PYTHON PROJECT ON NETFLIX DATASET

PROJECT NUMBER 1
BAHADURSHA A V L SAINADH
SCALER-JULY 2022 BEGINNER BATCH

CHAPTER 1: DEFINITION OF THE PROBLEM STATEMENT AND ANALYZING BASIC METRICS

1.1 INTRODUCTION TO NETFLIX

Netflix, Inc. is an American media company based in `Los Gatos, California.` It was founded in `1997` by `Reed Hastings` and `Marc Randolph`. It is a subscription-based streaming service that allows subscribed members to watch TV shows and movies on an internet-connected device.

They have over `10000 movies or tv shows` available on their platform, as of `mid-2021`, they have over `222M Subscribers` globally. This tabular dataset consists of listings of all the movies and tv shows available on Netflix, along with details such as - cast, directors, ratings, release year, duration, etc.

1.2 PROBLEM DEFINITION

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

Analyse the dataset using `PYTHON` programming and its libraries like `NUMPY, PANDAS, MATPLOTLIB, SEABORN`.

Import all the required libraries:

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

Load and read the provided dataset:

```
!gdown 1p03WPc1HT-gtNM4seYhxXdo0fQ1KjG1X

Downloading...
From: https://drive.google.com/uc?id=1p03WPc1HT-gtNM4seYhxXdo0fQ1KjG1X
To: /content/netflix.csv
   0% 0.00/3.40M [00:00<?, ?B/s] 100% 3.40M/3.40M [00:00<00:00,
136MB/s]

netflix = pd.read_csv("netflix.csv") # reading the csv file into df
variable
```

Actual provided data looks like:

```
netflix.head(10) # first 10 rows of the dataframe
```

```
  show_id     type                                 title  \
0      s1    Movie              Dick Johnson Is Dead
1      s2  TV Show                     Blood & Water
2      s3  TV Show                         Ganglands
3      s4  TV Show                Jailbirds New Orleans
4      s5  TV Show                         Kota Factory
5      s6  TV Show                        Midnight Mass
6      s7    Movie  My Little Pony: A New Generation
7      s8    Movie                               Sankofa
8      s9  TV Show     The Great British Baking Show
9     s10    Movie                          The Starling

                         director  \
0                Kirsten Johnson
1                             NaN
2                Julien Leclercq
3                             NaN
4                             NaN
5                  Mike Flanagan
6  Robert Cullen, José Luis Ucha
7                   Haile Gerima
8                Andy Devonshire
9                 Theodore Melfi
```

```
                                                  cast  \
0                                                  NaN
1  Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...
2  Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...
3                                                  NaN
4  Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...
5  Kate Siegel, Zach Gilford, Hamish Linklater, H...
6  Vanessa Hudgens, Kimiko Glenn, James Marsden, ...
7  Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...
8  Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho...
9  Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...

                                           country
date_added  \
0                                    United States  September 25,
2021
1                                     South Africa  September 24,
2021
2                                              NaN  September 24,
2021
3                                              NaN  September 24,
2021
4                                            India  September 24,
2021
5                                              NaN  September 24,
2021
6                                              NaN  September 24,
2021
7  United States, Ghana, Burkina Faso, United Kin...  September 24,
2021
8                                   United Kingdom  September 24,
2021
9                                    United States  September 24,
2021

   release_year rating   duration  \
0          2020  PG-13     90 min
1          2021  TV-MA  2 Seasons
2          2021  TV-MA   1 Season
3          2021  TV-MA   1 Season
4          2021  TV-MA  2 Seasons
5          2021  TV-MA   1 Season
6          2021     PG     91 min
7          1993  TV-MA    125 min
8          2021  TV-14  9 Seasons
9          2021  PG-13    104 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 ...
5                 TV Dramas, TV Horror, TV Mysteries
6                           Children & Family Movies
7   Dramas, Independent Movies, International Movies
8                        British TV Shows, Reality TV
9                                   Comedies, Dramas
```

```
                                           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...
5  The arrival of a charismatic young priest brin...
6  Equestria's divided. But a bright-eyed hero be...
7  On a photo shoot in Ghana, an American model s...
8  A talented batch of amateur bakers face off in...
9  A woman adjusting to life after a loss contend...
```

Description about Columns or Attributes available in the dataset:

1. Show_id: Unique ID for every Movie / Tv Show
1. Type: Identifier - A Movie or TV Show
2. Title: Title of the Movie / Tv Show
3. Director: Director of the Movie
4. Cast: Actors involved in the movie/show
5. Country: Country where the movie/show was produced
6. Date_added: Date it was added on Netflix
7. Release_year: Actual Release year of the movie/show
8. Rating: TV Rating of the movie/show
9. Duration: Total Duration - in minutes or number of seasons
10. Listed_in: Genre
11. Description: The summary description

`description` is unwanted column. It can be dropped from table. To analyse description column, Natural language processing is required (out of scope).

1.3 ANALYZING BASIC METRICS OF DATASET

Metrics are the numbers that are tracked in dataset by organisation which measures the performance or progress of the company.

Number of unique quantities in each column:

```
netflix.shape
```

```
(8807, 12)
```

8807 rows and 12 columns in given dataset

Attribute names

```
netflix.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country',
'date_added',
       'release_year', 'rating', 'duration', 'listed_in',
'description'],
      dtype='object')
```

Number of unique values in each column

```
netflix.nunique()
```

```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1767
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
```

`show_id and title` has 8807 unique values which is equal to number of rows. This implies that these two columns can act as unique identifiers or primary key.

`type` column has only two categories. TV Show and Movie

Number of non-null values and datatype Info related to all columns

```
netflix.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
```

Except `release_year` all are `object` data type. `release year` is `int64` data type.

`director, cast, country, date_added, rating, duration` are not having 8807 non null values. Should be verified for null values.

Number of null values in each column

```
netflix.isnull().sum()
```

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

Statistical description regarding numerical attributes (only one i.e., release year)

```
netflix.describe()
```

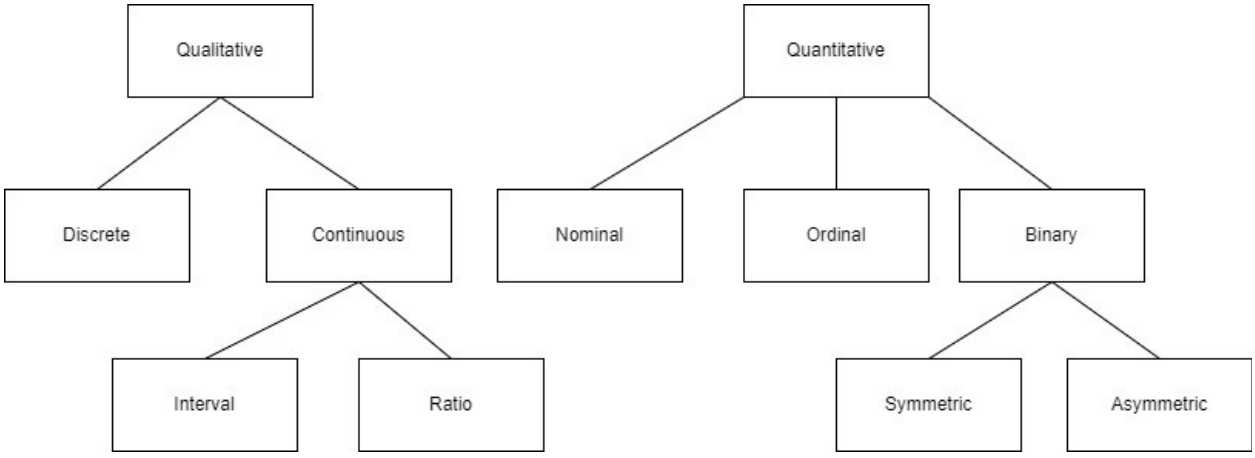
```
       release_year
count   8807.000000
mean    2014.180198
std        8.819312
min     1925.000000
25%     2013.000000
50%     2017.000000
75%     2019.000000
max     2021.000000
```

Statistical description regarding object datatype attributes

```
netflix.describe(include = "object")
```

```
       show_id   type                  title       director  \
count     8807   8807                   8807           6173
unique    8807      2                   8807           4528
top         s1  Movie  Dick Johnson Is Dead  Rajiv Chilaka
freq         1   6131                      1             19

                        cast        country       date_added rating
duration  \
count                   7982           7976             8797   8803
8804
unique                  7692            748             1767     17
220
top     David Attenborough  United States  January 1, 2020  TV-MA  1
Season
freq                      19           2818              109   3207
1793

                            listed_in  \
count                            8807
unique                            514
top     Dramas, International Movies
freq                              362

                                                 description
count                                                   8807
unique                                                  8775
top     Paranormal activity at a lush, abandoned prope...
freq                                                       4
```

Observations:



1. Nominal qualitative attributes: `"type", "title", "director", "cast", "country", "rating", "listed_in", "description"`
1. Ordinal qualitative attributes: `"show_id", "date_added", "release_year"`
2. Continuous quantitative attributes: `"duration"`

3. There are no Symmetric Binary qualitative attributes, Asymmetric Binary qualitative attributes, Discrete quantitative attributes
4. Nested data is available in `"director","cast","country","listed_in"`
5. Unwanted columns for data analysis are `"description"`
6. `duration` column should be divided to separate columns for tv_shows and movies. Data type should be `int64`. Those two divided columns will act as Continuous quantitative attributes. (min and seasons terms should not be present in column, only numbers should be present)

1.4 CLEANING THE DATASET (WITHOUT MISSING VALUE TREATMENT)

Delete the `description` column: (unwanted column)

```
netflix.drop(["description"],axis = 1,inplace = True)
```

`description` column deleted in netflix dataframe.

Unnesting the data

Unnesting should be done on `cast,director,country, listed_in`

splitting the nested data

```
netflix.shape
```

```
(8807, 11)
```

```
netflix.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country',
'date_added',
       'release_year', 'rating', 'duration', 'listed_in'],
      dtype='object')
```

There are 8807 Rows and 11 columns (after deleting `description` column.) (Before unnesting)

Splitting the `cast` column and making Pandas series as list

```
cast_list = netflix["cast"].apply(lambda
x:str(x).split(",")).to_list()

netflix_cast=pd.DataFrame(cast_list,index=netflix['title'])

type(netflix_cast)

pandas.core.frame.DataFrame

netflix_cast = netflix_cast.stack()
```

```
type(netflix_cast)
```

```
pandas.core.series.Series
```

If dataframe is single level, Stack return series as output If dataframe is multi level, Stack returns dataframe with one less level of multi-indexes

converting the netflix_new series to dataframe

```
netflix_cast = pd.DataFrame(netflix_cast)

netflix_cast
```

```
                                                 0
title
Dick Johnson Is Dead 0                         nan
Blood & Water        0                  Ama Qamata
                     1                 Khosi Ngema
                     2               Gail Mabalane
                     3              Thabang Molaba
...                                           ...
Zubaan               3             Manish Chaudhary
                     4                 Meghna Malik
                     5                Malkeet Rauni
                     6               Anita Shabdish
                     7   Chittaranjan Tripathy

[64951 rows x 1 columns]
```

resetting the index to numeric values

```
netflix_cast.reset_index(inplace = True)

netflix_cast
```

```
                     title  level_1                          0
0      Dick Johnson Is Dead        0                        nan
1             Blood & Water        0                 Ama Qamata
2             Blood & Water        1                Khosi Ngema
3             Blood & Water        2              Gail Mabalane
4             Blood & Water        3             Thabang Molaba
...                     ...      ...                        ...
64946               Zubaan        3            Manish Chaudhary
64947               Zubaan        4                Meghna Malik
64948               Zubaan        5               Malkeet Rauni
64949               Zubaan        6              Anita Shabdish
64950               Zubaan        7       Chittaranjan Tripathy

[64951 rows x 3 columns]
```

deleting `level_1` column

```
netflix_cast=netflix_cast[['title',0]]

netflix_cast
```

```
                      title                         0
0      Dick Johnson Is Dead                       nan
1             Blood & Water               Ama Qamata
2             Blood & Water              Khosi Ngema
3             Blood & Water            Gail Mabalane
4             Blood & Water           Thabang Molaba
...                     ...                       ...
64946               Zubaan          Manish Chaudhary
64947               Zubaan              Meghna Malik
64948               Zubaan             Malkeet Rauni
64949               Zubaan            Anita Shabdish
64950               Zubaan     Chittaranjan Tripathy

[64951 rows x 2 columns]
```

```
netflix_cast.rename(columns = {0:"cast"},inplace = True)
```

```
/usr/local/lib/python3.8/dist-packages/pandas/core/frame.py:5039:
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
  return super().rename(
```

```
netflix_cast = netflix_cast.explode("cast")

netflix_cast.columns
```

```
Index(['title', 'cast'], dtype='object')
```

```
netflix_cast.isnull().sum()
```

```
title    0
cast     0
dtype: int64
```

```
netflix_cast.shape
```

```
(64951, 2)
```

`cast` column from netflix dataset has taken out and unnested. It is finally stored in netflix_cast variable, which consists of 64951 row and ["title" and "cast"] columns with no null values

Similar code will be implemented on `director, country, listed_in,`

Unnesting of `director, country, listed_in`

```
#column split and changing to list
director_list = netflix["director"].apply(lambda
x:str(x).split(",")).tolist()
country_list = netflix["country"].apply(lambda
x:str(x).split(",")).tolist()
listed_in_list = netflix["listed_in"].apply(lambda
x:str(x).split(",")).tolist()

#changing list to dataframe
netflix_director=pd.DataFrame(director_list,index=netflix['title'])
netflix_country=pd.DataFrame(country_list,index=netflix['title'])
netflix_listed_in=pd.DataFrame(listed_in_list,index=netflix['title'])

#stacking the nested list
netflix_director = netflix_director.stack()
netflix_country = netflix_country.stack()
netflix_listed_in = netflix_listed_in.stack()

#converting series to dataframe
netflix_director = pd.DataFrame(netflix_director)
netflix_country = pd.DataFrame(netflix_country)
netflix_listed_in = pd.DataFrame(netflix_listed_in)

#resetting the index to numericals
netflix_director.reset_index(inplace = True)
netflix_country.reset_index(inplace = True)
netflix_listed_in.reset_index(inplace = True)

#removing the unwanted level_1 column
netflix_director=netflix_director[['title',0]]
netflix_country=netflix_country[['title',0]]
netflix_listed_in=netflix_listed_in[['title',0]]

#changing the attribute name
netflix_director.rename(columns = {0:"director"},inplace = True)
netflix_country.rename(columns = {0:"country"},inplace = True)
netflix_listed_in.rename(columns = {0:"listed_in"},inplace = True)

#explode the column
netflix_director= netflix_director.explode("director")
netflix_country = netflix_country.explode("country")
netflix_listed_in = netflix_listed_in.explode("listed_in")
```

> Four new tables are created successfully: `netflix_cast, netflix_director, netflix_country, netflix_listed_in`. All the table contains `Title` column which act as `foreign key` while merging the tables

Shape of the all the tables

```
print(netflix.shape)
print(netflix_cast.shape)
print(netflix_director.shape)
print(netflix_country.shape)
print(netflix_listed_in.shape)
```

```
(8807, 11)
(64951, 2)
(9612, 2)
(10850, 2)
(19323, 2)
```

Merging the unnested tables with main dataset

```
# Merging the four tables one after another
netflix_merge = netflix_director.merge(netflix[['show_id', 'type',
'title', 'date_added', 'release_year', 'rating', 'duration']],how =
"inner", on = "title")
netflix_merge = netflix_country.merge(netflix_merge,how="inner",on =
"title")
netflix_merge = netflix_listed_in.merge(netflix_merge,how="inner",on =
"title")
netflix_merge = netflix_cast.merge(netflix_merge,how="inner",on =
"title")

# Changing the order of attributes as initial table
netflix_merge =
netflix_merge[["show_id","type","title","director","cast","country","d
ate_added","release_year","rating","duration","listed_in"]]

netflix_merge
```

```
       show_id     type                  title         director  \
0           s1    Movie  Dick Johnson Is Dead  Kirsten Johnson
1           s2  TV Show         Blood & Water              nan
2           s2  TV Show         Blood & Water              nan
3           s2  TV Show         Blood & Water              nan
4           s2  TV Show         Blood & Water              nan
...        ...      ...                   ...              ...
202060   s8807    Movie                Zubaan      Mozez Singh
202061   s8807    Movie                Zubaan      Mozez Singh
202062   s8807    Movie                Zubaan      Mozez Singh
202063   s8807    Movie                Zubaan      Mozez Singh
202064   s8807    Movie                Zubaan      Mozez Singh

                         cast        country          date_added  \
0                         nan  United States  September 25, 2021
1                 Ama Qamata   South Africa  September 24, 2021
2                 Ama Qamata   South Africa  September 24, 2021
3                 Ama Qamata   South Africa  September 24, 2021
```

```
4                  Khosi Ngema   South Africa  September 24, 2021
...                        ...            ...                 ...
202060          Anita Shabdish          India       March 2, 2019
202061          Anita Shabdish          India       March 2, 2019
202062   Chittaranjan Tripathy          India       March 2, 2019
202063   Chittaranjan Tripathy          India       March 2, 2019
202064   Chittaranjan Tripathy          India       March 2, 2019

        release_year rating   duration               listed_in
0               2020  PG-13     90 min           Documentaries
1               2021  TV-MA  2 Seasons  International TV Shows
2               2021  TV-MA  2 Seasons              TV Dramas
3               2021  TV-MA  2 Seasons            TV Mysteries
4               2021  TV-MA  2 Seasons  International TV Shows
...              ...    ...        ...                    ...
202060          2015  TV-14    111 min    International Movies
202061          2015  TV-14    111 min        Music & Musicals
202062          2015  TV-14    111 min                 Dramas
202063          2015  TV-14    111 min    International Movies
202064          2015  TV-14    111 min        Music & Musicals

[202065 rows x 11 columns]
```

Deletion of any duplicated rows

```
netflix_merge.duplicated().sum()

7
```

There are 7 duplicate rows

```
netflix_merge[netflix_merge.duplicated()]

       show_id   type                title        director  \
39354    s1632  Movie           Rust Creek     Jen McGowan
135656   s6014  Movie  300 Miles to Heaven  Maciej Dejczer
135657   s6014  Movie  300 Miles to Heaven  Maciej Dejczer
135658   s6014  Movie  300 Miles to Heaven  Maciej Dejczer
135659   s6014  Movie  300 Miles to Heaven  Maciej Dejczer
135660   s6014  Movie  300 Miles to Heaven  Maciej Dejczer
135661   s6014  Movie  300 Miles to Heaven  Maciej Dejczer

                          cast        country         date_added
release_year  \
39354          Micah Hauptman  United States  November 30, 2020
2018
135656   Adrianna Biedrzyńska        Denmark    October 1, 2019
1989
135657   Adrianna Biedrzyńska         France    October 1, 2019
1989
```

```
135658   Adrianna Biedrzyńska          Poland    October 1, 2019
1989
135659   Adrianna Biedrzyńska         Denmark    October 1, 2019
1989
135660   Adrianna Biedrzyńska          France    October 1, 2019
1989
135661   Adrianna Biedrzyńska          Poland    October 1, 2019
1989
```

```
       rating duration              listed_in
39354       R  108 min              Thrillers
135656  TV-14   93 min                 Dramas
135657  TV-14   93 min                 Dramas
135658  TV-14   93 min                 Dramas
135659  TV-14   93 min   International Movies
135660  TV-14   93 min   International Movies
135661  TV-14   93 min   International Movies
```

```
netflix_merge.drop_duplicates(keep = "first",inplace = True)
```

Separating duration column into two columns: duration of TV Shows, duration of Movies

```
netflix_tv_show = netflix_merge[netflix_merge["type"] == "TV Show"]
netflix_movie = netflix_merge[netflix_merge["type"] == "Movie"]
netflix_merge['duration_tv_show']=netflix_tv_show.duration.str.extract
('(^\d*)')
netflix_merge['duration_movie']=netflix_movie.duration.str.extract('(^
\d*)')
netflix_merge.drop(["duration"],axis = 1, inplace = True)

netflix_merge.head()
```

```
  show_id     type                  title         director
cast  \
0      s1    Movie  Dick Johnson Is Dead  Kirsten Johnson
nan
1      s2  TV Show         Blood & Water              nan    Ama
Qamata
2      s2  TV Show         Blood & Water              nan    Ama
Qamata
3      s2  TV Show         Blood & Water              nan    Ama
Qamata
4      s2  TV Show         Blood & Water              nan   Khosi
Ngema

          country          date_added  release_year rating  \
0  United States  September 25, 2021          2020  PG-13
1   South Africa  September 24, 2021          2021  TV-MA
2   South Africa  September 24, 2021          2021  TV-MA
```

```
3   South Africa  September 24, 2021          2021  TV-MA
4   South Africa  September 24, 2021          2021  TV-MA

                 listed_in duration_tv_show duration_movie
0           Documentaries               NaN             90
1  International TV Shows                 2            NaN
2              TV Dramas                  2            NaN
3            TV Mysteries                 2            NaN
4  International TV Shows                 2            NaN
```

CHAPTER 2: OBSERVATIONS ON THE SHAPE OF THE DATA, DATA TYPES OF ALL THE ATTRIBUTES, CONVERSION OF CATEGORICAL ATTRIBUTES TO "CATEGORY" (IF REQUIRED), MISSING VALUE DETECTION, STATISTICAL SUMMARY

2.1 SHAPE OF DATA

```
netflix_merge.shape
```

```
(202058, 12)
```

There are 202058 Rows and 12 Columns in netflix_merge table after unnesting, merging, deleting duplicates and separating duration column into two columns

2.2 DATA TYPES OF ALL THE ATTRIBUTES

```
netflix_merge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 202058 entries, 0 to 202064
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   show_id           202058 non-null  object
 1   type              202058 non-null  object
 2   title             202058 non-null  object
 3   director          202058 non-null  object
 4   cast              202058 non-null  object
 5   country           202058 non-null  object
 6   date_added        201900 non-null  object
```

```
 7   release_year      202058 non-null  int64
 8   rating            201991 non-null  object
 9   listed_in         202058 non-null  object
 10  duration_tv_show  56148 non-null   object
 11  duration_movie    145907 non-null  object
dtypes: int64(1), object(11)
memory usage: 20.0+ MB
```

Except `release_year` remaining all attributes are `object data type`. `release_year` is `int64 data type`.

Null values are present in `date_added, rating and duration_tv_show, duration_movie` columns, As these columns are not modified into string format.

2.3 CONVERSION OF CATEGORICAL ATTRIBUTES TO "CATEGORY" (IF REQUIRED)

Conversion of `object` to `category` data type

Unique counts - to calculate which columns can be converted to `category` data type using `astype()`

```
unique_counts = pd.DataFrame.from_records([(col,
netflix_merge[col].nunique()) for col in netflix_merge.columns],
                          columns=['Column_Name',
'Num_Unique']).sort_values(by=['Num_Unique'])
unique_counts
```

	Column_Name	Num_Unique
1	type	2
10	duration_tv_show	15
8	rating	17
9	listed_in	73
7	release_year	74
5	country	198
11	duration_movie	205
6	date_added	1767
3	director	5121
0	show_id	8807
2	title	8807
4	cast	39297

Less than 200 unique counts columns can be converted to `category` data type.

`type,rating,listed_in,release_year,country` columns will be converted to category data type.

Categorical attributes occupy `less memory and helpful in Data visualization`.

`data_added` must not be converted to `category` data type. It should be converted to `datetime` data type.

`duration_tv_show and duration_movie` data type will be converted to `int64` as it is numerical attribute.

Conversion of datatypes to `category` using loops

```
#for col in netflix_merge.columns:
#    if (netflix_merge[col].nunique() < 200):
#        netflix_merge[col] = netflix_merge[col].astype('category')

for col in netflix_merge.columns:
    if (col not in ["duration_tv_show","duration_movie"])and
(netflix_merge[col].nunique() < 200):
        netflix_merge[col] = netflix_merge[col].astype('category')
```

Conversion of `duration_tv_show and duration_movie` columns to `float` data type

```
netflix_merge[["duration_tv_show","duration_movie"]] =
netflix_merge[["duration_tv_show","duration_movie"]].astype("float")
```

Conversion `date_added` to `datetime` data type

`date_added` column is in `object` data type. It can be converted to `datetime` data type

```
netflix_merge["date_added"] =
pd.to_datetime(netflix_merge["date_added"])

netflix_merge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 202058 entries, 0 to 202064
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   show_id           202058 non-null  object
 1   type              202058 non-null  category
 2   title             202058 non-null  object
 3   director          202058 non-null  object
 4   cast              202058 non-null  object
 5   country           202058 non-null  category
 6   date_added        201900 non-null  datetime64[ns]
 7   release_year      202058 non-null  category
 8   rating            201991 non-null  category
 9   listed_in         202058 non-null  category
 10  duration_tv_show  56148 non-null   float64
 11  duration_movie    145907 non-null  float64
```

```
dtypes: category(5), datetime64[ns](1), float64(2), object(4)
memory usage: 13.5+ MB
```

`date_added` converted to `datetime64[ns]` data type.

2.4 MISSING VALUE DETECTION

Detection of Null values in netflix_merge Table

```
netflix_merge.isnull().sum()
```

```
show_id                  0
type                     0
title                    0
director                 0
cast                     0
country                  0
date_added             158
release_year             0
rating                  67
listed_in                0
duration_tv_show    145910
duration_movie       56151
dtype: int64
```

```
netflix_merge["director"].head(10)
```

```
0    Kirsten Johnson
1                nan
2                nan
3                nan
4                nan
5                nan
6                nan
7                nan
8                nan
9                nan
Name: director, dtype: object
```

```
type(netflix_merge["director"][1])
```

```
str
```

null values like None or NaN are not shown in

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

but nan values are present in other columns such as "director". Those nan values are in string format. We have to deal hidden "nan" values too.

```
netflix_merge.replace("nan",np.nan,inplace = True)
netflix_merge.replace("NaN",np.nan,inplace = True)
```

Total Missing values in the merged table considering both NaN and Str(NaN)

```
netflix_merge.isna().sum()
```

```
show_id                  0
type                     0
title                    0
director             50643
cast                  2149
country              11897
date_added             158
release_year             0
rating                  67
listed_in                0
duration_tv_show    145910
duration_movie       56151
dtype: int64
```

By replace string format "NaN" values with np.nan, netflix_merge table consists of `50643 director null values, 2149 cast null values, 11897 country null values, 158 date_added null values, 67 rating null values, 145910 duration_tv_show null values and 56151 duration_movie null values`.

`show_id, type, title, release_year, listed_in` columns do not have any null values.

2.5 STATISTICAL SUMMARY

Statistical summary after unnesting and merging

```
netflix_merge.describe(include = "all")
```

```
<ipython-input-619-1e484ca3d2d5>:1: FutureWarning: Treating datetime
data as categorical rather than numeric in `.describe` is deprecated
and will be removed in a future version of pandas. Specify
`datetime_is_numeric=True` to silence this warning and adopt the
future behavior now.
  netflix_merge.describe(include = "all")
```

```
       show_id    type                          title
director  \
count   202058  202058                         202058           151415

unique    8807       2                           8807             5120
```

```
top      s7165   Movie  Kahlil Gibran's The Prophet  Martin Scorsese
freq       700  145910                          700              419
first      NaN     NaN                          NaN              NaN
last       NaN     NaN                          NaN              NaN
mean       NaN     NaN                          NaN              NaN
std        NaN     NaN                          NaN              NaN
min        NaN     NaN                          NaN              NaN
25%        NaN     NaN                          NaN              NaN
50%        NaN     NaN                          NaN              NaN
75%        NaN     NaN                          NaN              NaN
max        NaN     NaN                          NaN              NaN


                    cast        country           date_added
release_year  \
count             199909         190161               201900
202058.0
unique             39296            197                 1714
74.0
top       Alfred Molina  United States  2020-01-01 00:00:00
2018.0
freq                 160          49867                 3748
24440.0
first                NaN            NaN  2008-01-01 00:00:00
NaN
last                 NaN            NaN  2021-09-25 00:00:00
NaN
mean                 NaN            NaN                  NaN
NaN
std                  NaN            NaN                  NaN
NaN
min                  NaN            NaN                  NaN
NaN
25%                  NaN            NaN                  NaN
NaN
50%                  NaN            NaN                  NaN
NaN
75%                  NaN            NaN                  NaN
NaN
max                  NaN            NaN                  NaN
```

```
NaN

        rating               listed_in  duration_tv_show
duration_movie
count   201991                  202058      56148.000000
145907.000000
unique      17                      73               NaN
NaN
top      TV-MA   International Movies                NaN
NaN
freq     73915                   27138               NaN
NaN
first      NaN                     NaN               NaN
NaN
last       NaN                     NaN               NaN
NaN
mean       NaN                     NaN          1.928101
106.840947
std        NaN                     NaN          1.811729
24.709828
min        NaN                     NaN          1.000000
3.000000
25%        NaN                     NaN          1.000000
93.000000
50%        NaN                     NaN          1.000000
104.000000
75%        NaN                     NaN          2.000000
119.000000
max        NaN                     NaN         17.000000
312.000000
```

New Statistical summary points regarding netflix.merge table

1. `Kahilil Gibran's The Prophet` is highly repeated element (700 times) in `title` column
2. `Martin Scorsese` is highly repeated element (419 times) in `director` column
3. `Alfred Molina` is highly repeated element (160 times) in `cast` column
4. `United States` is highly repeated element (49867 times) in `country` column
5. `2018` is highly repeated element (24440 times) in `country` column
6. `TV-MA` is highly repeated element (73915 times) in `rating` column
7. `1 Season` is highly repeated element (35035 times) in `duration` column
8. `Internation Movies` is highly repeated element (27138 times) in `listed_In` column
9. Starting date of this dataset is `2008-01-01` and Ending date of this dataset is `2021-09-25`
10. Maximum number of seasons for a tv show is `17` seasons. Mean = `1.928` seasons
11. Highest duration for a movie is `312` minutes. lowest duration is `3` min. Mean = `106.84` minutes

CHAPTER-3: NON-GRAPHICAL ANALYSIS: VALUE COUNTS AND UNIQUE ATTRIBUTES

3.1 VALUE COUNTS

Value counts of all columns in netflix_merge table

```
for col in netflix_merge.columns:
    print('-' * 40 + col + '-' * 40 , end=' - ')
    display(netflix_merge[col].value_counts(dropna = True))
```

```
----------------------------------------
show_id---------------------------------------- -

s7165    700
s6985    504
s7516    468
s2554    416
s5306    378
        ...
s6330      1
s8176      1
s937       1
s3387      1
s1         1
Name: show_id, Length: 8807, dtype: int64

----------------------------------------
type---------------------------------------- -

Movie      145910
TV Show     56148
Name: type, dtype: int64

----------------------------------------
title---------------------------------------- -

Kahlil Gibran's The Prophet    700
Holidays                       504
Movie 43                       468
The Eddy                       416
Narcos                         378
                              ...
Blackfish                        1
The 2000s                        1
Miniforce: Super Dino Power      1
Dancing with the Birds           1
Dick Johnson Is Dead             1
Name: title, Length: 8807, dtype: int64
```

```
----------------------------------------
director---------------------------------------- -

Martin Scorsese         419
Youssef Chahine         409
Cathy Garcia-Molina     356
Steven Spielberg        355
Lars von Trier          336
                       ...
Sue Kim                   1
Bryce Wagoner             1
Brittany Andrews          1
Ariel Boles               1
Kirsten Johnson           1
Name: director, Length: 5120, dtype: int64

----------------------------------------
cast---------------------------------------- -

 Alfred Molina               160
 Salma Hayek                 130
 Frank Langella              128
 John Rhys-Davies            125
 John Krasinski              121
                             ...
Bukky Bakray                   1
 Kosar Ali                     1
 D'angelou Osei Kissiedu       1
 Jill Hennessy                 1
Charlie Shotwell               1
Name: cast, Length: 39296, dtype: int64

----------------------------------------
country---------------------------------------- -

United States     49867
India             22139
United Kingdom     9733
 United States     9482
Japan              7317
                  ...
 Sri Lanka            2
 Afghanistan          2
 Nicaragua            1
 Uganda               1
 Kazakhstan           1
Name: country, Length: 197, dtype: int64

----------------------------------------
date_added---------------------------------------- -
```

```
2020-01-01    3748
2019-11-01    2258
2021-07-01    2219
2017-10-01    1899
2021-09-01    1756
              ...
2014-11-14       1
2017-01-24       1
2020-11-18       1
2017-01-23       1
2021-09-25       1
Name: date_added, Length: 1714, dtype: int64

----------------------------------------
release_year----------------------------------------- -

2018    24440
2019    21931
2017    20516
2020    19697
2016    18465
        ...
1947        8
1946        6
1942        6
1943        5
1925        1
Name: release_year, Length: 74, dtype: int64

----------------------------------------
rating----------------------------------------- -

TV-MA       73915
TV-14       43951
R           25859
PG-13       16246
TV-PG       14926
PG          10919
TV-Y7        6304
TV-Y         3665
TV-G         2779
NR           1573
G            1530
NC-17         149
TV-Y7-FV       86
UR             86
74 min          1
84 min          1
66 min          1
Name: rating, dtype: int64
```

```
----------------------------------------
listed_in------------------------------------------ -

 International Movies    27138
Dramas                   19654
Comedies                 13894
Action & Adventure       12216
 Dramas                  10149
                         ...
 Stand-Up Comedy            24
Romantic Movies             20
TV Sci-Fi & Fantasy          7
LGBTQ Movies                 5
Sports Movies                3
Name: listed_in, Length: 73, dtype: int64

----------------------------------------
duration_tv_show----------------------------------------- -

1.0     35035
2.0      9559
3.0      5084
4.0      2134
5.0      1698
7.0       843
6.0       633
8.0       286
9.0       257
10.0      220
13.0      132
12.0      111
15.0       96
17.0       30
11.0       30
Name: duration_tv_show, dtype: int64

----------------------------------------
duration_movie------------------------------------------ -

94.0     4343
106.0    4040
97.0     3624
95.0     3560
96.0     3511
         ...
20.0        4
5.0         3
9.0         2
8.0         2
```

```
11.0        2
Name: duration_movie, Length: 205, dtype: int64
```

```
netflix_merge["rating"].value_counts()
```

```
TV-MA       73915
TV-14       43951
R           25859
PG-13       16246
TV-PG       14926
PG          10919
TV-Y7        6304
TV-Y         3665
TV-G         2779
NR           1573
G            1530
NC-17         149
TV-Y7-FV       86
UR             86
74 min          1
84 min          1
66 min          1
Name: rating, dtype: int64
```

`rating` column consists three irrelevant value counts. Those are `74 min, 84 min and 66 min`.

These values should be transfered to respective row `duration_movie` column.

Handling data typing error in `rating` column

```
netflix_merge[netflix_merge["rating"].isin(["74 min","84 min","66 min"])]
```

```
       show_id   type                                    title director  \
126582   s5542  Movie                      Louis C.K. 2017  Louis C.K.
131648   s5795  Movie                Louis C.K.: Hilarious  Louis C.K.
131782   s5814  Movie  Louis C.K.: Live at the Comedy Store  Louis C.K.

                cast        country date_added release_year  rating listed_in  \
126582  Louis C.K.  United States 2017-04-04         2017  74 min  Movies
131648  Louis C.K.  United States 2016-09-16         2010  84 min  Movies
131782  Louis C.K.  United States 2016-08-15         2015  66 min  Movies
```

```
        duration_tv_show  duration_movie
126582               NaN             NaN
131648               NaN             NaN
131782               NaN             NaN
```

`duration` column has NaN values in above table.

exchanging the values between `rating` and `duration_movie` columns in above three rows

```
Idx = netflix_merge[netflix_merge["rating"].isin(["74 min","84
min","66 min"])].index
netflix_merge.loc[Idx,"duration_movie"] = [74,84,66]
netflix_merge.loc[Idx,"rating"] = np.nan

netflix_merge.loc[Idx,:]
```

```
       show_id   type                                    title
director  \
126582   s5542  Movie                       Louis C.K. 2017  Louis
C.K.
131648   s5795  Movie                 Louis C.K.: Hilarious  Louis
C.K.
131782   s5814  Movie  Louis C.K.: Live at the Comedy Store  Louis
C.K.

              cast        country date_added release_year rating
listed_in  \
126582  Louis C.K.  United States 2017-04-04         2017    NaN
Movies
131648  Louis C.K.  United States 2016-09-16         2010    NaN
Movies
131782  Louis C.K.  United States 2016-08-15         2015    NaN
Movies

        duration_tv_show  duration_movie
126582               NaN            74.0
131648               NaN            84.0
131782               NaN            66.0
```

```
netflix_merge.dtypes
```

```
show_id                     object
type                      category
title                       object
director                    object
cast                        object
country                   category
date_added          datetime64[ns]
release_year              category
```

```
rating                    category
listed_in                 category
duration_tv_show           float64
duration_movie             float64
dtype: object
```

```
netflix_merge["rating"].value_counts()
```

```
TV-MA       73915
TV-14       43951
R           25859
PG-13       16246
TV-PG       14926
PG          10919
TV-Y7        6304
TV-Y         3665
TV-G         2779
NR           1573
G            1530
NC-17         149
TV-Y7-FV       86
UR             86
74 min          0
84 min          0
66 min          0
Name: rating, dtype: int64
```

Typing error has rectified as `74 min, 84 min, 66 min` count = 0

3.2 UNIQUE ATTRIBUTES

Number of unique values in each column

```
netflix_merge.nunique()
```

```
show_id              8807
type                    2
title                8807
director             5120
cast                39296
country               197
date_added           1714
release_year           74
rating                 14
listed_in              73
duration_tv_show       15
duration_movie        205
dtype: int64
```

There are `8807 unique movies/tvshows, 5121 unique directors, 39297 unique actors/actress, 198 unique countries, 17 unique rating categories and 73 unique genres`(listed_in) are involved in this `NETFLIX Inc`.

Number of unique values grouped by each column listed in dataset with reference to `title` column

```
for col in netflix_merge.columns:
    print('-' * 40 + col + '-' * 40 , end=' - ')
    display(netflix_merge.groupby(col)
["title"].nunique().sort_values())
```

```
----------------------------------------
show_id---------------------------------------- -
```

```
show_id
s1       1
s6290    1
s629     1
s6289    1
s6288    1
        ..
s3641    1
s3642    1
s3643    1
s3638    1
s999     1
Name: title, Length: 8807, dtype: int64
```

```
----------------------------------------
type---------------------------------------- -
```

```
type
TV Show    2676
Movie      6131
Name: title, dtype: int64
```

```
----------------------------------------
title---------------------------------------- -
```

```
title
#Alive                        1
Rishta.com                    1
Rise: Ini Kalilah             1
Rise of the Zombie            1
Rise of Empires: Ottoman      1
                            ..
Happy Times                   1
Happy Valley                  1
Happy as Lazzaro              1
```

```
Happy Hunting                1
    :       1
Name: title, Length: 8807, dtype: int64

----------------------------------------
director---------------------------------------- -

director
Joanna Lombardi     1
K. Subhash          1
K.C. Bokadia        1
K.S. Ravikumar      1
KVR Mahendra        1
                   ..
Suhas Kadav        16
Marcus Raboy       16
Raúl Campos        18
 Jan Suter         18
Rajiv Chilaka      22
Name: title, Length: 5120, dtype: int64

----------------------------------------
cast---------------------------------------- -

cast
 Marc Hayashi         1
 Paul L. Smith        1
 Paul Kwo             1
 Paul Kenny           1
 Paul Kasey           1
                     ..
 Om Puri             27
 Julie Tejwani       28
 Takahiro Sakurai    30
 Rupa Bhimani        31
 Anupam Kher         39
Name: title, Length: 39296, dtype: int64

----------------------------------------
country---------------------------------------- -

country
Zimbabwe             1
Guatemala            1
 Ethiopia            1
Greece               1
 Ghana               1
                    ...
Canada             271
 United States     479
United Kingdom     628
```

```
India             1008
United States     3211
Name: title, Length: 197, dtype: int64

----------------------------------------
date_added------------------------------------------ -

date_added
2008-01-01      1
2018-04-14      1
2018-04-18      1
2018-05-02      1
2018-05-03      1
              ...
2018-10-01     71
2019-12-31     74
2018-03-01     75
2019-11-01     91
2020-01-01    110
Name: title, Length: 1714, dtype: int64

----------------------------------------
release_year----------------------------------------- -

release_year
1925       1
1966       1
1947       1
1961       1
1959       1
         ...
2016     902
2020     953
2019    1030
2017    1032
2018    1147
Name: title, Length: 74, dtype: int64

----------------------------------------
rating----------------------------------------- -

rating
66 min          0
74 min          0
84 min          0
NC-17           3
UR              3
TV-Y7-FV        6
G              41
NR             80
TV-G          220
```

```
PG            287
TV-Y          307
TV-Y7         334
PG-13         490
R             799
TV-PG         863
TV-14        2160
TV-MA        3207
Name: title, dtype: int64

----------------------------------------
listed_in---------------------------------------- -

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

----------------------------------------
duration_tv_show---------------------------------------- -

duration_tv_show
17.0        1
11.0        2
12.0        2
15.0        2
13.0        3
10.0        7
9.0         9
8.0        17
7.0        23
6.0        33
5.0        65
4.0        95
3.0       199
2.0       425
1.0      1793
Name: title, dtype: int64

----------------------------------------
duration_movie---------------------------------------- -
```

```
duration_movie
3.0        1
167.0      1
178.0      1
186.0      1
189.0      1
         ...
91.0     144
93.0     146
94.0     146
97.0     146
90.0     152
Name: title, Length: 205, dtype: int64
```

Observations from above result

1. `6131 Movies` and `2676 TV Show` present in the data set. Total `8807 unique fields`
2. `Rajiv Chilaka` has directed `22 Movies/TV Shows` stands in first position. `Jan Suter and Raul Campos` has directed `18 Movies/TV Shows` stands in second position.
3. `Anupam Kher` has listed in `39 Different Movies/TV Shows` stands in first position. `Rupa Bhimani` has listed in `31 Different Movies/TV Shows` stands in second position.
4. `3211` Movies/TVShows are released/produced in `United States` stands in first position. `1008` Movies/TVShows are released/produced in `India` stands in second position.
5. ON `2020-01-01`, `110` Movies/TV shows added into netflix servers which stands in first position.
6. During `2018`, `1147` Movies/TV shows released which stands first position.
7. `TV-MA` rating category has `3207` Movies/TV Shows which stands first position. `TV-14` rating category has `2160` Movies/TV Shows which stands second position.
8. Among `2676` TV Shows, `1793` shows are having only 1 season. `425` Shows are having 2 seasons. `199` Shows are having 3 seasons.
9. Among `6131` Movies, `152` Movies has the duration of `90 min` which stands in first position.
10. `2624` Movie/TV shows listed in `International Movies` Genre which stands in first position. `1600` Movie/TV shows listed in `Dramas` Genre which stands in second position. `1210` Movie/TV shows listed in `Comedies` Genre which stands in third position.

3.3 OTHER EXPLORATORY DATA ANALYSIS ON MODIFIED DATA

```
netflix_merge.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country',
'date_added',
       'release_year', 'rating', 'listed_in', 'duration_tv_show',
       'duration_movie'],
      dtype='object')
```

Comparison of tv shows vs movies

```
netflix_merge.groupby("type")
["title","director","cast","country","listed_in"].nunique()
```

```
<ipython-input-630-1b28ad370513>:1: FutureWarning: Indexing with
multiple keys (implicitly converted to a tuple of keys) will be
deprecated, use a list instead.
  netflix_merge.groupby("type")
["title","director","cast","country","listed_in"].nunique()
```

type	title	director	cast	country	listed_in
Movie	6131	4886	27879	187	37
TV Show	2676	300	15501	102	36

Above analysis per movie/tv show

```
netflix_merge.groupby("type")
["title","director","cast","country","listed_in"].apply(lambda x:
x.nunique()/x["title"].nunique())
```

```
<ipython-input-631-3157f76eaa37>:1: FutureWarning: Indexing with
multiple keys (implicitly converted to a tuple of keys) will be
deprecated, use a list instead.
  netflix_merge.groupby("type")
["title","director","cast","country","listed_in"].apply(lambda x:
x.nunique()/x["title"].nunique())
```

type	title	director	cast	country	listed_in
Movie	1.0	0.796934	4.547219	0.030501	0.006035
TV Show	1.0	0.112108	5.792601	0.038117	0.013453

INSIGHTS FROM above comparison analysis:

> `TV Show` dominates the `Movies` according to number of persons are casted, number of countries released, number of genres listed_in.

`director ratio per movie`(0.7969) is higher compared to `director ration per tv show`(0.1121). This indicates that `One TV Show director directing mulitiple tv shows`. Whereas, movies require dedicated directors. `Movie directors are not directing as frequently as TV Show directors`.

Analysis of actors/directors of different types of shows/movies

```
netflix_merge["cast"]= netflix_merge["cast"].str.replace('\$\$','')
netflix_merge["director"] = netflix_merge["director"].str.replace('\$\
$',"")
```

```
<ipython-input-632-300c13f10f73>:1: FutureWarning: The default value
of regex will change from True to False in a future version.
  netflix_merge["cast"]= netflix_merge["cast"].str.replace('\$\$','')
<ipython-input-632-300c13f10f73>:2: FutureWarning: The default value
of regex will change from True to False in a future version.
  netflix_merge["director"] = netflix_merge["director"].str.replace('\
$\$',"")
```

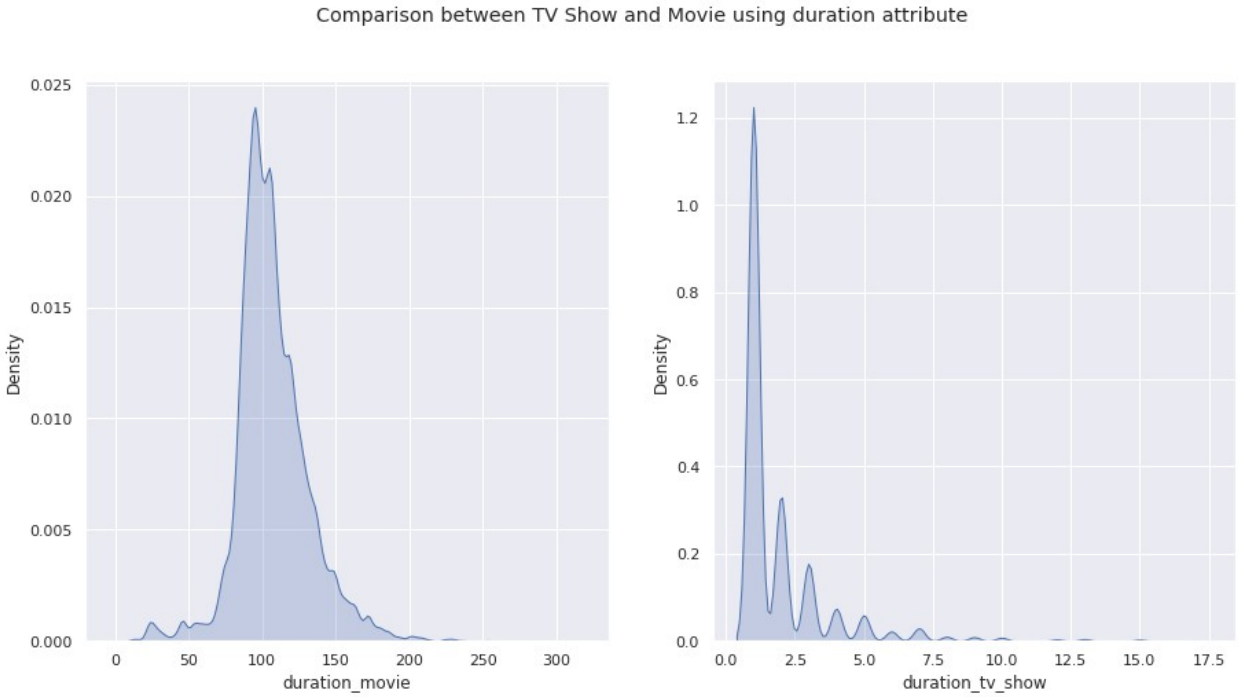
CHAPTER 4: VISUAL ANALYSIS - UNIVARIATE, BIVARIATE AFTER PRE-PROCESSING OF THE DATA

4.1 UNIVARIATE PLOTS

Comparison between TV Show and Movie duration using `kde plot`

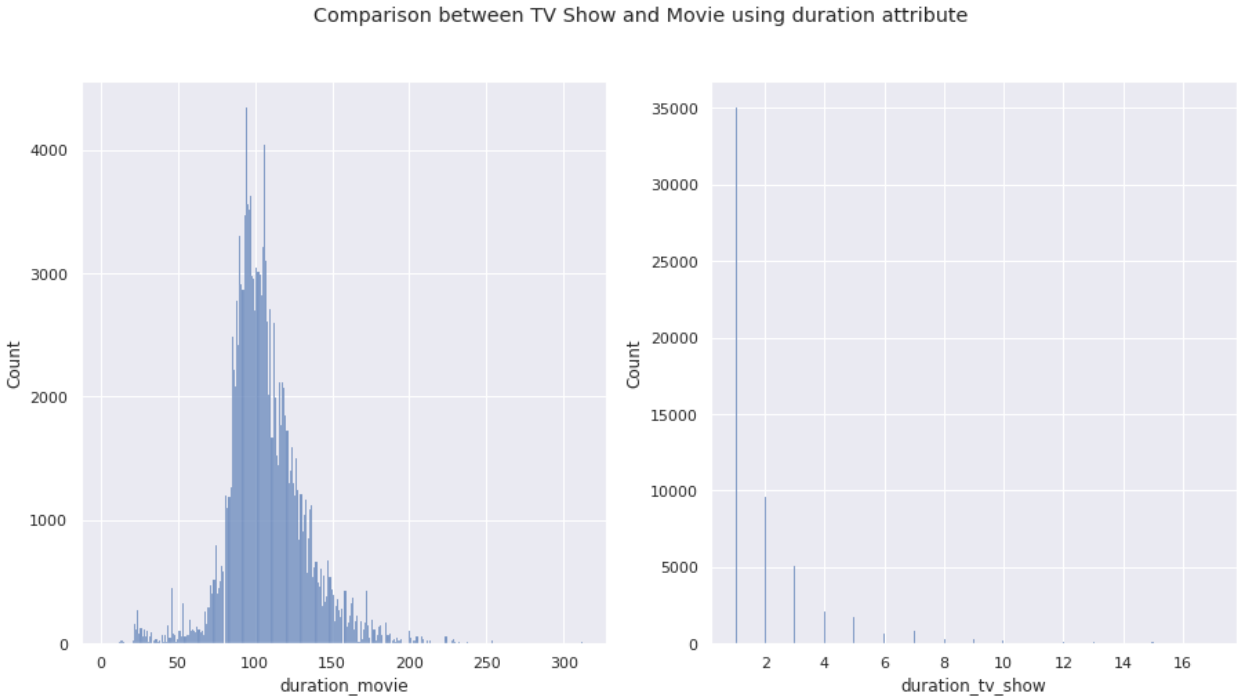
```
fig = plt.figure(figsize = (15,7.5))
fig.suptitle("Comparison between TV Show and Movie using duration
attribute")
plt.subplot(1,2,1)
sns.kdeplot(data = netflix_merge, x = "duration_movie",fill = True)
plt.subplot(1,2,2)
sns.kdeplot(data = netflix_merge, x = "duration_tv_show",fill = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff751acfbb0>
```




```
fig = plt.figure(figsize = (15,7.5))
fig.suptitle("Comparison between TV Show and Movie using duration
attribute")
plt.subplot(1,2,1)
sns.histplot(data = netflix_merge, x = "duration_movie",fill = True)
plt.subplot(1,2,2)
sns.histplot(data = netflix_merge, x = "duration_tv_show",fill = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff744f10af0>
```



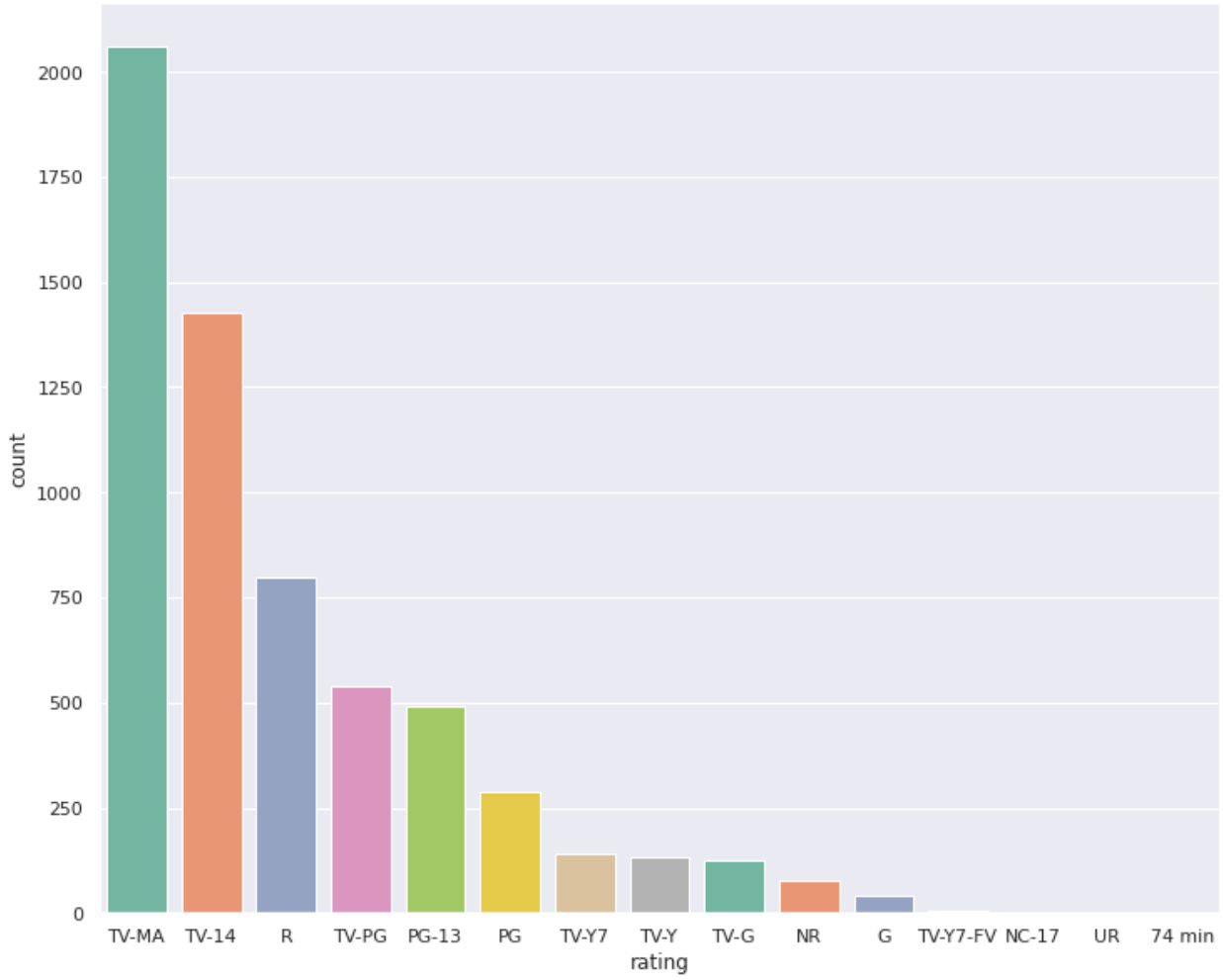
INSIGHTS:

For most of the `TV Show`, number of seasons in range of `1 to 3`

For most of the `Movies`, duration is in range of `70 to 130 min`

```
movie_netflix = netflix[netflix['type'] == 'Movie']
tv_netflix = netflix[netflix['type'] == 'TV Show']

#MOVIES RATINGS
plt.figure(figsize=(12,10))
sns.set(style="darkgrid")
sns.countplot(x="rating", data= movie_netflix, palette="Set2",
order=movie_netflix['rating'].value_counts().index[0:15])
```

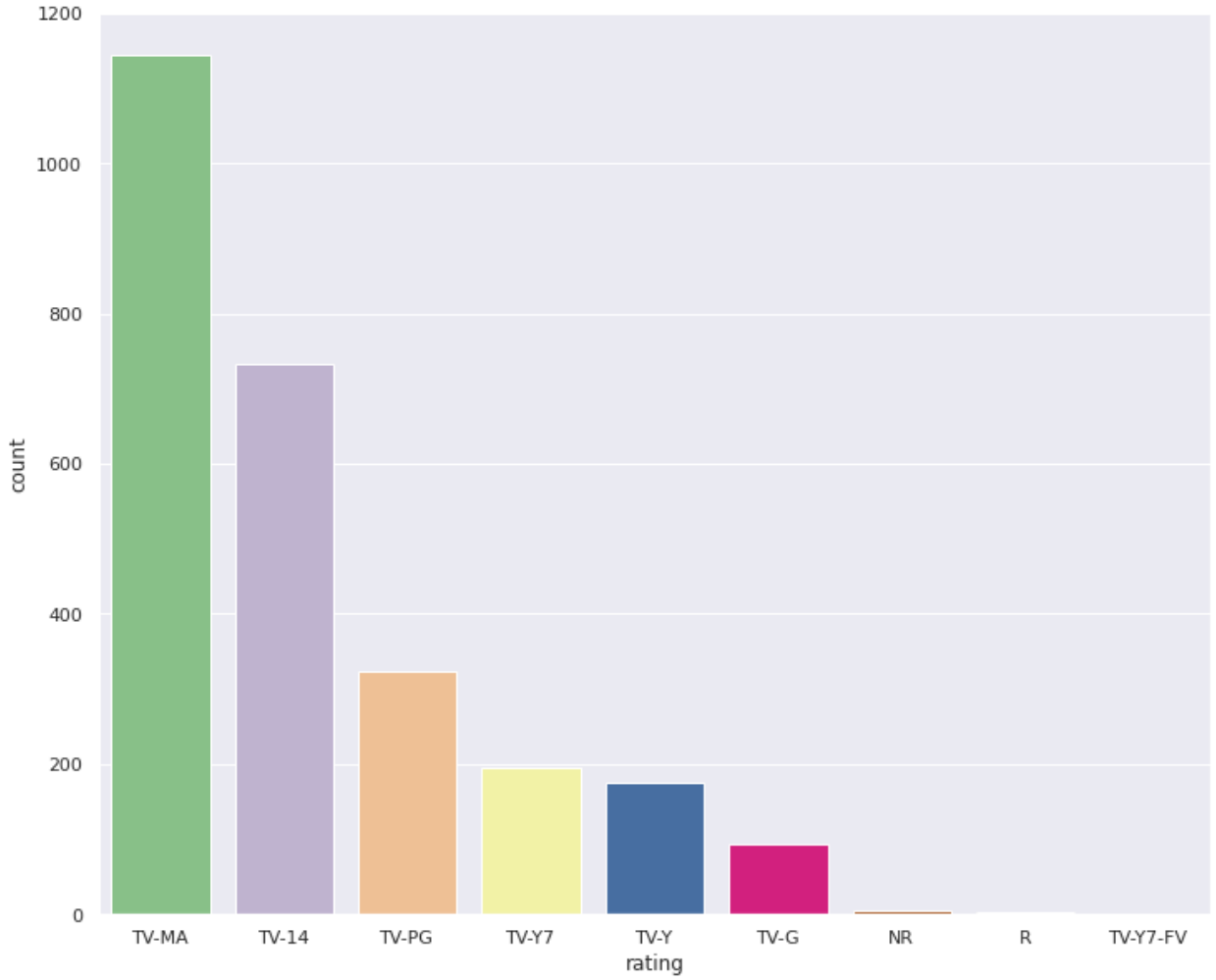
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff73f14cd90>
```



`TV-MA` Category has more movies

```
# TV SHOWS RATINGS
plt.figure(figsize=(12,10))
sns.set(style="darkgrid")
sns.countplot(x="rating", data=tv_netflix, palette="Accent",
order=tv_netflix['rating'].value_counts().index[0:15])
```

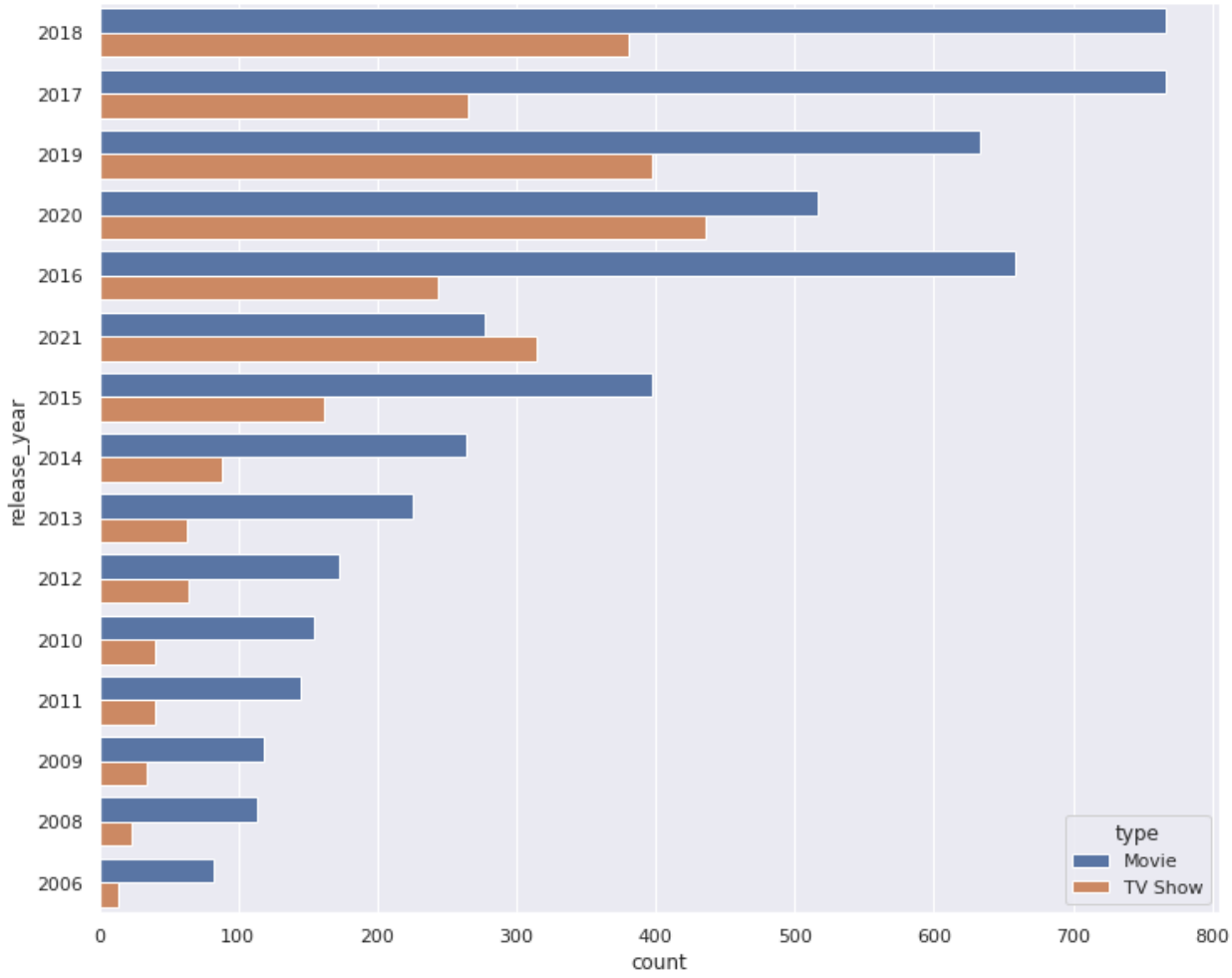
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff75279e220>
```



`TV-MA` genre has more TV Shows

```
plt.figure(figsize=(12,10))
sns.set(style="darkgrid")
sns.countplot(y="release_year", data= netflix, order=
netflix['release_year'].value_counts().index[0:15],hue=netflix['type']
)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff73e461c40>
```



During 2021, Both TV Show and Movie count reduced due to COVID.

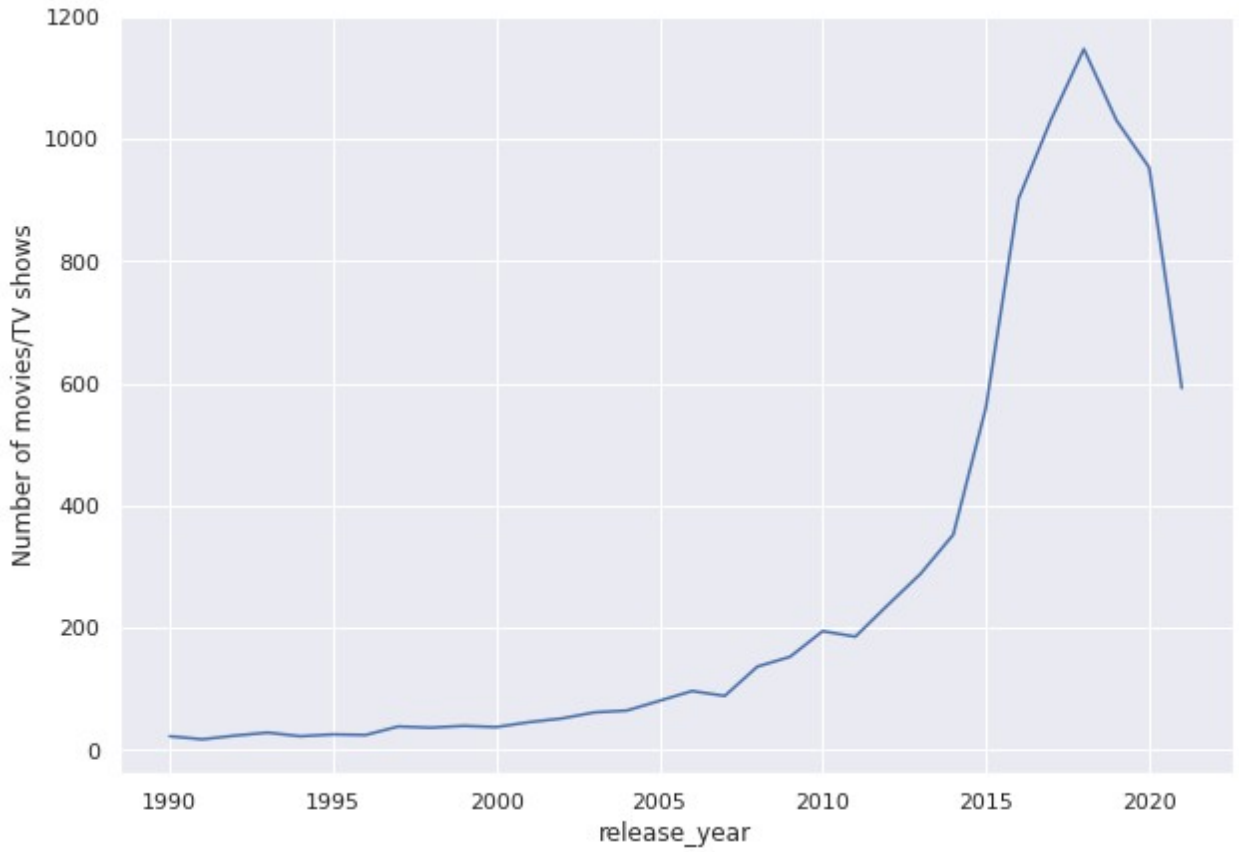
Both TV SHOW and Movie count gradually increased year by year

4.2 BIVARIATE PLOTS

Trend of number of movies/TV shows released per year over the last 20-30 years

```
#converting categoricy data type of release_year to int64 data type
arr = netflix_merge
arr[["release_year"]] = arr[["release_year"]].astype("int64")

arr = arr.groupby("release_year")
["title"].nunique().reset_index().sort_values("release_year")
arr = arr[arr["release_year"]>=1990] # from last 30 years
plt.figure(figsize = (10,7))
plt.ylabel("Number of movies/TV shows")
sns.lineplot(data = arr, x = "release_year", y = "title")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff744f64f40>
```



Number of movies/TV shows released per year is higher during `2015 to 2020` period

Box Plots on duration and other columns (statistical comparison)

```
netflix_merge.head()
```

```
  show_id     type                  title         director
cast  \
0      s1    Movie  Dick Johnson Is Dead  Kirsten Johnson
NaN
1      s2  TV Show         Blood & Water              NaN    Ama
Qamata
2      s2  TV Show         Blood & Water              NaN    Ama
Qamata
3      s2  TV Show         Blood & Water              NaN    Ama
Qamata
4      s2  TV Show         Blood & Water              NaN   Khosi
Ngema

         country date_added release_year rating
listed_in  \
```

```
0  United States 2021-09-25         2020  PG-13
Documentaries
1   South Africa 2021-09-24         2021  TV-MA  International TV
Shows
2   South Africa 2021-09-24         2021  TV-MA               TV
Dramas
3   South Africa 2021-09-24         2021  TV-MA            TV
Mysteries
4   South Africa 2021-09-24         2021  TV-MA  International TV
Shows

   duration_tv_show  duration_movie
0               NaN            90.0
1               2.0             NaN
2               2.0             NaN
3               2.0             NaN
4               2.0             NaN
```

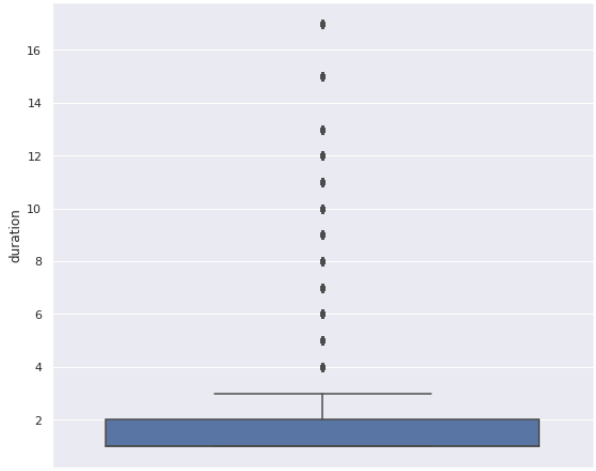
```
arr = netflix_merge
arr[["duration_tv_show","duration_movie"]] =
arr[["duration_tv_show","duration_movie"]].fillna(0)
arr["duration"] = arr["duration_tv_show"]+arr["duration_movie"]
arr
```

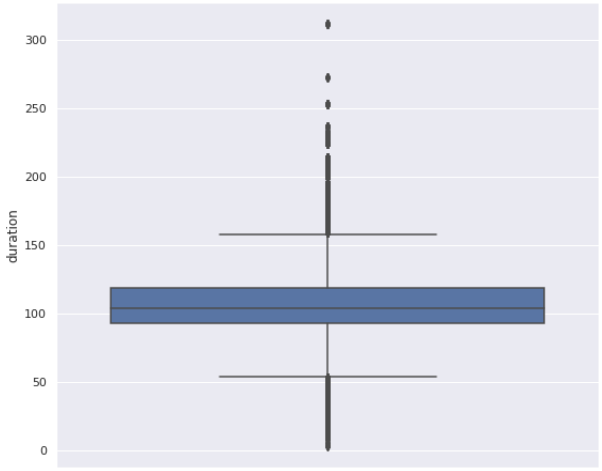
```
       show_id     type                  title         director  \
0           s1    Movie  Dick Johnson Is Dead  Kirsten Johnson
1           s2  TV Show         Blood & Water              NaN
2           s2  TV Show         Blood & Water              NaN
3           s2  TV Show         Blood & Water              NaN
4           s2  TV Show         Blood & Water              NaN
...        ...      ...                   ...              ...
202060   s8807    Movie                Zubaan      Mozez Singh
202061   s8807    Movie                Zubaan      Mozez Singh
202062   s8807    Movie                Zubaan      Mozez Singh
202063   s8807    Movie                Zubaan      Mozez Singh
202064   s8807    Movie                Zubaan      Mozez Singh

                          cast        country date_added release_year
rating  \
0                          NaN  United States 2021-09-25         2020
PG-13
1                  Ama Qamata   South Africa 2021-09-24         2021
TV-MA
2                  Ama Qamata   South Africa 2021-09-24         2021
TV-MA
3                  Ama Qamata   South Africa 2021-09-24         2021
TV-MA
4                Khosi Ngema   South Africa 2021-09-24         2021
TV-MA
...                       ...           ...        ...          ...
```

```
...
202060          Anita Shabdish          India 2019-03-02         2015
TV-14
202061          Anita Shabdish          India 2019-03-02         2015
TV-14
202062   Chittaranjan Tripathy          India 2019-03-02         2015
TV-14
202063   Chittaranjan Tripathy          India 2019-03-02         2015
TV-14
202064   Chittaranjan Tripathy          India 2019-03-02         2015
TV-14

                      listed_in  duration_tv_show  duration_movie
duration
0                 Documentaries               0.0            90.0
90.0
1       International TV Shows               2.0             0.0
2.0
2                     TV Dramas               2.0             0.0
2.0
3                   TV Mysteries              2.0             0.0
2.0
4       International TV Shows               2.0             0.0
2.0
...                         ...               ...             ...
...
202060    International Movies               0.0           111.0
111.0
202061        Music & Musicals               0.0           111.0
111.0
202062                  Dramas               0.0           111.0
111.0
202063    International Movies               0.0           111.0
111.0
202064        Music & Musicals               0.0           111.0
111.0

[202058 rows x 13 columns]
```

```
plt.figure(figsize = (20,8))
plt.subplot(1,2,1)
sns.boxplot(data = arr[arr["type"]=="TV Show"],y = "duration")
plt.subplot(1,2,2)
sns.boxplot(data = arr[arr["type"]=="Movie"],y = "duration")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff74b8b33d0>
```



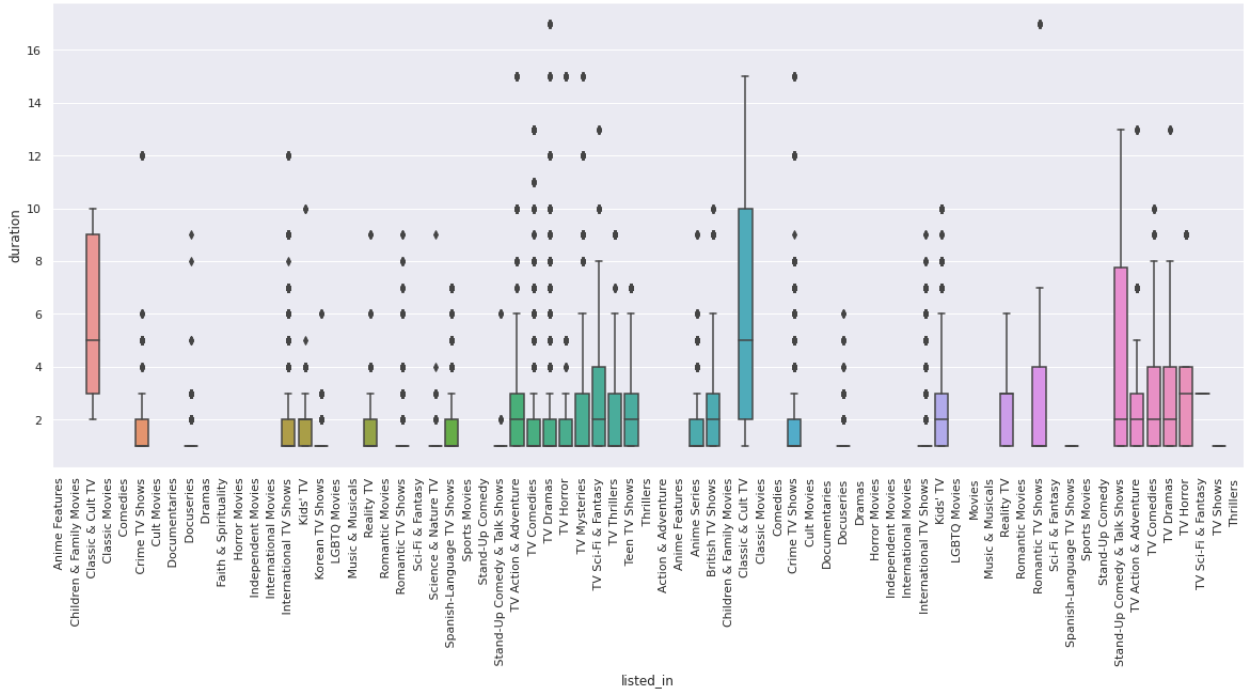


For TV Show , Mean duration is near 1 season

For Movie, Mean duration is near 110 min

random outliers present in both TV Show and Movies

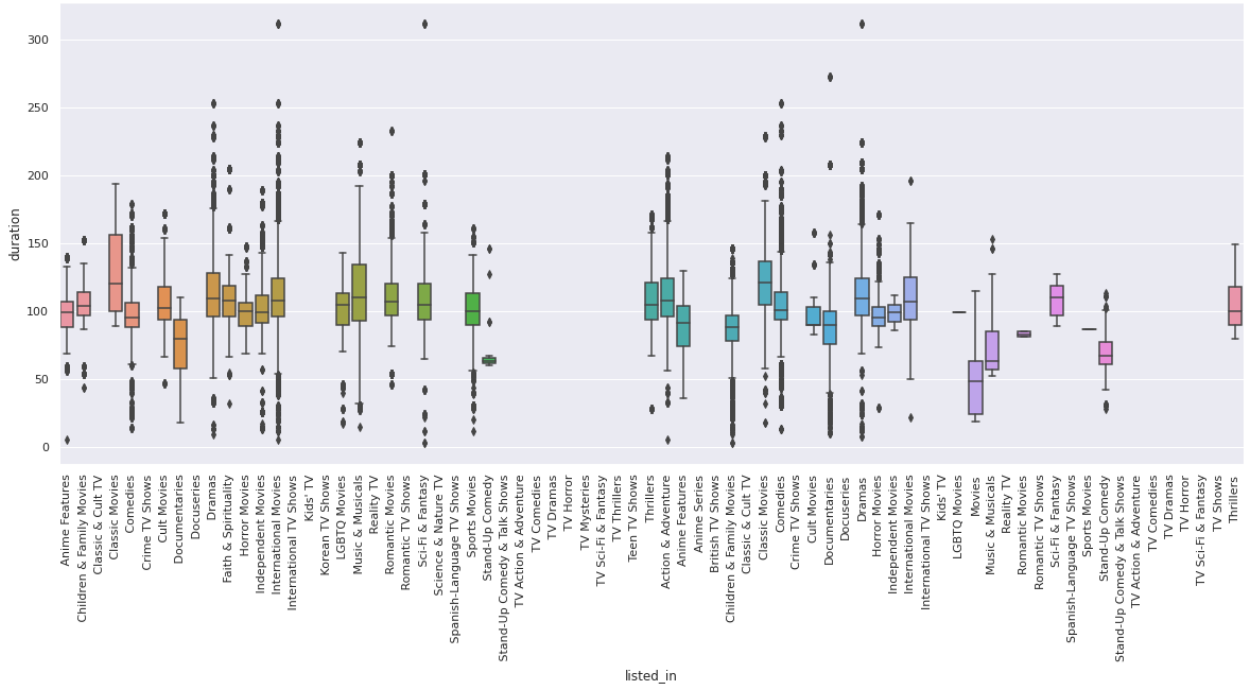
```
plt.figure(figsize = (20,8))
sns.boxplot(data = arr[arr["type"]=="TV Show"],y = "duration",x = "listed_in")
plt.xticks(rotation = 90)
plt.show()
```



Mean duration of `Classic & Cult TV` dominates other genres.

`Stand-Up Comedy and Talk shows` stands in second position.

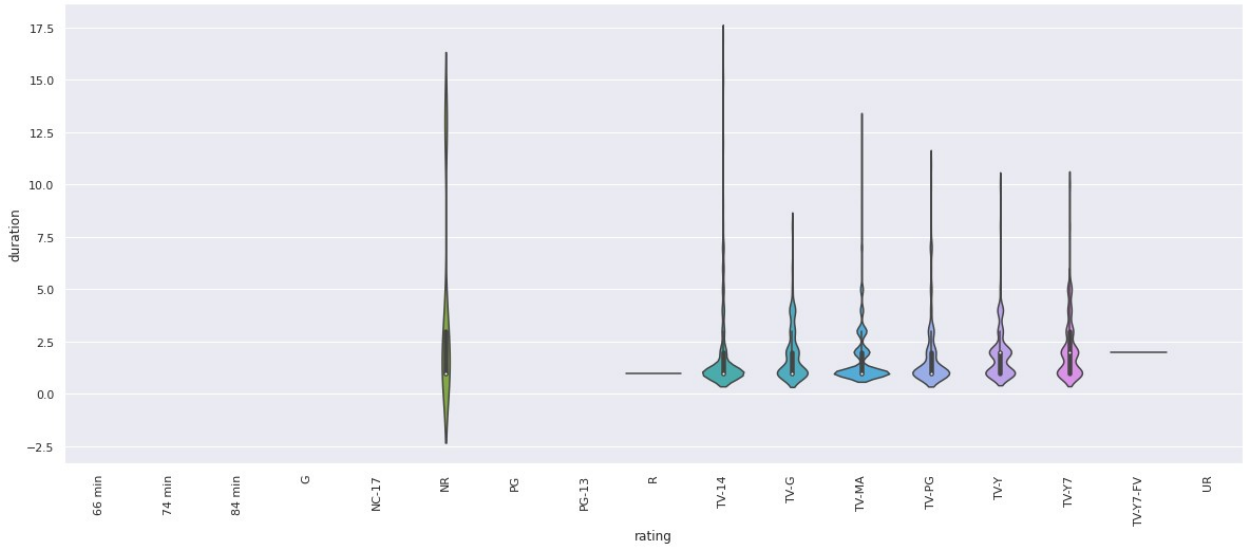
```
plt.figure(figsize = (20,8))
sns.boxplot(data = arr[arr["type"]=="Movie"],y = "duration",x =
"listed_in")
plt.xticks(rotation = 90)
plt.show()
```



In Movies, `Classic Movies` dominates the other categories

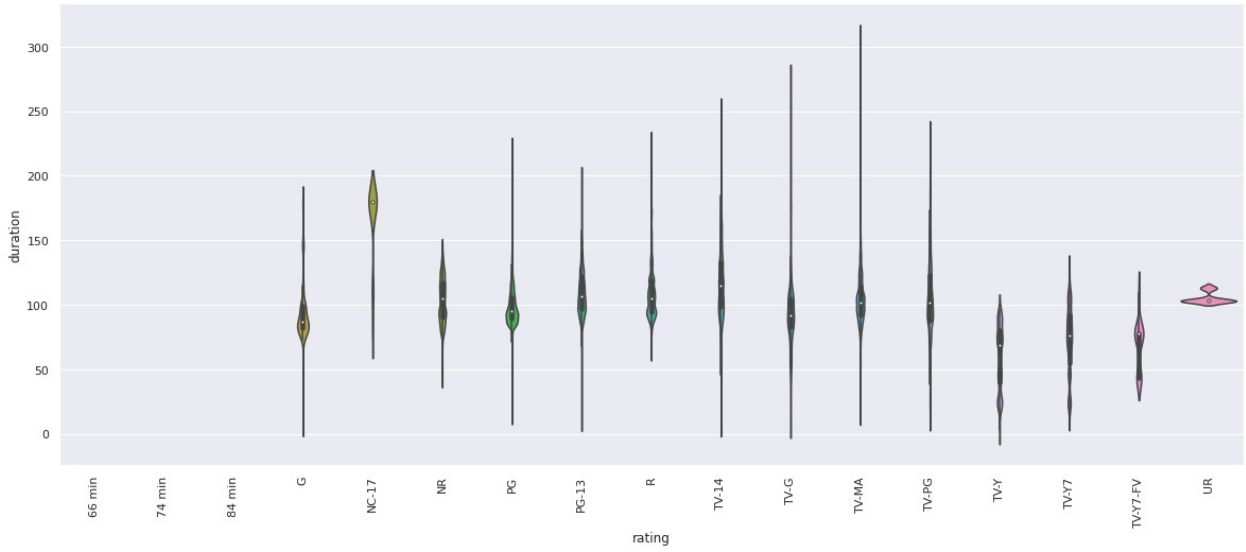
Almost all categories maintain, duration in range of 90 to 130 min

```
plt.figure(figsize = (20,8))
sns.violinplot(data = arr[arr["type"]=="TV Show"],y = "duration",x =
"rating")
plt.xticks(rotation = 90)
plt.show()
```



`TV-MA` and `TV-14` are high frequent ratings.

```
plt.figure(figsize = (20,8))
sns.violinplot(data = arr[arr["type"]=="Movie"],y = "duration",x =
"rating")
plt.xticks(rotation = 90)
plt.show()
```

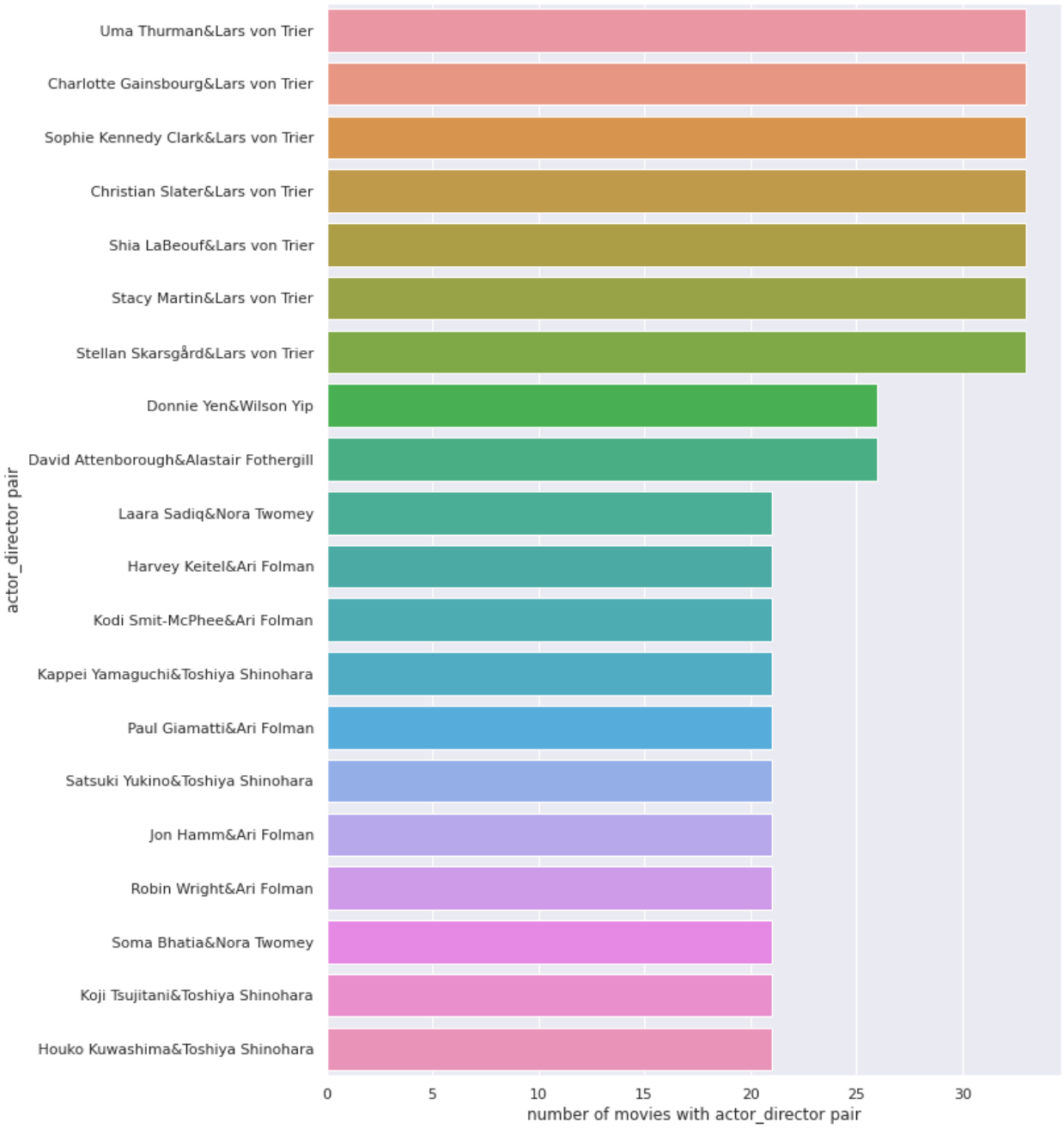


High dispersion is observed in these movies duration vs ratings box plot.

Best actor-director pair

```
arr = netflix_merge[["title","cast","director"]].dropna()
arr["actor_director"] = arr["cast"] + "&"+ arr["director"]
arr2 = arr["actor_director"].value_counts().reset_index()
```

```
arr2 = arr2[0:20]
plt.figure(figsize = (10,15))
sns.barplot(data = arr2, y = "index", x = "actor_director")
plt.xlabel("number of movies with actor_director pair")
plt.ylabel("actor_director pair")
plt.show()
```



`Uma Thurmand & Lars Von Tier` is the best actor-director pair

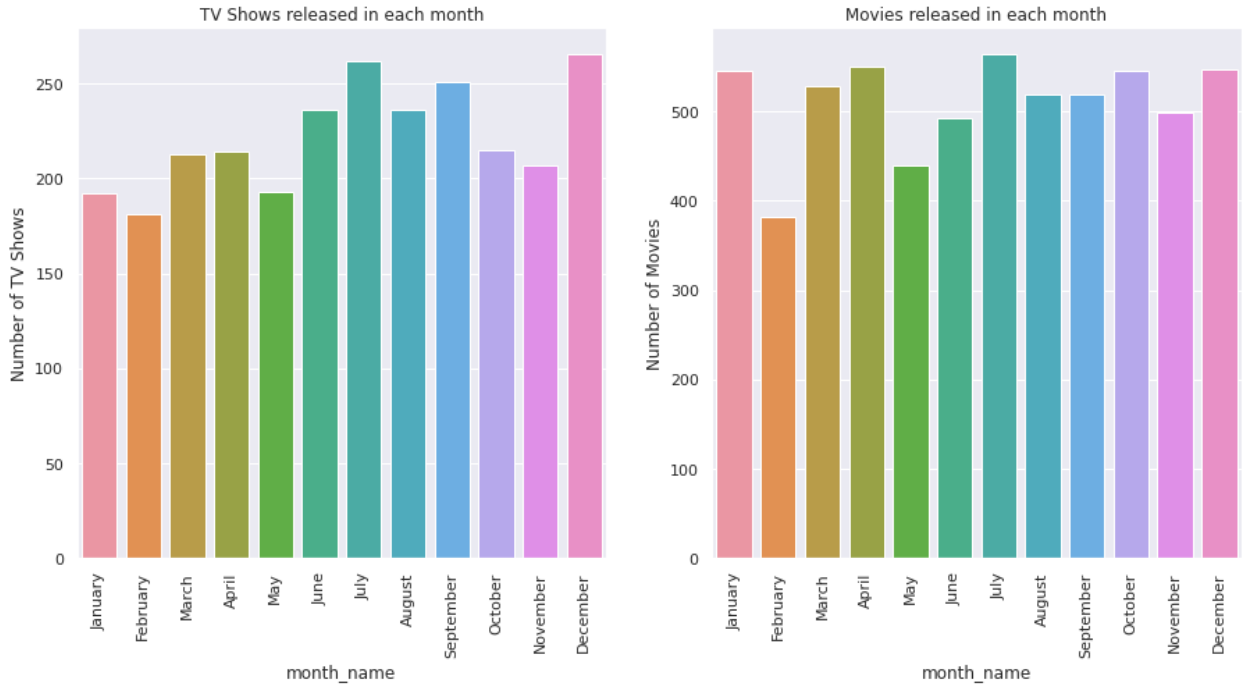
Best month to launch a TV show

```
month_dict = {'January':1,'February':2,'March':3, 'April':4, 'May':5,
'June':6, 'July':7, 'August':8, 'September':9, 'October':10,
'November':11, 'December':12}
arr = netflix_merge
arr["month_name"] = arr["date_added"].dt.month_name()
arr["month"] = arr["date_added"].dt.month
arr.groupby(["month_name","type"])
["title"].nunique().reset_index().sort_values('month_name', key =
lambda x : x.apply (lambda x : month_dict[x]))
```

	month_name	type	title
8	January	Movie	546
9	January	TV Show	192
6	February	Movie	382
7	February	TV Show	181
15	March	TV Show	213
14	March	Movie	529
0	April	Movie	550
1	April	TV Show	214
17	May	TV Show	193
16	May	Movie	439
13	June	TV Show	236
12	June	Movie	492
11	July	TV Show	262
10	July	Movie	565
3	August	TV Show	236
2	August	Movie	519
22	September	Movie	519
23	September	TV Show	251
20	October	Movie	545
21	October	TV Show	215
18	November	Movie	498
19	November	TV Show	207
5	December	TV Show	266
4	December	Movie	547

```
P = arr.groupby(["month_name","type"])
["title"].nunique().reset_index().sort_values('month_name', key =
lambda x : x.apply (lambda x : month_dict[x]))
Q = P[P["type"]=="TV Show"]
fig = plt.figure(figsize=(15,7))
plt.subplot(1,2,1)
sns.barplot(data = Q, x= "month_name",y = "title")
plt.xticks(rotation = 90)
plt.title("TV Shows released in each month")
plt.ylabel("Number of TV Shows")
plt.subplot(1,2,2)
R = P[P["type"]=="Movie"]
```

```
sns.barplot(data = R, x= "month_name",y = "title")
plt.title("Movies released in each month")
plt.ylabel("Number of Movies")
plt.xticks(rotation = 90)
plt.show()
```



OBSERVATIONS: TV SHOWS:

1. `February` is the month with lowest number of TV shows. (Month with less competition)
2. `December` is the month with highest number of TV shows. (Month with high competetion but view count may be high)

MOVIES:

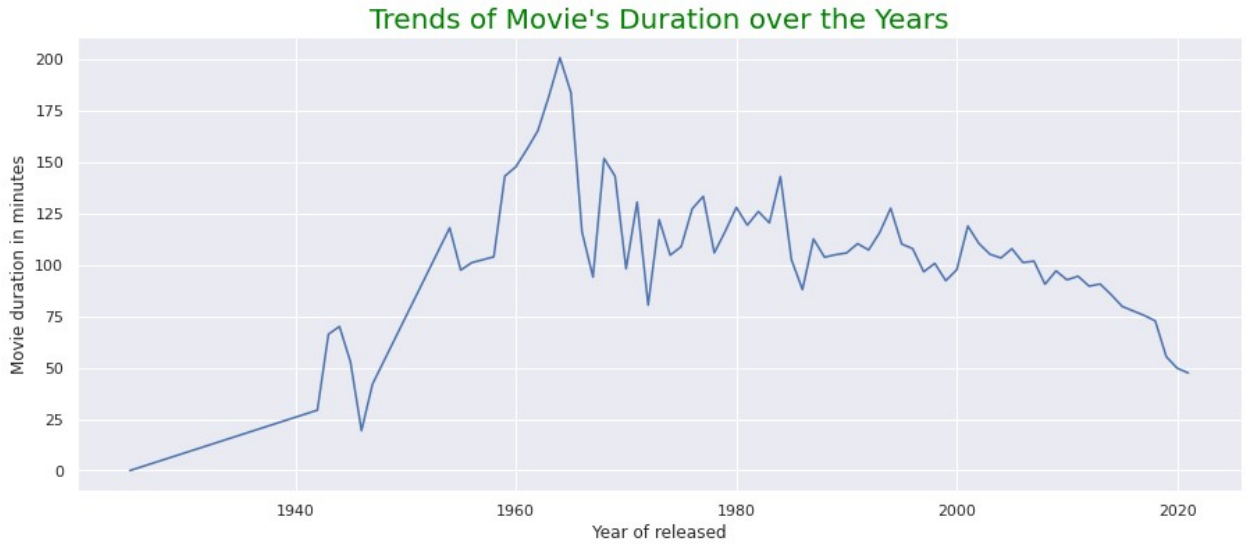
1. `February` is the month with lowest number of Movies. (Month with less competition)
2. `July` is the month with highest number of Movies. (Month with high competetion but view count may be high)

```
 netflix_merge.groupby(['release_year']).mean()
```

release_year	duration_tv_show	duration_movie	duration	month
1925	1.000000	0.000000	1.000000	12.000000
1942	0.000000	29.333333	29.333333	3.000000
1943	0.000000	66.200000	66.200000	3.000000
1944	0.000000	70.000000	70.000000	3.000000
1945	0.076923	52.769231	52.846154	2.923077
...	...	...	...	...
2017	0.544063	75.362693	75.906756	6.405098

```
2018                0.568453       72.691162  73.259615   6.889644
2019                1.031462       55.313848  56.345310   7.056131
2020                0.919074       49.684571  50.603645   6.584150
2021                0.894148       47.386161  48.280309   5.559946

[74 rows x 4 columns]
```

```
duration_year = netflix_merge.groupby(['release_year']).mean()
duration_year = duration_year.sort_index()

plt.figure(figsize=(15,6))
sns.lineplot(x=duration_year.index,
y=duration_year.duration_movie.values)
plt.ylabel('Movie duration in minutes');
plt.xlabel('Year of released');
plt.title("Trends of Movie's Duration over the Years", fontsize=20,
color='Green');
```

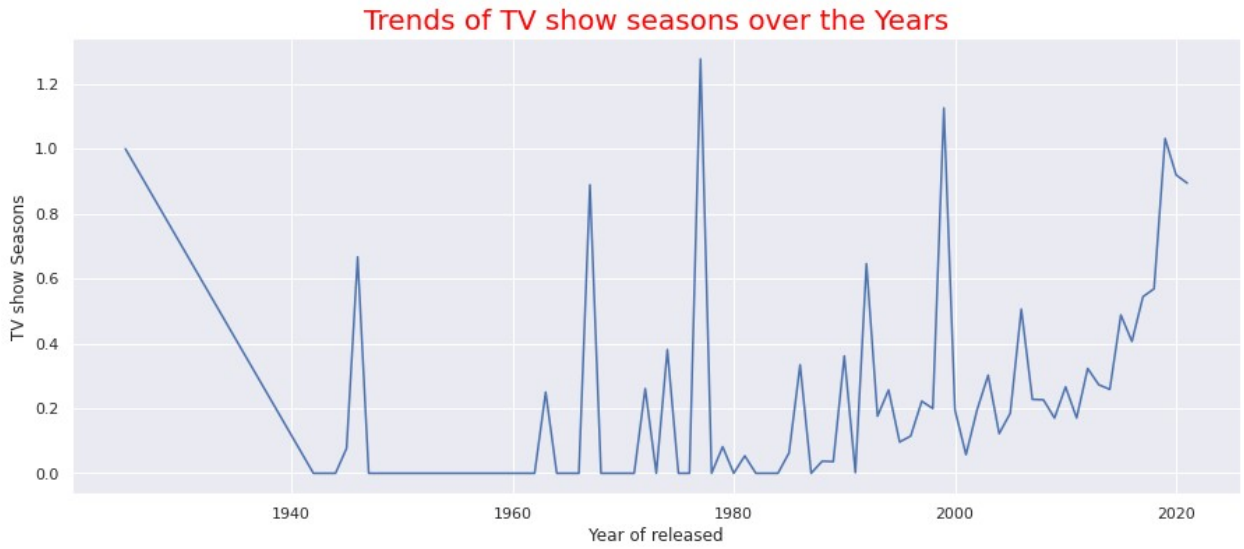


Duration of movies is higher during `1960 to 1970` period

After 1990, Duration maintain average of `100 to 115 min`

In the years of 1960 to 1965, Movies durations were over 200 minutes, after 1965 the durations became comparitively shorter. From the year 1980, we can see consistent trend of movie durations, of which duration time is around in between 100-150 minutes.

```
plt.figure(figsize=(15,6))
sns.lineplot(x=duration_year.index,
y=duration_year.duration_tv_show.values)
plt.ylabel('TV show Seasons');
plt.xlabel('Year of released');
plt.title("Trends of TV show seasons over the Years", fontsize=20,
color='Red');
```

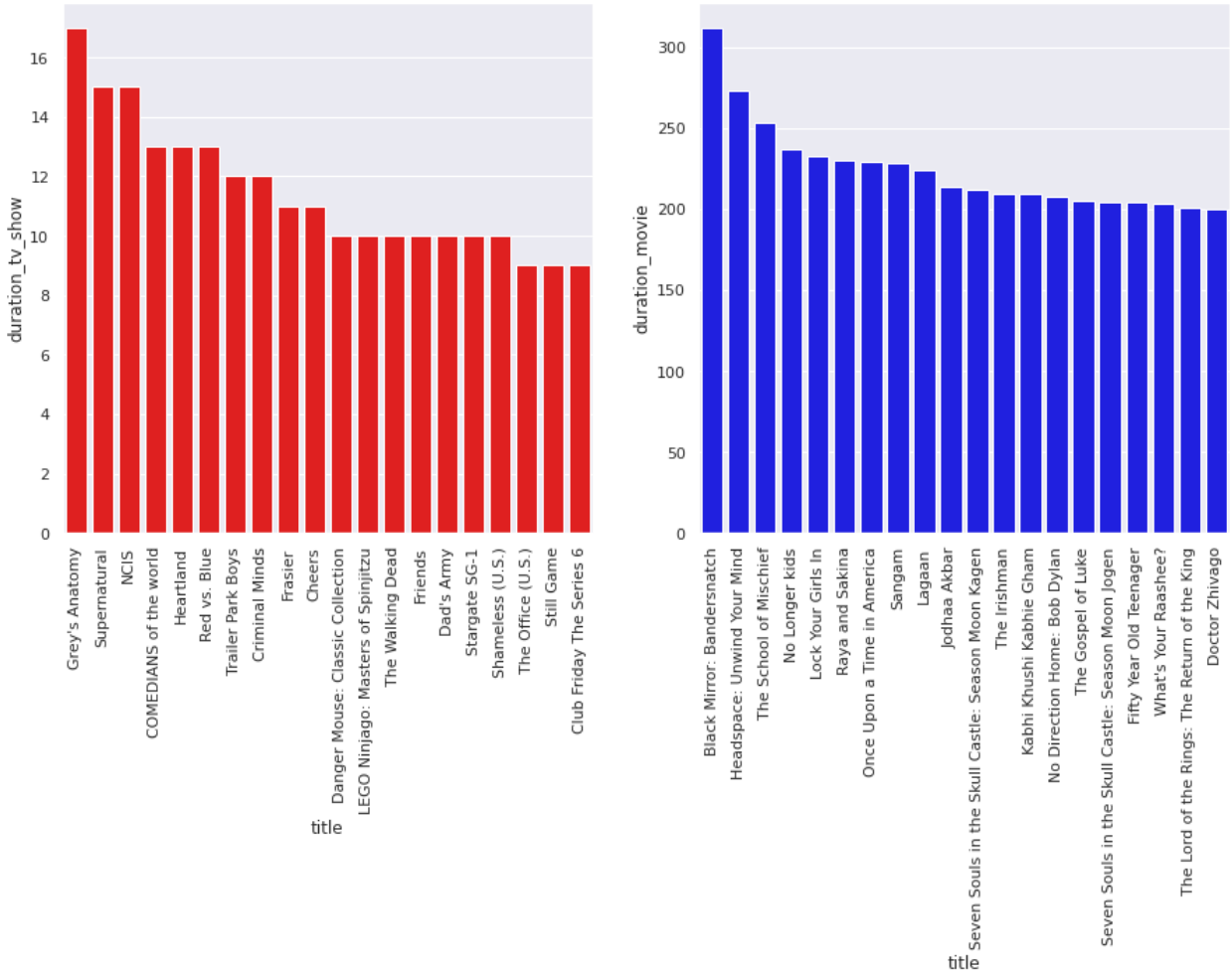


MOre number of seasons are released during `1960 to 2000`

Recent Trend is around `2 seasons` only

Best TV Show and Movie

```
arr = netflix_merge
arr[["duration_movie","duration_tv_show"]]=arr[["duration_movie","dura
tion_tv_show"]].fillna(0)
netflix_merge_mean =
arr.groupby(["title","type"]).mean().reset_index()
tv_shows =
netflix_merge_mean.sort_values(by='duration_tv_show',ascending=False)
movie =
netflix_merge_mean.sort_values(by='duration_movie',ascending=False)
tv20 = tv_shows[0:20]
movie20 = movie[0:20]

plt.figure(figsize=(15,7))
plt.subplot(1,2,1)
sns.barplot(data= tv20, x= "title",y="duration_tv_show",color = "Red")
plt.xticks(rotation = 90)
plt.subplot(1,2,2)
sns.barplot(data= movie20, x= "title",y="duration_movie",color =
"Blue")
plt.xticks(rotation = 90)
plt.show()
```



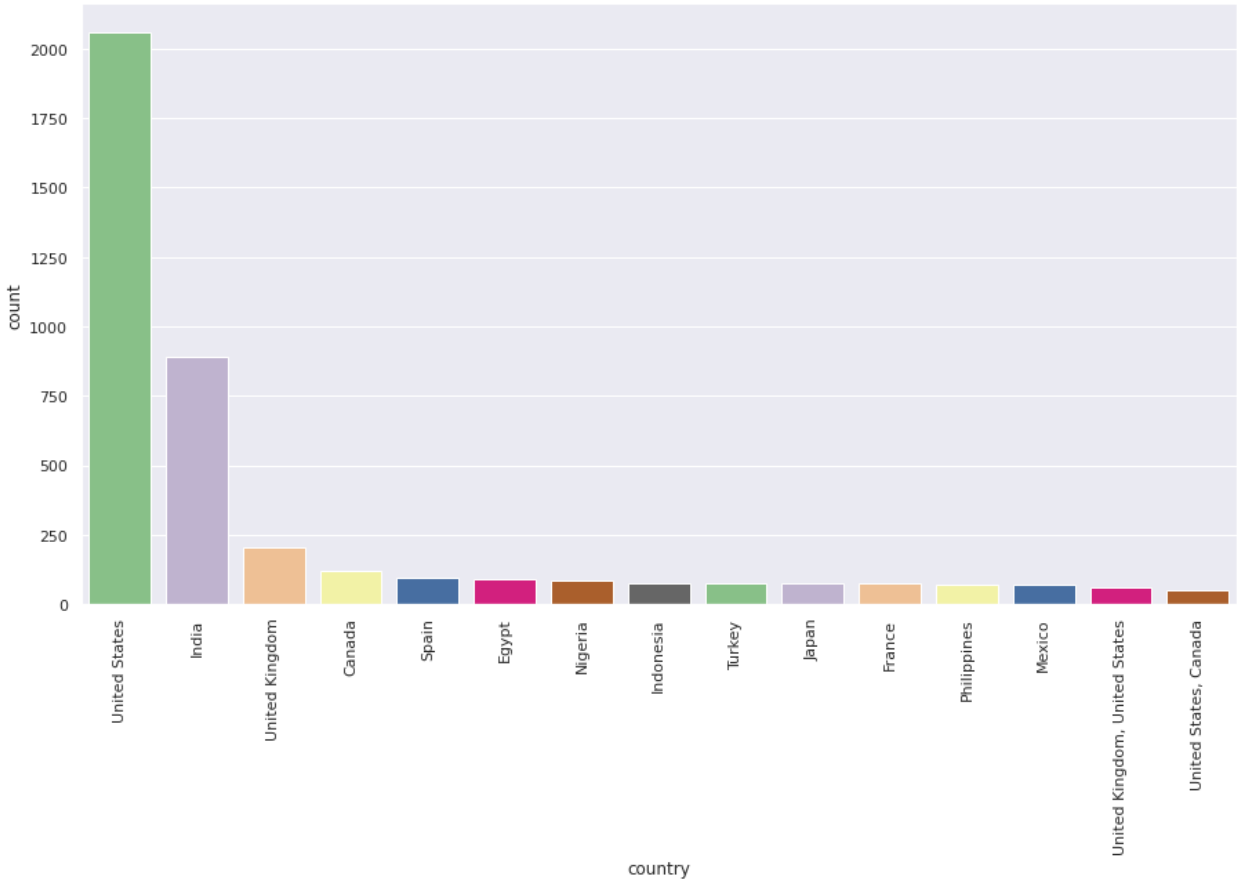
Highest number of season `TV Show` is `Grey's Anatomy`

Highest Duration movies `Black Mirror: Bandersnatch`

Country with more number of movies

```
plt.figure(figsize=(15,8))
sns.set(style="darkgrid")
sns.countplot(x="country", data=movie_netflix, palette="Accent",
order=movie_netflix["country"].value_counts().index[0:15])
plt.xticks(rotation = 90)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]),
 <a list of 15 Text major ticklabel objects>)
```



`United states` and `India` produces large number of movies compared to other countries

4.3 TRIVARIATE PLOTS

Heat map between countries and listed_in categories

```
arr = netflix_merge.groupby(["country"])["listed_in"].value_counts()

netflix_merge.shape

(202058, 15)

arr = arr.reset_index()

arr
```

	country	level_1	listed_in
0		International Movies	40
1		Dramas	30
2		Classic Movies	9
3		Dramas	9
4		Independent Movies	8
...	...	...	...

```
14376  Zimbabwe               Sports Movies          0
14377  Zimbabwe   Spanish-Language TV Shows          0
14378  Zimbabwe         Science & Nature TV          0
14379  Zimbabwe            Sci-Fi & Fantasy          0
14380  Zimbabwe                    Thrillers          0

[14381 rows x 3 columns]
```

```
arr1 = arr.pivot("country","level_1","listed_in")

for i in arr1.columns:
    if sum(arr1[i]) == 0:
        arr1 = arr1.drop([i],axis = 1)
arr1
```

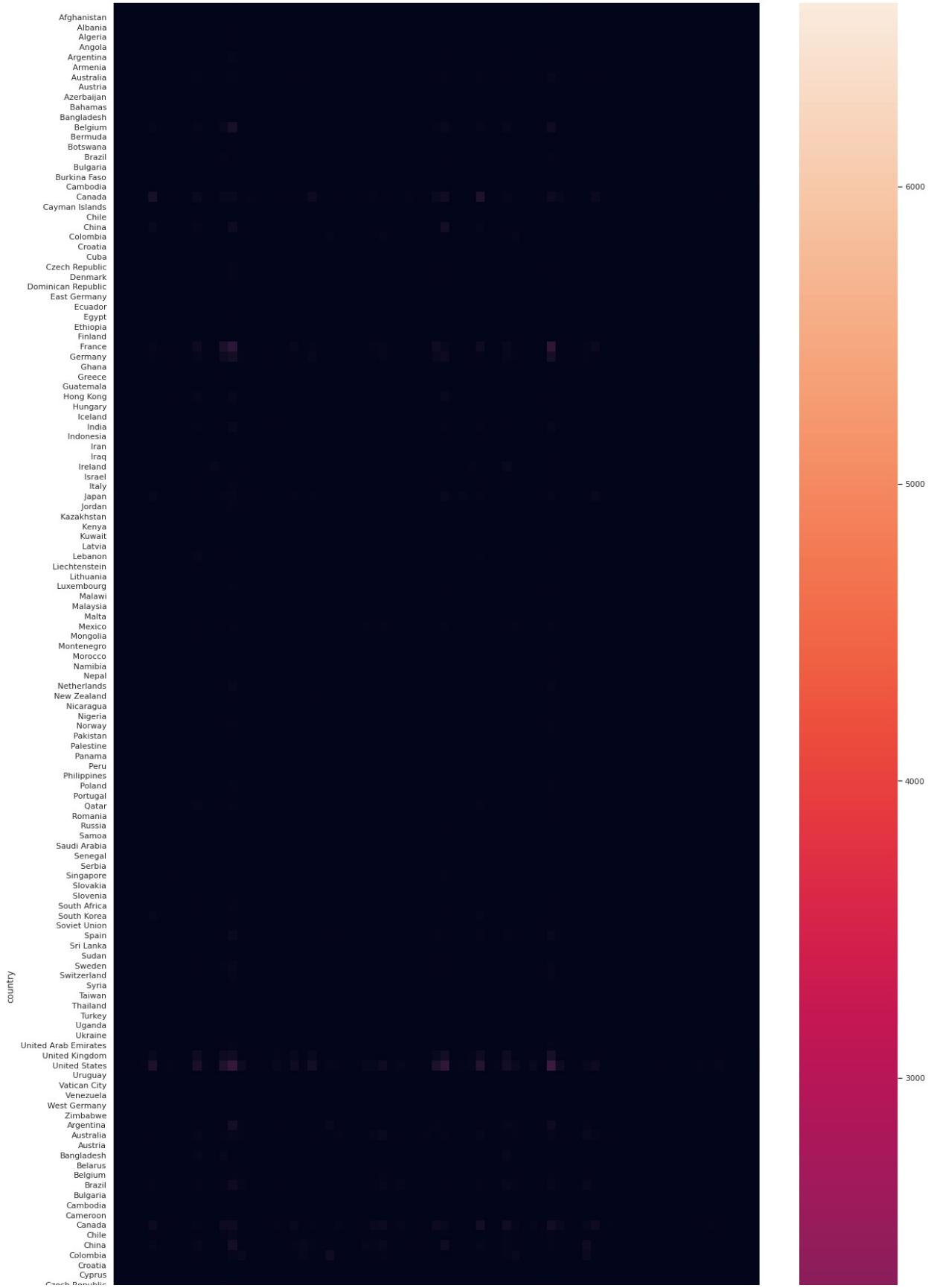
```
level_1        Anime Features   Children & Family Movies   Classic &
Cult TV  \
country

                            0                          0
0
 Afghanistan                0                          0
0
 Albania                    0                          0
0
 Algeria                    0                          0
0
 Angola                     0                          0
0
...                       ...                        ...
...
Uruguay                     0                          0
0
Venezuela                   0                          0
0
Vietnam                     0                          0
0
West Germany                0                          0
0
Zimbabwe                    0                          0
0

level_1        Classic Movies   Comedies   Crime TV Shows   Cult
Movies  \
country

                            0          0                0
0
 Afghanistan                0          0                0
0
```

```
 Albania                        0          0                0
0
 Algeria                        0          0                0
0
 Angola                         0          0                0
0
...                           ...        ...              ...            ..
.
Uruguay                         0          0                0
0
Venezuela                       0          0                0
0
Vietnam                         0          6                0
0
West Germany                    0          0                0
0
Zimbabwe                        0          0                0
0

level_1        Documentaries   Docuseries   Dramas  ...  Sports Movies
\
country                                                 ...

                               0            0        9  ...              0

 Afghanistan                   0            0        0  ...              0

 Albania                       0            0        0  ...              0

 Algeria                       0            0       11  ...              0

 Angola                        0            0        0  ...              0

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

Uruguay                        0            0       20  ...              0

Venezuela                      0            0        0  ...              0

Vietnam                        0            0        9  ...              0

West Germany                   0            0        0  ...              0

Zimbabwe                       0            0        0  ...              0


level_1       Stand-Up Comedy  Stand-Up Comedy & Talk Shows  \
country
                               0                             0
 Afghanistan                   0                             0
 Albania                       0                             0
```

```
 Algeria                      0                                0
 Angola                       0                                0
...                         ...                              ...
Uruguay                      0                                0
Venezuela                    0                                0
Vietnam                      0                                0
West Germany                 0                                0
Zimbabwe                     0                                0

level_1       TV Action & Adventure  TV Comedies  TV Dramas  TV Horror
\
country

                                   0            0          0          0

 Afghanistan                       0            0          0          0

 Albania                           0            0          0          0

 Algeria                           0            0          0          0

 Angola                            0            0          0          0

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

Uruguay                            0            0          0          0

Venezuela                          0            0          0          0

Vietnam                            0            0          0          0

West Germany                       0            0          0          0

Zimbabwe                           0            0          0          0


level_1       TV Sci-Fi & Fantasy  TV Shows  Thrillers
country
                                 0         0          0
 Afghanistan                     0         0          0
 Albania                         0         0          0
 Algeria                         0         0          0
 Angola                          0         0          0
...                            ...       ...        ...
Uruguay                          0         0          0
Venezuela                        0         0          0
Vietnam                          0         0          0
West Germany                     0         0          0
Zimbabwe                         0         0          0

[197 rows x 73 columns]
```

```
plt.figure(figsize = (20,50))
sns.heatmap(data = arr1)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff6e98beee0>
```


Afghanistan
Albania
Algeria
Angola
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahamas
Bangladesh
Belgium
Bermuda
Botswana
Brazil
Bulgaria
Burkina Faso
Cambodia
Canada
Cayman Islands
Chile
China
Colombia
Croatia
Cuba
Czech Republic
Denmark
Dominican Republic
East Germany
Ecuador
Egypt
Ethiopia
Finland
France
Germany
Ghana
Greece
Guatemala
Hong Kong
Hungary
Iceland
India
Indonesia
Iran
Iraq
Ireland
Israel
Italy
Japan
Jordan
Kazakhstan
Kenya
Kuwait
Latvia
Lebanon
Liechtenstein
Lithuania
Luxembourg
Malawi
Malaysia
Malta
Mexico
Mongolia
Montenegro
Morocco
Namibia
Nepal
Netherlands
New Zealand
Nicaragua
Nigeria
Norway
Pakistan
Palestine
Panama
Peru
Philippines
Poland
Portugal
Qatar
Romania
Russia
Samoa
Saudi Arabia
Senegal
Serbia
Singapore
Slovakia
Slovenia
South Africa
South Korea
Soviet Union
Spain
Sri Lanka
Sudan
Sweden
Switzerland
Syria
Taiwan
Thailand
Turkey
Uganda
Ukraine
United Arab Emirates
United Kingdom
United States
Uruguay
Vatican City
Venezuela
West Germany
Zimbabwe
Argentina
Australia
Austria
Bangladesh
Belarus
Belgium
Brazil
Bulgaria
Cambodia
Cameroon
Canada
Chile
China
Colombia
Croatia
Cyprus
Czech Republic
country
6000
5000
4000
3000

Heat map highlights the `International Movies Genre` in India.

`Classic and Cult movies` in `United States`

CHAPTER 5: MISSING VALUE & OUTLIER CHECK (TREATMENT OPTIONAL)

5.1 MISSING VALUE TREATMENT

```
netflix_mode_imputation = netflix_merge

netflix_mode_imputation['director'] =
netflix_mode_imputation['director'].fillna(netflix_mode_imputation['di
rector'].mode()[0])

netflix_mode_imputation['cast'] =
netflix_mode_imputation['cast'].fillna(netflix_mode_imputation['cast']
.mode()[0])

netflix_mode_imputation['country'] =
netflix_mode_imputation['country'].fillna(netflix_mode_imputation['cou
ntry'].mode()[0])

netflix_mode_imputation['date_added'] =
netflix_mode_imputation['date_added'].fillna(netflix_mode_imputation['
date_added'].mode()[0])

netflix_mode_imputation['rating'] =
netflix_mode_imputation['rating'].fillna(netflix_mode_imputation['rati
ng'].mode()[0])

netflix_mode_imputation.drop(["month_name","month"],axis = 1,inplace =
True)

netflix_mode_imputation.isnull().sum()
```

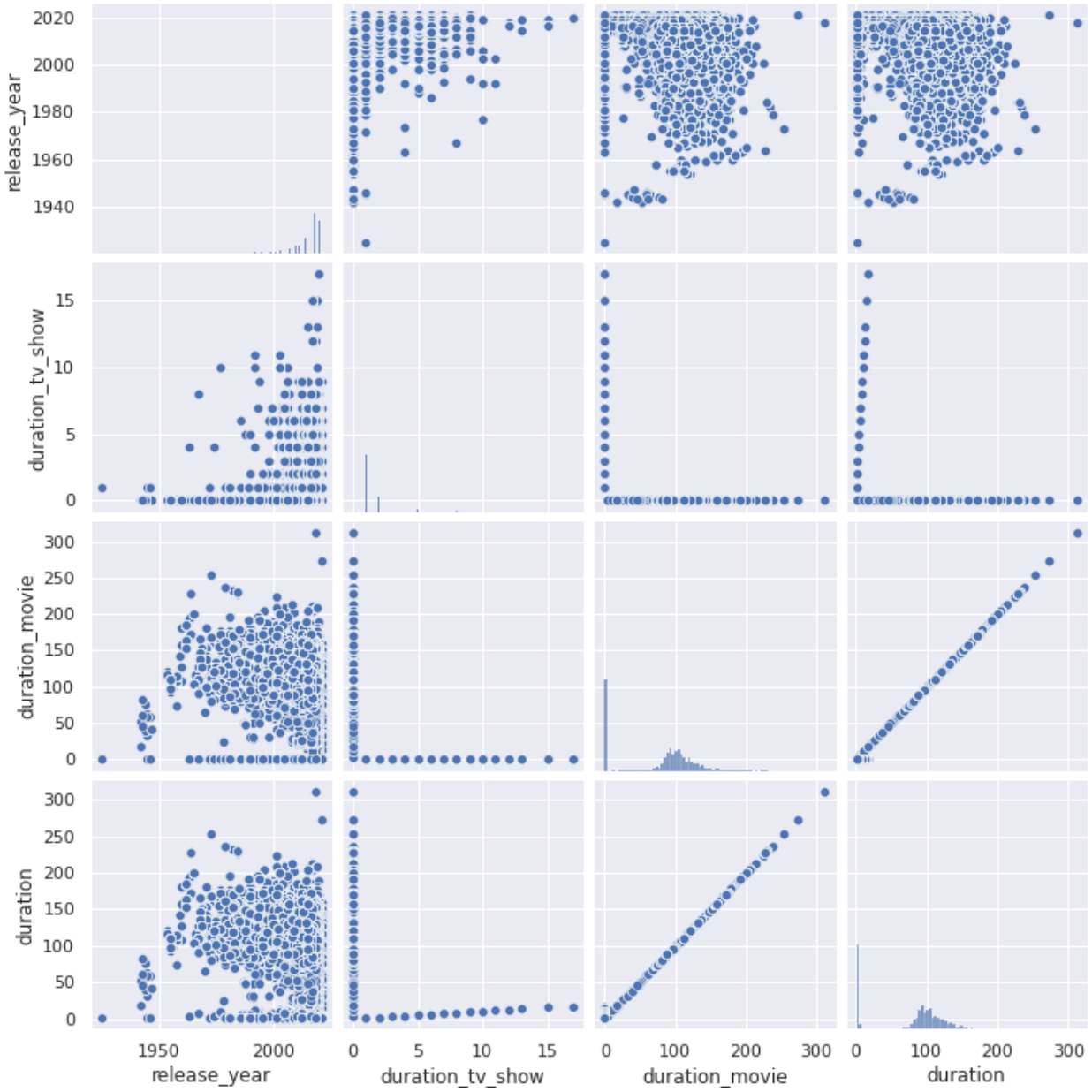
```
show_id             0
type                0
title               0
director            0
cast                0
country             0
date_added          0
release_year        0
rating              0
listed_in           0
duration_tv_show    0
duration_movie      0
```

```
duration              0
dtype: int64
```

5.2 OUTLIER CHECK

```
sns.pairplot(netflix_mode_imputation)
```

```
<seaborn.axisgrid.PairGrid at 0x7ff6ec05a0a0>
```



CHAPTER 6 INSIGHTS BASED ON NON-GRAPHICAL AND VISUAL ANALYSIS

6.1 COMMENTS ON THE RANGE OF ATTRIBUTES

There are `8807 Title fields`, only `two type' of entertainment,`5121`unique`directors`,`39297actor/actress`,`198 countries`,`73 Genres`and`17 rating fields are involved in this dataset.

Starting data of dataset is `2008-01-01` to `2021-09-25`. SO around 13 years data provided

There are `202058 Rows and 12 Columns` in netflix_merge table after unnesting, merging, deleting duplicates and separating duration column into two columns

netflix_merge table consists of `50643 director null values, 2149 cast null values, 11897 country null values, 158 date_added null values, 67 rating null values, 145910 duration_tv_show null values and 56151 duration_movie null values`.

6.2 COMMENTS ON THE DISTRIBUTION OF THE VARIABLES AND RELATIONSHIP BETWEEN THEM

`6131 Movies` and `2676 TV Show` present in the data set. Total `8807 unique fields`

`Rajiv Chilaka` has directed `22 Movies/TV Shows` stands in first position. `Jan Suter and Raul Campos` has directed `18 Movies/TV Shows` stands in second position.

`Anupam Kher` has listed in `39 Different Movies/TV Shows` stands in first position. `Rupa Bhimani` has listed in `31 Different Movies/TV Shows` stands in second position.

`3211` Movies/TVShows are released/produced in `United States` stands in first position. `1008` Movies/TVShows are released/produced in `India` stands in second position.

ON `2020-01-01`, `110` Movies/TV shows added into netflix servers which stands in first position.

During `2018`, `1147` Movies/TV shows released which stands first position.

`TV-MA` rating category has `3207` Movies/TV Shows which stands first position. `TV-14` rating category has `2160` Movies/TV Shows which stands second position.

Among `2676` TV Shows, `1793` shows are having only 1 season. `425` Shows are having 2 seasons. `199` Shows are having 3 seasons.

Among `6131` Movies, `152` Movies has the duration of `90 min` which stands in first position.

`2624` Movie/TV shows listed in `International Movies` Genre which stands in first position. `1600` Movie/TV shows listed in `Dramas` Genre which stands in second

position. `1210` Movie/TV shows listed in `Comedies` Genre which stands in third position.

`TV Show` dominates the `Movies` according to number of persons are casted, number of countries released, number of genres listed_in.

`director ratio per movie`(0.7969) is higher compared to `director ration per tv show`(0.1121). This indicates that `One TV Show director directing mulitiple tv shows`. Whereas, movies require dedicated directors. `Movie directors are not directing as frequently as TV Show directors`.

In Movies, `Classic Movies` dominates the other categories. Almost all categories maintain, duration in range of 90 to 130 min

`TV-MA` and `TV-14` are high frequent ratings.High dispersion is observed in these movies duration vs ratings box plot.

`TV Show` dominates the `Movies` according to number of persons are casted, number of countries released, number of genres listed_in.

`director ratio per movie`(0.7969) is higher compared to `director ration per tv show`(0.1121). This indicates that `One TV Show director directing mulitiple tv shows`. Whereas, movies require dedicated directors. `Movie directors are not directing as frequently as TV Show directors`.

6.3 COMMENTS FOR EACH UNIVARIATE AND BIVARIATE PLOT

the `International Movies Genre` in India. `Classic and Cult movies` in `United States` are popular.

`United states` and `India` produces large number of movies compared to other countries

Highest number of season `TV Show` is `Grey's Anatomy`. Highest Duration movies `Black Mirror: Bandersnatch`

Duration of movies is higher during `1960 to 1970` period. After 1990, Duration maintain average of `100 to 115 min`. More number of seasons are released during `1960 to 2000`

`February` is the month with lowest number of TV shows.`December` is the month with highest number of TV shows `February` is the month with lowest number of Movies.`July` is the month with highest number of Movies. (Month with high competetion but view count may be high)

`TV-MA` Category has more movies and more TV Shows

During 2021, Both TV Show and Movie count reduced due to `COVID`. Both TV SHOW and Movie count gradually increased year by year

Number of movies/TV shows released per year is higher during `2015 to 2020` period

For TV Show , Mean duration is near `1 season`.For Movie, Mean duration is near `110 min`. Random outliers present in both TV Show and Movies

Mean duration of `Classic & Cult TV` dominates other genres.

In Movies, `Classic Movies` dominates the other categories

`Uma Thurmand & Lars Von Tier` is the best actor-director pair

CHAPTER 7: BUSINESS INSIGHTS

INSTRUCTIONS: **`Should include patterns observed in the data along with what you can infer from it`**

Netflix concentrates highly on two countries - `United States and India`.

High number of movies are present `International Movies and Classical&Cult Movie` Genres. So these Genres are prefereable.

From last 20 year, On Average, Duration of TV Show is `1 to 1.9` Season and Duration of Movies `90 to 110` Minutes. So Ideal Duration is around 2 Hrs.

Pair like `Uma Thurmand & Lars Von Tier` have more number of movies. So Actor Director pair from India and United States created high impact.

`director ratio per movie`(0.7969) is higher compared to `director ration per tv show`(0.1121). This indicates that `One TV Show director directing mulitiple tv shows`. Whereas, movies require dedicated directors. `Movie directors are not directing as frequently as TV Show directors`.

CHAPTER 8: RECOMMENDATIONS

INSTRUCTIONS: **`ACTIONABLE ITEMS FOR BUSINESS. NO TECHNICAL JARGON. NO COMPLICATIONS. SIMPLE ACTION ITEMS THAT EVERYONE CAN UNDERSTAND`**

Netflix movies from countries other than India and United states helps to improve diversity.

Higher seasons TV Shows are not likely to get high view count. Netflix should maintain balance between TV Show and Movie growth. According to dataset, Netflix should slightly concentrate more on Movies.

Multi lingual/Diverse Culture Actor Director Pair movies or TV Shows are preferable.