration range, with few outliers exceedingly approximately 2.5 hours. 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 four seasons, with very few outliers having longer durations. This aligns with the earlier trends, indicating that Netflix focuses on shorter series formats.

What are Missing values?

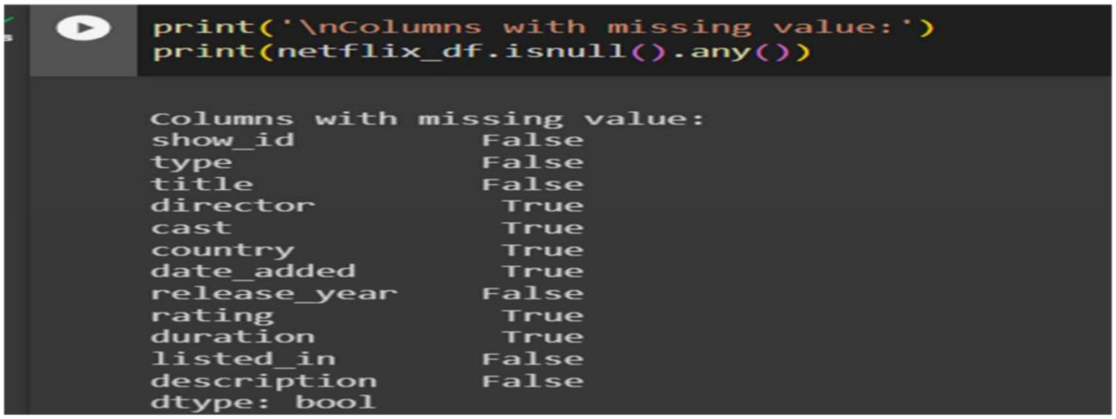
In a dataset, we often see the presence of empty cells, rows, and columns, also referred to as Missing values. They make the dataset inconsistent and unable to work on. Many machine learning algorithms return an error if parsed with a dataset containing null values. Detecting and treating missing values is essential while analyzing and formulating data for any purpose.

Detecting missing values

There are several ways to detect missing values in Python. isnull() function is widely used for the same purpose.

dataframe.isnull().values.any() allows us to find whether we have any null values in the dataframe.

```
print('\nColumns with missing value:')
print(netflix_df.isnull().any())
```



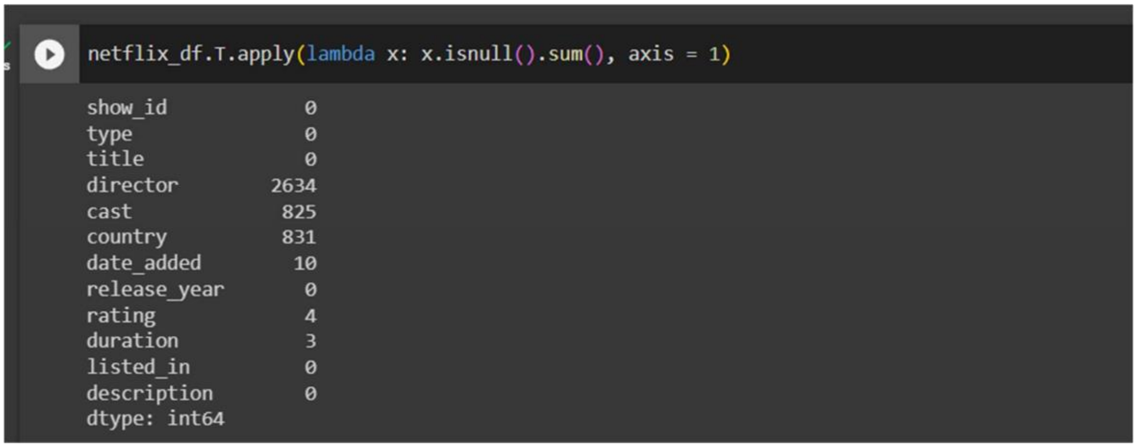
```
print('\nColumns with missing value:')
print(netflix_df.isnull().any())
```

```
Columns with missing value:
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
```

From the info, we know that there are 8807 entries and 12 columns to work with for this EDA. There are a few columns that contain null values, “director,” “cast,” “country,” “date_added,” “rating.”

dataframe.isnull().sum() this function displays the total number of null values in each column.

netflix_df.T.apply(lambda x: x.isnull().sum(), axis = 1)



```
netflix_df.T.apply(lambda x: x.isnull().sum(), axis = 1)
```

```
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
```

netflix_df.isnull().sum().sum()

4307
There are a total of 4307 null values across the entire dataset with 2634 missing points under "director", 825 under "cast", 831 under "country", 11 under "date_added", 4 under "rating" and 3 under “duration ”. We will have to handle all null data points before we can dive into EDA and modelling.

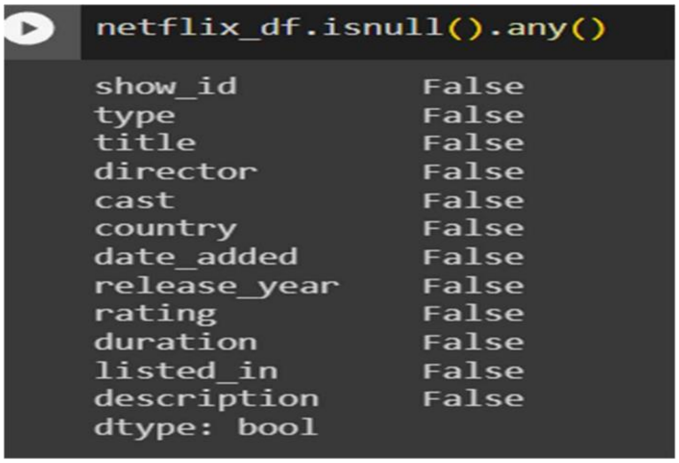
Remedies to the outliers and missing values

Imputation is a treatment method for missing value by filling it in using certain techniques.

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.

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

Check missing value



```
netflix_df.isnull().any()

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

Ioí missi→g :al"cs, tkc casicst waQ to gct íid or tkcm wo"ld bc to dclctc tkc íows witk tkc missi→g data. Howc:cí, tkis wo"ld→'t bc bc→cficial to o"í EKA si→cc tkc is a loss or i→roímatio→. Si→cc "diícctoí", "cast", a→d "co"→tíQ" co→tai→ tkc majoíitQ or →"ll :al"cs, wc ckosc to tícat cack missi→ g :al"c is "→ a:ailablc. l'kc otkcí two labcl "datc_addcd"," d"íatio→ " a→ d "íati→ g" co→ tai→ a→ i→ sig→ ifica→ t poítio→ or tkc data so it díops ríom tkc datasct. Ii→ allQ, wc ca→ scc tkat tkcíc aíc → o moíc missi→g :al"cs i→ tkc data ríamc.

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.