42047: Data Processing with Python

Assignment (Part C)

Report: Data Analysis and Visualization

Student Name and ID: [Kushal Ahuja (14191922)]

Date: [28th October 2022]

Table of Contents

Abstract

The purpose of the data analysis and visualisation in this dataset is to provide insights of the content available on Netflix.

I have performed Exploratory Data Analysis on the given dataset to create insights and understand it better by creating various types of plots, through different visualisation techniques and thus resulting information will be beneficial to stakeholders, decision-makers or producers and entertainment production companies

The business problems can be solved by the data analysis on this dataset which includes data cleaning, data wrangling and data visualisation.

1. Introduction and Background

The dataset chosen is the Netflix Movies and TV Shows data; this dataset contains information of all the movies and tv shows available on Netflix over the years; it consists of the attributes like country of production, type, cast, director, date added, release year, duration, genre and rating.

1.1The problem you tried to solve

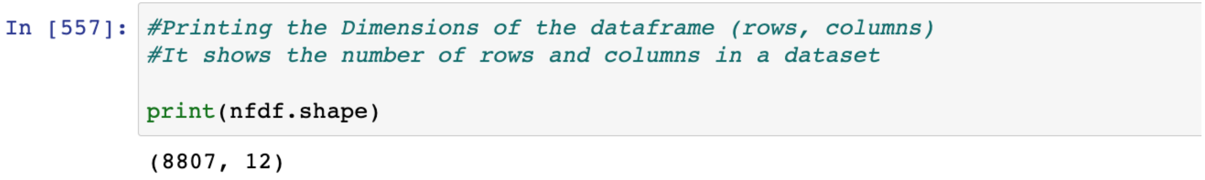
The problem which I tried to solve is getting to know the distribution of content according to its type, ratings and country thus creating insights for production companies or the public to make data-driven decisions i.e. for production companies to focus on certain content by analysing the insights and for the public to know the content available which is popular and according to the demographic data.

1.2 Business Question

The business question that I want to answer by performing exploratory data analysis is the Distribution of Movies and TV shows according to ratings and country. And interpreting the time that content is released and available on Netflix.

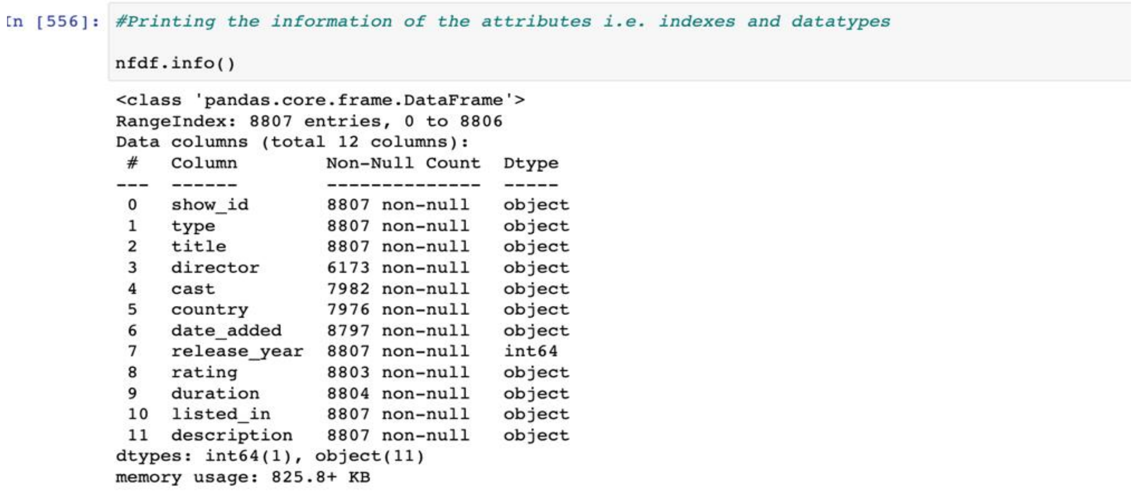
1.3 Dataset

Netflix dataset consists of 8,807 rows and 12 columns, this information can be accessed by the shape function.



```
In [557]: #Printing the Dimensions of the dataframe (rows, columns)
          #It shows the number of rows and columns in a dataset

          print(nfdf.shape)

          (8807, 12)
```

The data types of the attributes or columns can be accessed using info() function.



```
In [556]: #Printing the information of the attributes i.e. indexes and datatypes

          nfdf.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
```

Column details

Show_id – This column act as a unique key popularly known as the Primary Key of the dataset; it can also be used while plotting the count of any variable.

type – This column shows the type of content i.e. Movies or TV Shows

title- Title of the movie or tv show

director- The name of the director who have directed the movie or TV show.

cast- Cast members acted in the movie or TV show.

country- This column consists of the country names. It is evident that the content available is made in which country.

date_added- This column shows the date at which the content is available on Netflix.

release_year- This column shows the year at which the movie or TV show is released.

rating – This column shows the rating given by authority to the movie or TV shows on the basis of the age of the viewer, i.e. the content can be watched by people on the of certain age.

duration- This column shows the information of seasons of TV shows and duration of Movies.

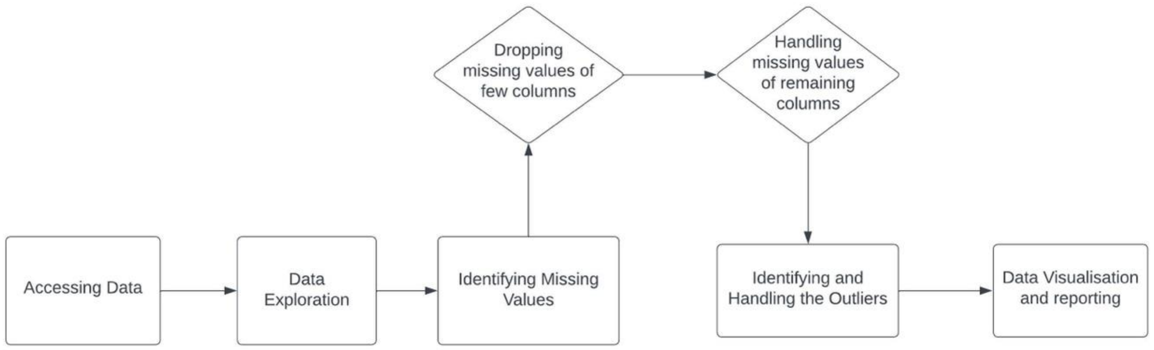
Listed_in – This column shows the information of genres of Movies and TV shows. For example, Documentaries, Thriller, Drama etc.

description- This column shows the brief description of the content i.e. the plot of the story.

The attributes which I have explored are rating, country, type, release_year and date_added.

2 Overview of the Data Analysis Pipeline

2.1 Flowchart


Dropping missing values of few columns
Handling missing values of remaining columns
Accessing Data
Data Exploration
Identifying Missing Values
Identifying and Handling the Outliers
Data Visualisation and reporting

Accessing Data- Loading the dataset in the jupyter notebook for analysis.

Data Exploration – Understanding the data by using various functions and thus getting the approach to solving the business problem.

Identifying the missing values- As a part of data cleaning the missing values are identified,

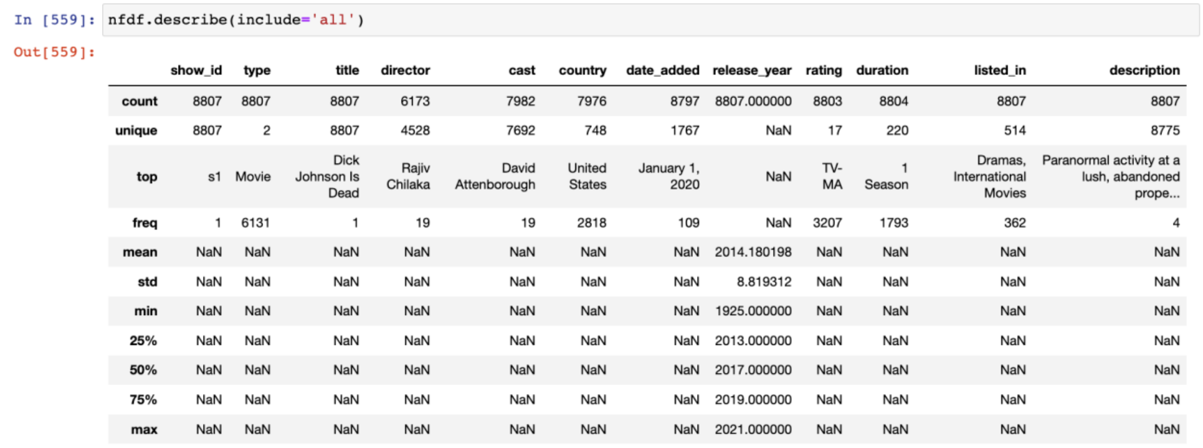
After finding the missing values, a few column's missing values are dropped and the remaining columns missing values are replaced due to their importance in solving the business question.

Identifying and handling the outliers- Finding the extreme values by visualisation technique and removing them by Inter Quartile Range method.

Data Visualisation and Reporting- Presenting insights through various visualisation techniques.

2.2 Data Preparation

I have used the describe() function to know the statistical data for the dataframe, Additionally included the particular attribute in the describe function.

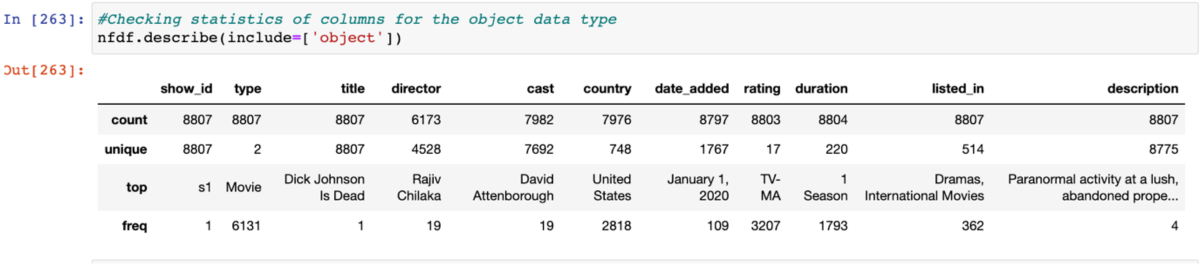


```
In [559]: nfdf.describe(include='all')
```

Out[559]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
count	8807	8807	8807	6173	7982	7976	8797	8807.000000	8803	8804	8807	8807
unique	8807	2	8807	4528	7692	748	1767	NaN	17	220	514	8775
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	NaN	TV-MA	1 Season	Dramas, International Movies	Paranormal activity at a lush, abandoned prope...
freq	1	6131	1	19	19	2818	109	NaN	3207	1793	362	4
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014.180198	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	8.819312	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1925.000000	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2013.000000	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2017.000000	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2019.000000	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2021.000000	NaN	NaN	NaN	NaN

Used describe to check the statistical data of datatype “object”

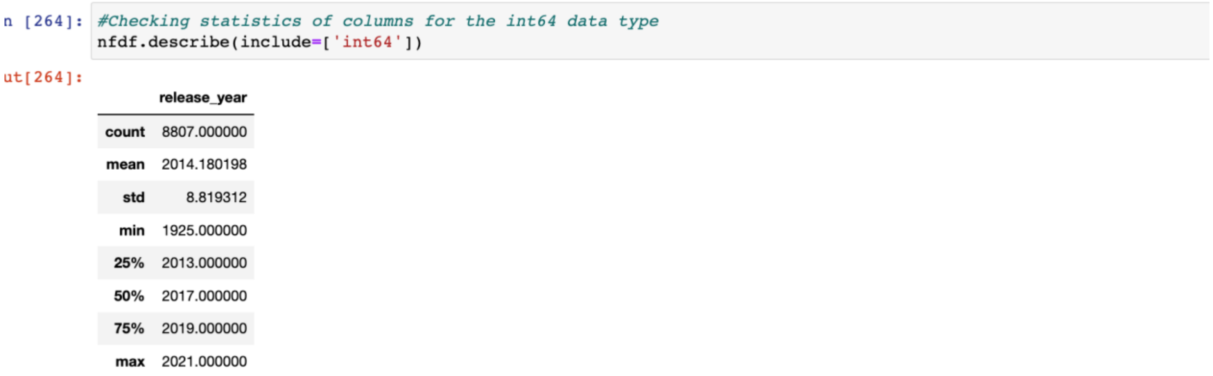


```
In [263]: #Checking statistics of columns for the object data type
          nfdf.describe(include=['object'])
```

Out[263]:

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

Used describe function to check the statistical data of datatype integer i.e. numerical data without the decimals.

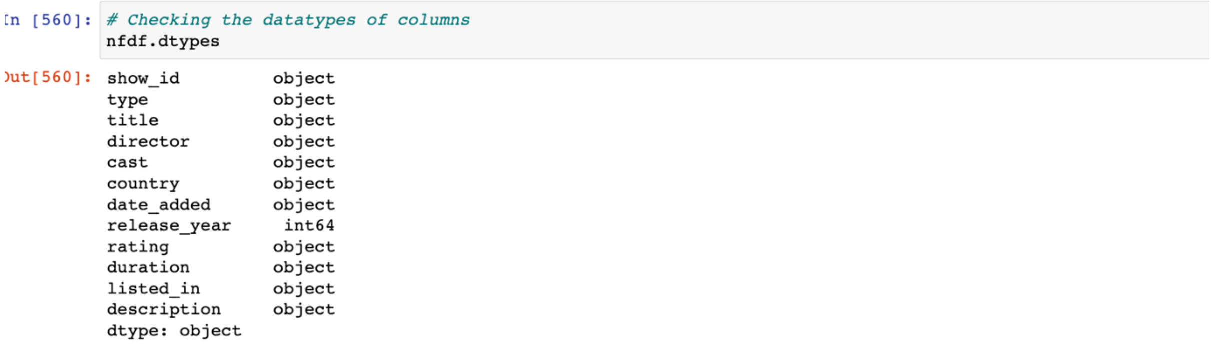


```
n [264]: #Checking statistics of columns for the int64 data type
         nfdf.describe(include=['int64'])
```

ut[264]:

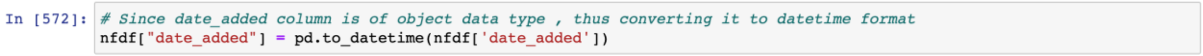
	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

Checking the datatypes of the attributes by using dtypes function



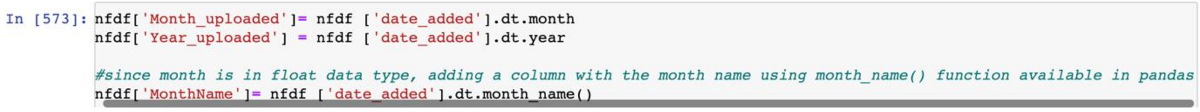
```
In [560]: # Checking the datatypes of columns
          nfdf.dtypes

Out[560]: show_id         object
          type            object
          title           object
          director        object
          cast            object
          country         object
          date_added      object
          release_year     int64
          rating          object
          duration        object
          listed_in       object
          description     object
          dtype: object
```

As the date_added column is of object datatype, converting it to the date time format.



```
In [572]: # Since date_added column is of object data type , thus converting it to datetime format
          nfdf["date_added"] = pd.to_datetime(nfdf['date_added'])
```

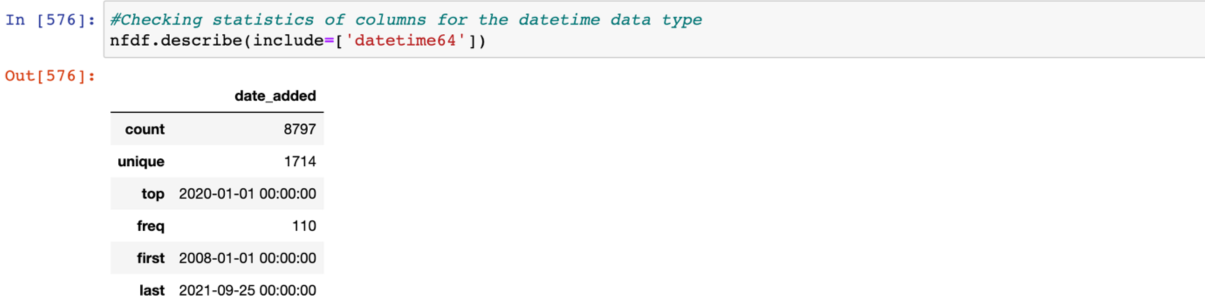
Splitting the date_added column in the column year and months.



```
In [573]: nfdf['Month_uploaded']= nfdf ['date_added'].dt.month
          nfdf['Year_uploaded'] = nfdf ['date_added'].dt.year

          #since month is in float data type, adding a column with the month name using month_name() function available in pandas
          nfdf['MonthName']= nfdf ['date_added'].dt.month_name()
```

This split is done using the dt.month and dt.year function and as the dt.month shows the values in number, dt.month_name() function is used to convert it into the months name.

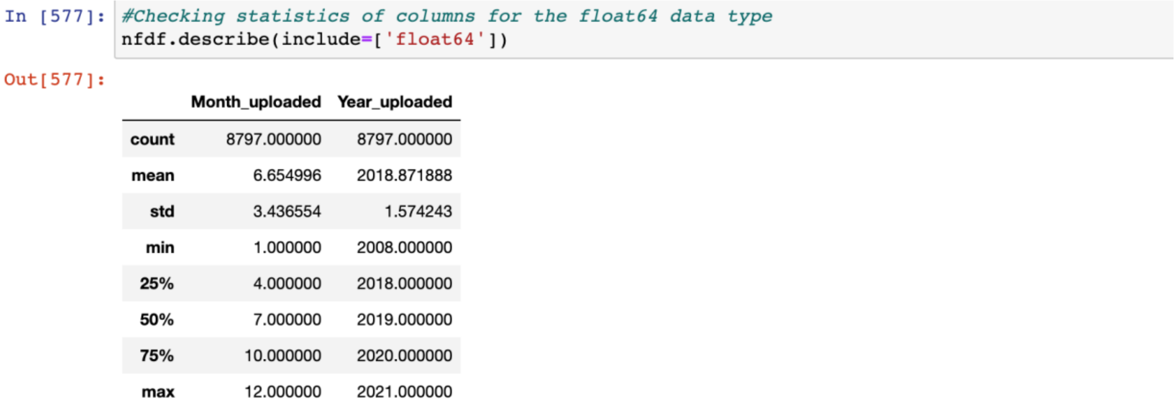
Checking the statistical data of the datatype- datetime



```
In [576]: #Checking statistics of columns for the datetime data type
          nfdf.describe(include=['datetime64'])

Out[576]:
```

	date_added
count	8797
unique	1714
top	2020-01-01 00:00:00
freq	110
first	2008-01-01 00:00:00
last	2021-09-25 00:00:00

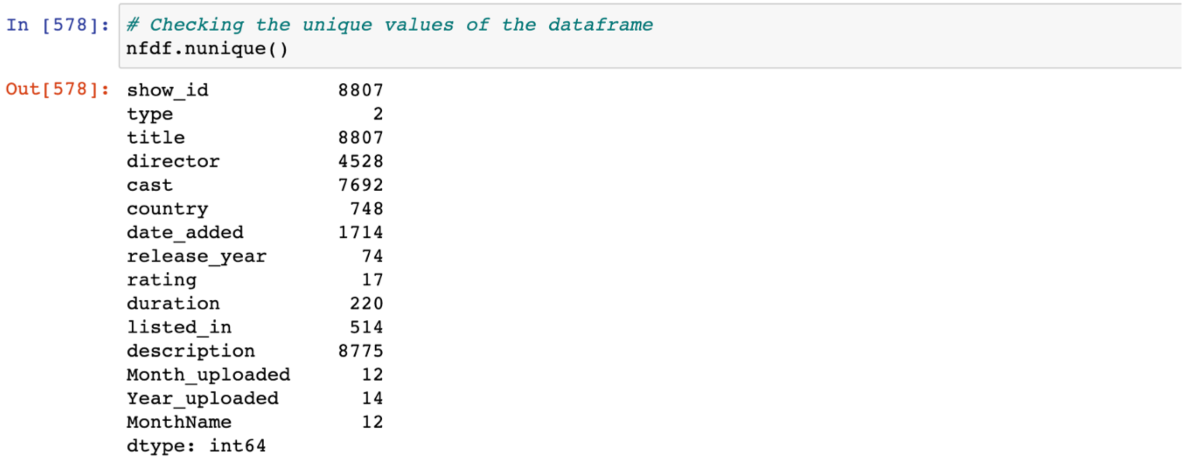
Checking the statistical data of the datatype- float, the new columns made are of float datatype



```
In [577]: #Checking statistics of columns for the float64 data type
          nfdf.describe(include=['float64'])

Out[577]:
```

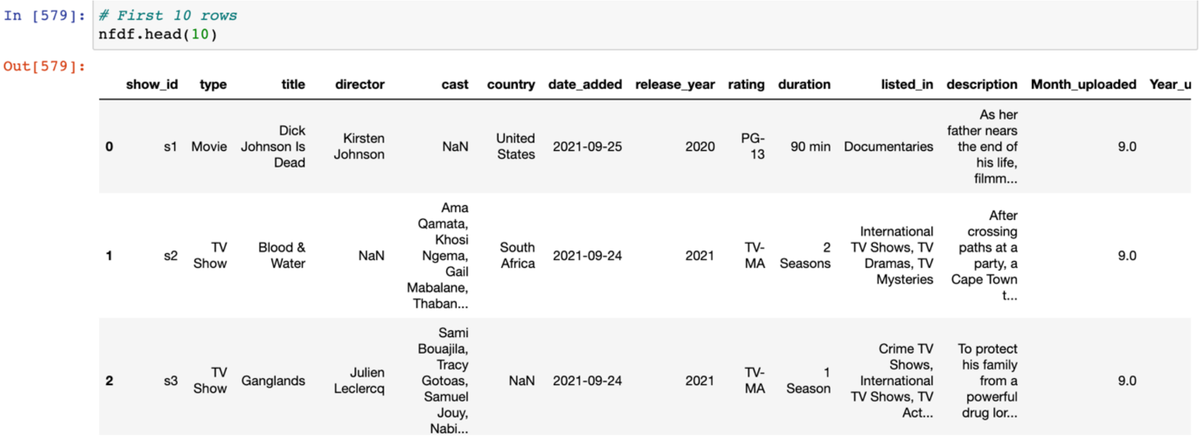
	Month_uploaded	Year_uploaded
count	8797.000000	8797.000000
mean	6.654996	2018.871888
std	3.436554	1.574243
min	1.000000	2008.000000
25%	4.000000	2018.000000
50%	7.000000	2019.000000
75%	10.000000	2020.000000
max	12.000000	2021.000000

Checking the unique values in the dataset, the function used for performing this nunique()



```
In [578]: # Checking the unique values of the dataframe
          nfdf.nunique()

Out[578]: show_id           8807
          type                 2
          title             8807
          director          4528
          cast              7692
          country            748
          date_added        1714
          release_year        74
          rating              17
          duration           220
          listed_in          514
          description       8775
          Month_uploaded      12
          Year_uploaded       14
          MonthName           12
          dtype: int64
```

head() function is used to display the first n rows

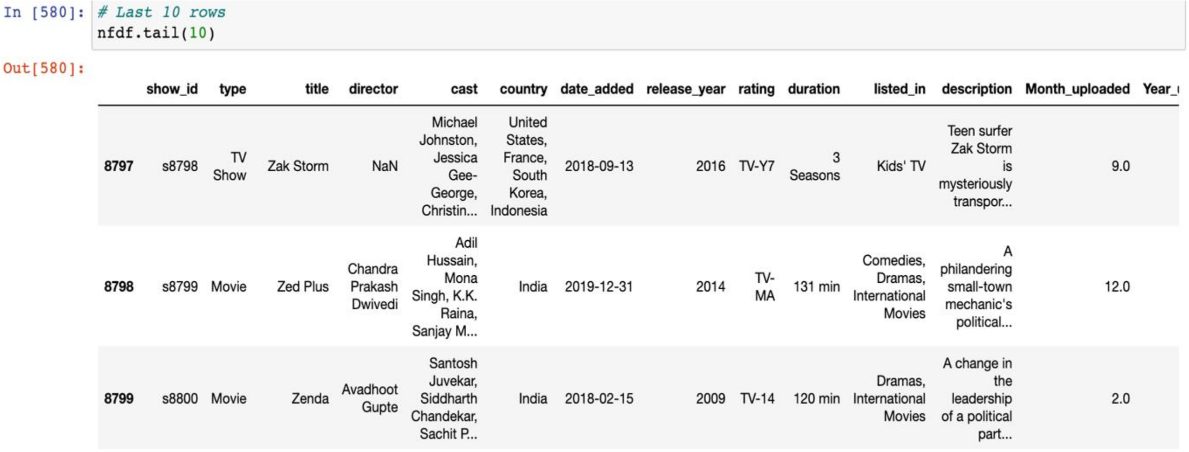


```
In [579]: # First 10 rows
          nfdf.head(10)
```

Out[579]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	Month_uploaded	Year_u
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	9.0	
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	9.0	
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	9.0	

tail() function is used to display the last n rows

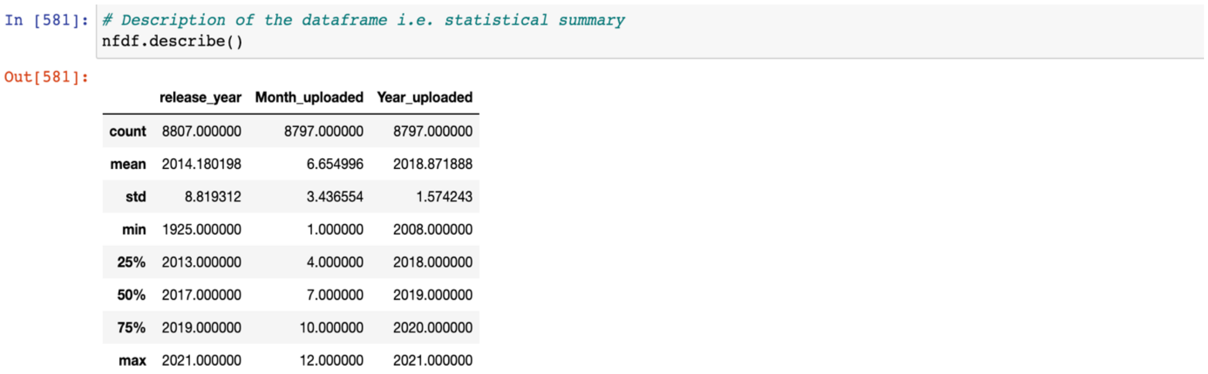


```
In [580]: # Last 10 rows
          nfdf.tail(10)
```

Out[580]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	Month_uploaded	Year_
8797	s8798	TV Show	Zak Storm	NaN	Michael Johnston, Jessica Gee-George, Christin...	United States, France, South Korea, Indonesia	2018-09-13	2016	TV-Y7	3 Seasons	Kids' TV	Teen surfer Zak Storm is mysteriously transpor...	9.0	
8798	s8799	Movie	Zed Plus	Chandra Prakash Dwivedi	Adil Hussain, Mona Singh, K.K. Raina, Sanjay M...	India	2019-12-31	2014	TV-MA	131 min	Comedies, Dramas, International Movies	A philandering small-town mechanic's political...	12.0	
8799	s8800	Movie	Zenda	Avadhoot Gupte	Santosh Juvekar, Siddharth Chandekar, Sachit P...	India	2018-02-15	2009	TV-14	120 min	Dramas, International Movies	A change in the leadership of a political part...	2.0	

describe() function is used for the statistical summary of numerical variables.



```
In [581]: # Description of the dataframe i.e. statistical summary
          nfdf.describe()
```

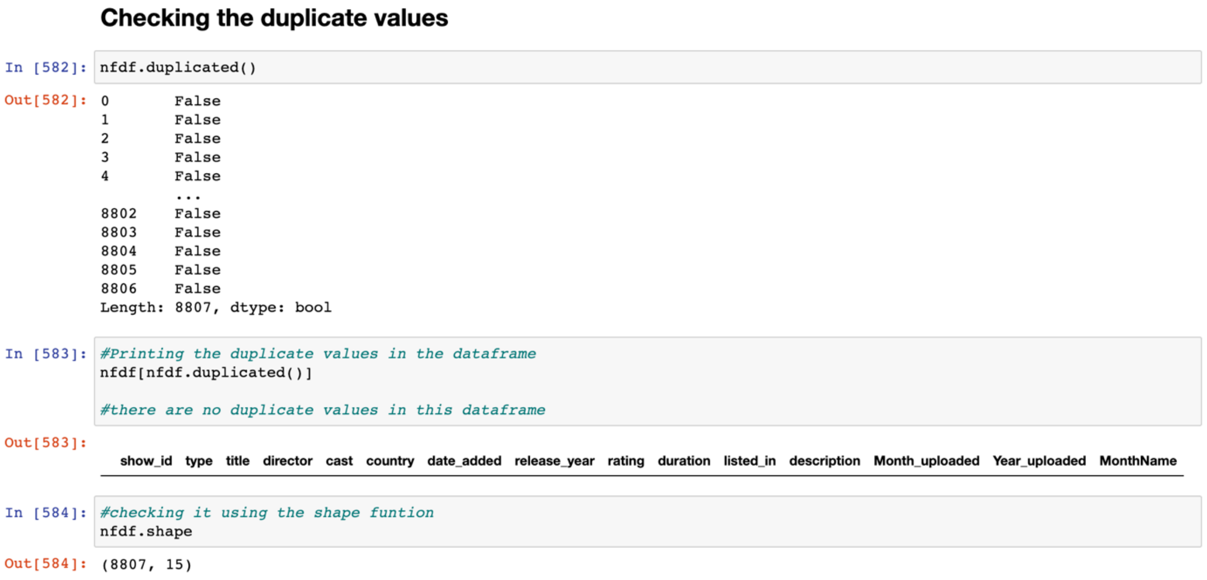
Out[581]:

	release_year	Month_uploaded	Year_uploaded
count	8807.000000	8797.000000	8797.000000
mean	2014.180198	6.654996	2018.871888
std	8.819312	3.436554	1.574243
min	1925.000000	1.000000	2008.000000
25%	2013.000000	4.000000	2018.000000
50%	2017.000000	7.000000	2019.000000
75%	2019.000000	10.000000	2020.000000
max	2021.000000	12.000000	2021.000000

2.3 Missing value exploration

Checking duplicate values

Duplicate values can be checked using the duplicated() function.



Checking the duplicate values

```
In [582]: nfdf.duplicated()
```

```
Out[582]: 0       False
          1       False
          2       False
          3       False
          4       False
                  ...
          8802    False
          8803    False
          8804    False
          8805    False
          8806    False
          Length: 8807, dtype: bool
```

```
In [583]: #Printing the duplicate values in the dataframe
          nfdf[nfdf.duplicated()]

          #there are no duplicate values in this dataframe
```

Out[583]:

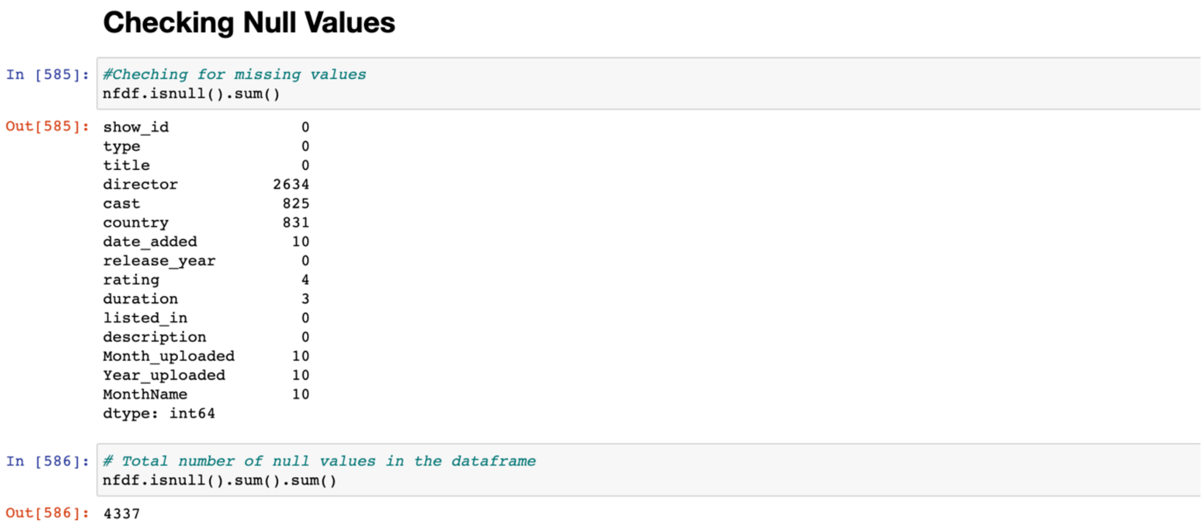
show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	Month_uploaded	Year_uploaded	MonthName

```
In [584]: #checking it using the shape funtion
          nfdf.shape
```

```
Out[584]: (8807, 15)
```

This dataset doesn’t contain any duplicate values.

Checking Missing Values



Checking Null Values

In [585]:
```
#Cheching for missing values
nfdf.isnull().sum()
```

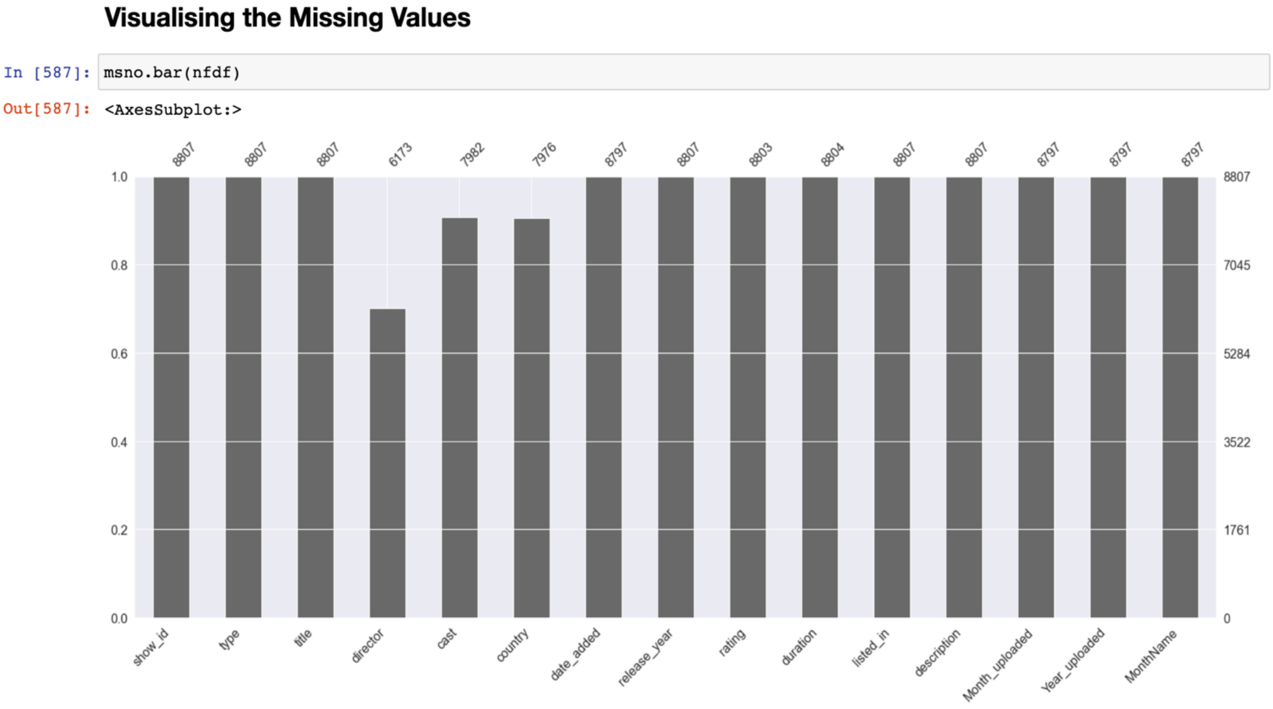
Out[585]:
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
Month_uploaded    10
Year_uploaded     10
MonthName         10
dtype: int64
```

In [586]:
```
# Total number of null values in the dataframe
nfdf.isnull().sum().sum()
```

Out[586]:
```
4337
```

Total number of 4337 missing values are found in the dataset.

Missing values can be visualised using missingno library and seaborn library’s heatmap function.

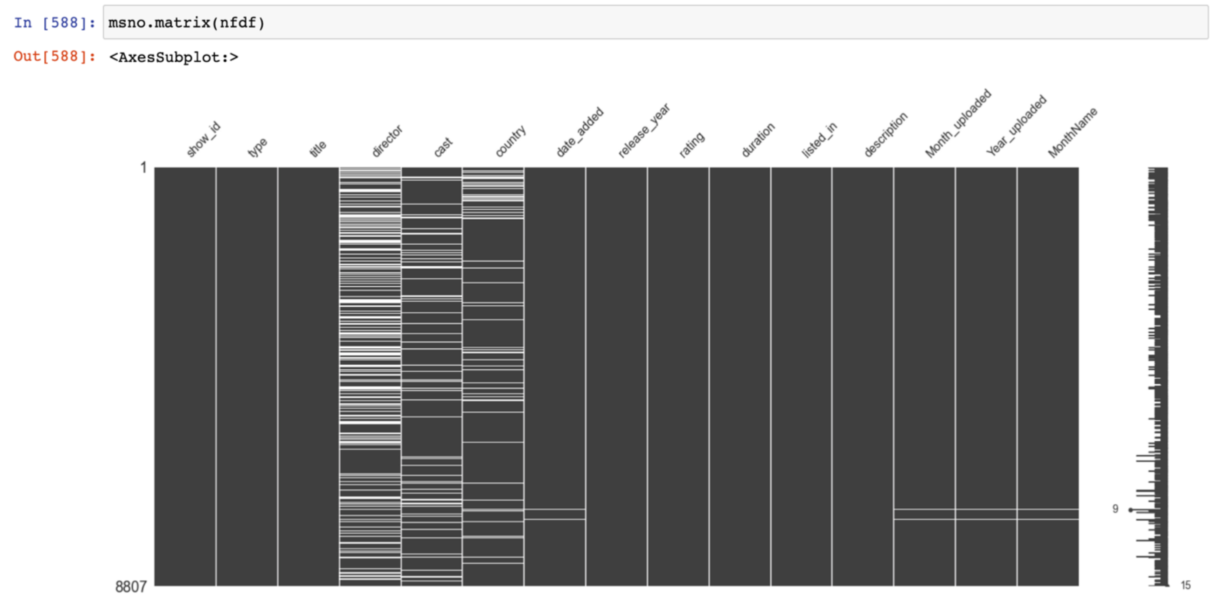


Visualising the Missing Values

In [587]:
```
msno.bar(nfdf)
```

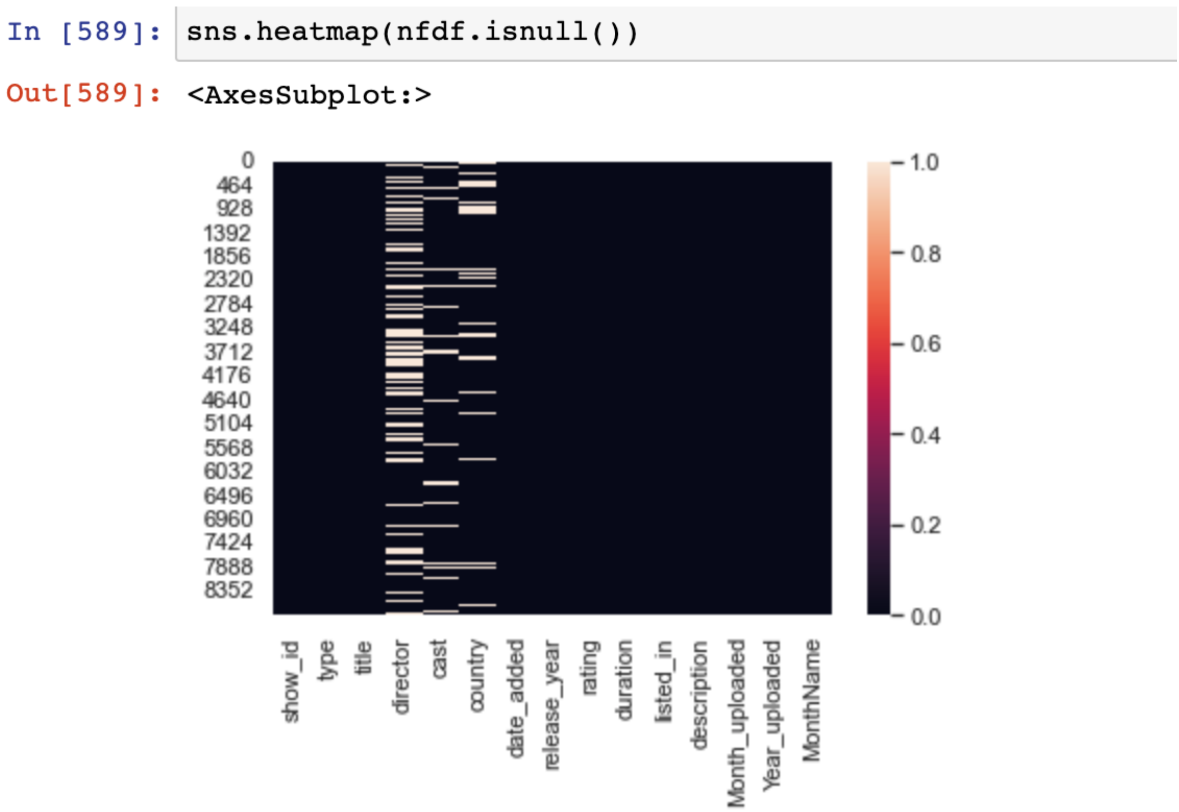
Out[587]:
```
<AxesSubplot:>
```

Visualising the missing values using the matrix plot



```
In [588]: msno.matrix(nfdf)
Out[588]: <AxesSubplot:>
```

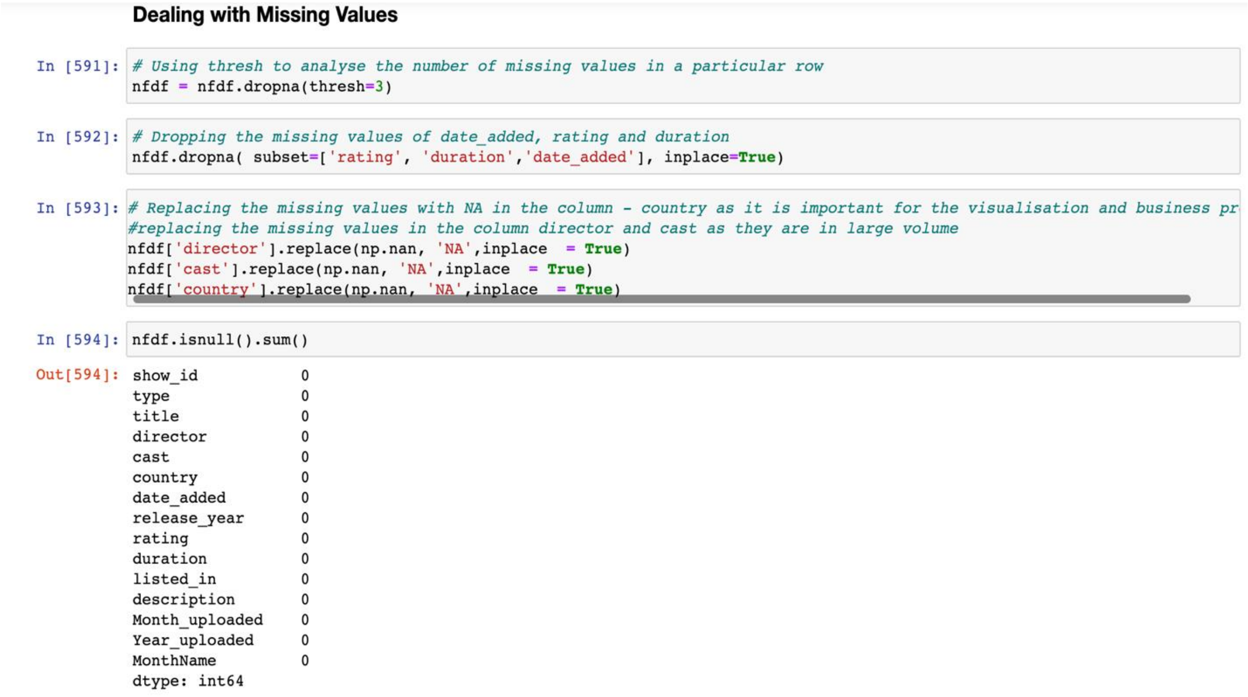
Visualising the missing values using the heatmap



```
In [589]: sns.heatmap(nfdf.isnull())
Out[589]: <AxesSubplot:>
```

It is evident from the graph and the output of code that there are missing values in the columns director, cast, country, date_added, rating and duration.

Handling Missing Values



Dealing with Missing Values

In [591]:
```
# Using thresh to analyse the number of missing values in a particular row
nfdf = nfdf.dropna(thresh=3)
```

In [592]:
```
# Dropping the missing values of date_added, rating and duration
nfdf.dropna( subset=['rating', 'duration','date_added'], inplace=True)
```

In [593]:
```
# Replacing the missing values with NA in the column - country as it is important for the visualisation and business pr
#replacing the missing values in the column director and cast as they are in large volume
nfdf['director'].replace(np.nan, 'NA',inplace  = True)
nfdf['cast'].replace(np.nan, 'NA',inplace  = True)
nfdf['country'].replace(np.nan, 'NA',inplace  = True)
```

In [594]:
```
nfdf.isnull().sum()
```

Out[594]:
```
show_id           0
type              0
title             0
director          0
cast              0
country           0
date_added        0
release_year      0
rating            0
duration          0
listed_in         0
description       0
Month_uploaded    0
Year_uploaded     0
MonthName         0
dtype: int64
```

Handling missing values is an important task in the data preparation process as without clearing or handling the missing values, the output or the visualisations created can give false insights; thus before working with the dataset, it is essential to handle the missing values.

I have used dropna() function to drop the missing values found in the dataset,

Dropped the missing values of column rating, duration and date_added because they were low in the volume of the missing values.

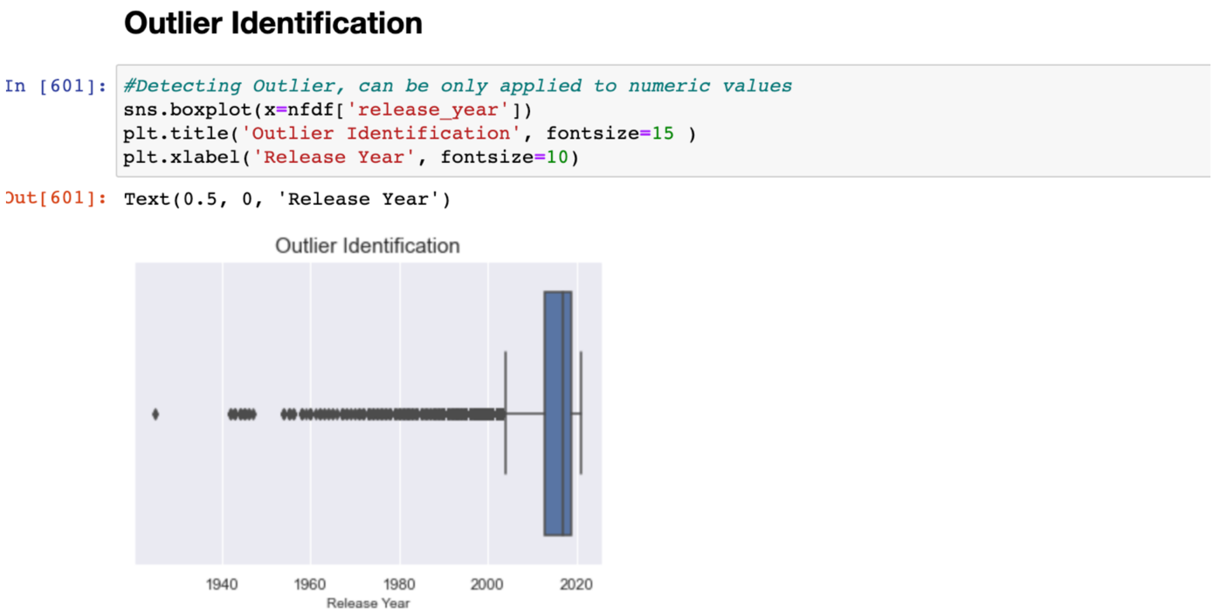
Additionally, I have replaced the missing values of column country with ‘NA’ as the country column data is important for solving the business problem.

I have also replaced the missing values of columns named cast and director because the missing values were high in volume.

In code, Numpy NaN is used to replace the missing values with the replace() function and inplace= True is used to update the data frame.

2.4 Outlier identification

To check the outlier in the given data frame, I used the boxplot to visualise and identify the outlier. Since this dataset consists of only one numerical data type, i.e. release_year the outlier can be identified with this column.

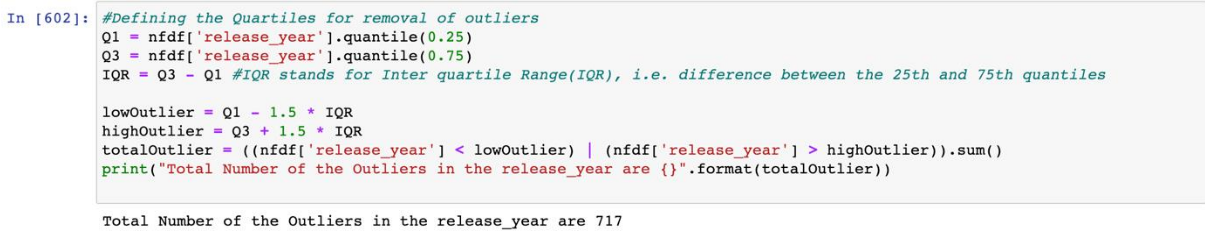


Outlier Identification

```
In [601]: #Detecting Outlier, can be only applied to numeric values
          sns.boxplot(x=nfdf['release_year'])
          plt.title('Outlier Identification', fontsize=15 )
          plt.xlabel('Release Year', fontsize=10)

Out[601]: Text(0.5, 0, 'Release Year')
```

With the above output, it is evident that there are outliers in the release_year column; the box plot helps to identify the outlier i.e. the data which is either low or high to the desired relevant information.

Handling the outliers



```
In [602]: #Defining the Quartiles for removal of outliers
          Q1 = nfdf['release_year'].quantile(0.25)
          Q3 = nfdf['release_year'].quantile(0.75)
          IQR = Q3 - Q1 #IQR stands for Inter quartile Range(IQR), i.e. difference between the 25th and 75th quantiles

          lowOutlier = Q1 - 1.5 * IQR
          highOutlier = Q3 + 1.5 * IQR
          totalOutlier = ((nfdf['release_year'] < lowOutlier) | (nfdf['release_year'] > highOutlier)).sum()
          print("Total Number of the Outliers in the release_year are {}".format(totalOutlier))

          Total Number of the Outliers in the release_year are 717
```

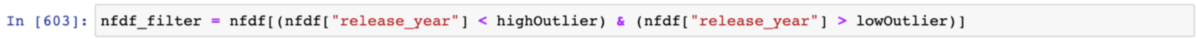
I have used the Inter Quartile Range method to find the number of outliers in the release_year.

The lowOutlier is being calculated using the formula i.e. Q1 – 1.5 * IQR, while the highOutlier (75 percentile) is calculated by Q3 + 1.5 * IQR.

In this method, the outlier can be found by calculating the difference between the 1st quartile i.e. the 25 percentile and the 3rd quartile, i.e. 75 percentile. And the remaining values are known as the middle 50% of the values of the attribute.

The total number of outliers found in the release_year are 717.

Removing the outliers

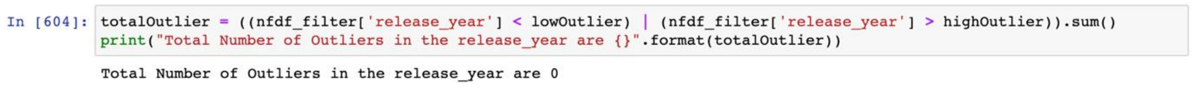


```
In [603]: nfdf_filter = nfdf[(nfdf["release_year"] < highOutlier) & (nfdf["release_year"] > lowOutlier)]
```

The outliers are removed by the above code stating that the resulting values should be lower than highOutlier and higher than lowOutlier.

Validating the removing of outliers

Printing the "totalOutlier" created to check if there are any duplicate values.

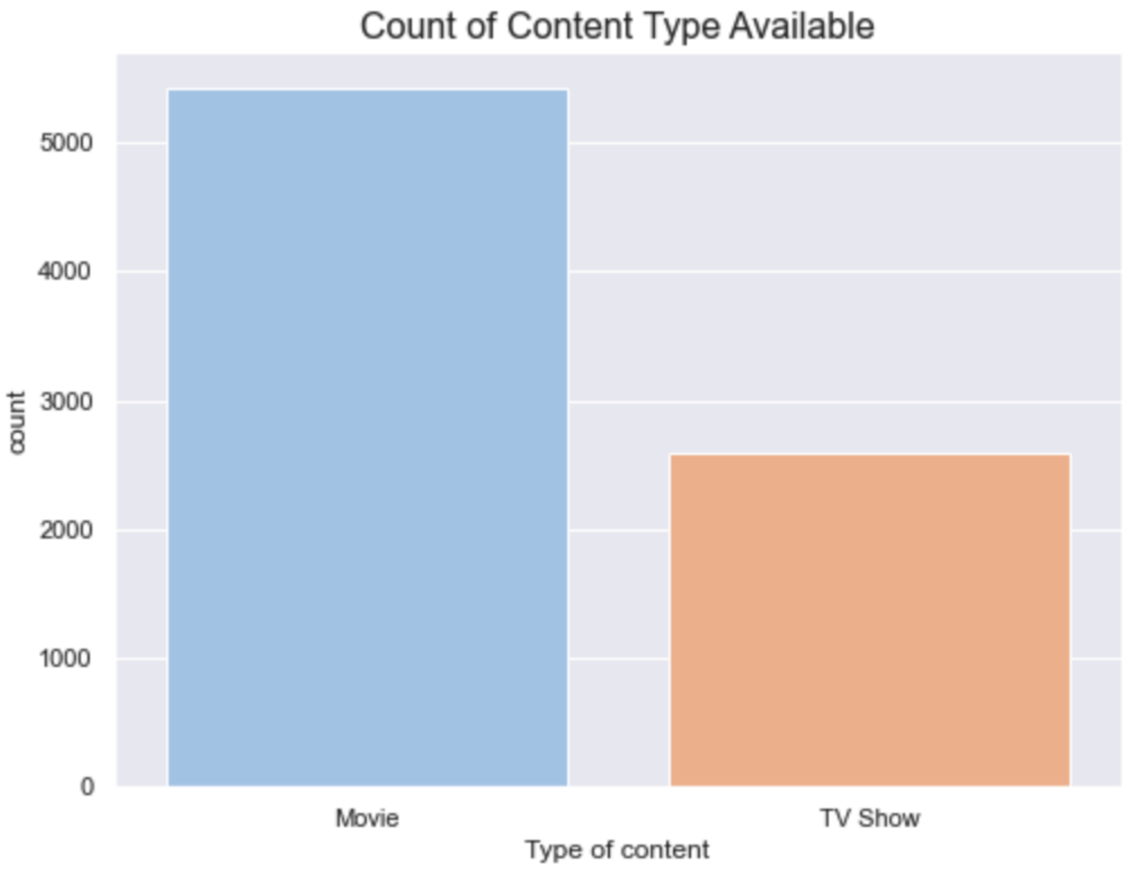


```
In [604]: totalOutlier = ((nfdf_filter['release_year'] < lowOutlier) | (nfdf_filter['release_year'] > highOutlier)).sum()
          print("Total Number of Outliers in the release_year are {}".format(totalOutlier))

          Total Number of Outliers in the release_year are 0
```

2.5 Data Visualization

The first visualisation made to solve the business problem is the bar-plot which is made the count plot of a seaborn library for interpreting the distribution of content according to the type.

This visualisation helps to understand the distribution of content with the two types i.e. Movies and TV shows.

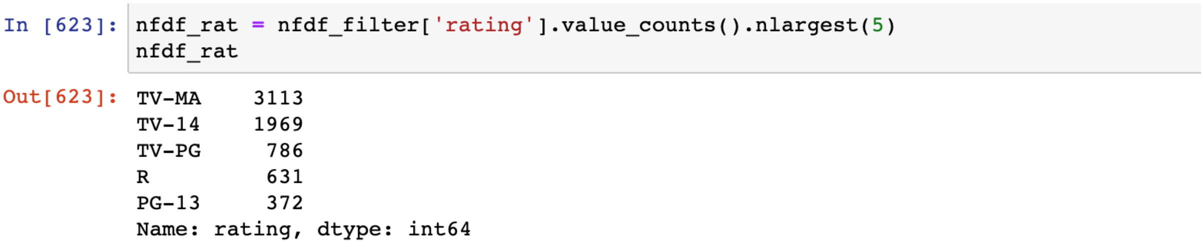


It is evident from the above graph that content in the type of movies is higher in a significant amount than in TV shows. As people could like to watch movies more often than tv shows due to the shorter duration.

Content available according to the rating

As there are many ratings, I have visualised for the content available to top five ratings which means the content available to this rating is higher than others.

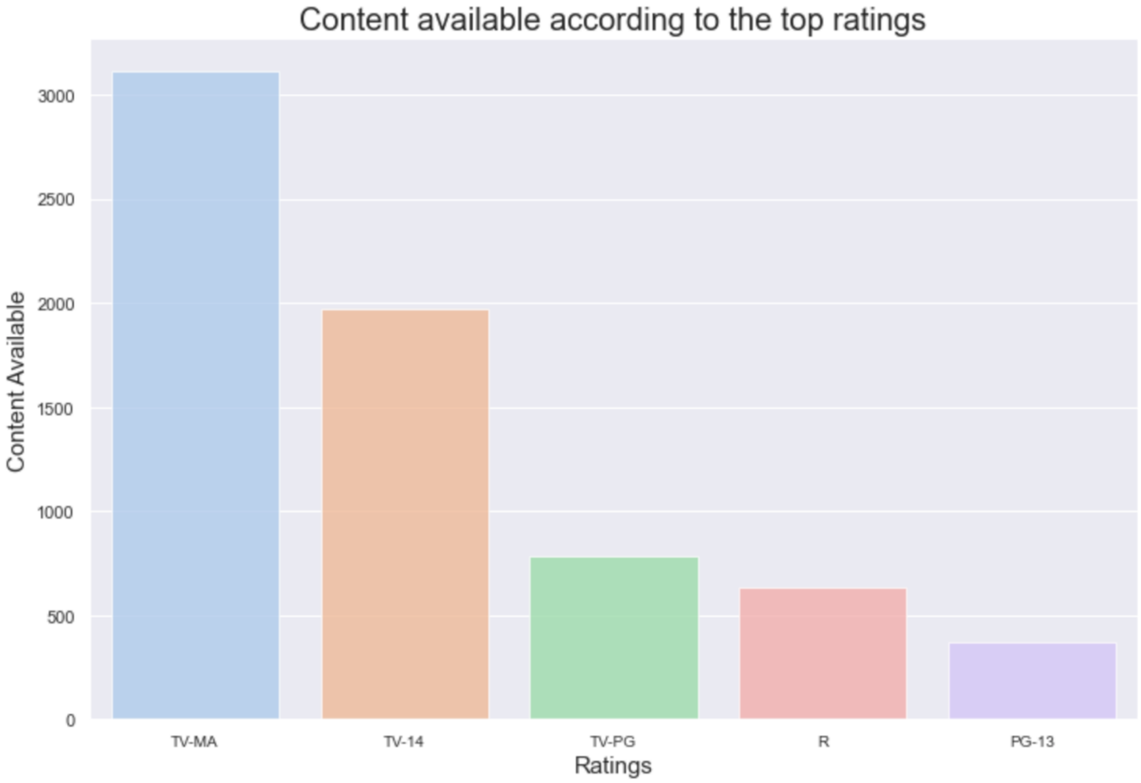
To calculate the highest ratings I have used the value_counts() function and the nlargest() function and then used the visualisation technique to make the bar-plot.

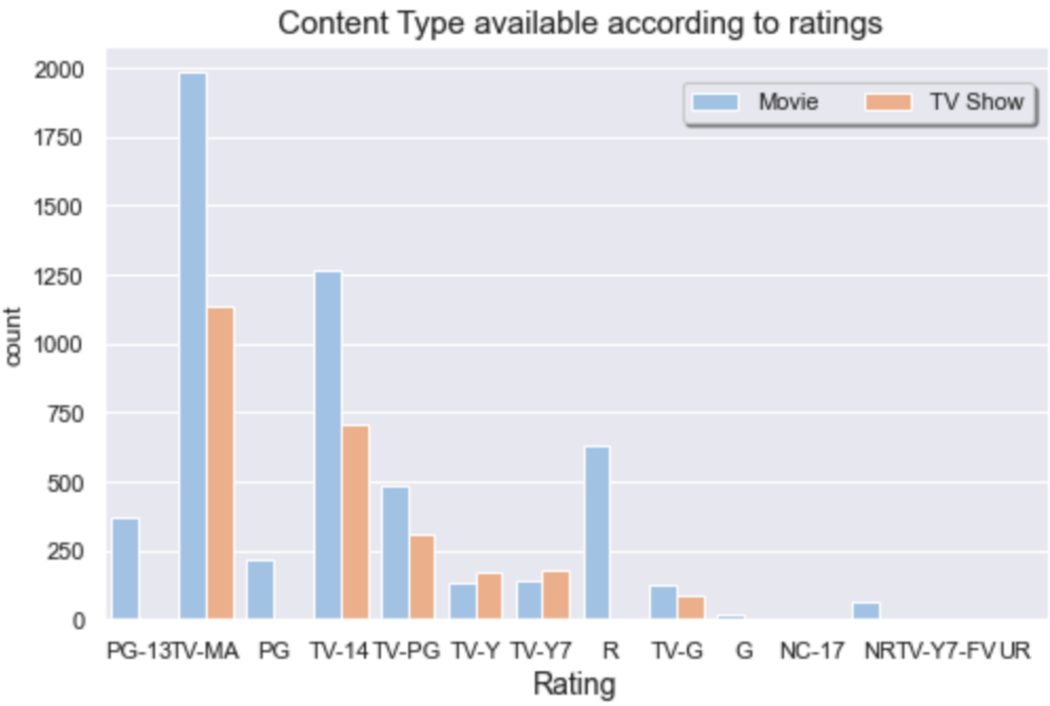


```
In [623]: nfdf_rat = nfdf_filter['rating'].value_counts().nlargest(5)
          nfdf_rat

Out[623]: TV-MA     3113
          TV-14     1969
          TV-PG      786
          R          631
          PG-13      372
          Name: rating, dtype: int64
```

The above graph states that the content available to TV-MA rating is the highest which means the content available to 15+ i.e. Mature Accompanied followed by TV-14, TV-PG, R and PG-13.

This information can be beneficial to the industry leader in deciding the distribution of content in respect of the rating of viewers according to the available content. Additionally, it is helpful for people to analyse the movies and tv shows content in respect of the rating they got from the authorized certification committee.



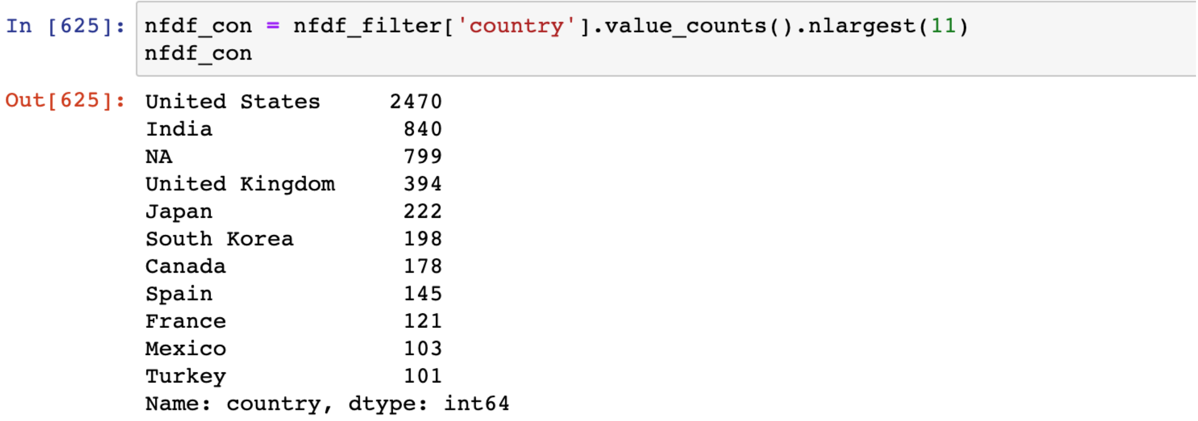


The above plot is made for bivariate analysis for displaying the ratings with the type of the content available on Netflix.

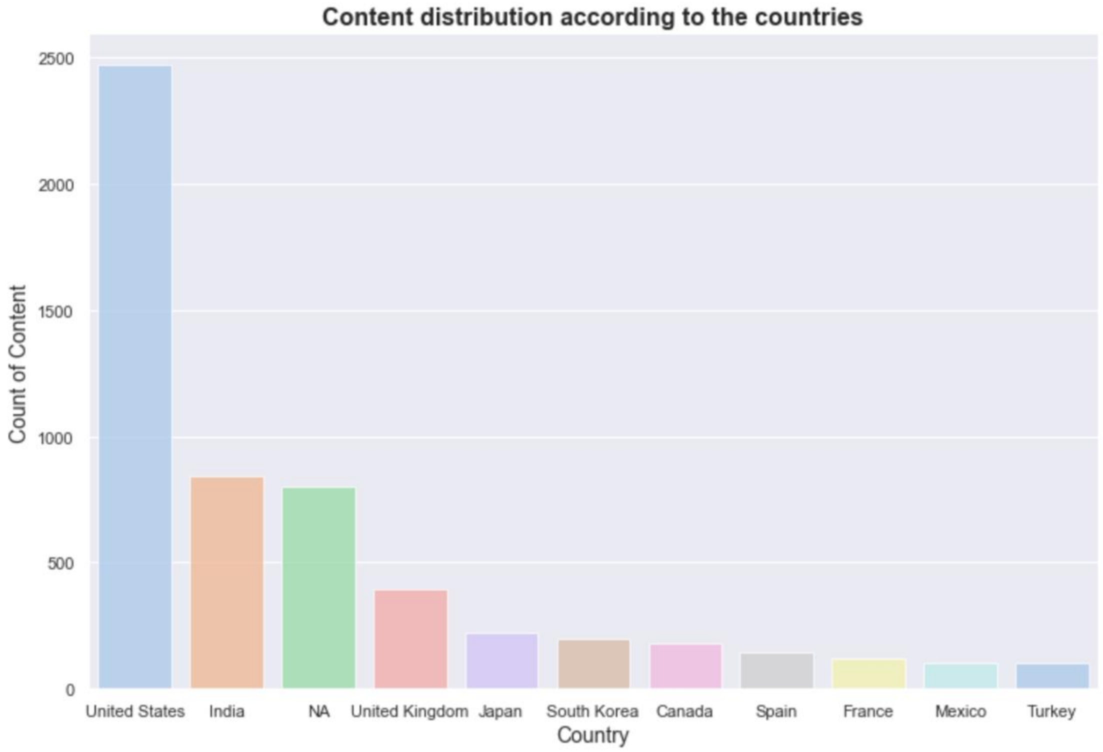
Content available according to the country

By visualising the content available according to the geographical data, it would be beneficial to industry leaders to decide which type of content is famous and liked by people to watch in which region.

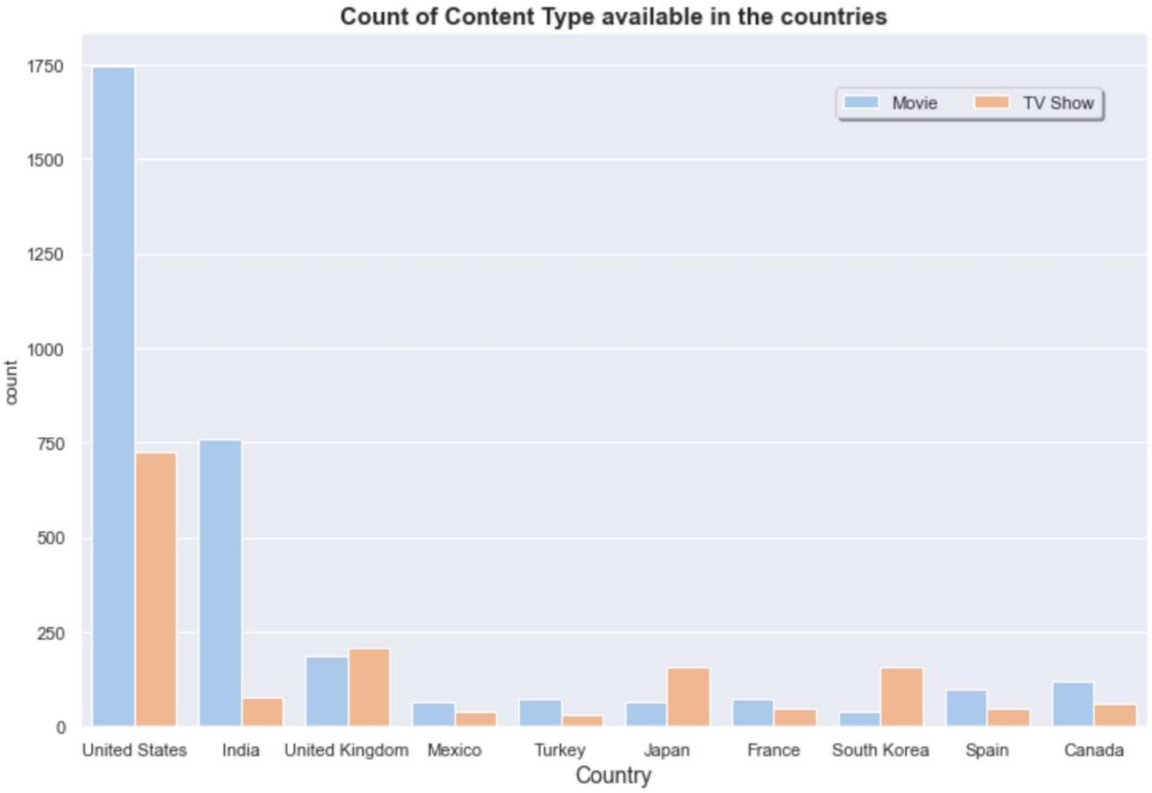
For visualising the content in respect of the countries I have made a data frame named nfdf_con for visualising the countries with the most content, i.e. the content made in the country, thus used the value_counts() and nlargest() function.



```
In [625]: nfdf_con = nfdf_filter['country'].value_counts().nlargest(11)
          nfdf_con

Out[625]: United States     2470
          India              840
          NA                 799
          United Kingdom     394
          Japan              222
          South Korea        198
          Canada             178
          Spain              145
          France             121
          Mexico             103
          Turkey             101
          Name: country, dtype: int64
```



It is evident from the plot that the USA is the major contributor to the entertainment industry followed by India. NA denotes “not available” which I have replaced while dealing with missing values.



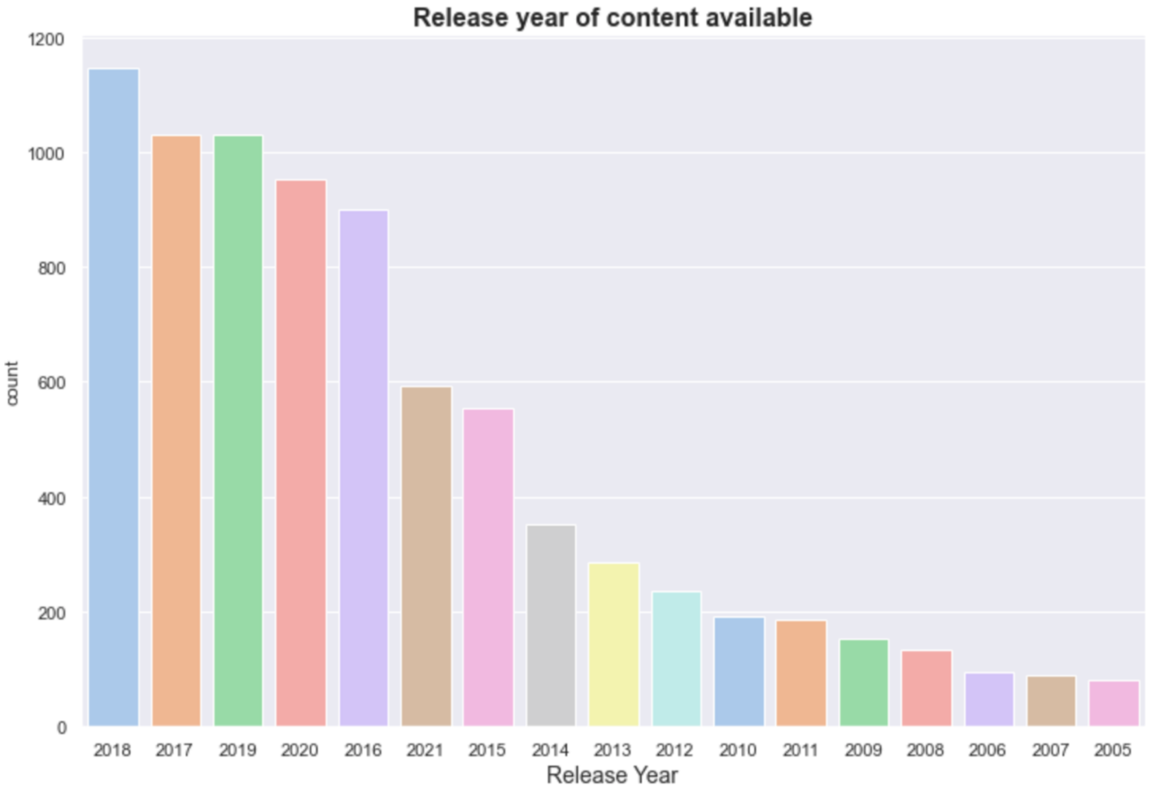
The above plot represents the content according to their types, it can be inference that Japan, South Korea and the United Kingdom are the only countries in which TV shows have more content than movies; it can be thought of as the people tend to watch more TV shows than movies in these countries. In contrast, the content available on Netflix is majorly movies.

Analysis of content released and uploaded

The content on Netflix can be analysed on the basis of two ways in respect of time associated with it-

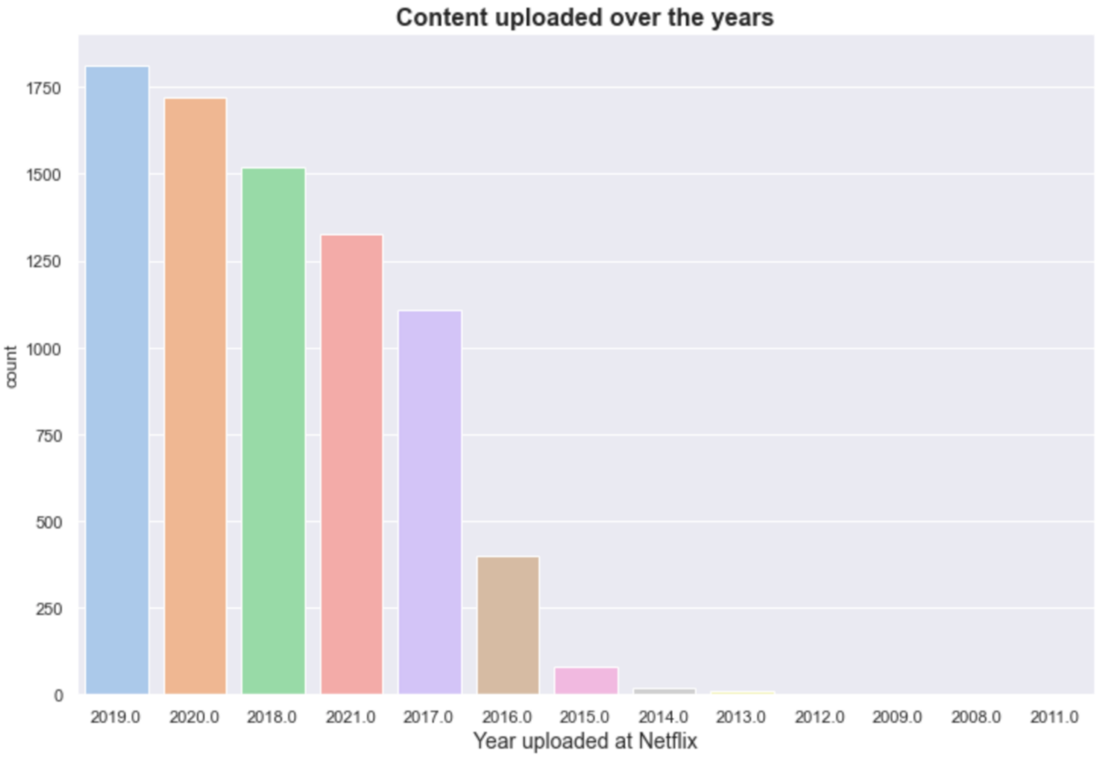
- **Release_year** - The year at which the content is being released is the release year.
- **Year_uploaded** – This column consists of the year values stating the content added to the Netflix, this column is added by splitting the date_added column.

The graph below shows that every year the content released gets increased except the year 2019, 2020 and 2021 which is due to the pandemic of coronavirus.



It is evident from the plot below that content uploaded on Netflix is in recent years stating the information that people like to watch few old movies on Netflix. Moreover, the same pattern of content reduction in years 2020 and 2021 can be seen in the uploading of content.

This information can help industry advisors in deciding which content to upload to Netflix.



A few other plots like histogram, correlation plot and pair plot are added in the notebook.

3 Discussion and Conclusions

The Netflix dataset has been cleaned and prepared to create data-driven insights; data cleaning also includes handling the outliers, missing values and duplicate values; without handling them, the insights created can give false or inaccurate information.

The steps included to create the valuable insights are Data Exploration, Data Cleaning And Data Visualisation.

The insights created solve the business problem of not knowing the content distribution in respect of the type, country, ratings duration i.e. seasons of TV shows and minutes of movies. And thus answer the business question of acknowledging the popular rating content, content most seen in certain countries i.e. popular in the world. Further answering the content distribution in respect of the duration so as to decide which content is popular according to duration and where should industry advisor focus on.

4 References

Lucid visual collaboration suite: Log in. (n.d.). Retrieved October 30, 2022, from https://lucid.app/users/login?referredProduct=lucidchart&returnUrlOverride=%2Flucidchart%2F65d52d50-da57-4e6a-9cc5-17fae6711293%2Fedit%3FbeaconFlowId%3DAA0648AAF179DF94