PRATEEK **RA2011003010491**

Q. Create an account in kaggle / github and take any bench mark dataset and do all data pre-processing technique using python code.
Submit the kaggle / github link as well as submit the entire worksheet in pdf format.

Netflix : Data Pre-processing And Analysis

Github Link - https://github.com/Prateek-sn-coder/Data-PreProcessing

1. Discussing the dataset

This data set consist on contents added to Netflix from 2008 to 2021. The variables of this data set are:

- *show_id*: Netflix ID of the media.
- *Type*: Movie or TV Show.
- *title*: Title of the media.
- *director*: Director of the media.
- *country*: Country in which the movie was made.
- *date_added*: Date in which the media was added.
- *release_year*: Year in which the media was released.
- *rating*: Age rating of the media.
- *duration*: Duration of the media.
- *listen_in*: Classification given by Netflix.

Our data is uploaded to Google Drive. It is saved in the root directory.
To use this data, we need to give Google Colab access to Google Drive. So let’s type and run the code below in Google Colab.

```
from google.colab import drive
drive.mount("/content/drive/")
```

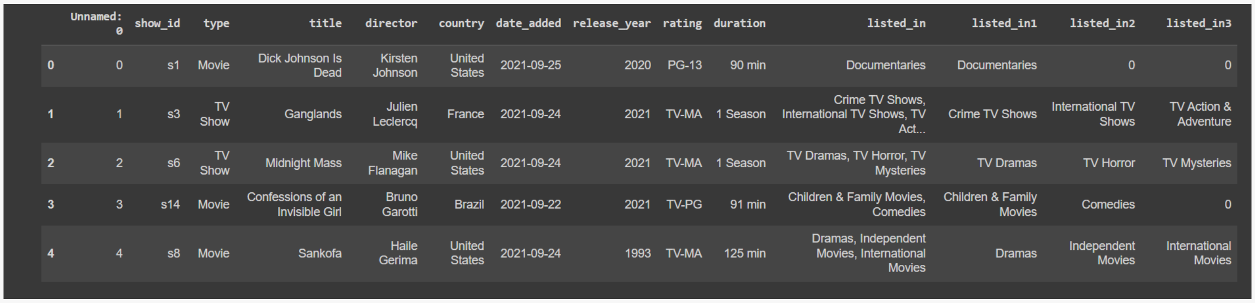
2. Importing Libraries

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

3. Gathering

We gather the dataset and turn it into a DataFrame.

```
# Importing the data from a csv file to a DataFrame
df =
pd.read_csv("../input/netflix-data-cleaning-analysis-and-visualization/net
flix1.csv")
# Showing the first five values of the DataFrame
df.head()
```



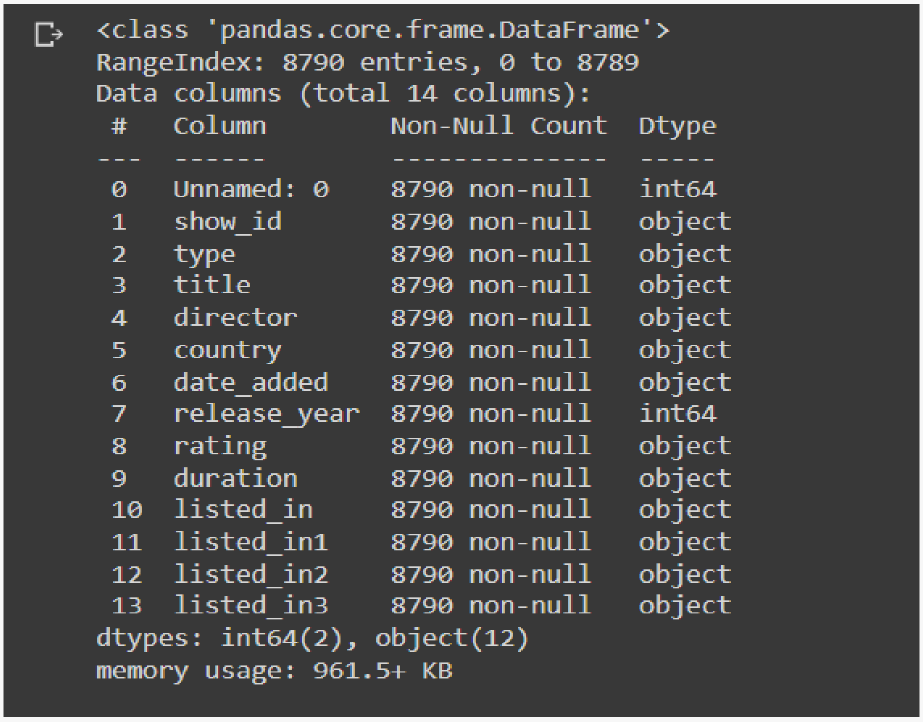
	Unnamed: 0	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in	listed_in1	listed_in2	listed_in3
0	0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	2021-09-25	2020	PG-13	90 min	Documentaries	Documentaries	0	0
1	1	s3	TV Show	Ganglands	Julien Leclercq	France	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	Crime TV Shows	International TV Shows	TV Action & Adventure
2	2	s6	TV Show	Midnight Mass	Mike Flanagan	United States	2021-09-24	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	TV Dramas	TV Horror	TV Mysteries
3	3	s14	Movie	Confessions of an Invisible Girl	Bruno Garotti	Brazil	2021-09-22	2021	TV-PG	91 min	Children & Family Movies, Comedies	Children & Family Movies	Comedies	0
4	4	s8	Movie	Sankofa	Haile Gerima	United States	2021-09-24	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	Dramas	Independent Movies	International Movies

4.0 Assessing

This section of the report, we will assess any issues the data may have.

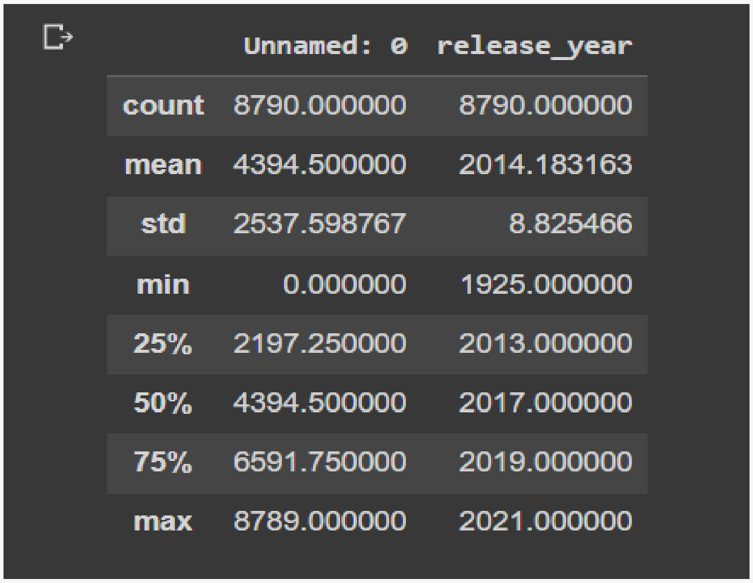
```
# Let's check the status of the data
```

```
df.info()
```



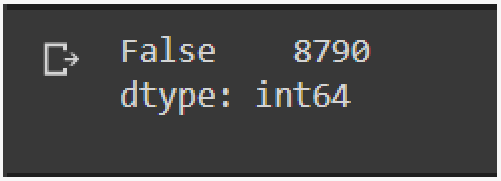
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Unnamed: 0    8790 non-null   int64 
 1   show_id       8790 non-null   object
 2   type          8790 non-null   object
 3   title         8790 non-null   object
 4   director      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  listed_in1    8790 non-null   object
 12  listed_in2    8790 non-null   object
 13  listed_in3    8790 non-null   object
dtypes: int64(2), object(12)
memory usage: 961.5+ KB
```

```
df.describe()
```



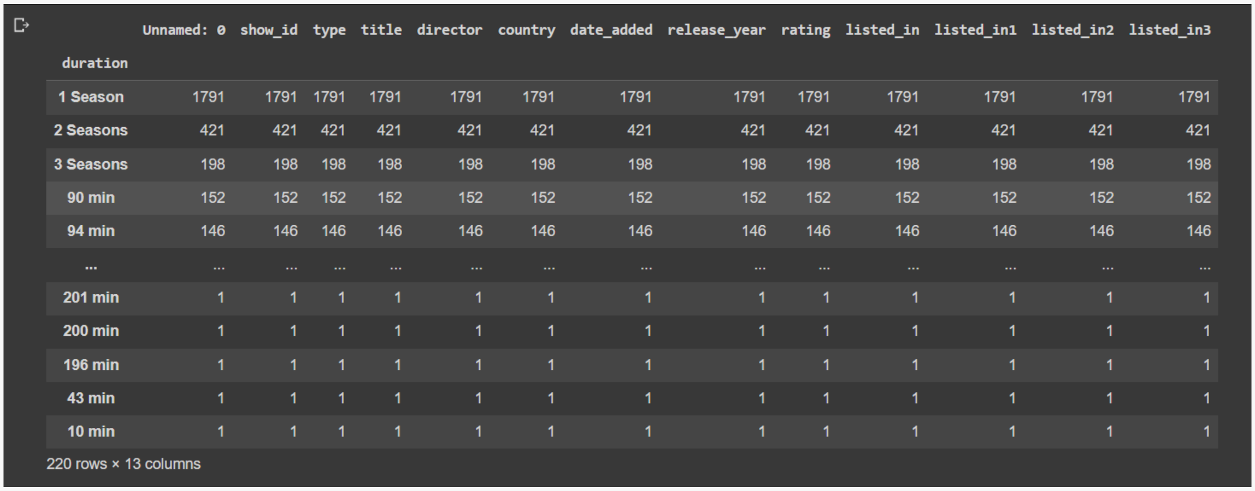
	Unnamed: 0	release_year
count	8790.000000	8790.000000
mean	4394.500000	2014.183163
std	2537.598767	8.825466
min	0.000000	1925.000000
25%	2197.250000	2013.000000
50%	4394.500000	2017.000000
75%	6591.750000	2019.000000
max	8789.000000	2021.000000

```
# Checking if there are any duplicates

df.duplicated().value_counts()
```



```
False    8790
dtype: int64
```

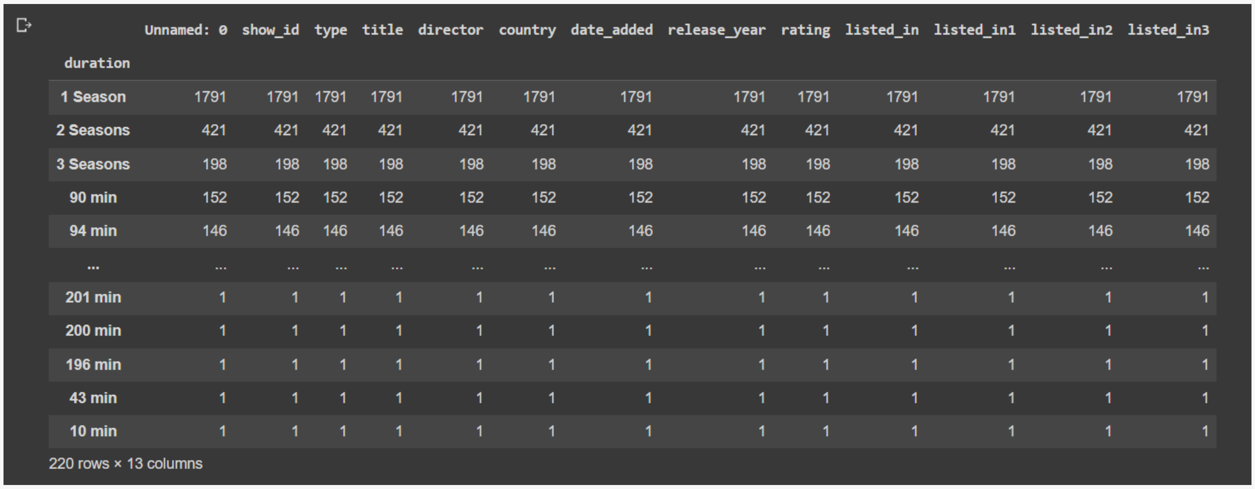
```
df.groupby('duration').count().sort_values(by='show_id',ascending=False)
```



duration	Unnamed: 0	show_id	type	title	director	country	date_added	release_year	rating	listed_in	listed_in1	listed_in2	listed_in3
1 Season	1791	1791	1791	1791	1791	1791	1791	1791	1791	1791	1791	1791	1791
2 Seasons	421	421	421	421	421	421	421	421	421	421	421	421	421
3 Seasons	198	198	198	198	198	198	198	198	198	198	198	198	198
90 min	152	152	152	152	152	152	152	152	152	152	152	152	152
94 min	146	146	146	146	146	146	146	146	146	146	146	146	146
...	...	...	...	...	...	...	...	...	...	...	...	...	...
201 min	1	1	1	1	1	1	1	1	1	1	1	1	1
200 min	1	1	1	1	1	1	1	1	1	1	1	1	1
196 min	1	1	1	1	1	1	1	1	1	1	1	1	1
43 min	1	1	1	1	1	1	1	1	1	1	1	1	1
10 min	1	1	1	1	1	1	1	1	1	1	1	1	1

220 rows × 13 columns

```
# Let's perform a basic visual analysis of the data.
# pd.set_option('display.max_rows', 220)
df.groupby('duration').count().sort_values(by='show_id',ascending=False)
```



duration	Unnamed: 0	show_id	type	title	director	country	date_added	release_year	rating	listed_in	listed_in1	listed_in2	listed_in3
1 Season	1791	1791	1791	1791	1791	1791	1791	1791	1791	1791	1791	1791	1791
2 Seasons	421	421	421	421	421	421	421	421	421	421	421	421	421
3 Seasons	198	198	198	198	198	198	198	198	198	198	198	198	198
90 min	152	152	152	152	152	152	152	152	152	152	152	152	152
94 min	146	146	146	146	146	146	146	146	146	146	146	146	146
...	...	...	...	...	...	...	...	...	...	...	...	...	...
201 min	1	1	1	1	1	1	1	1	1	1	1	1	1
200 min	1	1	1	1	1	1	1	1	1	1	1	1	1
196 min	1	1	1	1	1	1	1	1	1	1	1	1	1
43 min	1	1	1	1	1	1	1	1	1	1	1	1	1
10 min	1	1	1	1	1	1	1	1	1	1	1	1	1

220 rows × 13 columns

4.1 Assessment & Categorizing

4.1.1 Quality issue

- Variable 'date_added' has the wrong data type.
- Variable 'duration' has the wrong data type.

4.1.2 Tidiness issue

- The 'listed_in' variable has several categories in a single observation.
- There are two types of observations, TV shows and movies.

5. Cleaning

In this section of the report we will solve the quality and tidiness issues mentioned in the assessment.

```
# Before cleaning, lets make a copy of the dataframe.
```

```
df_clean = df.copy()
```

5.1 'date_added' variable has wrong data type

5.1.1 Define

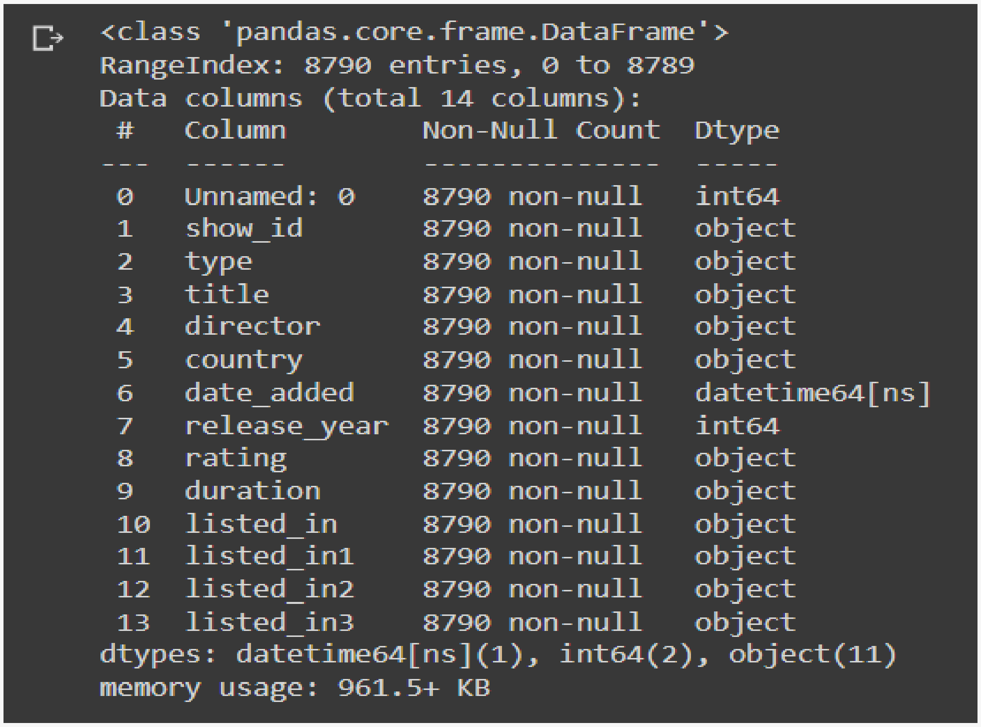
The variable 'date_added' has been categorised as an object (string), the most appropriate type of data for this variable would be datetime.

5.1.2 Code

```
df_clean.date_added = pd.to_datetime(df_clean.date_added)
```

5.1.3 Test

```
df_clean.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Unnamed: 0     8790 non-null   int64
 1   show_id        8790 non-null   object
 2   type           8790 non-null   object
 3   title          8790 non-null   object
 4   director       8790 non-null   object
 5   country        8790 non-null   object
 6   date_added     8790 non-null   datetime64[ns]
 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  listed_in1     8790 non-null   object
 12  listed_in2     8790 non-null   object
 13  listed_in3     8790 non-null   object
dtypes: datetime64[ns](1), int64(2), object(11)
memory usage: 961.5+ KB
```

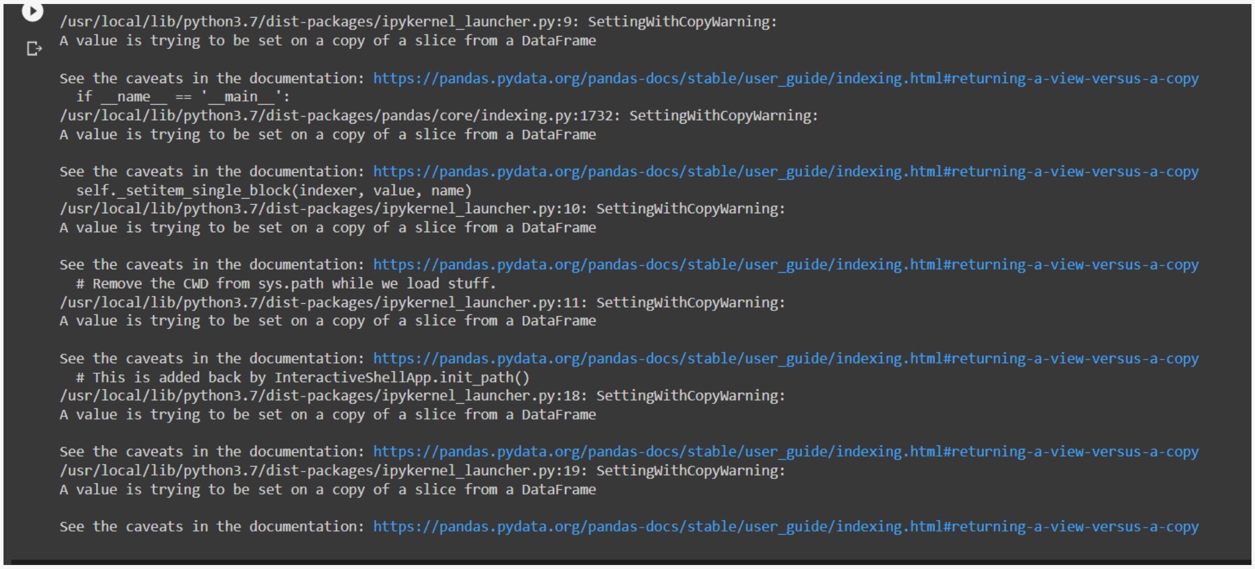
5.2 'listed_in' variable has several variables

5.2.1 Define

The 'listed_in' variable can have several categories per media, we would like to create new variables to be able to extract this and correctly filter the data. We will assume that the first category would be the 'main' category of the movie.

5.2.2 Code

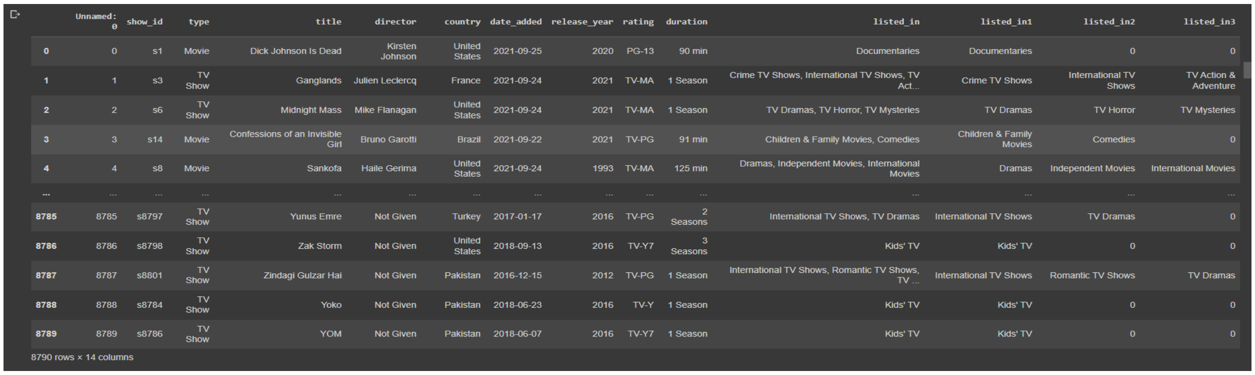
```
df_clean['listed_in1'] = 0
df_clean['listed_in2'] = 0
df_clean['listed_in3'] = 0
temp_cat = df_clean.listed_in.str.split(',')
i=0
for i in range (8790):
    t_cat = temp_cat[i]
    if len(t_cat) == 1:
        df_clean['listed_in1'][i] = temp_cat[i][0]
        df_clean['listed_in2'][i] = 0
        df_clean['listed_in3'][i] = 0
    if len(t_cat) == 2:
        df_clean['listed_in1'][i] = temp_cat[i][0]
        df_clean['listed_in2'][i] = temp_cat[i][1]
        df_clean['listed_in3'][i] = 0
    if len(t_cat) == 3:
        df_clean['listed_in1'][i] = temp_cat[i][0]
        df_clean['listed_in2'][i] = temp_cat[i][1]
        df_clean['listed_in3'][i] = temp_cat[i][2]
```



```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  if __name__ == '__main__':
/usr/local/lib/python3.7/dist-packages/pandas/core/indexing.py:1732: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:19: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

5.2.3 Test

```
df_clean
```



	Unnamed: 0	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in	listed_in1	listed_in2	listed_in3
0	0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	2021-09-25	2020	PG-13	90 min	Documentaries	Documentaries	0	0
1	1	s3	TV Show	Ganglands	Julien Leclercq	France	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	Crime TV Shows	International TV Shows	TV Action & Adventure
2	2	s6	TV Show	Midnight Mass	Mike Flanagan	United States	2021-09-24	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	TV Dramas	TV Horror	TV Mysteries
3	3	s14	Movie	Confessions of an Invisible Girl	Bruno Garotti	Brazil	2021-09-22	2021	TV-PG	91 min	Children & Family Movies, Comedies	Children & Family Movies	Comedies	0
4	4	s8	Movie	Sankofa	Haile Gerima	United States	2021-09-24	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	Dramas	Independent Movies	International Movies
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8785	8785	s8797	TV Show	Yunus Emre	Not Given	Turkey	2017-01-17	2016	TV-PG	2 Seasons	International TV Shows, TV Dramas	International TV Shows	TV Dramas	0
8786	8786	s8798	TV Show	Zak Storm	Not Given	United States	2018-09-13	2016	TV-Y7	3 Seasons	Kids' TV	Kids' TV	0	0
8787	8787	s8801	TV Show	Zindagi Gulzar Hai	Not Given	Pakistan	2016-12-15	2012	TV-PG	1 Season	International TV Shows, Romantic TV Shows, TV ...	International TV Shows	Romantic TV Shows	TV Dramas
8788	8788	s8784	TV Show	Yoko	Not Given	Pakistan	2018-06-23	2016	TV-Y	1 Season	Kids' TV	Kids' TV	0	0
8789	8789	s8786	TV Show	YOM	Not Given	Pakistan	2018-06-07	2016	TV-Y7	1 Season	Kids' TV	Kids' TV	0	0

8790 rows × 14 columns

5.3 Two different observation on a single dataset

5.3.1 Define

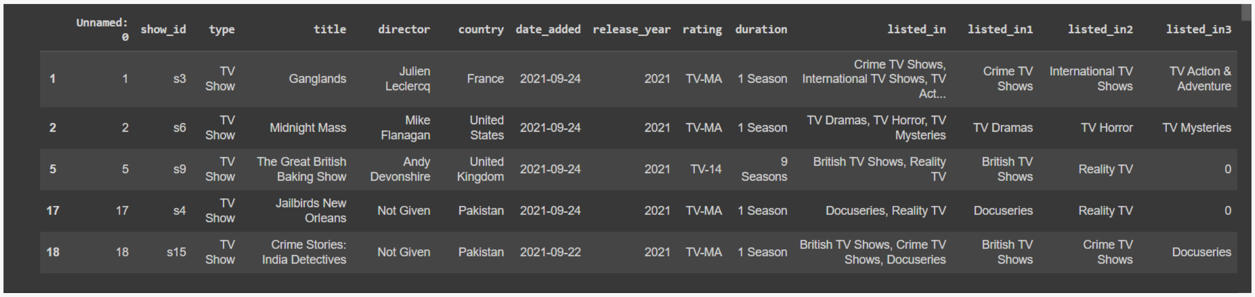
There are two different types of observations in a single data set: TV Shows and Movies. The solution would be to split the dataset into two.

5.3.2 Code

```
df_tv = df_clean[df_clean.type == 'TV Show']
df_movie = df_clean[df_clean.type == 'Movie']
```

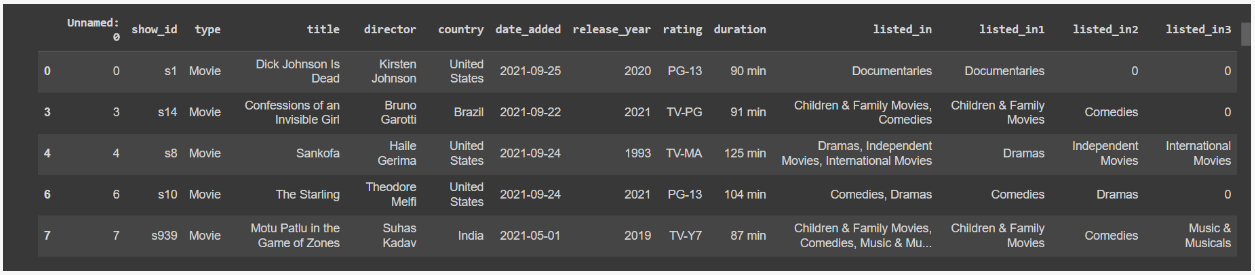
5.3.3 Test

```
df_tv.head()
```



	Unnamed: 0	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in	listed_in1	listed_in2	listed_in3
1	1	s3	TV Show	Ganglands	Julien Leclercq	France	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	Crime TV Shows	International TV Shows	TV Action & Adventure
2	2	s6	TV Show	Midnight Mass	Mike Flanagan	United States	2021-09-24	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	TV Dramas	TV Horror	TV Mysteries
5	5	s9	TV Show	The Great British Baking Show	Andy Devonshire	United Kingdom	2021-09-24	2021	TV-14	9 Seasons	British TV Shows, Reality TV	British TV Shows	Reality TV	0
17	17	s4	TV Show	Jailbirds New Orleans	Not Given	Pakistan	2021-09-24	2021	TV-MA	1 Season	Docuseries, Reality TV	Docuseries	Reality TV	0
18	18	s15	TV Show	Crime Stories: India Detectives	Not Given	Pakistan	2021-09-22	2021	TV-MA	1 Season	British TV Shows, Crime TV Shows, Docuseries	British TV Shows	Crime TV Shows	Docuseries

```
df_movie.head()
```



	Unnamed: 0	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in	listed_in1	listed_in2	listed_in3
0	0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	2021-09-25	2020	PG-13	90 min	Documentaries	Documentaries	0	0
3	3	s14	Movie	Confessions of an Invisible Girl	Bruno Garotti	Brazil	2021-09-22	2021	TV-PG	91 min	Children & Family Movies, Comedies	Children & Family Movies	Comedies	0
4	4	s8	Movie	Sankofa	Haile Gerima	United States	2021-09-24	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	Dramas	Independent Movies	International Movies
6	6	s10	Movie	The Starling	Theodore Melfi	United States	2021-09-24	2021	PG-13	104 min	Comedies, Dramas	Comedies	Dramas	0
7	7	s939	Movie	Motu Patlu in the Game of Zones	Suhas Kadav	India	2021-05-01	2019	TV-Y7	87 min	Children & Family Movies, Comedies, Music & Mu...	Children & Family Movies	Comedies	Music & Musicals

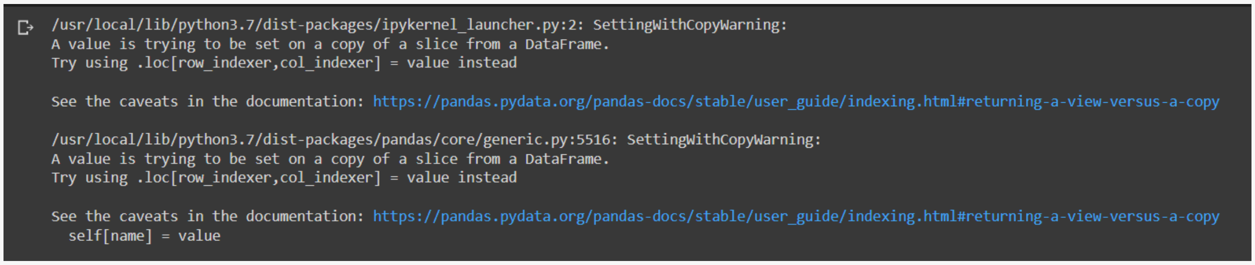
5.4 Variable 'duration' has the wrong data type

5.4.1 Define

Whilst movies and TV shows were combined into a single dataframe, it was not possible to easily compare the length of these medias. However, now that they are separated each of these variables are not require to be kept as an object; but instead they should be integers.

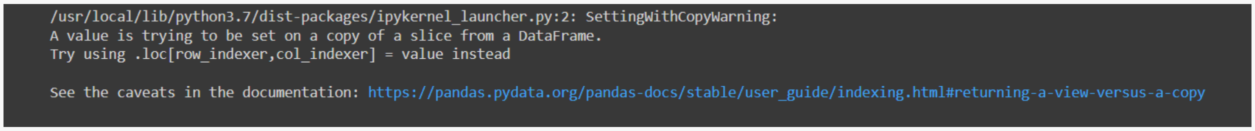
5.4.2 Code

```
temp_dur = df_tv.duration.str.split(' ',expand=True)
df_tv['duration_seasons'] = temp_dur[0]
df_tv.duration_seasons = pd.to_numeric(df_tv.duration_seasons)
```



```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/usr/local/lib/python3.7/dist-packages/pandas/core/generic.py:5516: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self[name] = value
```

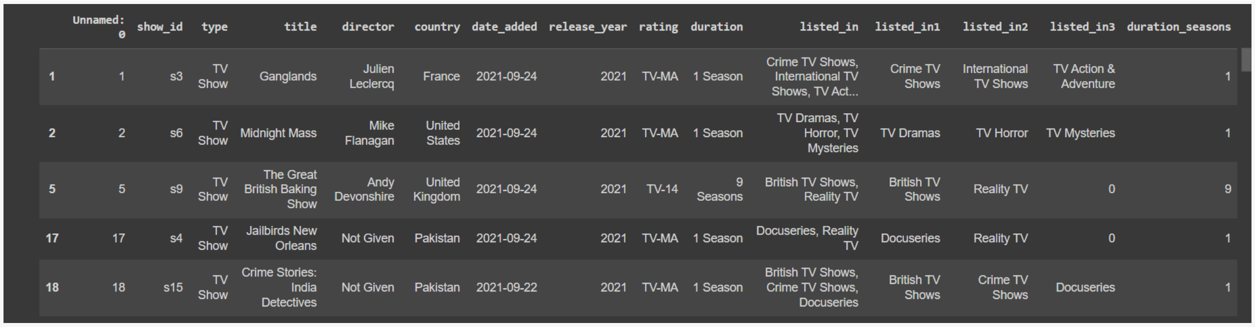
```
temp_dur = df_movie.duration.str.split(' ',expand=True)

df_movie['duration_minutes'] = temp_dur[0]

df_movie.duration_minutes = pd.to_numeric(df_movie.duration_minutes)
```



```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

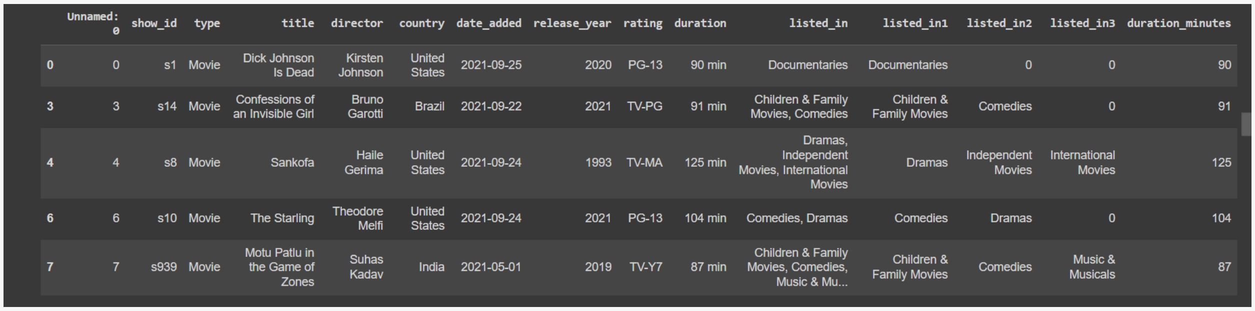
5.4.2 Test

```
df_tv.head()
```



	Unnamed: 0	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in	listed_in1	listed_in2	listed_in3	duration_seasons
1	1	s3	TV Show	Ganglands	Julien Leclercq	France	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	Crime TV Shows	International TV Shows	TV Action & Adventure	1
2	2	s6	TV Show	Midnight Mass	Mike Flanagan	United States	2021-09-24	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	TV Dramas	TV Horror	TV Mysteries	1
5	5	s9	TV Show	The Great British Baking Show	Andy Devonshire	United Kingdom	2021-09-24	2021	TV-14	9 Seasons	British TV Shows, Reality TV	British TV Shows	Reality TV	0	9
17	17	s4	TV Show	Jailbirds New Orleans	Not Given	Pakistan	2021-09-24	2021	TV-MA	1 Season	Docuseries, Reality TV	Docuseries	Reality TV	0	1
18	18	s15	TV Show	Crime Stories: India Detectives	Not Given	Pakistan	2021-09-22	2021	TV-MA	1 Season	British TV Shows, Crime TV Shows, Docuseries	British TV Shows	Crime TV Shows	Docuseries	1

```
df_movie.head()
```



	Unnamed: 0	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in	listed_in1	listed_in2	listed_in3	duration_minutes
0	0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	2021-09-25	2020	PG-13	90 min	Documentaries	Documentaries	0	0	90
3	3	s14	Movie	Confessions of an Invisible Girl	Bruno Garotti	Brazil	2021-09-22	2021	TV-PG	91 min	Children & Family Movies, Comedies	Children & Family Movies	Comedies	0	91
4	4	s8	Movie	Sankofa	Haile Gerima	United States	2021-09-24	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	Dramas	Independent Movies	International Movies	125
6	6	s10	Movie	The Starling	Theodore Melfi	United States	2021-09-24	2021	PG-13	104 min	Comedies, Dramas	Comedies	Dramas	0	104
7	7	s939	Movie	Motu Patlu in the Game of Zones	Suhas Kadav	India	2021-05-01	2019	TV-Y7	87 min	Children & Family Movies, Comedies, Music & Mu...	Children & Family Movies	Comedies	Music & Musicals	87

6. Storing

In this step, we will store the dataframes into CSV files.

```
df_clean.to_csv('Netflix_DF_cleaned.csv')
```

```
df_tv.to_csv('Netflix_TV_cleaned.csv')
```

```
df_movie.to_csv('Netflix_Movie_cleaned.csv')
```

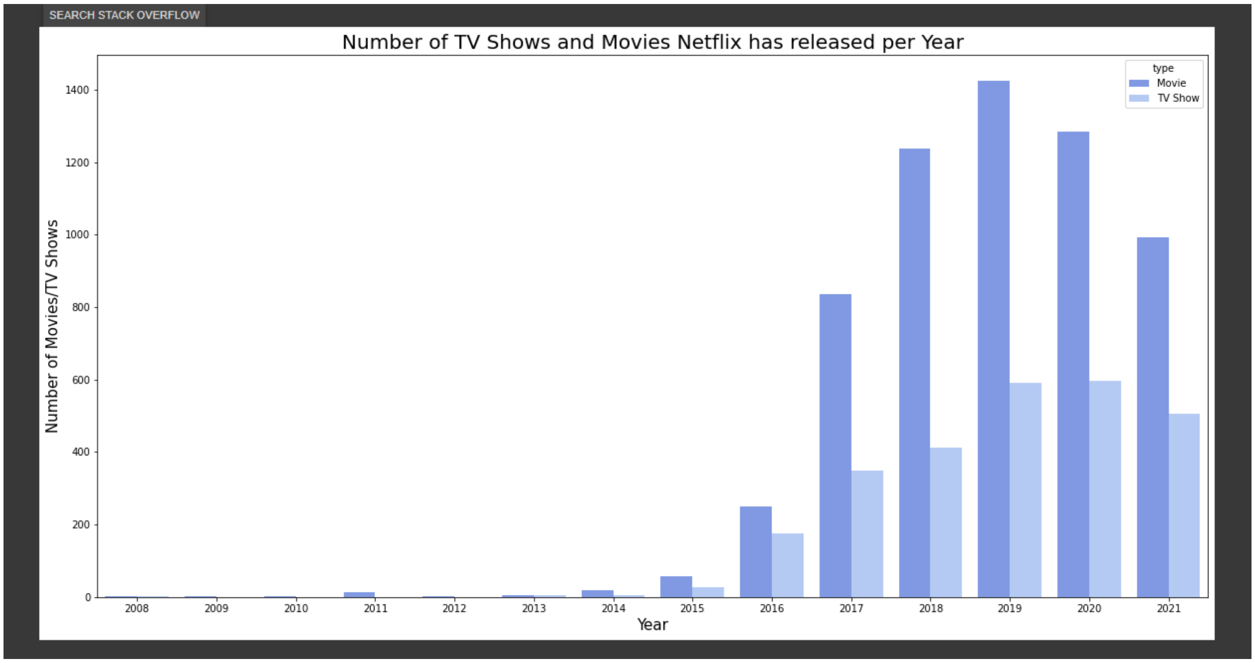
7. Analysing and Visualisation of Data

In this section of the report we will explore the answers for the following questions:

- What type of media has Netflix produced the most?
- Which country produced the most of Netflix's media?
- What are the most popular genres for countries that produced media?
- What is the relationship between the year a media was made and when added to the Netflix platform?
- Has Netflix's media classification changed over time?
- What are the most popular genres for Netflix media?
- Has the length of TV seasons or Movie's length changed over time?

7.1 General

```
# Considering there are too many countries, we will limit our study to just the top 10 countries.
plt.figure(figsize=[20,10])
base_color = sns.color_palette('coolwarm',n_colors=5)
tv_movie = sns.countplot(x=df_clean.date_added.dt.year, data=df_clean, hue='type', palette = base_color)
tv_movie.set_title("Number of TV Shows and Movies Netflix has released per Year",fontsize = 20)
tv_movie.set_xlabel('Year',fontsize = 15)
tv_movie.set_ylabel('Number of Movies/TV Shows',fontsize = 15)
for container in tv_movie.containers:
    tv_movie.bar_label(container)
```



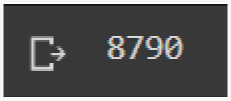
7.1.1 Comments on the Number of Netflix's media released per year.

It seems like for both TV shows and Movies there has been a steady increase since the start of 2008; the only big drop happening in 2021 possibly due to the economic impact of COVID.

Before 2017, the number of TV Shows and Movies brought to the streaming service was on par. However, after 2017 the company started introducing more movies into the service more than doubling TV Shows in the amount of content.

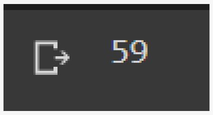
7.2 TV Shows

```
df_clean.date_added.dt.year.count()
```



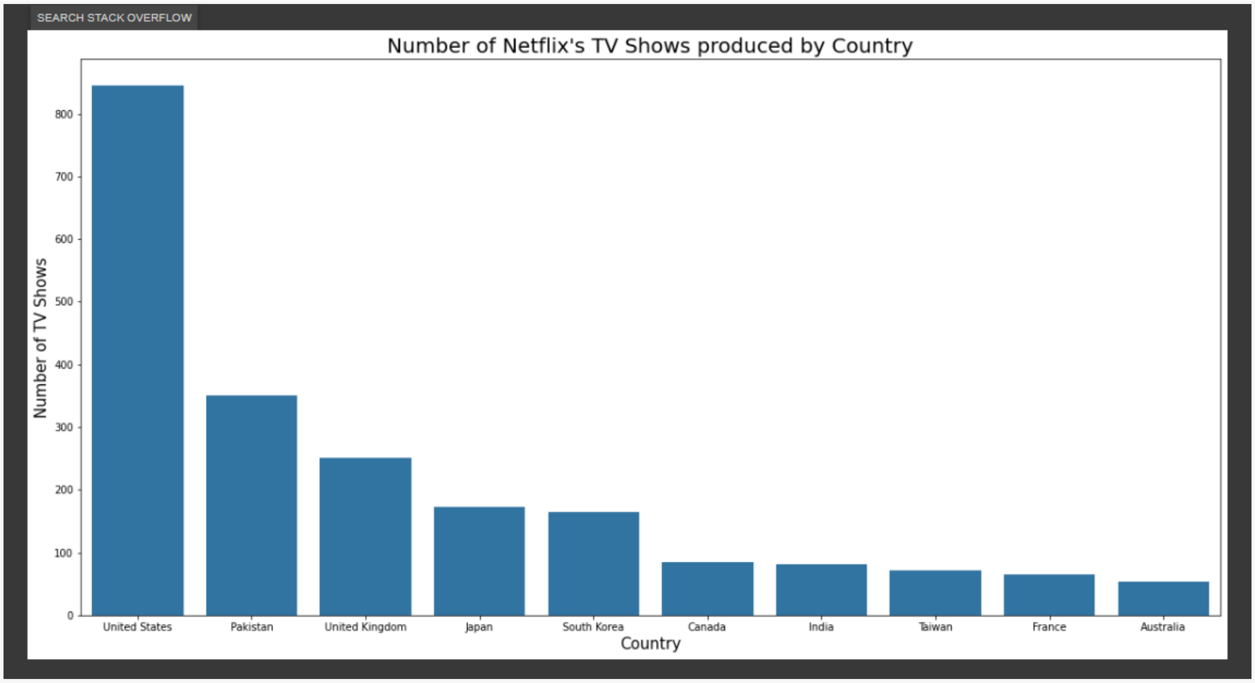
```
8790
```

```
#Let's check how many countries have produced a TV Show for Netlix
len(df_tv.groupby('country').count().index)
```



```
59
```

```
# Considering there are too many countries, we will limit our study to just
the top 10 countries.
sort_order = df_tv.groupby('country').count().sort_values(by =
'show_id',ascending=False)[0:10].index
df_tv_c = df_tv[df_tv['country'].isin(sort_order)]
base_color = base_color = sns.color_palette()[0]
plt.figure(figsize=[20,10])
tv_c = sns.countplot(x='country',data=df_tv_c,order=sort_order, color =
base_color)
tv_c.set_title("Number of Netflix's TV Shows produced by Country",fontsize
= 20)
tv_c.set_xlabel('Country',fontsize = 15)
tv_c.set_ylabel('Number of TV Shows',fontsize = 15)
for container in tv_c.containers:
    tv_c.bar_label(container)
```

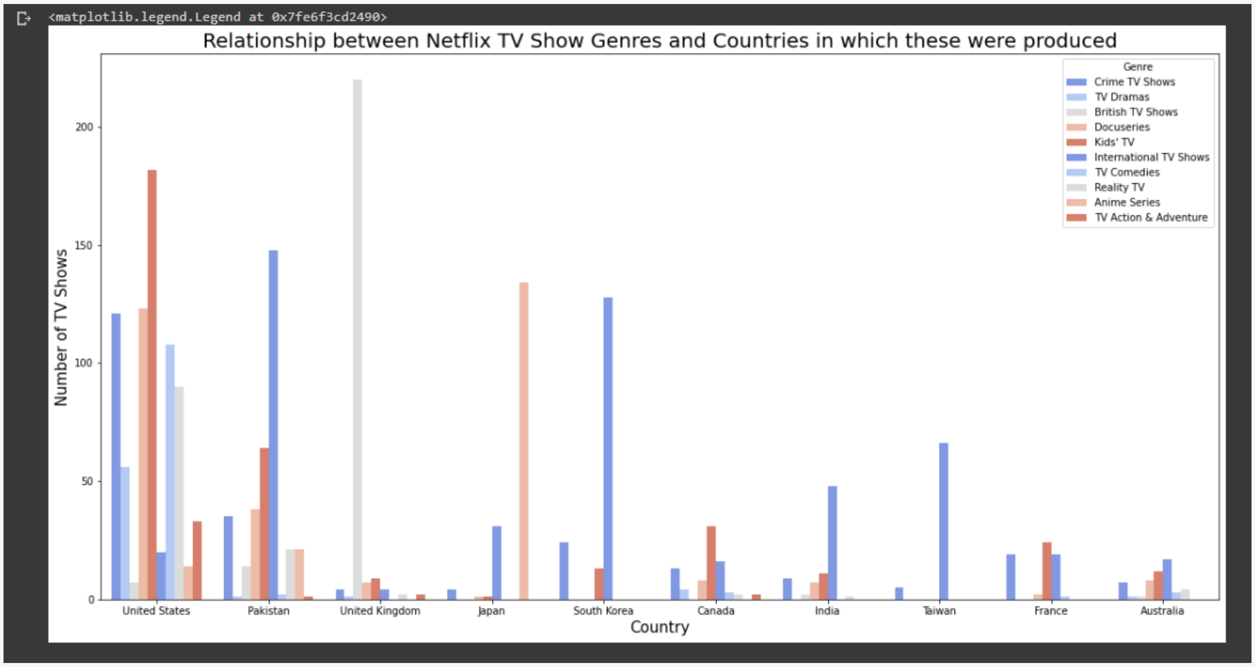


7.2.1 Comments on the number of Netflix's TV Shows produced by Country.

It is not unexpected that most of the TV shows that are brought to the streaming service were produced in the USA. However, the country with the second most production would be Pakistan - which one would normally expect the second place to belong to another english speaking country or western.

```
plt.figure(figsize=[20,10])

order1 = df_tv.groupby('listed_in1').count().sort_values(by =
'show_id',ascending=False)[0:10].index

df_tv_f = df_tv[df_tv['listed_in1'].isin(order1)]

order2 = df_tv_f.groupby('country').count().sort_values(by =
'show_id',ascending=False)[0:10].index

df_tv_f = df_tv_f[df_tv_f['country'].isin(order2)]
```

```
base_color = sns.color_palette('coolwarm',n_colors=5)

a=df_tv.date_added.dt.year

tv_g = sns.countplot(data=df_tv_f,x='country',hue='listed_in1',
palette=base_color, order=order2)

tv_g.set_xlabel('Country',fontsize = 15)

tv_g.set_ylabel('Number of TV Shows',fontsize = 15)

tv_g.set_title("Relationship between Netflix TV Show Genres and Countries
in which these were produced",fontsize = 20)

plt.legend(title = 'Genre', loc = 'upper right')
```



```
<matplotlib.legend.Legend at 0x7fe6f3cd2490>
```

7.2.2 Comments on the relationship vetween the top 10 TV Show Genres on Netflix and countries in which these were produced

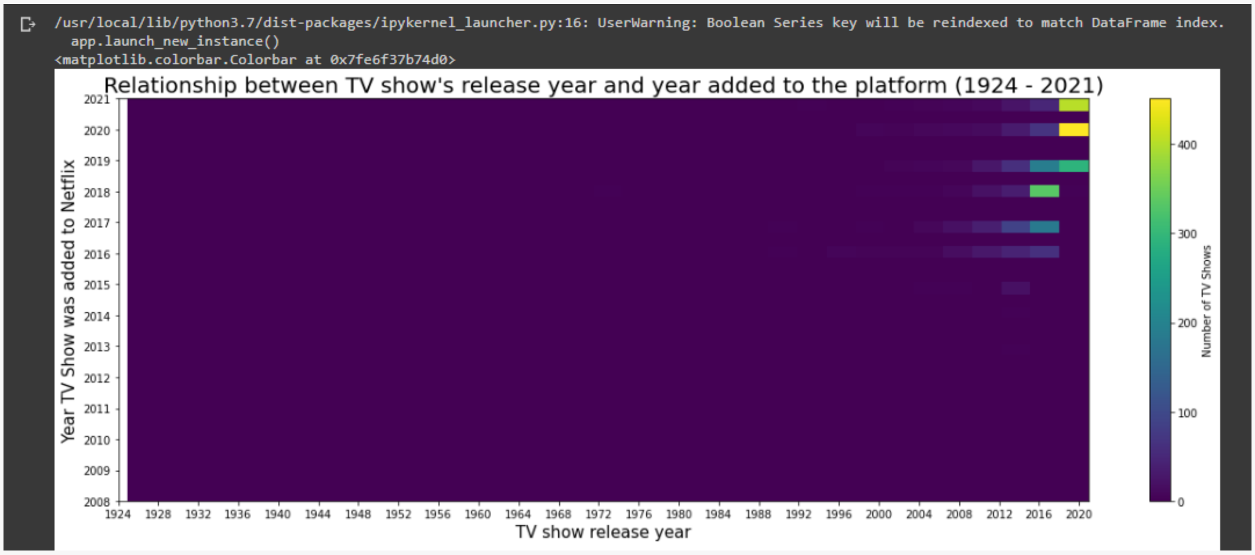
The most popular TV genres overall seem to be Kid’s TV and International TV Shows. Considering that Netflix is an American company, it makes sense that shows produced outside of the US are considered ‘International TV Shows.

US has a big diversity of shows produced but most of them were TV Action & Adventure, followed by Docuseries and Crime TV Shows.

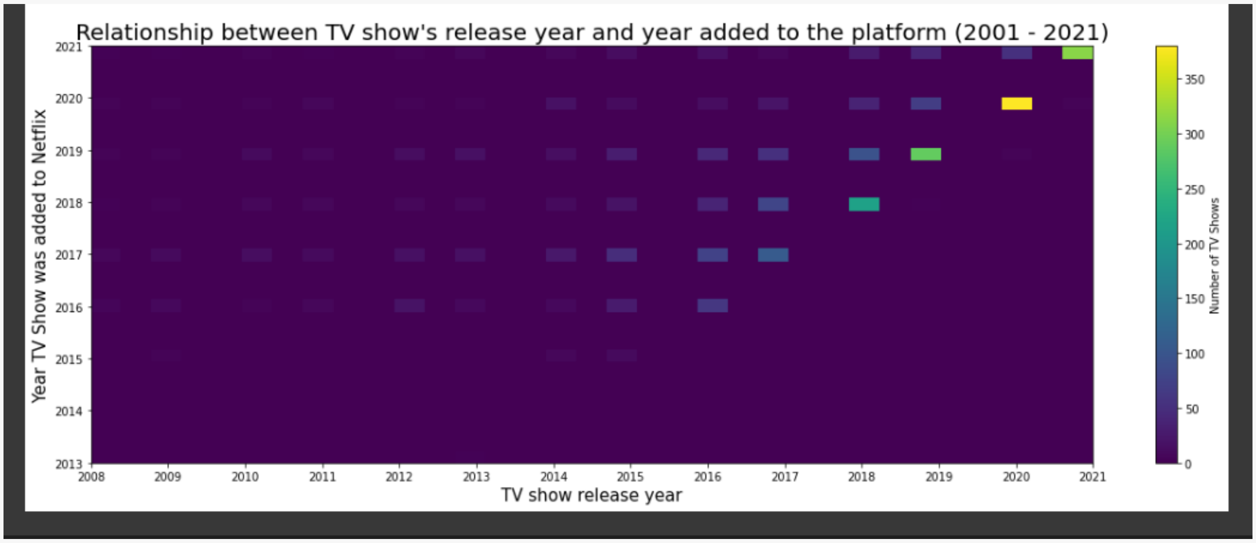
For Pakistan and South Korea, both the most produced genres were Kid’s TV and International TV Shows.

However, the UK and Japan have the biggest production of Reality TV shows and Anime series respectively.

```
#In this section we would see the relationship between Netflix adding TV show to their catalog and their respective release date

plt.figure(figsize=[20,15])

bins=np.arange(1924,2025,4)

plt.subplot(2,1,1)

tv_rd = plt.hist2d(data=df_tv,x='release_year',y=df_tv.date_added.dt.year, bins=33)

plt.xticks(np.arange(1924,2022,4));

plt.yticks(np.arange(2008,2022,1));

plt.xlabel('TV show release year',fontsize = 15)

plt.ylabel('Year TV Show was added to Netflix',fontsize = 15)
plt.title("Relationship between TV show's release year and year added to the platform (1924 - 2021)",fontsize = 20)
plt.colorbar(label = 'Number of TV Shows')


plt.subplot(2,1,2)
ry_f = df_tv.release_year>2007
da_f = df_tv.date_added.dt.year>2008
df_tv_f = df_tv[ry_f][da_f]
tv_rd1 = plt.hist2d(data=df_tv_f,x='release_year',y=df_tv_f.date_added.dt.year, bins=33)
plt.xticks(np.arange(2008,2022,1));
plt.yticks(np.arange(2013,2022,1));
plt.xlabel('TV show release year',fontsize = 15)
plt.ylabel('Year TV Show was added to Netflix',fontsize = 15)
plt.title("Relationship between TV show's release year and year added to the platform (2001 - 2021)",fontsize = 20)
plt.colorbar(label = 'Number of TV Shows')
```



```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  app.launch_new_instance()
<matplotlib.colorbar.Colorbar at 0x7fe6f37b74d0>
```

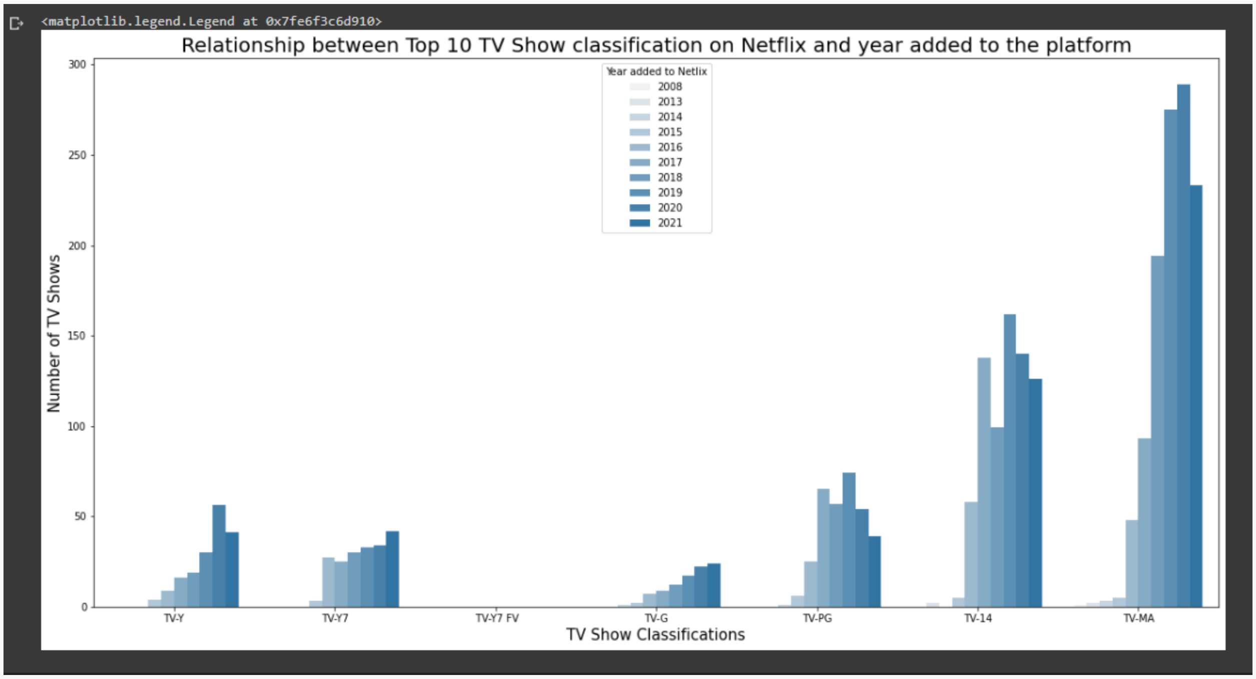


7.2.3 Comments on the relationship between TV shows release year and year added to Netflix

Initially, Netflix would not bring recently produced TV shows into the service. It was up until 2016, in which the service started producing their own TV shows and bringing recently filmed TV shows into their streaming service.

```
plt.figure(figsize=[20,10])

order = ['TV-Y', 'TV-Y7', ' TV-Y7 FV', 'TV-G', 'TV-PG', 'TV-14', 'TV-MA']
```

```
base_color = base_color = sns.color_palette()[0]

a=df_tv.date_added.dt.year

tv_g = sns.countplot(data=df_tv,x='rating',hue=a, order=order,
color=base_color)

tv_g.set_xlabel('TV Show Classifications',fontsize = 15)

tv_g.set_ylabel('Number of TV Shows',fontsize = 15)

tv_g.set_title("Relationship between Top 10 TV Show classification on
Netflix and year added to the platform",fontsize = 20)

plt.legend(title = 'Year added to Netlix',)
```



```
<matplotlib.legend.Legend at 0x7fe6f3c6d910>
```

7.2.4 Comments on the Relationship between Netflix TV Shows classification and year added to the platform.

Most of Netflix's TV shows are categorised as TV-MA (primarily) and TV-14 (secondarily).

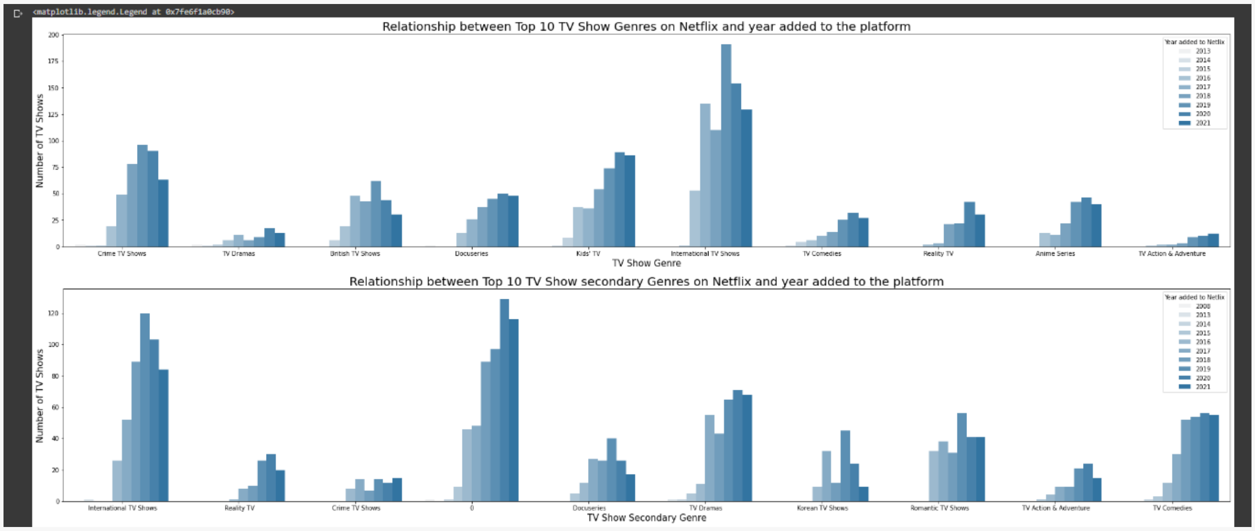
TV-MA had been steadily increasing up until 2021; whilst in the other hand TV-14 has had a sporadic growth with a decrease starting from 2020 - This possible due to TV Shows pushing their classification to a more mature audience (TV-MA).

The only shows that had a steady increase with no drops whatsoever were the ones classified as TV-G and TV-Y7.

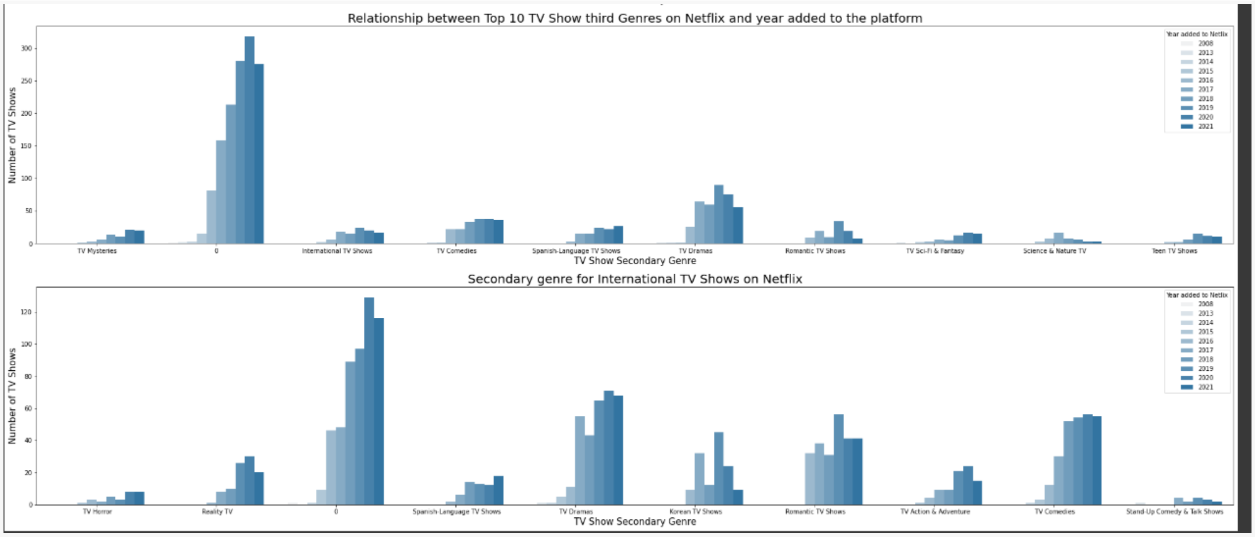
```
plt.figure(figsize=[35,30])

plt.subplot(4,1,1)



base_color = base_color = sns.color_palette()[0]

sort_order = df_tv.groupby('listed_in1').count().sort_values(by = 'show_id',ascending=False)[0:10].index

df_tv_g = df_tv[df_tv['listed_in1'].isin(sort_order)]

tv_g = sns.countplot(data=df_tv_g,x='listed_in1',hue=df_tv_g.date_added.dt.year, color=base_color)

tv_g.set_xlabel('TV Show Genre',fontsize = 15)

tv_g.set_ylabel('Number of TV Shows',fontsize = 15)

tv_g.set_title("Relationship between Top 10 TV Show Genres on Netflix and year added to the platform",fontsize = 20)

plt.legend(title = 'Year added to Netlix')
plt.subplot(4,1,2)


base_color = base_color = sns.color_palette()[0]

sort_order = df_tv.groupby('listed_in2').count().sort_values(by =
'show_id',ascending=False)[0:10].index

df_tv_g = df_tv[df_tv['listed_in2'].isin(sort_order)]

tv_g1 =
sns.countplot(data=df_tv_g,x='listed_in2',hue=df_tv_g.date_added.dt.year,
color=base_color)

tv_g1.set_xlabel('TV Show Secondary Genre',fontsize = 15)

tv_g1.set_ylabel('Number of TV Shows',fontsize = 15)

tv_g1.set_title("Relationship between Top 10 TV Show secondary Genres on
Netflix and year added to the platform",fontsize = 20)

plt.legend(title = 'Year added to Netlix')


plt.subplot(4,1,3)


base_color = base_color = sns.color_palette()[0]
```

```
sort_order = df_tv.groupby('listed_in3').count().sort_values(by =
'show_id',ascending=False)[0:10].index

df_tv_g = df_tv[df_tv['listed_in3'].isin(sort_order)]

tv_g3 =
sns.countplot(data=df_tv_g,x='listed_in3',hue=df_tv_g.date_added.dt.year,
color=base_color)

tv_g3.set_xlabel('TV Show Secondary Genre',fontsize = 15)

tv_g3.set_ylabel('Number of TV Shows',fontsize = 15)

tv_g3.set_title("Relationship between Top 10 TV Show third Genres on
Netflix and year added to the platform",fontsize = 20)

plt.legend(title = 'Year added to Netlix', loc='upper right')



plt.subplot(4,1,4)

df_tv_g1 = df_tv.listed_in1=='International TV Shows'

df_tv_g1 = df_tv[df_tv_g1]

base_color = base_color = sns.color_palette()[0]

sort_order = df_tv_g1.groupby('listed_in2').count().sort_values(by =
'show_id',ascending=False)[0:10].index

df_tv_g1 = df_tv[df_tv['listed_in2'].isin(sort_order)]
```

```
tv_g4 =
sns.countplot(data=df_tv_g1,x='listed_in2',hue=df_tv_g1.date_added.dt.year
, color=base_color)

tv_g4.set_xlabel('TV Show Secondary Genre',fontsize = 15)

tv_g4.set_ylabel('Number of TV Shows',fontsize = 15)

tv_g4.set_title("Secondary genre for International TV Shows on
Netflix",fontsize = 20)

plt.legend(title = 'Year added to Netlix', loc='upper right')
```



```
<matplotlib.legend.Legend at 0x7fe6f1a0cb90>
```



7.2.5 Comments on the relationship between Netflix's TV Shows genres and the year added to the platform.

For this analysis an assumption had to be made, the first category that appears on a TV show would be classified as the 'main' category, followed by the secondary and the third.

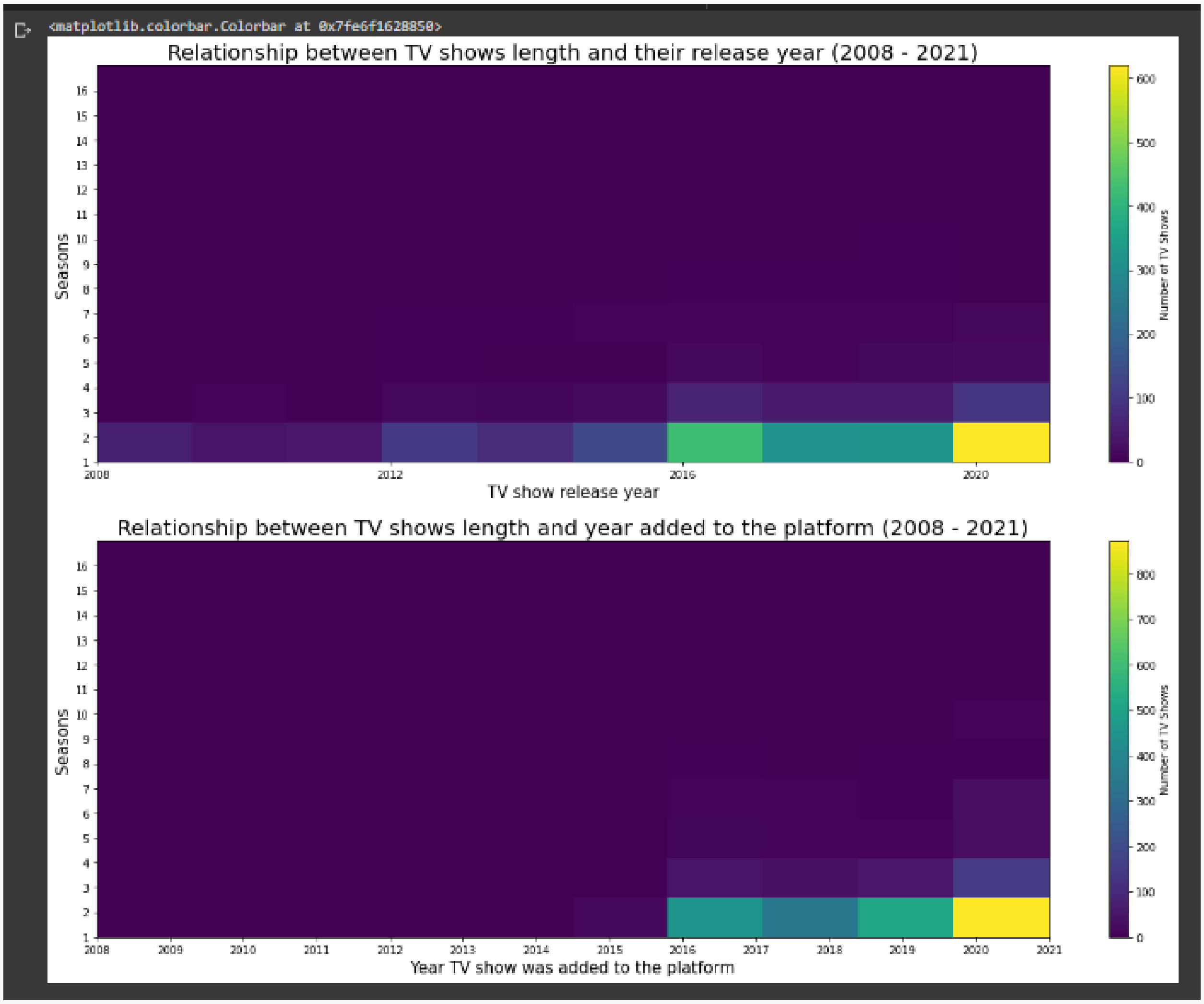
Most of the TV shows produced were International TV Shows followed by Crime shows and Kids TV.

For secondary genres, most of TV shows do not have a secondary genre . But the rest were mainly classified as International TV Shows, TV dramas and TV comedies.

For third genres, Most of TV shows do not have a third genre. But the rest were mainly classified as TV Dramas, TV comedies and Romantic TV shows.

Since International TV Shows is a broad genre, we also performed an analysis on the secondary genres for International TV Shows. The result was that most of them do not have a secondary genre. But the rest were mainly classified as TV Dramas, Romantic TV shows and TV comedies.

```
#In this section we would see the relationship between Netflix adding TV show to their catalog and their respective release date

plt.figure(figsize=[20,15])

plt.subplot(2,1,1)

da_f = df_tv.release_year>2007

df_tv_f = df_tv[da_f]

tv_rd = plt.hist2d(data=df_tv_f,x='release_year',y='duration_seasons')

plt.xticks(np.arange(2008,2022,4));
plt.yticks(np.arange(1,17,1));
plt.xlabel('TV show release year',fontsize = 15)
plt.ylabel('Seasons',fontsize = 15)
plt.title("Relationship between TV shows length and their release year (2008 - 2021)",fontsize = 20)
plt.colorbar(label = 'Number of TV Shows')


plt.subplot(2,1,2)
da_f = df_tv.date_added.dt.year>2007
df_tv_f = df_tv[da_f]
tv_rd1 = plt.hist2d(data=df_tv_f,x=df_tv_f.date_added.dt.year,y='duration_seasons')
plt.xticks(np.arange(2008,2022,1));
plt.yticks(np.arange(1,17,1));
plt.xlabel('Year TV show was added to the platform',fontsize = 15)
plt.ylabel('Seasons',fontsize = 15)
plt.title("Relationship between TV shows length and year added to the platform (2008 - 2021)",fontsize = 20)
plt.colorbar(label = 'Number of TV Shows')
```



```
<matplotlib.colorbar.Colorbar at 0x7fe6f1628850>
```

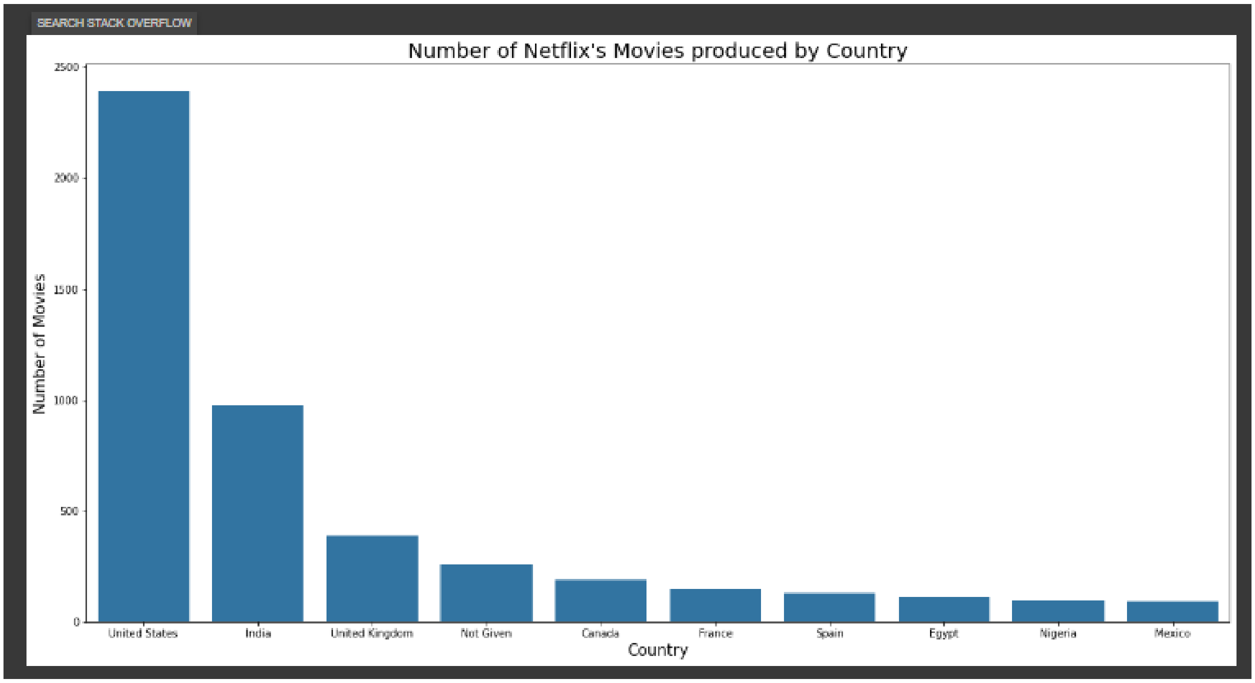
7.2.6 Comments on the relationship between TV Shows length and the release year / year added to the platform

For both release year and year added to the platform, it seems to be that most of the TV shows just last for a single season.

7.3 Movies

```
# Considering there are too many countries, we will limit our study to just the top 10 countries.
sort_order = df_movie.groupby('country').count().sort_values(by = 'show_id',ascending=False)[0:10].index
df_movie_c = df_movie[df_movie['country'].isin(sort_order)]
base_color = base_color = sns.color_palette()[0]
```

```
plt.figure(figsize=[20,10])
movie_c = sns.countplot(x='country',data=df_movie_c,order=sort_order,
color = base_color)
movie_c.set_title("Number of Netflix's Movies produced by
Country",fontsize = 20)
movie_c.set_xlabel('Country',fontsize = 15)
movie_c.set_ylabel('Number of Movies',fontsize = 15)
for container in movie_c.containers:
    movie_c.bar_label(container)
```



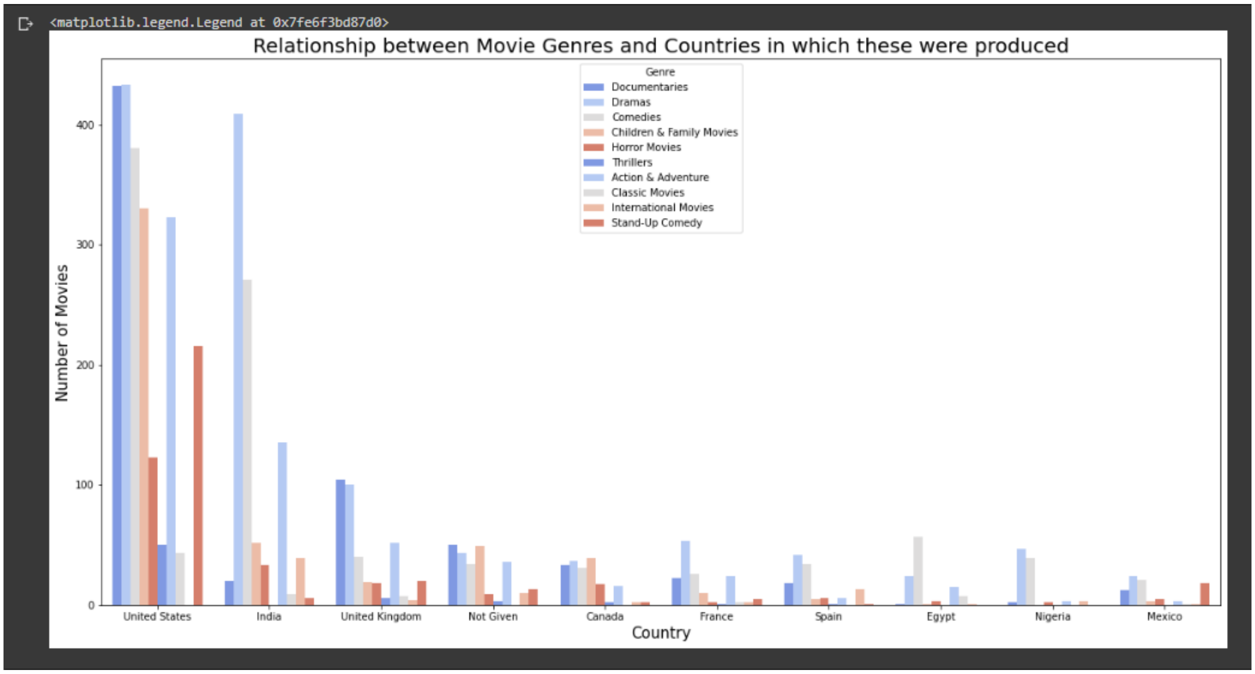
7.3.1 Comments on Netflix's Movies produced by country

Similarly to Netflix TV shows, most of the movies were also produced in the United States. However the second most popular country would be India, which it would be likely due to its big movie industry ('Bollywood').

Surprisingly, despite Pakistan producing most of the TV shows for Netflix; it is not considered between the top 10 countries that has produced most movies for Netflix.

```
plt.figure(figsize=[20,10])
```

```
order1 = df_movie.groupby('listed_in1').count().sort_values(by = 'show_id',ascending=False)[0:10].index

df_movie_f = df_movie[df_movie['listed_in1'].isin(order1)]

order2 = df_movie_f.groupby('country').count().sort_values(by = 'show_id',ascending=False)[0:10].index

df_movie_f = df_movie_f[df_movie_f['country'].isin(order2)]

base_color = sns.color_palette('coolwarm',n_colors=5)

a=df_movie.date_added.dt.year

tv_g = sns.countplot(data=df_movie_f,x='country',hue='listed_in1', palette=base_color, order=order2)

tv_g.set_xlabel('Country',fontsize = 15)

tv_g.set_ylabel('Number of Movies',fontsize = 15)

tv_g.set_title("Relationship between Movie Genres and Countries in which these were produced",fontsize = 20)

plt.legend(title = 'Genre',)
```



```
<matplotlib.legend.Legend at 0x7fe6f3bd87d0>
```

7.3.2 Comments on Netflix's Movies produced by country

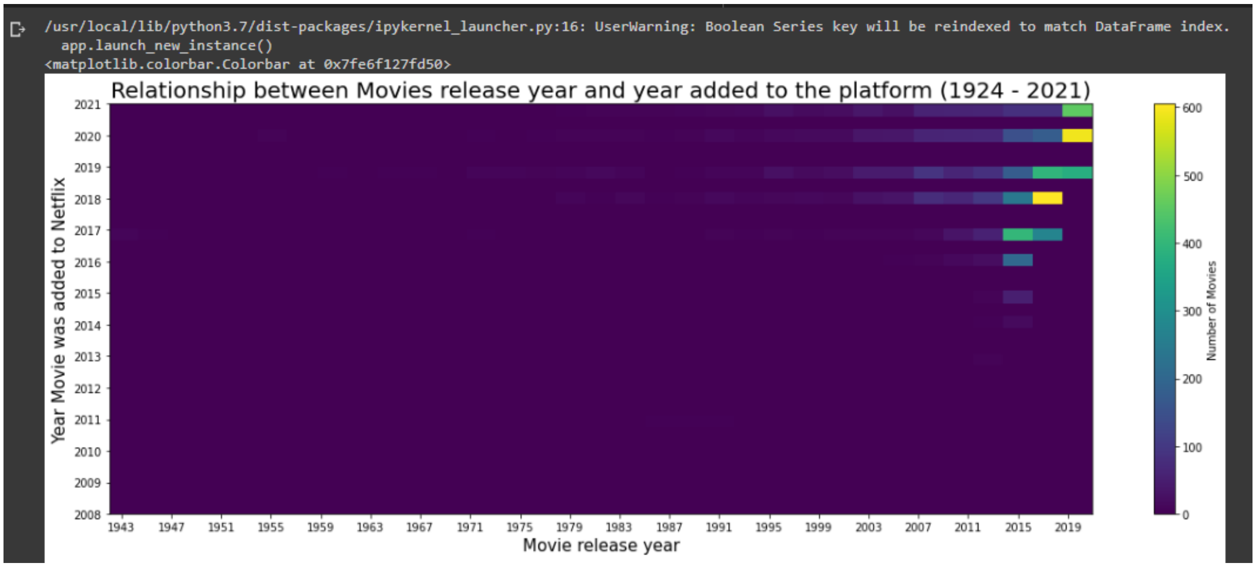
Most of the media produced in the US were documentaries, dramas and comedies. Whilst for India, it was mainly dramas followed by comedies.

US and the UK have one of the biggest ratio in movies produced against documentaries.

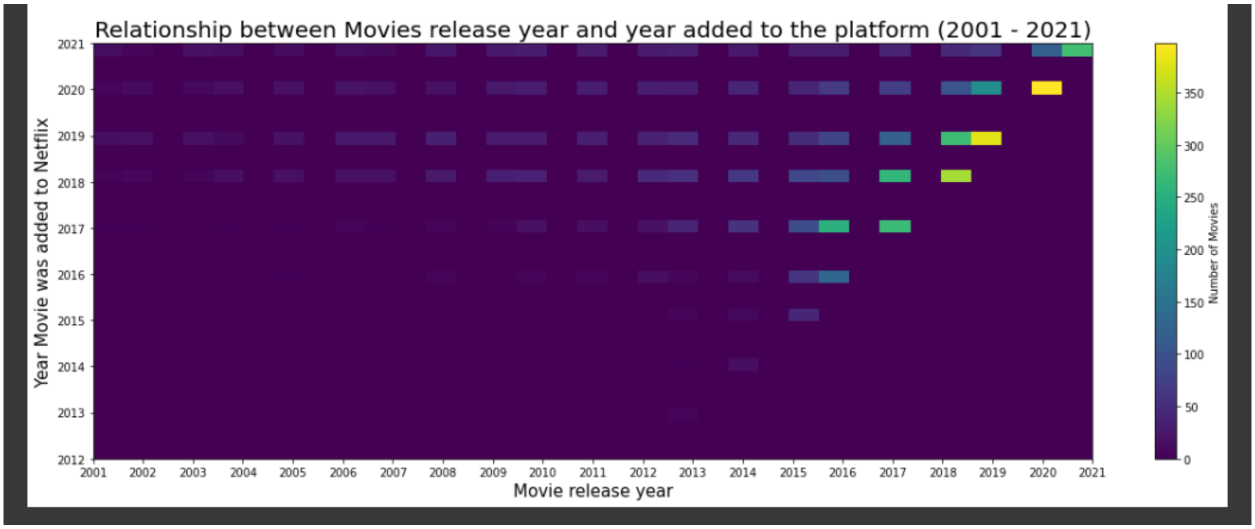
Furthermore, it seems that Stand-Up Comedy is also a big genre in the US; when compared to other countries.

```
#In this section we would see the relationship between Netflix adding Movie to their catalog and their respective release date

plt.figure(figsize=[20,15])

bins=np.arange(1943,2025,4)

plt.subplot(2,1,1)
```

```
movie_rd = plt.hist2d(data=df_movie,x='release_year',y=df_movie.date_added.dt.year, bins=33)
plt.xticks(np.arange(1943,2022,4));
plt.yticks(np.arange(2008,2022,1));
plt.xlabel('Movie release year',fontsize = 15)
plt.ylabel('Year Movie was added to Netflix',fontsize = 15)
plt.title("Relationship between Movies release year and year added to the platform (1924 - 2021)",fontsize = 20)
plt.colorbar(label = 'Number of Movies')

plt.subplot(2,1,2)
ry_f = df_movie.release_year>2000
da_f = df_movie.date_added.dt.year>2011
df_movie_f = df_movie[ry_f][da_f]
movie_rd1 = plt.hist2d(data=df_movie_f,x='release_year',y=df_movie_f.date_added.dt.year, bins=33)
plt.xticks(np.arange(2001,2022,1));
plt.yticks(np.arange(2012,2022,1));
plt.xlabel('Movie release year',fontsize = 15)

plt.ylabel('Year Movie was added to Netflix',fontsize = 15)

plt.title("Relationship between Movies release year and year added to the platform (2001 - 2021)",fontsize = 20)

plt.colorbar(label = 'Number of Movies')
```



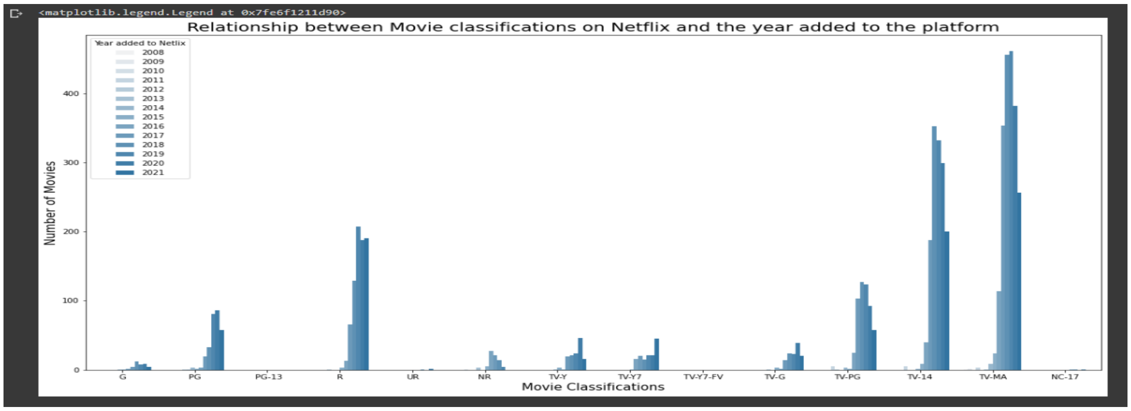
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  app.launch_new_instance()
<matplotlib.colorbar.Colorbar at 0x7fe6f127fd50>
```



7.3.3 Comments on the relationship between movies release year and year added to the platform

Starting from 2015, there seems to be a clear increase in the number of movies released in the same year as they were added to the platform. There is also a slight decrease in 2021 but this would have been likely due to COVID - since not many movies were made in 2020 and released in 2021.

```
plt.figure(figsize=[20,10])

order = ['G', 'PG', ' PG-13', 'R', 'UR', 'NR', 'TV-Y', 'TV-Y7', '
TV-Y7-FV', 'TV-G', 'TV-PG', 'TV-14', 'TV-MA', 'NC-17']

base_color = base_color = sns.color_palette()[0]

a=df_movie.date_added.dt.year

movie_g = sns.countplot(data=df_movie,x='rating',hue=a,order=order,
color=base_color)

movie_g.set_xlabel('Movie Classifications',fontsize = 15)

movie_g.set_ylabel('Number of Movies',fontsize = 15)

movie_g.set_title("Relationship between Movie classifications on Netflix
and the year added to the platform",fontsize = 20)

plt.legend(title = 'Year added to Netlix',)
```



```
<matplotlib.legend.Legend at 0x7fe6f1211d90>
```

7.3.4 Comments on the movie classifications and the year these were added to the platform.

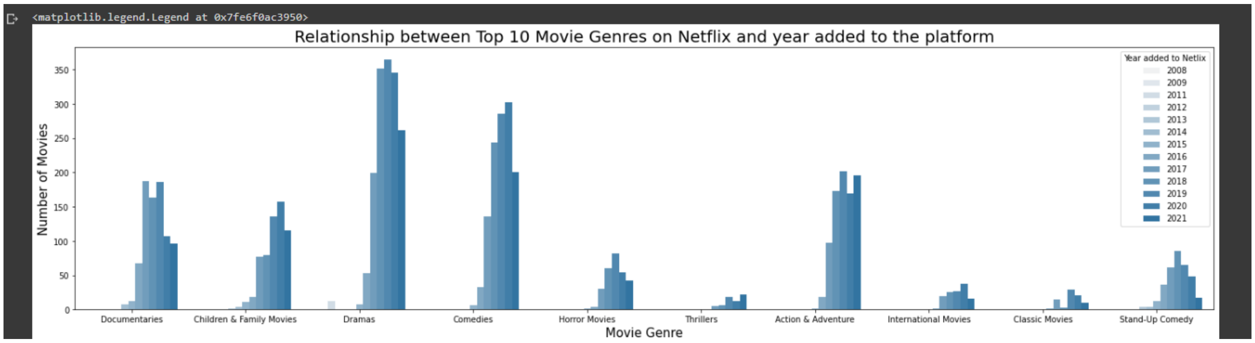
Movies had both movie classification and TV classification meaning that some of the movies that are part of Netflix were movies made for straight to TV.

There seems to be that most of the movies available on Netflix would be for mature aged people (18+, R and TV-MA).

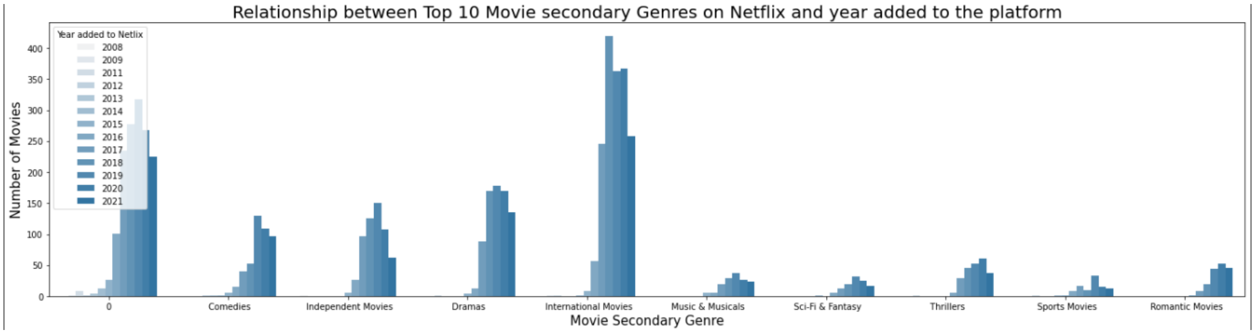
It seems that Netflix did not want to bring teen movies released in theatres (PG-13) to its platform but instead bring R-rated movies. This strategy is different to the straight to TV movies, since the second most popular classification would be TV-14.

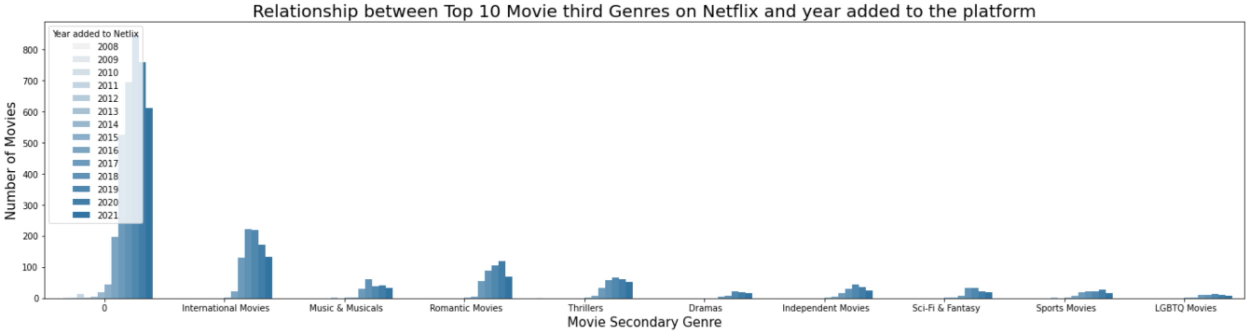
```
plt.figure(figsize=[25,20])

plt.subplot(3,1,1)



base_color = base_color = sns.color_palette()[0]

sort_order = df_movie.groupby('listed_in1').count().sort_values(by = 'show_id',ascending=False)[0:10].index

df_movie_g = df_movie[df_movie['listed_in1'].isin(sort_order)]

movie_g = sns.countplot(data=df_movie_g,x='listed_in1',hue=df_movie_g.date_added.dt.year, color=base_color)

movie_g.set_xlabel('Movie Genre',fontsize = 15)

movie_g.set_ylabel('Number of Movies',fontsize = 15)
movie_g.set_title("Relationship between Top 10 Movie Genres on Netflix and year added to the platform",fontsize = 20)

plt.legend(title = 'Year added to Netlix')



plt.subplot(3,1,2)



base_color = base_color = sns.color_palette()[0]

sort_order = df_movie.groupby('listed_in2').count().sort_values(by = 'show_id',ascending=False)[0:10].index

df_movie_g = df_movie[df_movie['listed_in2'].isin(sort_order)]

movie_g1 = sns.countplot(data=df_movie_g,x='listed_in2',hue=df_movie_g.date_added.dt.year, color=base_color)

movie_g1.set_xlabel('Movie Secondary Genre',fontsize = 15)

movie_g1.set_ylabel('Number of Movies',fontsize = 15)

movie_g1.set_title("Relationship between Top 10 Movie secondary Genres on Netflix and year added to the platform",fontsize = 20)

plt.legend(title = 'Year added to Netlix')



plt.subplot(3,1,3)
```

```
base_color = base_color = sns.color_palette()[0]

sort_order = df_movie.groupby('listed_in3').count().sort_values(by =
'show_id',ascending=False)[0:10].index

df_movie_g = df_movie[df_movie['listed_in3'].isin(sort_order)]

movie_g3 =
sns.countplot(data=df_movie_g,x='listed_in3',hue=df_movie_g.date_added.dt.
year, color=base_color)

movie_g3.set_xlabel('Movie Secondary Genre',fontsize = 15)

movie_g3.set_ylabel('Number of Movies',fontsize = 15)

movie_g3.set_title("Relationship between Top 10 Movie third Genres on
Netflix and year added to the platform",fontsize = 20)

plt.legend(title = 'Year added to Netlix')
```



```
<matplotlib.legend.Legend at 0x7fe6f0ac3950>
```





7.3.5 Comments on the movie genres and year added to the platform.

For this analysis an assumption had to be made, the first category/genre that appears on a Movie would be classified as the 'main' category, followed by the secondary and the third.

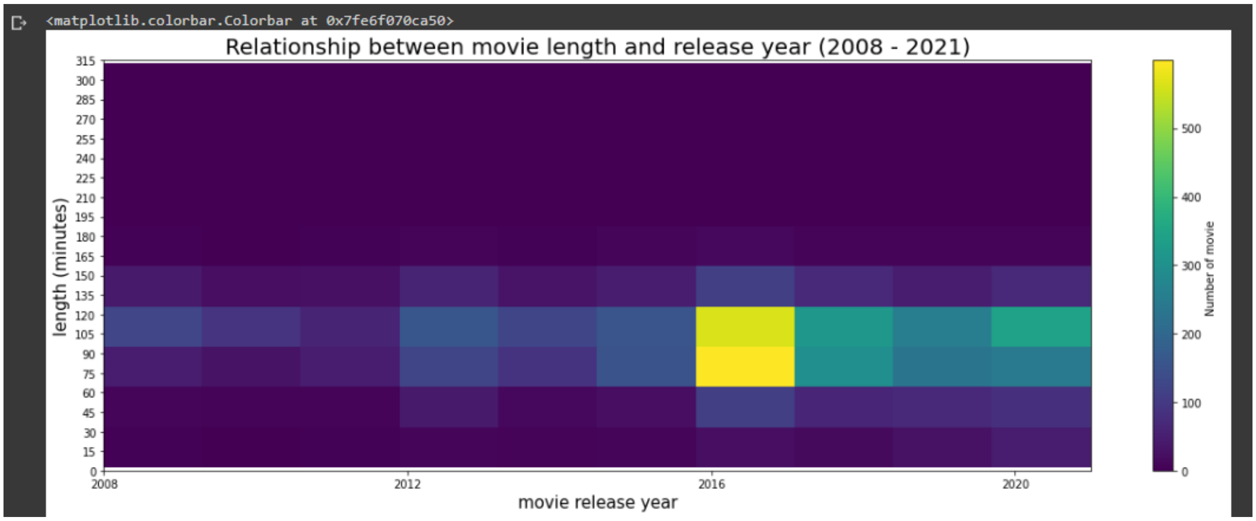
Most of the movies produced were Dramas, Comedies and Action & Adventure.

For the secondary genres, most of the TV shows are classified as International Movies. But the rest were mainly classified as None, dramas and independent movies.

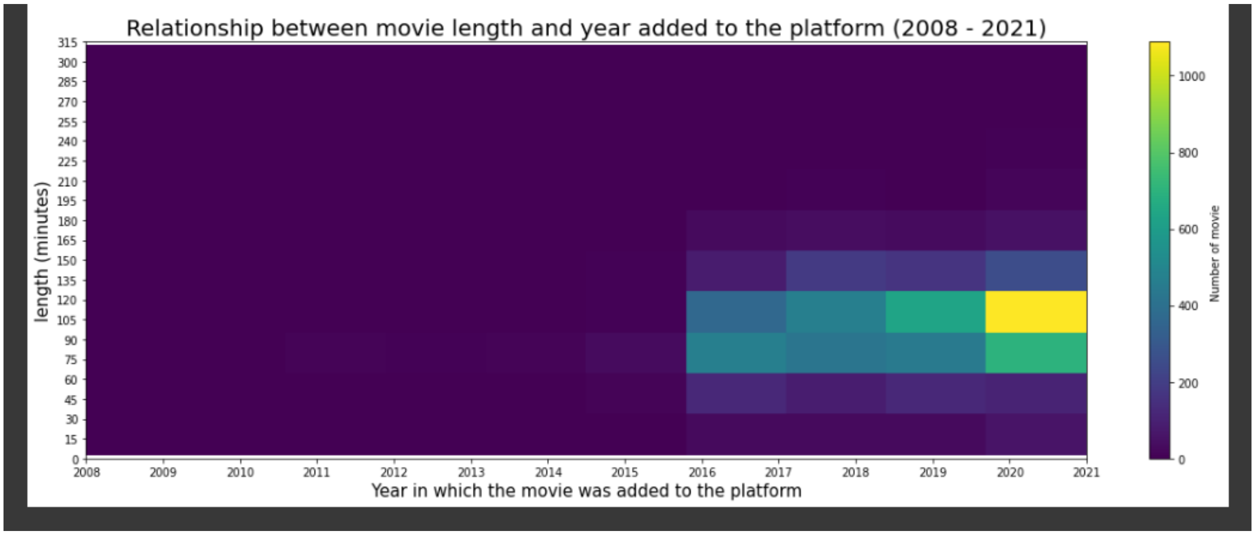
For the third genre, most TV shows do not have a third genre. But the rest were mainly classified as international movies.

```
#In this section we would see the relationship between Netflix adding movie to their catalog and their respective release date

plt.figure(figsize=[20,15])

plt.subplot(2,1,1)
```

```
da_f = df_movie.release_year>2007
df_movie_f = df_movie[da_f]
movie_rd = plt.hist2d(data=df_movie_f,x='release_year',y='duration_minutes')
plt.xticks(np.arange(2008,2022,4));
plt.yticks(np.arange(0,316,15));
plt.xlabel('movie release year',fontsize = 15)
plt.ylabel('length (minutes)',fontsize = 15)
plt.title("Relationship between movie length and release year (2008 - 2021)",fontsize = 20)
plt.colorbar(label = 'Number of movie ')


plt.subplot(2,1,2)
da_f = df_movie.date_added.dt.year>2007
df_movie_f = df_movie[da_f]
movie_rd1 = plt.hist2d(data=df_movie_f,x=df_movie_f.date_added.dt.year,y='duration_minutes')
plt.xticks(np.arange(2008,2022,1));
plt.yticks(np.arange(0,316,15));
plt.xlabel('Year in which the movie was added to the platform',fontsize = 15)

plt.ylabel('length (minutes)',fontsize = 15)

plt.title("Relationship between movie length and year added to the platform (2008 - 2021)",fontsize = 20)

plt.colorbar(label = 'Number of movie ')
```



```
<matplotlib.colorbar.Colorbar at 0x7fe6f070ca50>
```



7.3.6 Comments on the relationship between movie length and release year/year it was added to the platform.

For movies release year and length, most of the movie's length range between 70 and 120min, however overtime the length of the movies has decreased.

For movies length and year it was added to the platform, it follows a similar trend in which most of the movies last between 70 and 120min. However, it also seem like most of the movies seems to be gradually increasing in length.

In this way, data preprocessing can be done along with the analysis of the data.