Netflix Case study

October 2, 2023

[40]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

1 1. Importing Libraries , Loading the data and Basic Observations

[7]:
```
df = pd.read_csv("/Users/senth/Desktop/netflix.csv")
df
```

[7]:
```
     show_id     type                  title         director  \
0         s1    Movie   Dick Johnson Is Dead  Kirsten Johnson
1         s2  TV Show          Blood & Water              NaN
2         s3  TV Show              Ganglands  Julien Leclercq
3         s4  TV Show  Jailbirds New Orleans              NaN
4         s5  TV Show           Kota Factory              NaN
…      …      …                    …              …
8802   s8803    Movie                 Zodiac    David Fincher
8803   s8804  TV Show            Zombie Dumb              NaN
8804   s8805    Movie             Zombieland  Ruben Fleischer
8805   s8806    Movie                   Zoom     Peter Hewitt
8806   s8807    Movie                 Zubaan      Mozez Singh

                                                   cast        country  \
0                                                   NaN  United States
1     Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…   South Africa
2     Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…            NaN
3                                                   NaN            NaN
4     Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…          India
…                                                 …            …
8802  Mark Ruffalo, Jake Gyllenhaal, Robert Downey J…  United States
8803                                                NaN            NaN
8804  Jesse Eisenberg, Woody Harrelson, Emma Stone, …  United States
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma…  United States
```

```
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan…          India

              date_added  release_year rating   duration  \
0     September 25, 2021          2020  PG-13     90 min
1     September 24, 2021          2021  TV-MA  2 Seasons
2     September 24, 2021          2021  TV-MA   1 Season
3     September 24, 2021          2021  TV-MA   1 Season
4     September 24, 2021          2021  TV-MA  2 Seasons
…                  …           …    …        …
8802   November 20, 2019          2007      R    158 min
8803        July 1, 2019          2018  TV-Y7  2 Seasons
8804    November 1, 2019          2009      R     88 min
8805    January 11, 2020          2006     PG     88 min
8806       March 2, 2019          2015  TV-14    111 min

                                              listed_in  \
0                                         Documentaries
1       International TV Shows, TV Dramas, TV Mysteries
2     Crime TV Shows, International TV Shows, TV Act…
3                                Docuseries, Reality TV
4     International TV Shows, Romantic TV Shows, TV …
…                                                 …
8802                     Cult Movies, Dramas, Thrillers
8803             Kids' TV, Korean TV Shows, TV Comedies
8804                            Comedies, Horror Movies
8805                 Children & Family Movies, Comedies
8806     Dramas, International Movies, Music & Musicals

                                            description
0     As her father nears the end of his life, filmm…
1     After crossing paths at a party, a Cape Town t…
2     To protect his family from a powerful drug lor…
3     Feuds, flirtations and toilet talk go down amo…
4     In a city of coaching centers known to train I…
…                                                 …
8802  A political cartoonist, a crime reporter and a…
8803  While living alone in a spooky town, a young g…
8804  Looking to survive in a world taken over by zo…
8805  Dragged from civilian life, a former superhero…
8806  A scrappy but poor boy worms his way into a ty…

[8807 rows x 12 columns]
```

```
[21]: df.shape
```

```
[21]: (8807, 12)
```

[ ]:

```
### Now lets see the information of our data
```

[14]:

```
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      8807 non-null   object
 4   cast          8807 non-null   object
 5   country       8807 non-null   object
 6   date_added    8807 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8807 non-null   object
 9   duration      8807 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
```

[15]:

```
df.describe()
```

[15]:

```
       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
```

[16]:

```
df.head()
```

[16]:

```
  show_id     type                 title         director  \
0      s1    Movie  Dick Johnson Is Dead  Kirsten Johnson
1      s2  TV Show         Blood & Water  No Data Availabe
2      s3  TV Show             Ganglands   Julien Leclercq
3      s4  TV Show Jailbirds New Orleans  No Data Availabe
4      s5  TV Show          Kota Factory  No Data Availabe

                                                 cast         country  \
0                                   No Data Available   United States
1  Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…    South Africa
```

```
2  Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…  No Data Available
3                                  No Data Available  No Data Available
4  Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…              India

           date_added  release_year  rating  duration   \
0  September 25, 2021          2020  PG-13     90 min
1  September 24, 2021          2021  TV-MA  2 Seasons
2  September 24, 2021          2021  TV-MA   1 Season
3  September 24, 2021          2021  TV-MA   1 Season
4  September 24, 2021          2021  TV-MA  2 Seasons

                                           listed_in  \
0                                      Documentaries
1    International TV Shows, TV Dramas, TV Mysteries
2  Crime TV Shows, International TV Shows, TV Act…
3                             Docuseries, Reality TV
4  International TV Shows, Romantic TV Shows, TV …

                                         description
0  As her father nears the end of his life, filmm…
1  After crossing paths at a party, a Cape Town t…
2  To protect his family from a powerful drug lor…
3  Feuds, flirtations and toilet talk go down amo…
4  In a city of coaching centers known to train I…
```

[17]:
```python
df.tail()
```

[17]:
```
     show_id     type        title          director  \
8802   s8803    Movie       Zodiac     David Fincher
8803   s8804  TV Show  Zombie Dumb  No Data Availabe
8804   s8805    Movie   Zombieland   Ruben Fleischer
8805   s8806    Movie         Zoom      Peter Hewitt
8806   s8807    Movie       Zubaan       Mozez Singh

                                                   cast            country   \
8802  Mark Ruffalo, Jake Gyllenhaal, Robert Downey J…      United States
8803                                  No Data Available  No Data Available
8804  Jesse Eisenberg, Woody Harrelson, Emma Stone, …      United States
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma…      United States
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan…              India

             date_added  release_year rating   duration   \
8802  November 20, 2019          2007      R    158 min
8803       July 1, 2019          2018  TV-Y7  2 Seasons
8804   November 1, 2019          2009      R     88 min
8805   January 11, 2020          2006     PG     88 min
8806      March 2, 2019          2015  TV-14    111 min
```

```
                                           listed_in  \
8802                  Cult Movies, Dramas, Thrillers
8803          Kids' TV, Korean TV Shows, TV Comedies
8804                         Comedies, Horror Movies
8805              Children & Family Movies, Comedies
8806  Dramas, International Movies, Music & Musicals

                                            description
8802  A political cartoonist, a crime reporter and a…
8803  While living alone in a spooky town, a young g…
8804  Looking to survive in a world taken over by zo…
8805  Dragged from civilian life, a former superhero…
8806  A scrappy but poor boy worms his way into a ty…
```

[18]:
```
df.nunique()
```

[18]:
```
show_id         8807
type               2
title           8807
director        4529
cast            7693
country          749
date_added      1767
release_year      74
rating            17
duration         221
listed_in        514
description     8775
dtype: int64
```

2 2.DATA CLEANING

[22]:
```
df.isna().sum()
```

[22]:
```
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
```

```
dtype: int64
```

[29]:
```
df.drop(df.loc[df['date_added'].isna()].index , axis = 0 , inplace = True)
```

[30]:
```
df['date_added'].value_counts()
```

[30]:
```
January 1, 2020      119
November 1, 2019      89
March 1, 2018         75
December 31, 2019     74
October 1, 2018       71
                    …
December 4, 2016       1
November 21, 2016      1
November 19, 2016      1
November 17, 2016      1
January 11, 2020       1
Name: date_added, Length: 1767, dtype: int64
```

[31]:
```
df['date_added'] = pd.to_datetime(df['date_added'])
df['date_added']
```

[31]:
```
0      2021-09-25
1      2021-09-24
2      2021-09-24
3      2021-09-24
4      2021-09-24
          …
8802   2019-11-20
8803   2019-07-01
8804   2019-11-01
8805   2020-01-11
8806   2019-03-02
Name: date_added, Length: 8807, dtype: datetime64[ns]
```

[32]:
```
# total null values in each column
df.isna().sum()
```

[32]:
```
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
dtype: int64
```

[34]:
```
round((df.isna().sum()/ df.shape[0])*100)
```

[34]:
```
show_id         0.0
type            0.0
title           0.0
director        0.0
cast            0.0
country         0.0
date_added      0.0
release_year    0.0
rating          0.0
duration        0.0
listed_in       0.0
description     0.0
dtype: float64
```

3 3.Data Exploration and Non Graphical Analysis

[35]:
```
# 2 types of content present in dataset - either Movie or TV Show
df['type'].unique()
```

[35]:
```
array(['Movie', 'TV Show'], dtype=object)
```

[36]:
```
movies  = df.loc[df['type'] == 'Movie']
tv_shows = df.loc[df['type'] == 'TV Show']
```

[37]:
```
movies.duration.value_counts()
```

[37]:
```
90 min     152
94 min     146
93 min     146
97 min     146
91 min     144
          …
16 min       1
8 min        1
9 min        1
208 min      1
191 min      1
Name: duration, Length: 206, dtype: int64
```

[38]:
```
tv_shows.duration.value_counts()
```

```
[38]: 1 Season      1793
      2 Seasons      425
      3 Seasons      199
      4 Seasons       95
      5 Seasons       65
      6 Seasons       33
      7 Seasons       23
      8 Seasons       17
      9 Seasons        9
      10 Seasons       7
      13 Seasons       3
      15 Seasons       2
      12 Seasons       2
      11 Seasons       2
      17 Seasons       1
      Name: duration, dtype: int64
```

```
[42]: timeperiod = pd.Series((df['date_added'].min().strftime('%B %Y') ,␣
       ↪df['date_added'].max().strftime('%B %Y')))
      timeperiod.index = ['first' , 'Most Recent']
      timeperiod
```

```
[42]: first              January 2008
      Most Recent      September 2021
      dtype: object
```

```
[43]: df.release_year.min() , df.release_year.max()
```

```
[43]: (1925, 2021)
```

```
[44]: df.loc[(df.release_year == df.release_year.min()) | (df.release_year == df.
       ↪release_year.max())].sort_values('release_year')
```

```
[44]:      show_id     type                                         title  \
      4250   s4251  TV Show            Pioneers: First Women Filmmakers*
      966     s967    Movie                                 Get the Grift
      967     s968  TV Show                  Headspace Guide to Sleep
      968     s969  TV Show                                        Sexify
      972     s973  TV Show                                         Fatma
      …        …       …                                        …
      466     s467  TV Show                          My Unorthodox Life
      467     s468    Movie  Private Network: Who Killed Manuel Buendía?
      468     s469    Movie               The Guide to the Perfect Family
      471     s472    Movie                                Day of Destiny
      8437   s8438  TV Show                        The Netflix Afterparty

                          director  \
      4250        No Data Availabe
      966            Pedro Antonio
      967         No Data Availabe
      968         No Data Availabe
      972         No Data Availabe
      …                      …
      466         No Data Availabe
      467            Manuel Alcalá
      468            Ricardo Trogi
      471   Akay Mason, Abosi Ogba
      8437        No Data Availabe

                                                         cast            country  \
      4250                                  No Data Available  No Data Available
      966   Marcus Majella, Samantha Schmütz, Caito Mainie…             Brazil
      967                                 Evelyn Lewis Prieto  No Data Available
      968   Aleksandra Skraba, Maria Sobocińska, Sandra Dr…             Poland
      972   Burcu Biricik, Uğur Yücel, Mehmet Yılmaz Ak, H…             Turkey
      …                                                 …                …
      466                                   No Data Available  No Data Available
      467                                Daniel Giménez Cacho  No Data Available
      468   Louis Morissette, Émilie Bierre, Catherine Cha…  No Data Available
      471   Olumide Oworu, Denola Grey, Gbemi Akinlade, Ji…  No Data Available
      8437       David Spade, London Hughes, Fortune Feimster      United States

           date_added  release_year rating  duration  \
      4250 2018-12-30          1925  TV-14  1 Season
      966  2021-04-28          2021  TV-MA    95 min
      967  2021-04-28          2021   TV-G  1 Season
      968  2021-04-28          2021  TV-MA  1 Season
      972  2021-04-27          2021  TV-MA  1 Season
      …         …           …    …       …
      466  2021-07-14          2021  TV-MA  1 Season
      467  2021-07-14          2021  TV-MA   100 min
      468  2021-07-14          2021  TV-MA   102 min
      471  2021-07-13          2021  TV-PG   110 min
      8437 2021-01-02          2021  TV-MA  1 Season

                                                    listed_in  \
      4250                                           TV Shows
      966                      Comedies, International Movies
      967                     Docuseries, Science & Nature TV
      968      International TV Shows, TV Comedies, TV Dramas
      972     International TV Shows, TV Dramas, TV Thrillers
      …                                                 …
      466                                          Reality TV
      467                 Documentaries, International Movies
```

```
468              Comedies, Dramas, International Movies
471   Children & Family Movies, Dramas, Internationa…
8437           Stand-Up Comedy & Talk Shows, TV Comedies

                                            description
4250  This collection restores films from women who …
966   After a botched scam, Clóvis bumps into Lohane…
967   Learn how to sleep better with Headspace. Each…
968   To build an innovative sex app and win a tech …
972   Reeling from tragedy, a nondescript house clea…
…                                                   …
466   Follow Julia Haart, Elite World Group CEO and …
467   A deep dive into the work of renowned Mexican …
468   A couple in Québec deals with the pitfalls, pr…
471   With their family facing financial woes, two t…
8437  Hosts David Spade, Fortune Feimster and London…

[593 rows x 12 columns]
```

Working on the columns having maximum null values and the columns having comma separated multiple values for each record

1. Country column

[45]:
```
df['country'].value_counts()
```

[45]:
```
United States                                 2818
India                                          972
No Data Available                              831
United Kingdom                                 419
Japan                                          245
                                              …
Romania, Bulgaria, Hungary                       1
Uruguay, Guatemala                               1
France, Senegal, Belgium                         1
Mexico, United States, Spain, Colombia           1
United Arab Emirates, Jordan                     1
Name: country, Length: 749, dtype: int64
```

This makes it difficult to analyse how many movies were produced in each country. We can use explode function in pandas to split the country column into different rows.

we are Creating a separate table for country , to avoid the duplicasy of records in our origional table after exploding.

[46]:
```
country_tb = df[['show_id' , 'type' , 'country']]
country_tb.dropna(inplace = True)
country_tb['country'] = country_tb['country'].apply(lambda x : x.split(','))
country_tb = country_tb.explode('country')
```

```
country_tb
```

```
[46]:      show_id     type          country
      0          s1    Movie    United States
      1          s2  TV Show     South Africa
      2          s3  TV Show  No Data Available
      3          s4  TV Show  No Data Available
      4          s5  TV Show            India
      …        …       …                 …
      8802    s8803    Movie    United States
      8803    s8804  TV Show  No Data Available
      8804    s8805    Movie    United States
      8805    s8806    Movie    United States
      8806    s8807    Movie            India

      [10850 rows x 3 columns]
```

```
[47]: # some duplicate values are found, which have unnecessary spaces. some empty␣
       ↪strings found
      country_tb['country'] = country_tb['country'].str.strip()
```

```
[48]: country_tb.loc[country_tb['country'] == '']
```

```
[48]:      show_id     type country
      193      s194  TV Show
      365      s366    Movie
      1192    s1193    Movie
      2224    s2225    Movie
      4653    s4654    Movie
      5925    s5926    Movie
      7007    s7008    Movie
```

2. Director column

```
[49]: df['director'].value_counts()
```

```
[49]: No Data Availabe                  2634
      Rajiv Chilaka                       19
      Raúl Campos, Jan Suter              18
      Suhas Kadav                         16
      Marcus Raboy                        16
                                         …
      Raymie Muzquiz, Stu Livingston       1
      Joe Menendez                         1
      Eric Bross                           1
      Will Eisenberg                       1
      Mozez Singh                          1
      Name: director, Length: 4529, dtype: int64
```

There are some movies which are directed by multiple directors. Hence multiple names of directors are given in comma separated format. We will explode the director column as well. It will create many duplicate records in originaltable hence we created separate table for directors.

[50]:
```
dir_tb = df[['show_id' , 'type' , 'director']]
dir_tb.dropna(inplace = True)
dir_tb['director'] = dir_tb['director'].apply(lambda x : x.split(','))
dir_tb
```

[50]:
```
     show_id     type            director
0         s1    Movie   [Kirsten Johnson]
1         s2  TV Show  [No Data Availabe]
2         s3  TV Show   [Julien Leclercq]
3         s4  TV Show  [No Data Availabe]
4         s5  TV Show  [No Data Availabe]
…        …       …                   …
8802   s8803    Movie     [David Fincher]
8803   s8804  TV Show  [No Data Availabe]
8804   s8805    Movie   [Ruben Fleischer]
8805   s8806    Movie      [Peter Hewitt]
8806   s8807    Movie       [Mozez Singh]

[8807 rows x 3 columns]
```

3. 'listed_in' column to understand more about genres

[51]:
```
genre_tb = df[['show_id' , 'type', 'listed_in']]
```

[52]:
```
genre_tb['listed_in'] = genre_tb['listed_in'].apply(lambda x : x.split(','))
genre_tb = genre_tb.explode('listed_in')
genre_tb['listed_in'] = genre_tb['listed_in'].str.strip()
```

[53]:
```
genre_tb
```

[53]:
```
     show_id     type                 listed_in
0         s1    Movie             Documentaries
1         s2  TV Show    International TV Shows
1         s2  TV Show                 TV Dramas
1         s2  TV Show              TV Mysteries
2         s3  TV Show           Crime TV Shows
…        …       …                         …
8805   s8806    Movie  Children & Family Movies
8805   s8806    Movie                  Comedies
8806   s8807    Movie                    Dramas
8806   s8807    Movie      International Movies
8806   s8807    Movie          Music & Musicals

[19323 rows x 3 columns]
```

[54]:

```
genre_tb.listed_in.unique()
```

[54]:

```
array(['Documentaries', 'International TV Shows', 'TV Dramas',
       'TV Mysteries', 'Crime TV Shows', 'TV Action & Adventure',
       'Docuseries', 'Reality TV', 'Romantic TV Shows', 'TV Comedies',
       'TV Horror', 'Children & Family Movies', 'Dramas',
       'Independent Movies', 'International Movies', 'British TV Shows',
       'Comedies', 'Spanish-Language TV Shows', 'Thrillers',
       'Romantic Movies', 'Music & Musicals', 'Horror Movies',
       'Sci-Fi & Fantasy', 'TV Thrillers', "Kids' TV",
       'Action & Adventure', 'TV Sci-Fi & Fantasy', 'Classic Movies',
       'Anime Features', 'Sports Movies', 'Anime Series',
       'Korean TV Shows', 'Science & Nature TV', 'Teen TV Shows',
       'Cult Movies', 'TV Shows', 'Faith & Spirituality', 'LGBTQ Movies',
       'Stand-Up Comedy', 'Movies', 'Stand-Up Comedy & Talk Shows',
       'Classic & Cult TV'], dtype=object)
```

[55]:

```
genre_tb.listed_in.nunique()
```

[55]:

```
42
```

4. Casting Column

[56]:

```
cast_tb = df[['show_id' , 'type' ,'cast']]
cast_tb.dropna(inplace = True)
cast_tb['cast'] = cast_tb['cast'].apply(lambda x : x.split(','))
cast_tb = cast_tb.explode('cast')
cast_tb
```

[56]:

```
     show_id     type                    cast
0         s1    Movie       No Data Available
1         s2  TV Show              Ama Qamata
1         s2  TV Show             Khosi Ngema
1         s2  TV Show           Gail Mabalane
1         s2  TV Show          Thabang Molaba
…      …      …                     …
8806   s8807    Movie        Manish Chaudhary
8806   s8807    Movie            Meghna Malik
8806   s8807    Movie           Malkeet Rauni
8806   s8807    Movie          Anita Shabdish
8806   s8807    Movie   Chittaranjan Tripathy

[64951 rows x 3 columns]
```

[57]:

```
cast_tb['cast'] = cast_tb['cast'].str.strip()
```

[58]:

```
# checking empty strings
cast_tb[cast_tb['cast'] == '']
```

```
[58]: Empty DataFrame
      Columns: [show_id, type, cast]
      Index: []
```

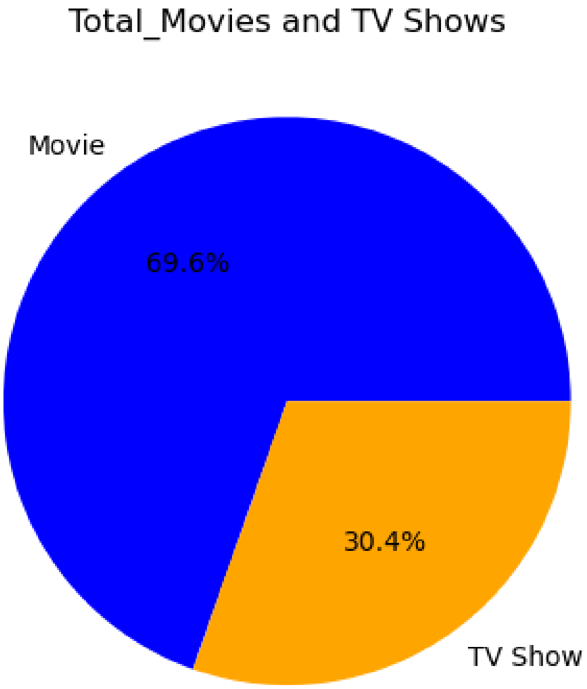
```
[59]: # Total actors on the Netflix
      cast_tb.cast.nunique()
```

```
[59]: 36440
```

4 4. Visual Analysis - Univariate & Bivariate

4.0.1 4.1. Distribution of content across the different types

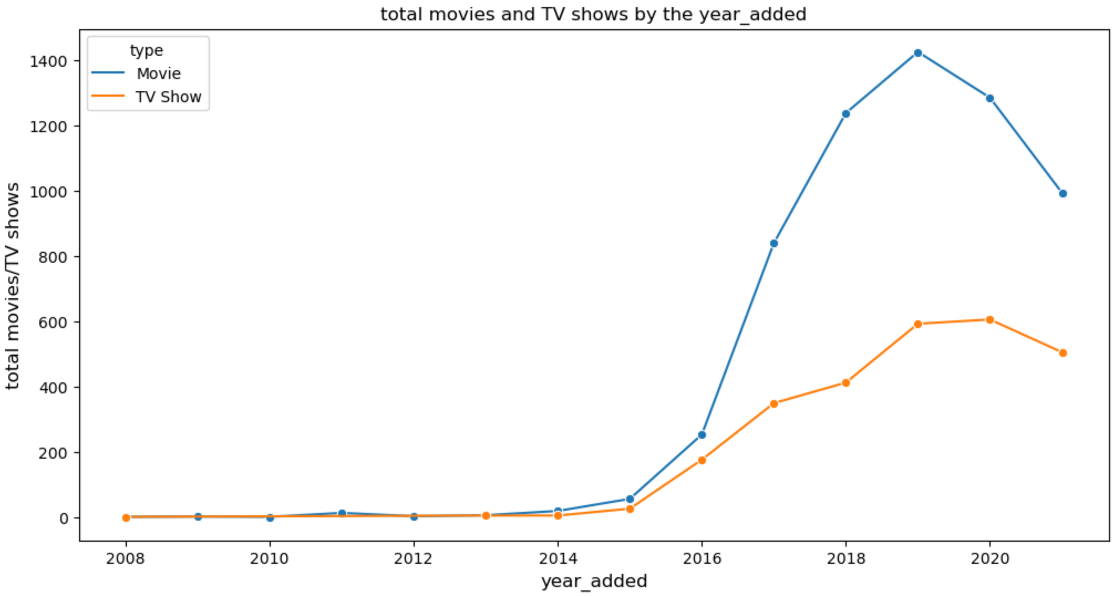
```
[60]: types = df.type.value_counts()
      plt.pie(types,  labels=types.index, autopct='%1.1f%%' , colors = ['blue' ,␣
        ↪'orange'])
      plt.title('Total_Movies and TV Shows')
      plt.show()
```



4.0.2 4.2 Distribution of ‘date_added’ column

[75]:
```
d = df.groupby(['year_added' ,'type' ])['show_id'].count().reset_index()
d.rename({'show_id' : 'total movies/TV shows'}, axis = 1 , inplace = True)
```

[76]:
```
plt.figure(figsize = (12,6))
sns.lineplot(data = d , x = 'year_added' , y = 'total movies/TV shows' , hue =␣
  ↪'type', marker = 'o'  , ms = 6)
plt.xlabel('year_added' , fontsize = 12)
plt.ylabel('total movies/TV shows' , fontsize = 12)
plt.title('total movies and TV shows by the year_added' , fontsize = 12)
plt.show()
```



Observation:

```
The content added on the Netflix surged drastically after 2015.
2019 marks the highest number of movies and TV shows added on the Netflix.
Year 2020 and 2021 has seen the drop in content added on Netflix, possibly because of Pandemic
```

But still , TV shows content have not dropped as drastic as movies. In recent years TV shows are focussed more than Movies.

4.0.3 4.3 Distribution of ‘Release_year’ column

[77]:
```
d = df.groupby(['type' , 'release_year'])['show_id'].count().reset_index()
d.rename({'show_id' : 'total movies/TV shows'}, axis = 1 , inplace = True)
d
```

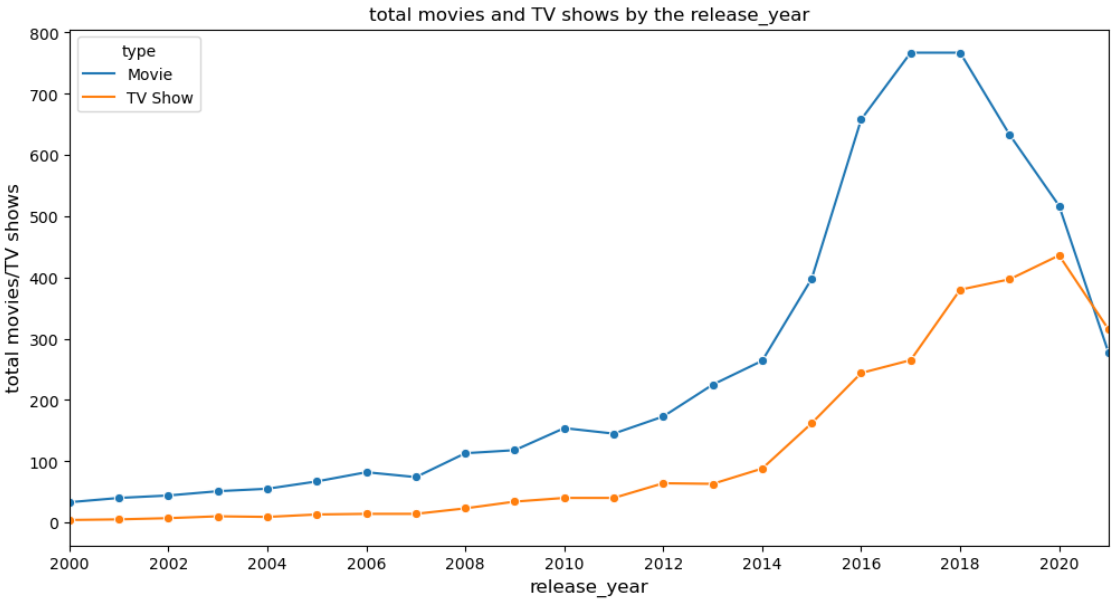
[77]:

```
          type  release_year  total movies/TV shows
0        Movie          1942                      2
1        Movie          1943                      3
2        Movie          1944                      3
3        Movie          1945                      3
4        Movie          1946                      1
..         …             …                      …
114    TV Show          2017                    265
115    TV Show          2018                    380
116    TV Show          2019                    397
117    TV Show          2020                    436
118    TV Show          2021                    315

[119 rows x 3 columns]
```

[78]:

```
plt.figure(figsize = (12,6))
sns.lineplot(data = d , x = 'release_year' , y = 'total movies/TV shows' , hue␣
 ↪= 'type' , marker = 'o'  , ms = 6 )
plt.xlabel('release_year' , fontsize = 12)
plt.ylabel('total movies/TV shows' , fontsize = 12)
plt.title('total movies and TV shows by the release_year' , fontsize = 12)
plt.xlim( left = 2000 , right = 2021)
plt.xticks(np.arange(2000 , 2021 , 2))
plt.show()
```



Observation:

`2018 marks the highest number of movie and TV show releases.`

```
Since 2018, A drop in movies is seen and rise in TV shows is observed clearly, and TV shows su
In recent years TV shows are focussed more than Movies.
The yearly number of releases has surged drastically from 2015.
```

4.0.4 4.4 Total movies/TV shows by each country

[82]:

```
# Lets check for top 10 countries
top_10_country = country_tb.country.value_counts().head(10).index
df_new = country_tb.loc[country_tb['country'].isin(top_10_country)]
```

[83]:

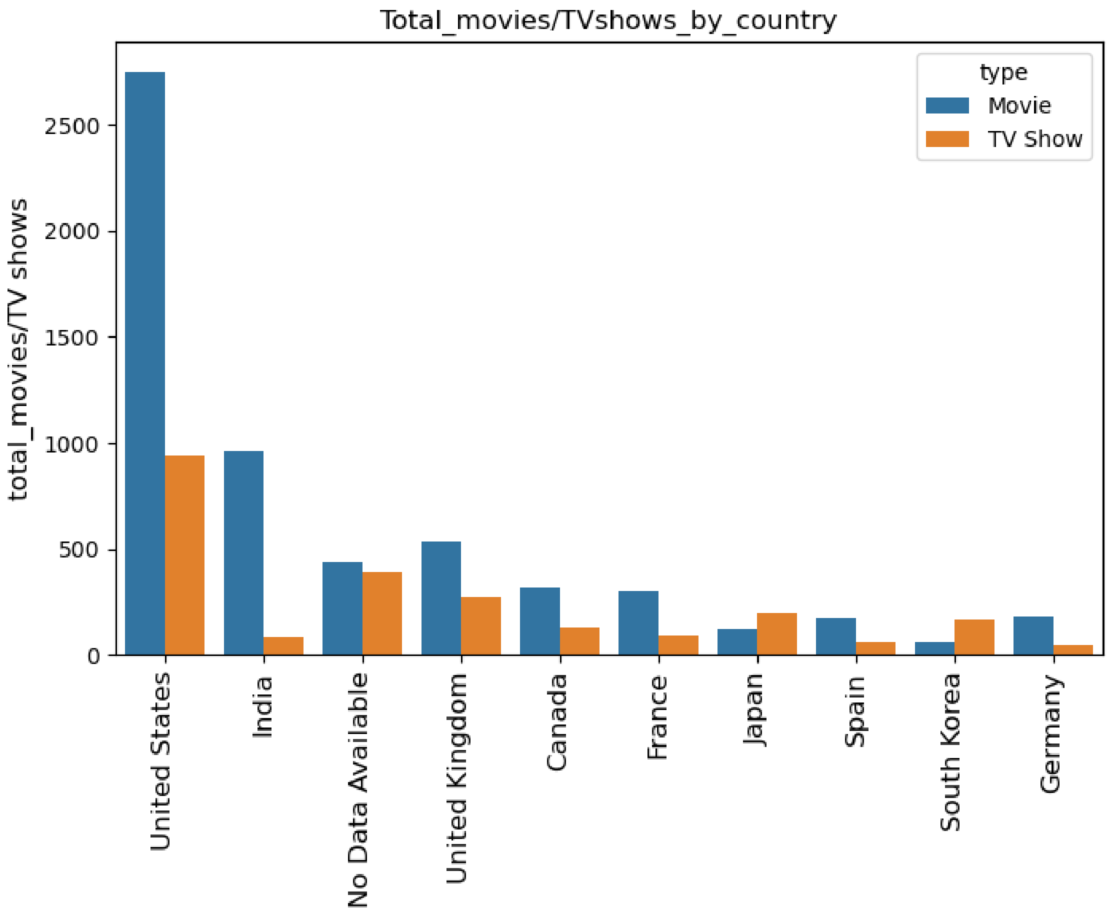
```
x = df_new.groupby(['country' , 'type'])['show_id'].count().reset_index()
x.pivot(index = 'country' , columns = 'type' , values = 'show_id').
 ↪sort_values('Movie',ascending = False)
```

[83]:

```
type                Movie  TV Show
country
United States        2752      938
India                 962       84
United Kingdom        534      272
No Data Available     440      391
Canada                319      126
France                303       90
Germany               182       44
Spain                 171       61
Japan                 119      199
South Korea            61      170
```

[84]:

```
plt.figure(figsize= (8,5))
sns.countplot(data = df_new , x = 'country' , order = top_10_country , hue =␣
 ↪'type')
plt.xticks(rotation = 90 , fontsize = 12)
plt.ylabel('total_movies/TV shows' , fontsize = 12)
plt.xlabel('')
plt.title('Total_movies/TVshows_by_country')
plt.show()
```



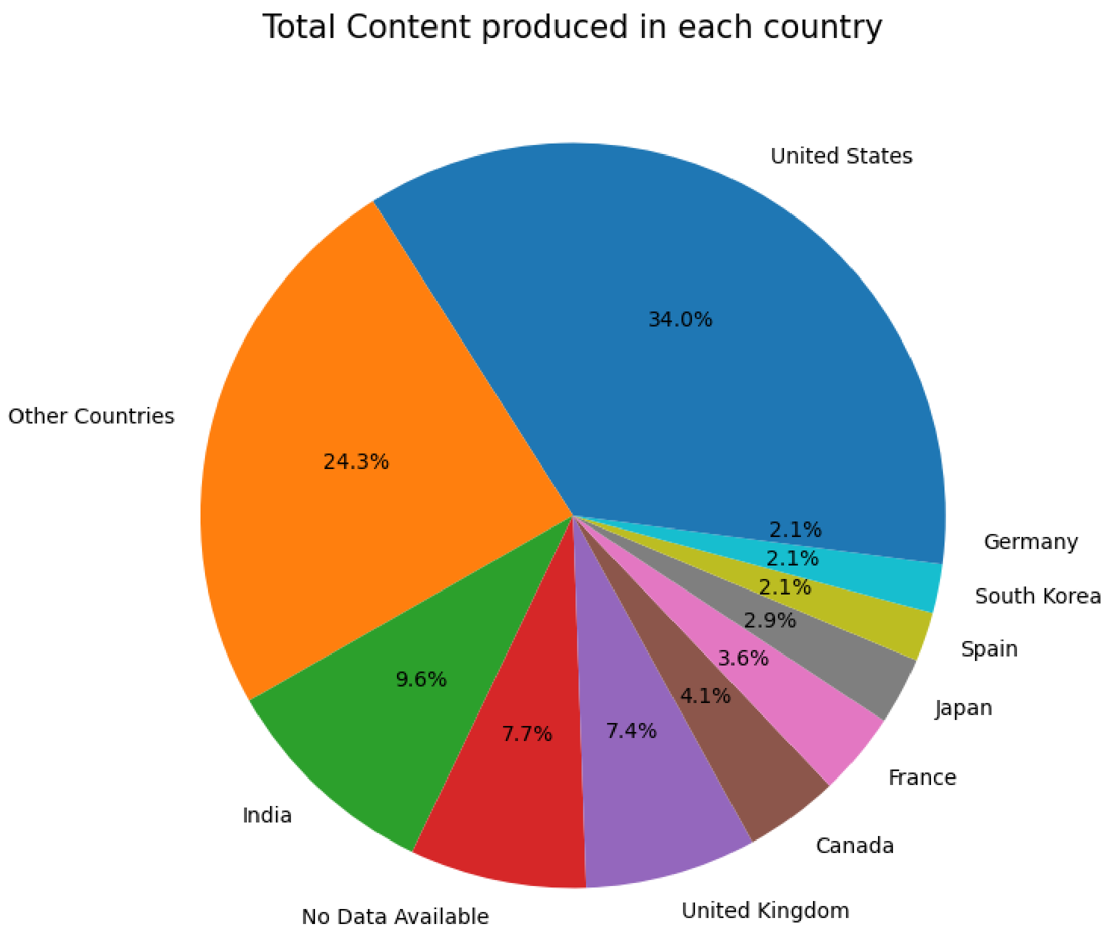
[85]:

```
top_10_country = country_tb.country.value_counts().head(10).index
country_tb['cat'] = country_tb['country'].apply(lambda x : x if x in␣
 ↪top_10_country else 'Other Countries' )
```

[86]:

```
x = country_tb.cat.value_counts()

plt.figure(figsize = (8,8))
plt.pie(x , labels = x.index, autopct='%1.1f%%')
plt.title('Total Content produced in each country' , fontsize = 15)
plt.show()
```



Observation:

```
United States is the HIGHEST contributor country on Netflix, followed by India and United King
Maximum content of Netflix which is around 75% , is coming from these top 10 countries. Rest o
```

5 5. Bivariate Analysis

5.0.1 5.1 Lets check popular genres in top 20 countries

[88]:

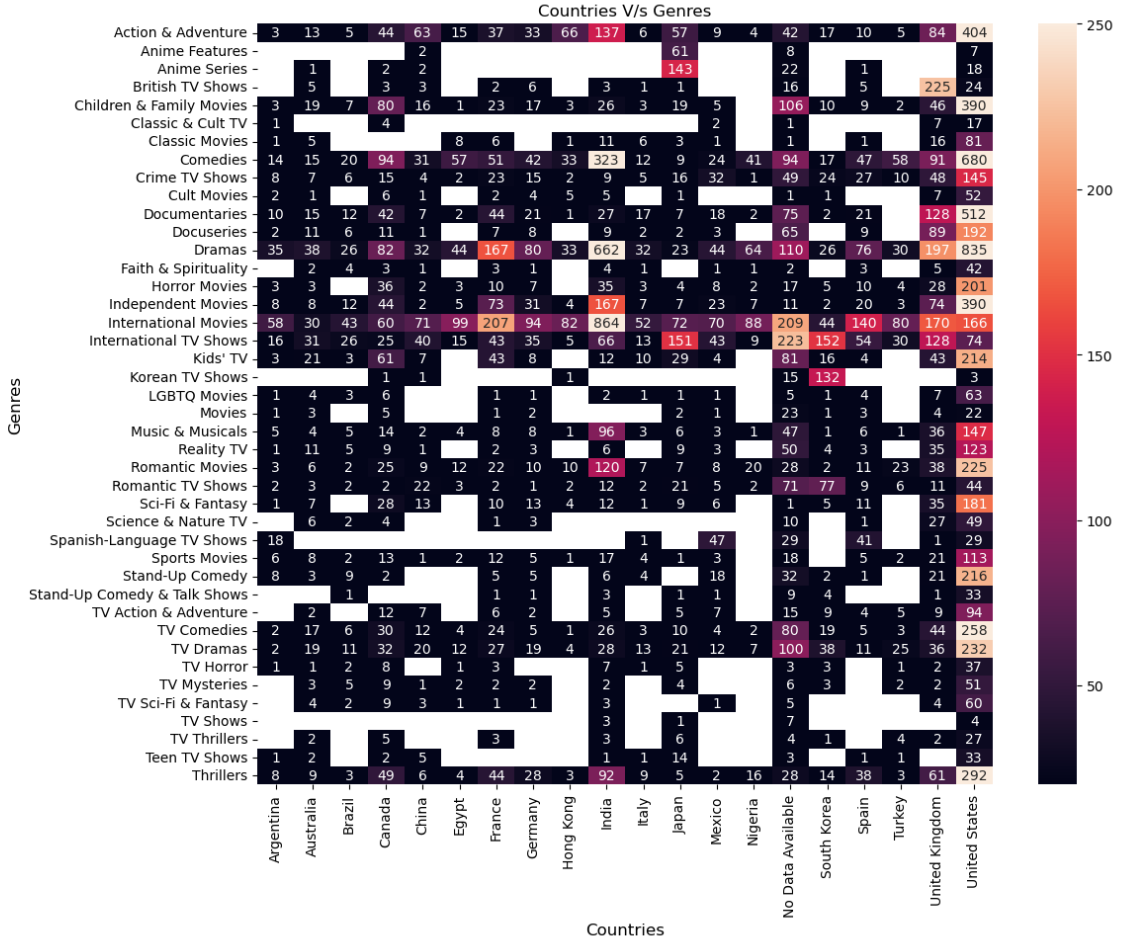
```
top_20_country = country_tb.country.value_counts().head(20).index
top_20_country = country_tb.loc[country_tb['country'].isin(top_20_country)]
```

[89]:

```
x = top_20_country.merge(genre_tb , on = 'show_id').drop_duplicates()
country_genre = x.groupby([ 'country' , 'listed_in'])['show_id'].count().
  ↪sort_values(ascending = False).reset_index()
country_genre = country_genre.pivot(index = 'listed_in' , columns = 'country' ,␣
  ↪values = 'show_id')
```

```
[90]: plt.figure(figsize = (12,10))
      sns.heatmap(data = country_genre , annot = True , fmt=".0f" , vmin = 20 , vmax␣
       ↪= 250 )
      plt.xlabel('Countries' , fontsize = 12)
      plt.ylabel('Genres' , fontsize = 12)
      plt.title('Countries V/s Genres' , fontsize = 12)
```

```
[90]: Text(0.5, 1.0, 'Countries V/s Genres')
```



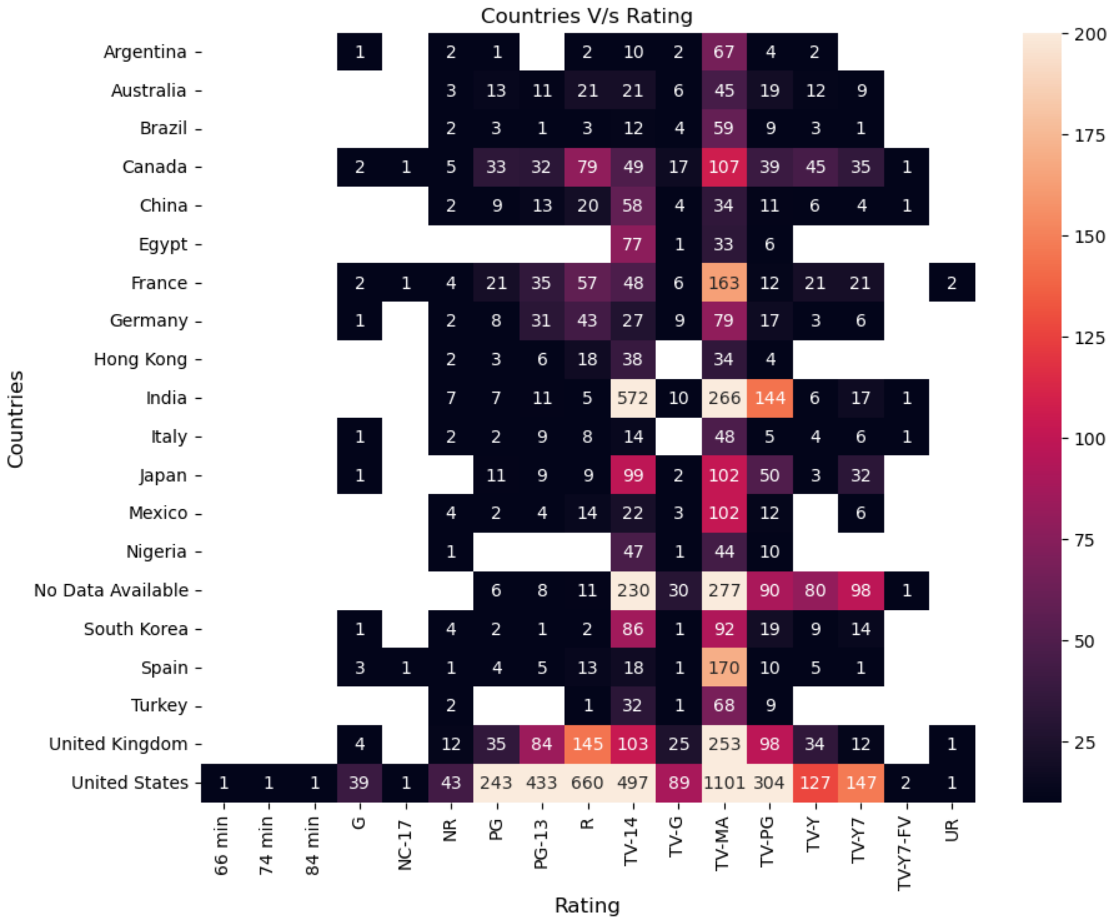
5.0.2 5.2 Country-wise Rating of Content

```
[91]: x = top_20_country.merge(df , on = 'show_id').groupby(['country_x' ,␣
       ↪'rating'])['show_id'].count().reset_index()
```

```
[92]: country_rating = x.pivot(index = ['country_x'] , columns = 'rating' , values =␣
       ↪'show_id')
```

[93]:
```
plt.figure(figsize = (10,8))
sns.heatmap(data = country_rating , annot = True , fmt=".0f"  , vmin = 10 ,␣
 ↪vmax=200)
plt.ylabel('Countries' , fontsize = 12)
plt.xlabel('Rating' , fontsize = 12)
plt.title('Countries V/s Rating' , fontsize = 12)
```

[93]:
```
Text(0.5, 1.0, 'Countries V/s Rating')
```



5.0.3 5.3 The top actors by country

[94]:
```
x = cast_tb.merge(country_tb , on = 'show_id').drop_duplicates()
x = x.groupby(['country' , 'cast'])['show_id'].count().reset_index()
x.loc[x['country'].isin(['United States'])].sort_values('show_id' , ascending =␣
 ↪False).head(5)
```

[94]:
```
                country                cast  show_id
50571  United States   No Data Available      407
```

```
53483  United States        Tara Strong       22
52408  United States  Samuel L. Jackson       22
44529  United States    Fred Tatasciore       21
39794  United States       Adam Sandler       20
```

[95]:

```
country_list = ['India'  , 'United Kingdom' , 'Canada' , 'France' , 'Japan']
top_5_actors = x.loc[x['country'].isin(['United States'])].
 ↪sort_values('show_id' , ascending = False).head(5)
```

[96]:

```
# top 5 actors in top countries and their movies/tv shows count
top_5_actors
```

[96]:

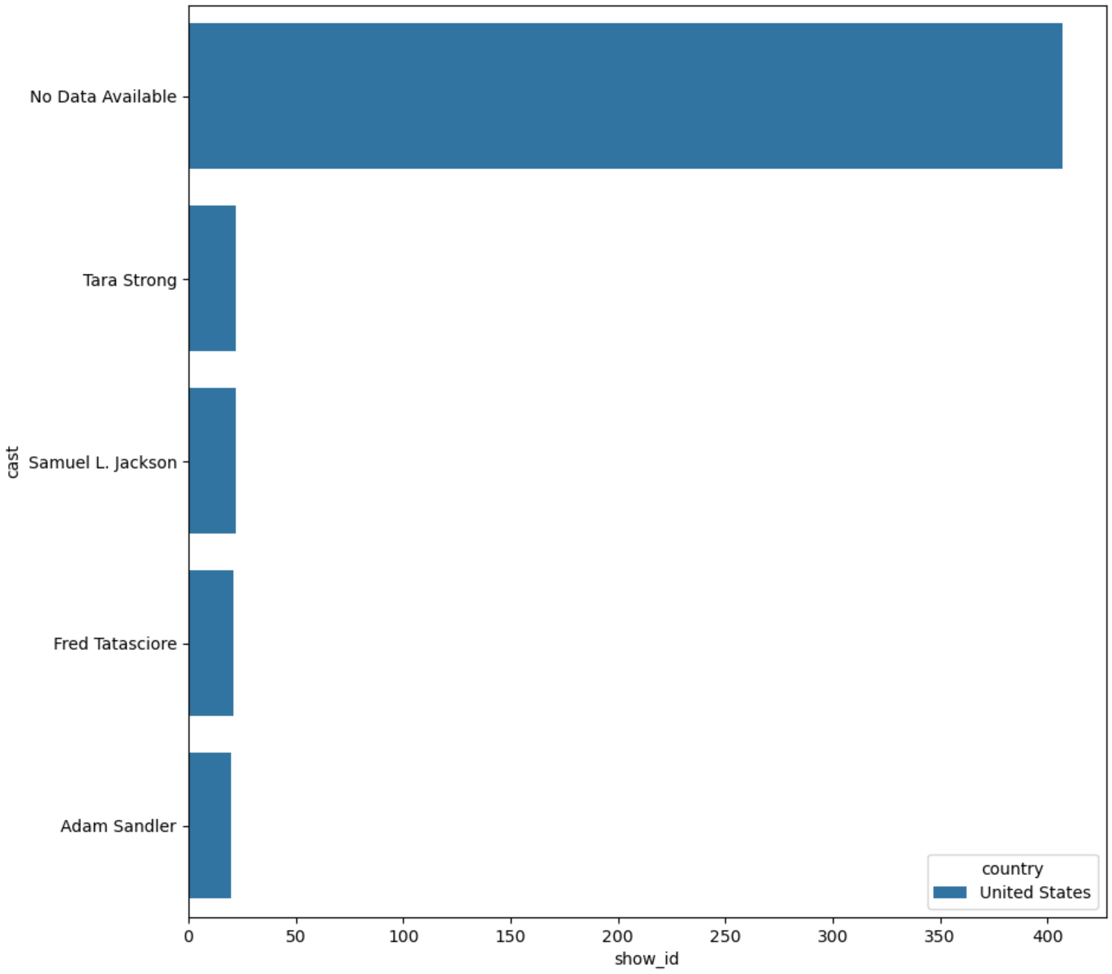
```
             country               cast  show_id
50571  United States  No Data Available      407
53483  United States        Tara Strong       22
52408  United States  Samuel L. Jackson       22
44529  United States    Fred Tatasciore       21
39794  United States       Adam Sandler       20
```

[97]:

```
plt.figure(figsize = (10,10))
sns.barplot(data = top_5_actors , y = 'cast' , x = 'show_id' , hue = 'country')
```

[97]:

```
<Axes: xlabel='show_id', ylabel='cast'>
```

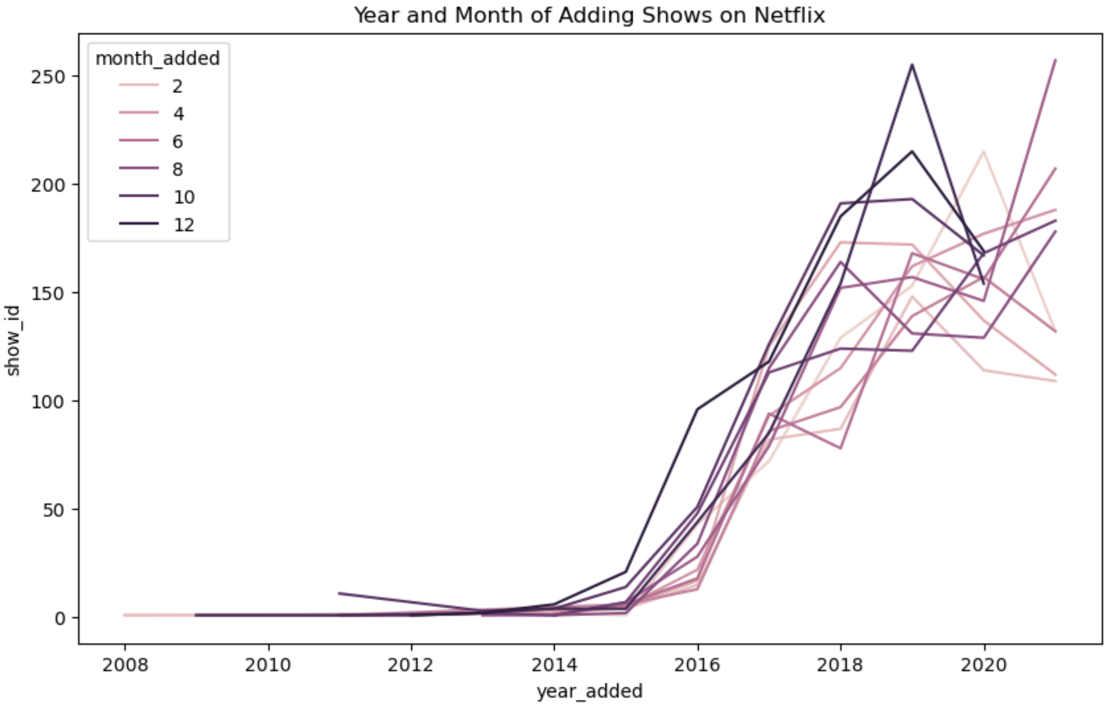


5.0.4 5.4 What is the best time of the year when maximum content get added on the Netflix?

```
[100]: month_year = df.groupby(['year_added' , 'month_added'])['show_id'].count().
         ↪reset_index()
```

```
[101]: plt.figure(figsize = (10,6))
       sns.lineplot(data=month_year, x = 'year_added', y = 'show_id',␣
         ↪hue='month_added')
       plt.title('Year and Month of Adding Shows on Netflix')
```

```
[101]: Text(0.5, 1.0, 'Year and Month of Adding Shows on Netflix')
```



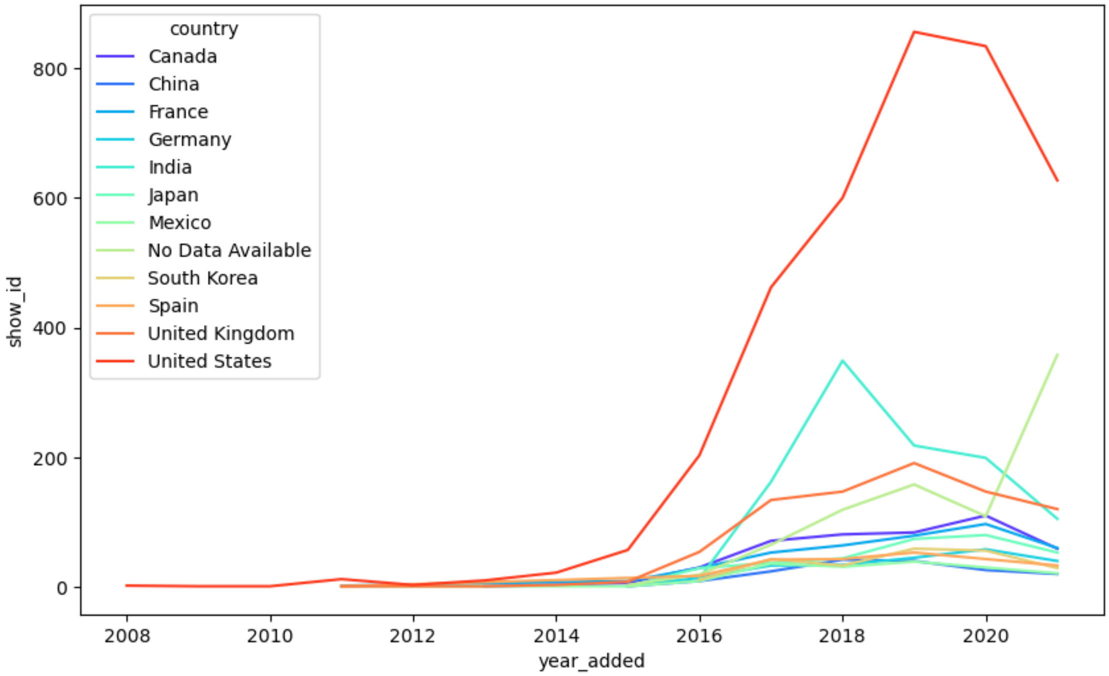
5.0.5 5.5 Which countries are adding more number of content over the time?

```
[102]: country_list = country_tb.country.value_counts().head(12).index
top_12_country = country_tb.loc[country_tb['country'].isin(country_list)]
country_year = top_12_country.merge(df , on = 'show_id')[['show_id','country_x',␣
↪,'type_x' , 'year_added' ]]
country_year.columns = ['show_id', 'country', 'type', 'year_added']
```

```
[103]: country_year = country_year.groupby(['country' , 'year_added'])['show_id'].
↪count().reset_index()
```

```
[104]: plt.figure(figsize = (10,6))
sns.lineplot(data = country_year , x = 'year_added' , y = 'show_id' , hue =␣
↪'country' , palette ='rainbow' )
```

```
[104]: <Axes: xlabel='year_added', ylabel='show_id'>
```



6 5. Outlier check

[66]:
```
def calculate_outliers(data):
    # Calculate the first quartile (Q1)
    q1 = np.percentile(data, 25)

    # Calculate the third quartile (Q3)
    q3 = np.percentile(data, 75)

    # Calculate the interquartile range (IQR)
    iqr = q3 - q1

    # Determine the lower and upper bounds for outliers
    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr

    # Identify outliers in the dataset
    outliers = [value for value in data if value < lower_bound or value >␣
 ↪upper_bound]

    return outliers


def calculate_max_occurred_value(data):
    # Calculate the unique values and their counts in the dataset
    unique_values, value_counts = np.unique(data, return_counts=True)

    # Find the index of the maximum count
    max_count_index = np.argmax(value_counts)

    # Retrieve the corresponding unique value with the maximum count
    max_occurred_value = unique_values[max_count_index]

    return max_occurred_value
```

```
[67]: outliers = calculate_outliers(x)  # Implement your outlier calculation method
      max_occurred_value = calculate_max_occurred_value(x)  # Implement your method␣
       ↪to find the maximum-occurred value
      set(outliers)
```

```
[67]: {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 19, 2634}
```

```
[68]: max_occurred_value
```

```
[68]: 1
```

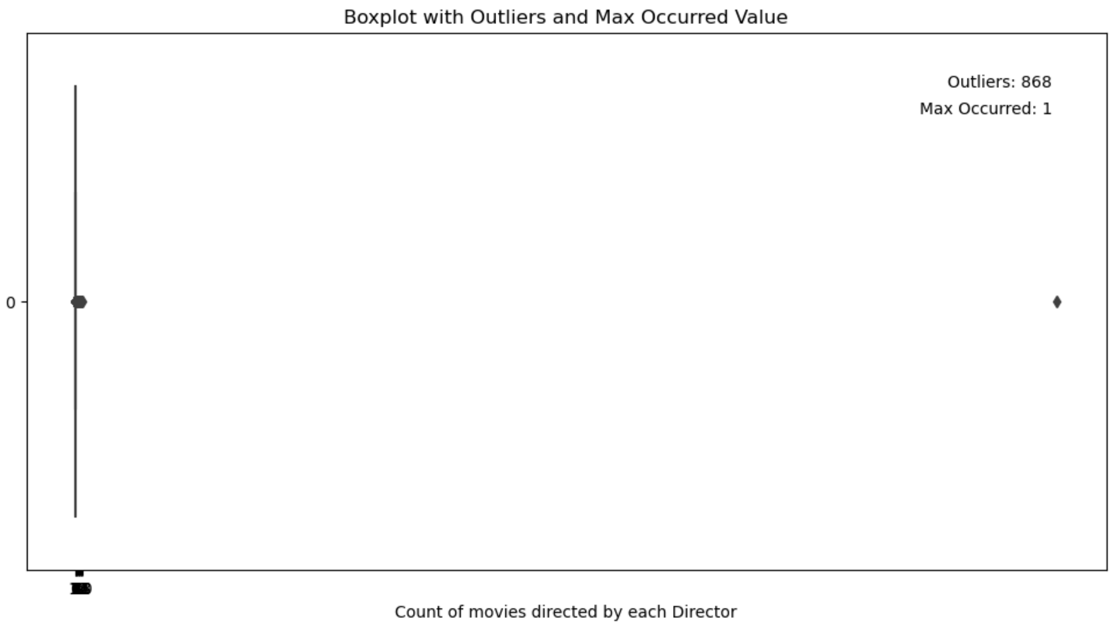
```
[69]: plt.figure(figsize = (12,6))
      sns.boxplot(data=x, showfliers=True, whis=1.5 , orient = 'h')

      # Calculate the outliers and maximum-occurred value
      outliers = calculate_outliers(x)  # Implement your outlier calculation method
      max_occurred_value = calculate_max_occurred_value(x)  # Implement your method␣
       ↪to find the maximum-occurred value

      # Annotate the plot
      plt.text(0.95, 0.9, f"Outliers: {len(outliers)}", transform=plt.gca().
       ↪transAxes, ha='right')
      plt.text(0.95, 0.85, f"Max Occurred: {max_occurred_value}", transform=plt.gca().
       ↪transAxes, ha='right')


      plt.xlabel("Count of movies directed by each Director")
      plt.xticks(np.arange(0,22,2))
      plt.title("Boxplot with Outliers and Max Occurred Value")

      # Show the plot
      plt.show()
```



7 6. Insights based on Non-Graphical and Visual Analysis

1. Around 70% content on Netflix is Movies and around 30% content is TV shows.
2. The movies and TV shows uploading on the Netflix started from the year 2008, It had very lesser content till 2014.
3. Year 2015 marks the drastic surge in the content getting uploaded on Netflix. It continues the uptrend since then and 2019 marks the highest number of movies and TV shows added on the Netflix. Year 2020 and 2021 has seen the drop in content added on Netflix, possibly because of Pandemic. But still , TV shows content have not dropped as drastic as movies.
4. Since 2018, A drop in the movies is seen , but rise in TV shows is observed clearly. Being in continuous uptrend , TV shows surpassed the movies count in mid 2020. It shows the rise in popularity of tv shows in recent years.
5. Netflix has movies from variety of directors. Around 4993 directors have their movies or tv shows on Netflix.
6. Netflix has movies from total 122 countries, United States being the highset contributor with almost 37% of all the content.
7. The release year for shows is concentrated in the range 2005-2021.
8. 50 mins - 150 mins is the range of movie durations, excluding potential outliers.
9. 1-3 seasons is the range for TV shows seasons, excluding potential outliers.
10. various ratings of content is avaialble on netfilx, for the various viewers categories like kids, adults , families. Highest number of movies and TV shows are rated TV-MA (for mature audiences).
11. Content in most of the ratings is available in lesser quanitity except in US. Ratings like TV-Y7 , TV-Y7 FV , PG ,TV-G , G , TV-Y , TV-PG are very less avaialble in all countries except US.
12. International Movies and TV Shows , Dramas , and Comedies are the top 3 genres on Netflix

for both Movies and TV shows.

13. Mostly country specific popular genres are observed in each country. Only United States have a good mix of almost all genres. Eg. Korean TV shows (Korea), British TV Shows (UK), Anime features and Anime series (Japan) and so on.
14. Indian Actors have been acted in maximum movies on netflix. Top 5 actors are in India based on quantity of movies.
15. Shorter duration movies have been popular in last 10 years.

8 7. Business Insights

1. Netflix have majority of content which is released after the year 2000. It is observed that the content older than year 2000 is very scarce on Netflix. Senior Citizen could be the target audience for such content, which is almost missing currently.
2. Most popular genres on Netflix are International Movies and TV Shows , Dramas , Comedies, Action & Adventure, Children & Family Movies, Thrillers.
3. Maximum content of Netflix which is around 75% , is coming from the top 10 countries. Rest of the world only contributes 25% of the content. More countries can be focussed in future to grow the business.
4. ing towards the shorter duration content is on the rise. (duration 75 to 150 minutes and seasons 1 to 3)

This can be considered while production of new content on Netflix.

```
drop in content is seen across all the countries and type of content in year 2020 and 2021, pos
```

9 8. Recommendations

Very limited genres are focussed in most of the countries except US. It seems the current available genres suits best for US and few countries but maximum countries need some more genres which are highly popular in the region.

Eg. Indian Mythological content is highly popular. We can create such more country specific genres and It might also be liked acorss the world just like Japanese Anime.

Country specific insights - The content need to be targetting the demographic of any country. Netflix can produce higher number of content in the perticular rating as per demographic of the country.

[ ]: