netflix-case-study

March 28, 2024

Import Libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/
↪original/netflix.csv -O netflix.csv
```

```
Downloading…
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/ori
ginal/netflix.csv
To: /content/netflix.csv
100% 3.40M/3.40M [00:00<00:00, 127MB/s]
```

[3]:
```
df=pd.read_csv("netflix.csv")
```

1 1.Problem Statement : Netflix wants to analyze the Movie/TV show data to get key insights on how to grow their business.

Analysing basic metrics

[4]:
```
df.head()
```

[4]:

	show_id	type	title	director	cast	country
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…	South Africa
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…	NaN
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN
4	s5	TV Show	Kota Factory	NaN		

```
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…
```

[5]:
```
df.tail()
```

[5]:
```
     show_id     type        title         director  \
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  \
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  \
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…
```

[6]:
```
len(df) #length
```

[6]:
```
8807
```

[7]:
```
df.columns
```

[7]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

[8]:
```
df.dtypes #datatypes of columns
```

[8]:
```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

[9]:
```
df.isnull().sum() # missing values in the dataset
```

[9]:
```
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
```

[10]:
```
df.duplicated().sum() #it shows each column represent data of individual title
```

[10]:
```
0
```

[11]:
```
df.describe()
```

[11]:
```
       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
```

[12]:
```
df.nunique() # unique values in each column
```

[12]:
```
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
```

[13]:
```
df['type'].unique()
```

[13]:
```
array(['Movie', 'TV Show'], dtype=object)
```

[14]:
```
df['rating'].unique()
```

[14]:
```
array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
       'TV-G', 'G', 'NC-17', '74 min', '84 min', '66 min', 'NR', nan,
       'TV-Y7-FV', 'UR'], dtype=object)
```

[15]:
```
df['rating'].value_counts()
```

[15]:
```
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64
```

2 2.Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' , missing value detection, statistical summary

[16]:
```
df.shape #shape of dataset #it show total data contains of 8807 titles released␣
↪in netflix
```

[16]:
```
(8807, 12)
```

[17]:
```
df.info()  #shows all info about dataset
```

```
<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
```

[18]:
```
df['rating'].value_counts()
```

[18]:
```
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64
```

[19]:
```
df['country'].unique()
```

[19]:
```
array(['United States', 'South Africa', nan, 'India',
       'United States, Ghana, Burkina Faso, United Kingdom, Germany, Ethiopia',
       'United Kingdom', 'Germany, Czech Republic', 'Mexico', 'Turkey',
       'Australia', 'United States, India, France', 'Finland',
       'China, Canada, United States',
       'South Africa, United States, Japan', 'Nigeria', 'Japan',
       'Spain, United States', 'France', 'Belgium',
       'United Kingdom, United States', 'United States, United Kingdom',
       'France, United States', 'South Korea', 'Spain',
       'United States, Singapore', 'United Kingdom, Australia, France',
       'United Kingdom, Australia, France, United States',
       'United States, Canada', 'Germany, United States',
       'South Africa, United States', 'United States, Mexico',
       'United States, Italy, France, Japan',
```

```
'United States, Italy, Romania, United Kingdom',
'Australia, United States', 'Argentina, Venezuela',
'United States, United Kingdom, Canada', 'China, Hong Kong',
'Russia', 'Canada', 'Hong Kong', 'United States, China, Hong Kong',
'Italy, United States', 'United States, Germany',
'United Kingdom, Canada, United States', ', South Korea',
'Ireland', 'India, Nepal',
'New Zealand, Australia, France, United States', 'Italy',
'Italy, Brazil, Greece', 'Argentina', 'Jordan', 'Colombia',
'United States, Japan', 'Belgium, United Kingdom',
'Switzerland, United Kingdom, Australia', 'Israel, United States',
'Canada, United States', 'Brazil', 'Argentina, Spain', 'Taiwan',
'United States, Nigeria', 'Bulgaria, United States',
'Spain, United Kingdom, United States', 'United States, China',
'United States, France',
'Spain, France, United Kingdom, United States',
', France, Algeria', 'Poland', 'Germany',
'France, Israel, Germany, United States, United Kingdom',
'New Zealand', 'Saudi Arabia', 'Thailand', 'Indonesia',
'Egypt, Denmark, Germany', 'United States, Switzerland',
'Hong Kong, Canada, United States', 'Kuwait, United States',
'France, Canada, United States, Spain',
'France, Netherlands, Singapore', 'France, Belgium',
'Ireland, United States, United Kingdom', 'Egypt', 'Malaysia',
'Israel', 'Australia, New Zealand', 'United Kingdom, Germany',
'Belgium, Netherlands', 'South Korea, Czech Republic',
'Australia, Germany', 'Vietnam', 'United Kingdom, Belgium',
'United Kingdom, Australia, United States',
'France, Japan, United States',
'United Kingdom, Germany, Spain, United States',
'United Kingdom, United States, France, Italy',
'United States, Germany, Canada',
'United States, France, Italy, United Kingdom',
'United States, United Kingdom, Germany, Hungary',
'United States, New Zealand', 'Sweden', 'China', 'Lebanon',
'Romania', 'Finland, Germany', 'Lebanon, Syria', 'Philippines',
'Iceland', 'Denmark', 'United States, India',
'Philippines, Singapore, Indonesia',
'China, United States, Canada', 'Lebanon, United Arab Emirates',
'Canada, United States, Denmark', 'United Arab Emirates',
'Mexico, France, Colombia', 'Netherlands',
'Germany, United States, France', 'United States, Bulgaria',
'United Kingdom, France, Germany, United States',
'Norway, Denmark', 'Syria, France, Lebanon, Qatar',
'United States, Czech Republic', 'Mauritius',
'Canada, South Africa', 'Austria', 'Mexico, Brazil',
'Germany, France', 'Mexico, United States',
```

```
        'United Kingdom, France, Spain, United States',
        'United States, Australia',
        'United States, United Kingdom, France', 'United States, Russia',
        'United States, United Kingdom, New Zealand',
        'Australia, United Kingdom', 'Canada, Nigeria, United States',
        'France, United States, United Kingdom, Canada',
        'France, United Kingdom', 'India, United Kingdom',
        'Canada, United States, Mexico',
        'United Kingdom, Germany, United States',
        'Czech Republic, United Kingdom, United States',
        'China, United Kingdom', 'Italy, United Kingdom', 'China, Taiwan',
        'United States, Brazil, Japan, Spain, India',
        'United States, China, United Kingdom', 'Cameroon',
        'Lebanon, Palestine, Denmark, Qatar', 'Japan, United States',
        'Uruguay, Germany', 'Egypt, Saudi Arabia',
        'United Kingdom, France, Poland, Germany, United States',
        'Ireland, Switzerland, United Kingdom, France, United States',
        'United Kingdom, South Africa, France',
        'Ireland, United Kingdom, France, Germany',
        'Russia, United States', 'United Kingdom, United States, France',
        'United Kingdom,', 'United States, India, United Kingdom', 'Kenya',
        'Spain, Argentina', 'India, United Kingdom, France, Qatar',
        'Belgium, France', 'Argentina, Chile', 'United States, Thailand',
        'Chile, Brazil', 'United States, Colombia',
        'Canada, United States, United Kingdom', 'Uruguay', 'Luxembourg',
        'United States, Cambodia, Romania', 'Bangladesh',
        'Spain, Belgium, United States',
        'United Kingdom, United States, Australia',
        'Canada, United States, France', 'Portugal, United States',
        'Portugal, Spain', 'India, United States',
        'United Kingdom, Ireland', 'United Kingdom, Spain, United States',
        'Hungary, United States', 'United States, South Korea',
        'Canada, United States, Cayman Islands', 'India, France',
        'France, Canada', 'Canada, Hungary, United States', 'Norway',
        'Canada, United Kingdom, United States',
        'United Kingdom, Germany, France, United States',
        'Denmark, United States', 'Senegal', 'France, Algeria',
        'United Kingdom, Finland, Germany, United States, Australia, Japan,
France, Ireland',
        'Philippines, Canada, United Kingdom, United States',
        'Ireland, France, Iceland, United States, Mexico, Belgium, United
Kingdom, Hong Kong',
        'Singapore', 'Kuwait', 'United States, France, Serbia',
        'United States, Italy', 'Spain, Italy',
        'United States, Ireland, United Kingdom, India',
        'United Kingdom, Singapore', 'Hong Kong, United States',
        'United States, Malta, France, United Kingdom',
```

```
       'United States, China, Canada', 'Canada, United States, Ireland',
       'Lebanon, Canada, France', 'Japan, Canada, United States',
       'Spain, France, Canada',
       'Denmark, Singapore, Canada, United States',
       'United States, France, Denmark', 'United States, China, Colombia',
       'Spain, Thailand, United States', 'Mexico, Spain',
       'Ireland, Luxembourg, Belgium', 'China, United States',
       'Canada, Belgium', 'Canada, United Kingdom',
       'Lebanon, United Arab Emirates, France, Switzerland, Germany',
       'France, Belgium, Italy',
       'Lebanon, United States, United Arab Emirates', 'Lebanon, France',
       'France, Lebanon', 'France, Lebanon, United Kingdom',
       'France, Norway, Lebanon, Belgium',
       'Sweden, Czech Republic, United Kingdom, Denmark, Netherlands',
       'United States, United Kingdom, India', 'Indonesia, Netherlands',
       'Turkey, South Korea', 'Serbia, United States', 'Namibia',
       'United Kingdom, Kenya', 'United Kingdom, France, Germany, Spain',
       'United Kingdom, France, United States, Belgium, Luxembourg, China,
Germany',
       'Thailand, United States',
       'United States, France, Canada, Belgium', 'United Kingdom, China',
       'Germany, China, United Kingdom',
       'Australia, New Zealand, United States',
       'Hong Kong, Iceland, United States', 'France, Australia, Germany',
       'United States, Belgium, Canada, France', 'South Africa, Angola',
       'United States, Philippines',
       'United States, United Kingdom, Canada, China',
       'United States, Canada, United Kingdom', 'Turkey, United States',
       'Peru, Germany, Norway', 'Mozambique', 'Brazil, France',
       'China, Spain, South Korea, United States', 'Spain, Germany',
       'Hong Kong, China', 'France, Belgium, Luxembourg, Cambodia,',
       'United Kingdom, Australia', 'Belarus',
       'Indonesia, United Kingdom',
       'Switzerland, France, Belgium, United States', 'Ghana',
       'Spain, France, Canada, United States', 'Chile, Italy',
       'United Kingdom, Nigeria', 'Chile', 'France, Egypt',
       'Egypt, France', 'France, Brazil, Spain, Belgium',
       'Egypt, Algeria', 'Canada, South Korea, United States',
       'Nigeria, United Kingdom', 'United States, France, Canada',
       'Poland, United States',
       'United Arab Emirates, Jordan, Lebanon, Saudi Arabia',
       'United States, Mexico, Spain, Malta',
       'Saudi Arabia, United Arab Emirates', 'Zimbabwe',
       'United Kingdom, Germany, United Arab Emirates, New Zealand',
       'Romania, United States', 'Canada, Nigeria',
       'Saudi Arabia, Netherlands, Germany, Jordan, United Arab Emirates, United
States',
```

```
        'United Kingdom, Spain', 'Finland, France',
        'United Kingdom, Germany, United States, France',
        'India, United Kingdom, China, Canada, Japan, South Korea, United
States',
        'Italy, United Kingdom, France', 'United States, Mexico, Colombia',
        'Turkey, India', 'Italy, Turkey',
        'United Kingdom, United States, Japan',
        'France, Belgium, United States',
        'Puerto Rico, United States, Colombia', 'Uruguay, Argentina',
        'United States, United Kingdom, Japan', 'United States, Argentina',
        'United Kingdom, Italy', 'Ireland, United Kingdom',
        'United Kingdom, France, Belgium, Canada, United States',
        'Netherlands, Germany, Denmark, United Kingdom', 'Hungary',
        'Austria, Germany', 'Taiwan, China',
        'United Kingdom, United States, Ireland',
        'South Korea, United States', 'Brazil, United Kingdom',
        'Pakistan, United States', 'Romania, France, Switzerland, Germany',
        'Romania, United Kingdom', 'France, Malta, United States',
        'Cyprus',
        'United Kingdom, France, Belgium, Ireland, United States',
        'United States, Norway, Canada', 'Kenya, United States',
        'France, South Korea, Japan, United States', 'Taiwan, Malaysia',
        'Uruguay, Argentina, Germany, Spain',
        'United States, United Kingdom, France, Germany, Japan',
        'United States, France, Japan',
        'United Kingdom, France, United States',
        'Spain, France, United States',
        'Indonesia, South Korea, Singapore', 'United States, Spain',
        'Netherlands, Germany, Italy, Canada',
        'Spain, Germany, Denmark, United States', 'Norway, Sweden',
        'South Korea, Canada, United States, China',
        'Argentina, Uruguay, Serbia', 'France, Japan',
        'Mauritius, South Africa', 'United States, Poland',
        'United Kingdom, United States, Germany, Denmark, Belgium, Japan',
        'India, Germany', 'India, United Kingdom, Canada, United States',
        'Philippines, United States', 'Romania, Bulgaria, Hungary',
        'Uruguay, Guatemala', 'France, Senegal, Belgium',
        'United Kingdom, Canada', 'Mexico, United States, Spain, Colombia',
        'Canada, Norway', 'Singapore, United States',
        'Finland, Germany, Belgium', 'United Kingdom, France',
        'United States, Chile', 'United Kingdom, Japan, United States',
        'Spain, United Kingdom', 'Argentina, United States, Mexico',
        'United States, South Korea, Japan', 'Canada, Australia',
        'United Kingdom, Hungary, Australia', 'Italy, Belgium',
        'United States, United Kingdom, Germany', 'Switzerland',
        'Singapore, Malaysia',
        'France, Belgium, Luxembourg, Romania, Canada, United States',
```

```
'South Africa, Nigeria', 'Spain, France',
'United Kingdom, Hong Kong', 'Pakistan', 'Brazil, United States',
'Denmark, Brazil, France, Portugal, Sweden', 'India, Turkey',
'Malaysia, Singapore, Hong Kong', 'Philippines, Singapore',
'Australia, Canada', 'Taiwan, China, France, United States',
'Germany, Italy', 'Colombia, Peru, United Kingdom',
'Thailand, China, United States', 'Argentina, United States',
'Sweden, United States', 'Uruguay, Spain, Mexico',
'France, Luxembourg, Canada', 'Denmark, Spain', 'Chile, Argentina',
'United Kingdom, Belgium, Sweden', 'Canada, Brazil',
'Italy, France', 'Canada, Germany',
'Pakistan, United Arab Emirates', 'Ghana, United States',
'Mexico, Finland', 'United Arab Emirates, United Kingdom, India',
'Netherlands, Belgium', 'United States, Taiwan',
'Austria, Iraq, United States', 'United Kingdom, Malawi',
'Paraguay, Argentina', 'United Kingdom, Russia, United States',
'India, Pakistan', 'Indonesia, Singapore', 'Spain, Belgium',
'Iceland, Sweden, Belgium', 'Croatia', 'Uruguay, Argentina, Spain',
'United Kingdom, Ireland, United States',
'Canada, Germany, France, United States', 'United Kingdom, Japan',
'Norway, Denmark, Netherlands, Sweden',
'Hong Kong, China, United States', 'Ireland, Canada',
'Italy, Switzerland, France, Germany', 'Mexico, Netherlands',
'United States, Sweden', 'Germany, France, Russia',
'France, Iran, United States', 'United Kingdom, India',
'Russia, Poland, Serbia', 'Spain, Portugal', 'Peru',
'Mexico, Argentina',
'United Kingdom, Canada, United States, Cayman Islands',
'Indonesia, United States',
'United States, Israel, United Kingdom, Canada',
'Norway, Iceland, United States', 'Czech Republic, United States',
'United Kingdom, India, United States',
'United Kingdom, West Germany', 'India, Australia',
'United States,', 'Belgium, United Kingdom, United States',
'India, Germany, Austria',
'United States, Brazil, South Korea, Mexico, Japan, Germany',
'Spain, Mexico', 'China, Japan', 'Argentina, France',
'China, United States, United Kingdom',
'France, Luxembourg, United States',
'China, United States, Australia', 'Colombia, Mexico',
'United States, Canada, Ireland', 'Chile, Peru',
'Argentina, Italy', 'Canada, Japan, United States',
'United Kingdom, Canada, United States, Germany',
'Italy, Switzerland, Albania, Poland',
'United States, Japan, Canada', 'Cambodia',
'Italy, United States, Argentina',
'Saudi Arabia, Syria, Egypt, Lebanon, Kuwait',
```

```
'United States, Canada, Indonesia, United Kingdom, China, Singapore',
'Spain, Colombia',
'United Kingdom, South Africa, Australia, United States',
'Bulgaria', 'Argentina, Brazil, France, Poland, Germany, Denmark',
'United Kingdom, Spain, United States, Germany',
'Philippines, Qatar', 'Netherlands, Belgium, Germany, Jordan',
'United Arab Emirates, United States', 'Norway, Germany, Sweden',
'South Korea, China', 'Georgia', 'Soviet Union, India',
'Australia, United Arab Emirates', 'Canada, Germany, South Africa',
'South Korea, China, United States', 'India, Soviet Union',
'India, Mexico', 'Georgia, Germany, France',
'United Arab Emirates, Romania', 'India, Malaysia',
'Germany, Jordan, Netherlands', 'Turkey, France, Germany, Poland',
'Greece, United States', 'France, United Kingdom, United States',
'Norway, Germany', 'France, Morocco', 'Cambodia, United States',
'United States, Denmark', 'United States, Colombia, Mexico',
'United Kingdom, Italy, Israel, Peru, United States',
'Argentina, Uruguay, Spain, France',
'United Kingdom, France, United States, Belgium',
'France, Canada, China, Cambodia',
'United Kingdom, France, Belgium, United States', 'Chile, France',
'Netherlands, United States', 'France, United Kingdom, India',
'Czech Republic, Slovakia', 'Singapore, France',
'Spain, Switzerland', 'United States, Australia, China',
'South Africa, United States, Germany',
'United States, United Kingdom, Australia',
'Spain, Italy, Argentina', 'Chile, Spain, Argentina, Germany',
'West Germany', 'Austria, Czech Republic', 'Lebanon, Qatar',
'United Kingdom, Jordan, Qatar, Iran',
'France, South Korea, Japan', 'Israel, Germany, France',
'Canada, Japan, Netherlands', 'United States, Hungary',
'France, Germany', 'France, Qatar',
'United Kingdom, Germany, Canada', 'Ireland, South Africa',
'Chile, United States, France', 'Belgium, France, Netherlands',
'United Kingdom, Ukraine, United States',
'Germany, Australia, France, China', 'Norway, United States',
'United States, Bermuda, Ecuador',
'United States, Hungary, Ireland, Canada',
'United Kingdom, Egypt, United States',
'United States, France, United Kingdom', 'Spain, Mexico, France',
'United States, South Africa', 'Hong Kong, China, Singapore',
'South Africa, China, United States', 'Denmark, France, Poland',
'New Zealand, United Kingdom',
'Netherlands, Denmark, South Africa', 'Iran, France',
'United Kingdom, United States, France, Germany',
'Australia, France', 'Ireland, United Kingdom, United States',
'United Kingdom, France, Germany', 'Canada, Luxembourg',
```

```
       'Brazil, Netherlands, United States, Colombia, Austria, Germany',
       'France, Canada, Belgium', 'Canada, France',
       'Bulgaria, United States, Spain, Canada', 'Sweden, Netherlands',
       'France, United States, Mexico',
       'Australia, United Kingdom, United Arab Emirates, Canada',
       'Australia, Armenia, Japan, Jordan, Mexico, Mongolia, New Zealand,
Philippines, South Africa, Sweden, United States, Uruguay',
       'India, Iran', 'France, Belgium, Spain',
       'Denmark, Sweden, Israel, United States', 'United States, Iceland',
       'United Kingdom, Russia',
       'United States, Israel, Italy, South Africa',
       'Netherlands, Denmark, France, Germany', 'South Korea, Japan',
       'United Kingdom, Pakistan', 'France, New Zealand',
       'United Kingdom, Czech Republic, United States, Germany, Bahamas',
       'China, Germany, India, United States', 'Germany, Sri Lanka',
       'United States, India, Bangladesh',
       'United States, Canada, France', 'Brazil, France, Germany',
       'Germany, United States, Hong Kong, Singapore',
       'France, Germany, Switzerland',
       'Germany, France, Luxembourg, United Kingdom, United States',
       'United Kingdom, Canada, Italy', 'Czech Republic, France',
       'Taiwan, Hong Kong, United States, China', 'Germany, Australia',
       'United Kingdom, Poland, United States', 'Denmark, Zimbabwe',
       'United Kingdom, South Africa',
       'Finland, Sweden, Norway, Latvia, Germany',
       'South Africa, United States, New Zealand, Canada',
       'United States, Italy, United Kingdom, Liechtenstein',
       'Denmark, France, Belgium, Italy, Netherlands, United States, United
Kingdom',
       'United States, Australia, Mexico',
       'United Kingdom, Czech Republic, Germany, United States',
       'France, China, Japan, United States',
       'United States, South Korea, China', 'Germany, Belgium',
       'Pakistan, Norway, United States',
       'United States, Canada, Belgium, United Kingdom', 'Venezuela',
       'Canada, France, Italy, Morocco, United States',
       'Canada, Spain, France', 'United States, Indonesia',
       'Spain, France, Italy',
       'United Arab Emirates, United States, United Kingdom',
       'United Kingdom, Israel, Russia', 'Spain, Cuba',
       'United States, Brazil', 'United States, France, Mexico',
       'United States, Nicaragua',
       'United Kingdom, United States, Spain, Germany, Greece, Canada',
       'Italy, Canada, France',
       'United Kingdom, Denmark, Canada, Croatia', 'Italy, Germany',
       'United States, France, United Kingdom, Japan',
       'United States, United Kingdom, Denmark, Sweden',
```

```
'United States, United Kingdom, Italy',
'United States, France, Canada, Spain',
'Russia, United States, China', 'United States, Canada, Germany',
'Ireland, United States', 'United States, United Arab Emirates',
'United States, Ireland',
'Ireland, United Kingdom, Italy, United States', 'Poland,',
'Slovenia, Croatia, Germany, Czech Republic, Qatar',
'Canada, United Kingdom, Netherlands',
'United States, Spain, Germany', 'India, Japan',
'China, South Korea, United States',
'United Kingdom, France, Belgium',
'Canada, Ireland, United States',
'United Kingdom, United States, Dominican Republic',
'United States, Senegal', 'Germany, United Kingdom, United States',
'South Africa, Germany, Netherlands, France',
'Canada, United States, United Kingdom, France, Luxembourg',
'Ireland, United States, France', 'Germany, United States, Canada',
'United Kingdom, Germany, Canada, United States',
'United States, France, Canada, Lebanon, Qatar',
'Netherlands, Belgium, United Kingdom, United States',
'France, Belgium, China, United States',
'United States, Chile, Israel',
'United Kingdom, Norway, Denmark, Germany, Sweden',
'Norway, Denmark, Sweden', 'China, India, Nepal',
'Colombia, Mexico, United States', 'United Kingdom, South Korea',
'Denmark, China', 'United States, Greece, Brazil',
'South Korea, France',
'United States, Australia, Samoa, United Kingdom',
'Germany, United Kingdom', 'Argentina, Chile, Peru',
'Turkey, Azerbaijan', 'Poland, West Germany',
'Germany, United States, Sweden', 'Canada, Spain',
'United States, Cambodia', 'United States, Greece',
'Norway, United Kingdom, France, Ireland',
'United Kingdom, Poland', 'Israel, Sweden, Germany, Netherlands',
'Switzerland, France', 'Italy, India', 'United States, Botswana',
'Chile, Argentina, France, Spain, United States',
'United States, India, South Korea, China',
'Denmark, Germany, Belgium, United Kingdom, France',
'Denmark, Germany, Belgium, United Kingdom, France, Sweden',
'France, Switzerland, Spain, United States, United Arab Emirates',
'Brazil, India, China, United States',
'Denmark, France, United States, Sweden', 'Australia, Iraq',
'China, Morocco, Hong Kong', 'Canada, United States, Germany',
'United Kingdom, Thailand', 'Venezuela, Colombia',
'Colombia, United States',
'France, Germany, Czech Republic, Belgium',
'Switzerland, Vatican City, Italy, Germany, France',
```

```
        'Portugal, France, Poland, United States',
        'United States, New Zealand, Japan',
        'United States, Netherlands, Japan, France', 'India, Switzerland',
        'Canada, India', 'United States, Morocco',
        'Singapore, Japan, France',
        'Canada, Mexico, Germany, South Africa',
        'United Kingdom, United States, Canada',
        'Germany, France, United States, Canada, United Kingdom',
        'United States, Uruguay', 'India, Canada',
        'Ireland, Canada, United Kingdom, United States',
        'United States, Germany, Australia', 'Australia, France, Ireland',
        'Australia, India', 'United States, United Kingdom, Canada, Japan',
        'Sweden, United Kingdom, Finland', 'Hong Kong, Taiwan',
        'United States, United Kingdom, Spain, South Korea', 'Guatemala',
        'Ukraine',
        'Italy, South Africa, West Germany, Australia, United States',
        'United States, Germany, United Kingdom, Australia',
        'Italy, France, Switzerland', 'Canada, France, United States',
        'Switzerland, United States', 'Thailand, Canada, United States',
        'China, Hong Kong, United States', 'United Kingdom, New Zealand',
        'Czech Republic, United Kingdom, France',
        'Australia, United Kingdom, Canada', 'Jamaica, United States',
        'Australia, United Kingdom, United States, New Zealand, Italy, France',
        'France, United States, Canada',
        'United Kingdom, France, Canada, Belgium, United States',
        'Denmark, United Kingdom, Sweden', 'United States, Hong Kong',
        'United States, Kazakhstan',
        'Argentina, France, United States, Germany, Qatar',
        'United States, Germany, United Kingdom',
        'United States, Germany, United Kingdom, Italy',
        'United States, New Zealand, United Kingdom',
        'Finland, United States', 'Spain, France, Uruguay',
        'France, Canada, United States', 'United States, Canada, China',
        'Ireland, Canada, Luxembourg, United States, United Kingdom, Philippines,
India',
        'United States, Czech Republic, United Kingdom', 'Israel, Germany',
        'Mexico, France',
        'Israel, Germany, Poland, Luxembourg, Belgium, France, United States',
        'Austria, United States', 'United Kingdom, Lithuania',
        'United States, Greece, United Kingdom',
        'United Kingdom, China, United States, India',
        'United States, Sweden, Norway',
        'United Kingdom, United States, Morocco',
        'United States, United Kingdom, Morocco',
        'Spain, Canada, United States',
        'United States, India, United Arab Emirates',
        'United Kingdom, Canada, France, United States',
```

```
        'India, Germany, France',
        'Belgium, Ireland, Netherlands, Germany, Afghanistan',
        'France, Canada, Italy, United States, China',
        'Ireland, United Kingdom, Greece, France, Netherlands',
        'Denmark, Indonesia, Finland, Norway, United Kingdom, Israel, France,
    United States, Germany, Netherlands',
        'New Zealand, United States',
        'United States, Australia, South Africa, United Kingdom',
        'United States, Germany, Mexico',
        'Somalia, Kenya, Sudan, South Africa, United States',
        'United States, Canada, Japan, Panama',
        'United Kingdom, Spain, Belgium', 'Serbia, South Korea, Slovenia',
        'Denmark, United Kingdom, South Africa, Sweden, Belgium',
        'Germany, Canada, United States',
        'Ireland, Canada, United States, United Kingdom',
        'New Zealand, United Kingdom, Australia',
        'United Kingdom, Australia, Canada, United States',
        'Germany, United States, Italy', 'United States, Venezuela',
        'United Kingdom, Canada, Japan',
        'United Kingdom, United States, Czech Republic',
        'United Kingdom, China, United States',
        'United Kingdom, Brazil, Germany',
        'United Kingdom, Namibia, South Africa, Zimbabwe, United States',
        'Canada, United States, India, United Kingdom',
        'Switzerland, United Kingdom, United States',
        'United Kingdom, India, Sweden',
        'United States, Brazil, India, Uganda, China',
        'Peru, United States, United Kingdom',
        'Germany, United States, United Kingdom, Canada',
        'Canada, India, Thailand, United States, United Arab Emirates',
        'United States, East Germany, West Germany',
        'France, Netherlands, South Africa, Finland',
        'Egypt, Austria, United States', 'Russia, Spain',
        'Croatia, Slovenia, Serbia, Montenegro', 'Japan, Canada',
        'United States, France, South Korea, Indonesia',
        'United Arab Emirates, Jordan'], dtype=object)
```

[20]:
```
# if we observe in above unique values in country where countries united␣
 ↪kingdom has repeated two times one with comma
df['country']=df['country'].str.replace("'",'')
```

[21]:
```
df['country'].unique()
```

[21]:
```
array(['United States', 'South Africa', nan, 'India',
        'United States, Ghana, Burkina Faso, United Kingdom, Germany, Ethiopia',
        'United Kingdom', 'Germany, Czech Republic', 'Mexico', 'Turkey',
        'Australia', 'United States, India, France', 'Finland',
```

'China, Canada, United States',
'South Africa, United States, Japan', 'Nigeria', 'Japan',
'Spain, United States', 'France', 'Belgium',
'United Kingdom, United States', 'United States, United Kingdom',
'France, United States', 'South Korea', 'Spain',
'United States, Singapore', 'United Kingdom, Australia, France',
'United Kingdom, Australia, France, United States',
'United States, Canada', 'Germany, United States',
'South Africa, United States', 'United States, Mexico',
'United States, Italy, France, Japan',
'United States, Italy, Romania, United Kingdom',
'Australia, United States', 'Argentina, Venezuela',
'United States, United Kingdom, Canada', 'China, Hong Kong',
'Russia', 'Canada', 'Hong Kong', 'United States, China, Hong Kong',
'Italy, United States', 'United States, Germany',
'United Kingdom, Canada, United States', ', South Korea',
'Ireland', 'India, Nepal',
'New Zealand, Australia, France, United States', 'Italy',
'Italy, Brazil, Greece', 'Argentina', 'Jordan', 'Colombia',
'United States, Japan', 'Belgium, United Kingdom',
'Switzerland, United Kingdom, Australia', 'Israel, United States',
'Canada, United States', 'Brazil', 'Argentina, Spain', 'Taiwan',
'United States, Nigeria', 'Bulgaria, United States',
'Spain, United Kingdom, United States', 'United States, China',
'United States, France',
'Spain, France, United Kingdom, United States',
', France, Algeria', 'Poland', 'Germany',
'France, Israel, Germany, United States, United Kingdom',
'New Zealand', 'Saudi Arabia', 'Thailand', 'Indonesia',
'Egypt, Denmark, Germany', 'United States, Switzerland',
'Hong Kong, Canada, United States', 'Kuwait, United States',
'France, Canada, United States, Spain',
'France, Netherlands, Singapore', 'France, Belgium',
'Ireland, United States, United Kingdom', 'Egypt', 'Malaysia',
'Israel', 'Australia, New Zealand', 'United Kingdom, Germany',
'Belgium, Netherlands', 'South Korea, Czech Republic',
'Australia, Germany', 'Vietnam', 'United Kingdom, Belgium',
'United Kingdom, Australia, United States',
'France, Japan, United States',
'United Kingdom, Germany, Spain, United States',
'United Kingdom, United States, France, Italy',
'United States, Germany, Canada',
'United States, France, Italy, United Kingdom',
'United States, United Kingdom, Germany, Hungary',
'United States, New Zealand', 'Sweden', 'China', 'Lebanon',
'Romania', 'Finland, Germany', 'Lebanon, Syria', 'Philippines',
'Iceland', 'Denmark', 'United States, India',

```
'Philippines, Singapore, Indonesia',
'China, United States, Canada', 'Lebanon, United Arab Emirates',
'Canada, United States, Denmark', 'United Arab Emirates',
'Mexico, France, Colombia', 'Netherlands',
'Germany, United States, France', 'United States, Bulgaria',
'United Kingdom, France, Germany, United States',
'Norway, Denmark', 'Syria, France, Lebanon, Qatar',
'United States, Czech Republic', 'Mauritius',
'Canada, South Africa', 'Austria', 'Mexico, Brazil',
'Germany, France', 'Mexico, United States',
'United Kingdom, France, Spain, United States',
'United States, Australia',
'United States, United Kingdom, France', 'United States, Russia',
'United States, United Kingdom, New Zealand',
'Australia, United Kingdom', 'Canada, Nigeria, United States',
'France, United States, United Kingdom, Canada',
'France, United Kingdom', 'India, United Kingdom',
'Canada, United States, Mexico',
'United Kingdom, Germany, United States',
'Czech Republic, United Kingdom, United States',
'China, United Kingdom', 'Italy, United Kingdom', 'China, Taiwan',
'United States, Brazil, Japan, Spain, India',
'United States, China, United Kingdom', 'Cameroon',
'Lebanon, Palestine, Denmark, Qatar', 'Japan, United States',
'Uruguay, Germany', 'Egypt, Saudi Arabia',
'United Kingdom, France, Poland, Germany, United States',
'Ireland, Switzerland, United Kingdom, France, United States',
'United Kingdom, South Africa, France',
'Ireland, United Kingdom, France, Germany',
'Russia, United States', 'United Kingdom, United States, France',
'United Kingdom,', 'United States, India, United Kingdom', 'Kenya',
'Spain, Argentina', 'India, United Kingdom, France, Qatar',
'Belgium, France', 'Argentina, Chile', 'United States, Thailand',
'Chile, Brazil', 'United States, Colombia',
'Canada, United States, United Kingdom', 'Uruguay', 'Luxembourg',
'United States, Cambodia, Romania', 'Bangladesh',
'Spain, Belgium, United States',
'United Kingdom, United States, Australia',
'Canada, United States, France', 'Portugal, United States',
'Portugal, Spain', 'India, United States',
'United Kingdom, Ireland', 'United Kingdom, Spain, United States',
'Hungary, United States', 'United States, South Korea',
'Canada, United States, Cayman Islands', 'India, France',
'France, Canada', 'Canada, Hungary, United States', 'Norway',
'Canada, United Kingdom, United States',
'United Kingdom, Germany, France, United States',
'Denmark, United States', 'Senegal', 'France, Algeria',
```

```
       'United Kingdom, Finland, Germany, United States, Australia, Japan,
France, Ireland',
       'Philippines, Canada, United Kingdom, United States',
       'Ireland, France, Iceland, United States, Mexico, Belgium, United
Kingdom, Hong Kong',
       'Singapore', 'Kuwait', 'United States, France, Serbia',
       'United States, Italy', 'Spain, Italy',
       'United States, Ireland, United Kingdom, India',
       'United Kingdom, Singapore', 'Hong Kong, United States',
       'United States, Malta, France, United Kingdom',
       'United States, China, Canada', 'Canada, United States, Ireland',
       'Lebanon, Canada, France', 'Japan, Canada, United States',
       'Spain, France, Canada',
       'Denmark, Singapore, Canada, United States',
       'United States, France, Denmark', 'United States, China, Colombia',
       'Spain, Thailand, United States', 'Mexico, Spain',
       'Ireland, Luxembourg, Belgium', 'China, United States',
       'Canada, Belgium', 'Canada, United Kingdom',
       'Lebanon, United Arab Emirates, France, Switzerland, Germany',
       'France, Belgium, Italy',
       'Lebanon, United States, United Arab Emirates', 'Lebanon, France',
       'France, Lebanon', 'France, Lebanon, United Kingdom',
       'France, Norway, Lebanon, Belgium',
       'Sweden, Czech Republic, United Kingdom, Denmark, Netherlands',
       'United States, United Kingdom, India', 'Indonesia, Netherlands',
       'Turkey, South Korea', 'Serbia, United States', 'Namibia',
       'United Kingdom, Kenya', 'United Kingdom, France, Germany, Spain',
       'United Kingdom, France, United States, Belgium, Luxembourg, China,
Germany',
       'Thailand, United States',
       'United States, France, Canada, Belgium', 'United Kingdom, China',
       'Germany, China, United Kingdom',
       'Australia, New Zealand, United States',
       'Hong Kong, Iceland, United States', 'France, Australia, Germany',
       'United States, Belgium, Canada, France', 'South Africa, Angola',
       'United States, Philippines',
       'United States, United Kingdom, Canada, China',
       'United States, Canada, United Kingdom', 'Turkey, United States',
       'Peru, Germany, Norway', 'Mozambique', 'Brazil, France',
       'China, Spain, South Korea, United States', 'Spain, Germany',
       'Hong Kong, China', 'France, Belgium, Luxembourg, Cambodia,',
       'United Kingdom, Australia', 'Belarus',
       'Indonesia, United Kingdom',
       'Switzerland, France, Belgium, United States', 'Ghana',
       'Spain, France, Canada, United States', 'Chile, Italy',
       'United Kingdom, Nigeria', 'Chile', 'France, Egypt',
       'Egypt, France', 'France, Brazil, Spain, Belgium',
```

```
       'Egypt, Algeria', 'Canada, South Korea, United States',
       'Nigeria, United Kingdom', 'United States, France, Canada',
       'Poland, United States',
       'United Arab Emirates, Jordan, Lebanon, Saudi Arabia',
       'United States, Mexico, Spain, Malta',
       'Saudi Arabia, United Arab Emirates', 'Zimbabwe',
       'United Kingdom, Germany, United Arab Emirates, New Zealand',
       'Romania, United States', 'Canada, Nigeria',
       'Saudi Arabia, Netherlands, Germany, Jordan, United Arab Emirates, United
States',
       'United Kingdom, Spain', 'Finland, France',
       'United Kingdom, Germany, United States, France',
       'India, United Kingdom, China, Canada, Japan, South Korea, United
States',
       'Italy, United Kingdom, France', 'United States, Mexico, Colombia',
       'Turkey, India', 'Italy, Turkey',
       'United Kingdom, United States, Japan',
       'France, Belgium, United States',
       'Puerto Rico, United States, Colombia', 'Uruguay, Argentina',
       'United States, United Kingdom, Japan', 'United States, Argentina',
       'United Kingdom, Italy', 'Ireland, United Kingdom',
       'United Kingdom, France, Belgium, Canada, United States',
       'Netherlands, Germany, Denmark, United Kingdom', 'Hungary',
       'Austria, Germany', 'Taiwan, China',
       'United Kingdom, United States, Ireland',
       'South Korea, United States', 'Brazil, United Kingdom',
       'Pakistan, United States', 'Romania, France, Switzerland, Germany',
       'Romania, United Kingdom', 'France, Malta, United States',
       'Cyprus',
       'United Kingdom, France, Belgium, Ireland, United States',
       'United States, Norway, Canada', 'Kenya, United States',
       'France, South Korea, Japan, United States', 'Taiwan, Malaysia',
       'Uruguay, Argentina, Germany, Spain',
       'United States, United Kingdom, France, Germany, Japan',
       'United States, France, Japan',
       'United Kingdom, France, United States',
       'Spain, France, United States',
       'Indonesia, South Korea, Singapore', 'United States, Spain',
       'Netherlands, Germany, Italy, Canada',
       'Spain, Germany, Denmark, United States', 'Norway, Sweden',
       'South Korea, Canada, United States, China',
       'Argentina, Uruguay, Serbia', 'France, Japan',
       'Mauritius, South Africa', 'United States, Poland',
       'United Kingdom, United States, Germany, Denmark, Belgium, Japan',
       'India, Germany', 'India, United Kingdom, Canada, United States',
       'Philippines, United States', 'Romania, Bulgaria, Hungary',
       'Uruguay, Guatemala', 'France, Senegal, Belgium',
```

```
'United Kingdom, Canada', 'Mexico, United States, Spain, Colombia',
'Canada, Norway', 'Singapore, United States',
'Finland, Germany, Belgium', 'United Kingdom, France',
'United States, Chile', 'United Kingdom, Japan, United States',
'Spain, United Kingdom', 'Argentina, United States, Mexico',
'United States, South Korea, Japan', 'Canada, Australia',
'United Kingdom, Hungary, Australia', 'Italy, Belgium',
'United States, United Kingdom, Germany', 'Switzerland',
'Singapore, Malaysia',
'France, Belgium, Luxembourg, Romania, Canada, United States',
'South Africa, Nigeria', 'Spain, France',
'United Kingdom, Hong Kong', 'Pakistan', 'Brazil, United States',
'Denmark, Brazil, France, Portugal, Sweden', 'India, Turkey',
'Malaysia, Singapore, Hong Kong', 'Philippines, Singapore',
'Australia, Canada', 'Taiwan, China, France, United States',
'Germany, Italy', 'Colombia, Peru, United Kingdom',
'Thailand, China, United States', 'Argentina, United States',
'Sweden, United States', 'Uruguay, Spain, Mexico',
'France, Luxembourg, Canada', 'Denmark, Spain', 'Chile, Argentina',
'United Kingdom, Belgium, Sweden', 'Canada, Brazil',
'Italy, France', 'Canada, Germany',
'Pakistan, United Arab Emirates', 'Ghana, United States',
'Mexico, Finland', 'United Arab Emirates, United Kingdom, India',
'Netherlands, Belgium', 'United States, Taiwan',
'Austria, Iraq, United States', 'United Kingdom, Malawi',
'Paraguay, Argentina', 'United Kingdom, Russia, United States',
'India, Pakistan', 'Indonesia, Singapore', 'Spain, Belgium',
'Iceland, Sweden, Belgium', 'Croatia', 'Uruguay, Argentina, Spain',
'United Kingdom, Ireland, United States',
'Canada, Germany, France, United States', 'United Kingdom, Japan',
'Norway, Denmark, Netherlands, Sweden',
'Hong Kong, China, United States', 'Ireland, Canada',
'Italy, Switzerland, France, Germany', 'Mexico, Netherlands',
'United States, Sweden', 'Germany, France, Russia',
'France, Iran, United States', 'United Kingdom, India',
'Russia, Poland, Serbia', 'Spain, Portugal', 'Peru',
'Mexico, Argentina',
'United Kingdom, Canada, United States, Cayman Islands',
'Indonesia, United States',
'United States, Israel, United Kingdom, Canada',
'Norway, Iceland, United States', 'Czech Republic, United States',
'United Kingdom, India, United States',
'United Kingdom, West Germany', 'India, Australia',
'United States,', 'Belgium, United Kingdom, United States',
'India, Germany, Austria',
'United States, Brazil, South Korea, Mexico, Japan, Germany',
'Spain, Mexico', 'China, Japan', 'Argentina, France',
```

```
'China, United States, United Kingdom',
'France, Luxembourg, United States',
'China, United States, Australia', 'Colombia, Mexico',
'United States, Canada, Ireland', 'Chile, Peru',
'Argentina, Italy', 'Canada, Japan, United States',
'United Kingdom, Canada, United States, Germany',
'Italy, Switzerland, Albania, Poland',
'United States, Japan, Canada', 'Cambodia',
'Italy, United States, Argentina',
'Saudi Arabia, Syria, Egypt, Lebanon, Kuwait',
'United States, Canada, Indonesia, United Kingdom, China, Singapore',
'Spain, Colombia',
'United Kingdom, South Africa, Australia, United States',
'Bulgaria', 'Argentina, Brazil, France, Poland, Germany, Denmark',
'United Kingdom, Spain, United States, Germany',
'Philippines, Qatar', 'Netherlands, Belgium, Germany, Jordan',
'United Arab Emirates, United States', 'Norway, Germany, Sweden',
'South Korea, China', 'Georgia', 'Soviet Union, India',
'Australia, United Arab Emirates', 'Canada, Germany, South Africa',
'South Korea, China, United States', 'India, Soviet Union',
'India, Mexico', 'Georgia, Germany, France',
'United Arab Emirates, Romania', 'India, Malaysia',
'Germany, Jordan, Netherlands', 'Turkey, France, Germany, Poland',
'Greece, United States', 'France, United Kingdom, United States',
'Norway, Germany', 'France, Morocco', 'Cambodia, United States',
'United States, Denmark', 'United States, Colombia, Mexico',
'United Kingdom, Italy, Israel, Peru, United States',
'Argentina, Uruguay, Spain, France',
'United Kingdom, France, United States, Belgium',
'France, Canada, China, Cambodia',
'United Kingdom, France, Belgium, United States', 'Chile, France',
'Netherlands, United States', 'France, United Kingdom, India',
'Czech Republic, Slovakia', 'Singapore, France',
'Spain, Switzerland', 'United States, Australia, China',
'South Africa, United States, Germany',
'United States, United Kingdom, Australia',
'Spain, Italy, Argentina', 'Chile, Spain, Argentina, Germany',
'West Germany', 'Austria, Czech Republic', 'Lebanon, Qatar',
'United Kingdom, Jordan, Qatar, Iran',
'France, South Korea, Japan', 'Israel, Germany, France',
'Canada, Japan, Netherlands', 'United States, Hungary',
'France, Germany', 'France, Qatar',
'United Kingdom, Germany, Canada', 'Ireland, South Africa',
'Chile, United States, France', 'Belgium, France, Netherlands',
'United Kingdom, Ukraine, United States',
'Germany, Australia, France, China', 'Norway, United States',
'United States, Bermuda, Ecuador',
```

```
       'United States, Hungary, Ireland, Canada',
       'United Kingdom, Egypt, United States',
       'United States, France, United Kingdom', 'Spain, Mexico, France',
       'United States, South Africa', 'Hong Kong, China, Singapore',
       'South Africa, China, United States', 'Denmark, France, Poland',
       'New Zealand, United Kingdom',
       'Netherlands, Denmark, South Africa', 'Iran, France',
       'United Kingdom, United States, France, Germany',
       'Australia, France', 'Ireland, United Kingdom, United States',
       'United Kingdom, France, Germany', 'Canada, Luxembourg',
       'Brazil, Netherlands, United States, Colombia, Austria, Germany',
       'France, Canada, Belgium', 'Canada, France',
       'Bulgaria, United States, Spain, Canada', 'Sweden, Netherlands',
       'France, United States, Mexico',
       'Australia, United Kingdom, United Arab Emirates, Canada',
       'Australia, Armenia, Japan, Jordan, Mexico, Mongolia, New Zealand,
Philippines, South Africa, Sweden, United States, Uruguay',
       'India, Iran', 'France, Belgium, Spain',
       'Denmark, Sweden, Israel, United States', 'United States, Iceland',
       'United Kingdom, Russia',
       'United States, Israel, Italy, South Africa',
       'Netherlands, Denmark, France, Germany', 'South Korea, Japan',
       'United Kingdom, Pakistan', 'France, New Zealand',
       'United Kingdom, Czech Republic, United States, Germany, Bahamas',
       'China, Germany, India, United States', 'Germany, Sri Lanka',
       'United States, India, Bangladesh',
       'United States, Canada, France', 'Brazil, France, Germany',
       'Germany, United States, Hong Kong, Singapore',
       'France, Germany, Switzerland',
       'Germany, France, Luxembourg, United Kingdom, United States',
       'United Kingdom, Canada, Italy', 'Czech Republic, France',
       'Taiwan, Hong Kong, United States, China', 'Germany, Australia',
       'United Kingdom, Poland, United States', 'Denmark, Zimbabwe',
       'United Kingdom, South Africa',
       'Finland, Sweden, Norway, Latvia, Germany',
       'South Africa, United States, New Zealand, Canada',
       'United States, Italy, United Kingdom, Liechtenstein',
       'Denmark, France, Belgium, Italy, Netherlands, United States, United
Kingdom',
       'United States, Australia, Mexico',
       'United Kingdom, Czech Republic, Germany, United States',
       'France, China, Japan, United States',
       'United States, South Korea, China', 'Germany, Belgium',
       'Pakistan, Norway, United States',
       'United States, Canada, Belgium, United Kingdom', 'Venezuela',
       'Canada, France, Italy, Morocco, United States',
       'Canada, Spain, France', 'United States, Indonesia',
```

```
'Spain, France, Italy',
'United Arab Emirates, United States, United Kingdom',
'United Kingdom, Israel, Russia', 'Spain, Cuba',
'United States, Brazil', 'United States, France, Mexico',
'United States, Nicaragua',
'United Kingdom, United States, Spain, Germany, Greece, Canada',
'Italy, Canada, France',
'United Kingdom, Denmark, Canada, Croatia', 'Italy, Germany',
'United States, France, United Kingdom, Japan',
'United States, United Kingdom, Denmark, Sweden',
'United States, United Kingdom, Italy',
'United States, France, Canada, Spain',
'Russia, United States, China', 'United States, Canada, Germany',
'Ireland, United States', 'United States, United Arab Emirates',
'United States, Ireland',
'Ireland, United Kingdom, Italy, United States', 'Poland,',
'Slovenia, Croatia, Germany, Czech Republic, Qatar',
'Canada, United Kingdom, Netherlands',
'United States, Spain, Germany', 'India, Japan',
'China, South Korea, United States',
'United Kingdom, France, Belgium',
'Canada, Ireland, United States',
'United Kingdom, United States, Dominican Republic',
'United States, Senegal', 'Germany, United Kingdom, United States',
'South Africa, Germany, Netherlands, France',
'Canada, United States, United Kingdom, France, Luxembourg',
'Ireland, United States, France', 'Germany, United States, Canada',
'United Kingdom, Germany, Canada, United States',
'United States, France, Canada, Lebanon, Qatar',
'Netherlands, Belgium, United Kingdom, United States',
'France, Belgium, China, United States',
'United States, Chile, Israel',
'United Kingdom, Norway, Denmark, Germany, Sweden',
'Norway, Denmark, Sweden', 'China, India, Nepal',
'Colombia, Mexico, United States', 'United Kingdom, South Korea',
'Denmark, China', 'United States, Greece, Brazil',
'South Korea, France',
'United States, Australia, Samoa, United Kingdom',
'Germany, United Kingdom', 'Argentina, Chile, Peru',
'Turkey, Azerbaijan', 'Poland, West Germany',
'Germany, United States, Sweden', 'Canada, Spain',
'United States, Cambodia', 'United States, Greece',
'Norway, United Kingdom, France, Ireland',
'United Kingdom, Poland', 'Israel, Sweden, Germany, Netherlands',
'Switzerland, France', 'Italy, India', 'United States, Botswana',
'Chile, Argentina, France, Spain, United States',
'United States, India, South Korea, China',
```

```
        'Denmark, Germany, Belgium, United Kingdom, France',
        'Denmark, Germany, Belgium, United Kingdom, France, Sweden',
        'France, Switzerland, Spain, United States, United Arab Emirates',
        'Brazil, India, China, United States',
        'Denmark, France, United States, Sweden', 'Australia, Iraq',
        'China, Morocco, Hong Kong', 'Canada, United States, Germany',
        'United Kingdom, Thailand', 'Venezuela, Colombia',
        'Colombia, United States',
        'France, Germany, Czech Republic, Belgium',
        'Switzerland, Vatican City, Italy, Germany, France',
        'Portugal, France, Poland, United States',
        'United States, New Zealand, Japan',
        'United States, Netherlands, Japan, France', 'India, Switzerland',
        'Canada, India', 'United States, Morocco',
        'Singapore, Japan, France',
        'Canada, Mexico, Germany, South Africa',
        'United Kingdom, United States, Canada',
        'Germany, France, United States, Canada, United Kingdom',
        'United States, Uruguay', 'India, Canada',
        'Ireland, Canada, United Kingdom, United States',
        'United States, Germany, Australia', 'Australia, France, Ireland',
        'Australia, India', 'United States, United Kingdom, Canada, Japan',
        'Sweden, United Kingdom, Finland', 'Hong Kong, Taiwan',
        'United States, United Kingdom, Spain, South Korea', 'Guatemala',
        'Ukraine',
        'Italy, South Africa, West Germany, Australia, United States',
        'United States, Germany, United Kingdom, Australia',
        'Italy, France, Switzerland', 'Canada, France, United States',
        'Switzerland, United States', 'Thailand, Canada, United States',
        'China, Hong Kong, United States', 'United Kingdom, New Zealand',
        'Czech Republic, United Kingdom, France',
        'Australia, United Kingdom, Canada', 'Jamaica, United States',
        'Australia, United Kingdom, United States, New Zealand, Italy, France',
        'France, United States, Canada',
        'United Kingdom, France, Canada, Belgium, United States',
        'Denmark, United Kingdom, Sweden', 'United States, Hong Kong',
        'United States, Kazakhstan',
        'Argentina, France, United States, Germany, Qatar',
        'United States, Germany, United Kingdom',
        'United States, Germany, United Kingdom, Italy',
        'United States, New Zealand, United Kingdom',
        'Finland, United States', 'Spain, France, Uruguay',
        'France, Canada, United States', 'United States, Canada, China',
        'Ireland, Canada, Luxembourg, United States, United Kingdom, Philippines,
India',
        'United States, Czech Republic, United Kingdom', 'Israel, Germany',
        'Mexico, France',
```

```
       'Israel, Germany, Poland, Luxembourg, Belgium, France, United States',
       'Austria, United States', 'United Kingdom, Lithuania',
       'United States, Greece, United Kingdom',
       'United Kingdom, China, United States, India',
       'United States, Sweden, Norway',
       'United Kingdom, United States, Morocco',
       'United States, United Kingdom, Morocco',
       'Spain, Canada, United States',
       'United States, India, United Arab Emirates',
       'United Kingdom, Canada, France, United States',
       'India, Germany, France',
       'Belgium, Ireland, Netherlands, Germany, Afghanistan',
       'France, Canada, Italy, United States, China',
       'Ireland, United Kingdom, Greece, France, Netherlands',
       'Denmark, Indonesia, Finland, Norway, United Kingdom, Israel, France,
United States, Germany, Netherlands',
       'New Zealand, United States',
       'United States, Australia, South Africa, United Kingdom',
       'United States, Germany, Mexico',
       'Somalia, Kenya, Sudan, South Africa, United States',
       'United States, Canada, Japan, Panama',
       'United Kingdom, Spain, Belgium', 'Serbia, South Korea, Slovenia',
       'Denmark, United Kingdom, South Africa, Sweden, Belgium',
       'Germany, Canada, United States',
       'Ireland, Canada, United States, United Kingdom',
       'New Zealand, United Kingdom, Australia',
       'United Kingdom, Australia, Canada, United States',
       'Germany, United States, Italy', 'United States, Venezuela',
       'United Kingdom, Canada, Japan',
       'United Kingdom, United States, Czech Republic',
       'United Kingdom, China, United States',
       'United Kingdom, Brazil, Germany',
       'United Kingdom, Namibia, South Africa, Zimbabwe, United States',
       'Canada, United States, India, United Kingdom',
       'Switzerland, United Kingdom, United States',
       'United Kingdom, India, Sweden',
       'United States, Brazil, India, Uganda, China',
       'Peru, United States, United Kingdom',
       'Germany, United States, United Kingdom, Canada',
       'Canada, India, Thailand, United States, United Arab Emirates',
       'United States, East Germany, West Germany',
       'France, Netherlands, South Africa, Finland',
       'Egypt, Austria, United States', 'Russia, Spain',
       'Croatia, Slovenia, Serbia, Montenegro', 'Japan, Canada',
       'United States, France, South Korea, Indonesia',
       'United Arab Emirates, Jordan'], dtype=object)
```

2.1 Filling null values of duration

[22]:
```
df[df['duration'].isnull()]
```

[22]:
```
     show_id   type                                 title    director  \
5541   s5542  Movie                   Louis C.K. 2017  Louis C.K.
5794   s5795  Movie               Louis C.K.: Hilarious  Louis C.K.
5813   s5814  Movie  Louis C.K.: Live at the Comedy Store  Louis C.K.

            cast        country          date_added  release_year  rating  \
5541  Louis C.K.  United States       April 4, 2017          2017  74 min
5794  Louis C.K.  United States  September 16, 2016          2010  84 min
5813  Louis C.K.  United States     August 15, 2016          2015  66 min

     duration listed_in                                        description
5541      NaN    Movies  Louis C.K. muses on religion, eternal love, gi…
5794      NaN    Movies  Emmy-winning comedy writer Louis C.K. brings h…
5813      NaN    Movies  The comic puts his trademark hilarious/thought…
```

[23]:
```
# rating cannot be in min when we observe rating with 74 min,84 min,66 min it␣
 ↪has null values in duration column happens due to poor DE
# Replace duration  with rating with min values
df['duration'].loc[df['rating']=='74 min']='74 min'
df[df['rating']=='74 min']
```

```
<ipython-input-23-7096ca40b69d>:3: 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
  df['duration'].loc[df['rating']=='74 min']='74 min'
```

[23]:
```
     show_id   type           title    director        cast        country  \
5541   s5542  Movie  Louis C.K. 2017  Louis C.K.  Louis C.K.  United States

         date_added  release_year  rating duration listed_in  \
5541  April 4, 2017          2017  74 min   74 min    Movies

                                        description
5541  Louis C.K. muses on religion, eternal love, gi…
```

[24]:
```
# same for 84 min and 66 min
df['duration'].loc[df['rating']=='84 min']='84 min'
df['duration'].loc[df['rating']=='66 min']='66 min'
```

```
<ipython-input-24-d96671e8143f>:2: 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
  df['duration'].loc[df['rating']=='84 min']='84 min'
<ipython-input-24-d96671e8143f>:3: 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
  df['duration'].loc[df['rating']=='66 min']='66 min'
```

2.2 Filling null values of rating

[25]:
```
# Ratings cannot CONTAIN min so making inplace as 'NR' i.e. Non Rated
df.loc[df['rating'].str.contains('min',na=False),'rating']='NR'
df['rating'].fillna('NR',inplace=True)
df.isna().sum()
```

[25]:
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             0
duration           0
listed_in          0
description        0
dtype: int64
```

2.3 conversion of datatypes

[26]:
```
df.nunique() # unique values in each column
```

[26]:
```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1767
release_year      74
rating            14
duration         220
listed_in        514
description     8775
```

```
dtype: int64
```

By seeing above unique values ,we can convert type

[27]:
```
#converting into 'categorical' data types
df['type']=df['type'].astype('category')
df['rating']=df['rating'].astype('category')
```

[28]:
```
df["date_added"] = df["date_added"].str.strip()
df["date_added"] = pd.to_datetime(df["date_added"])
```

[29]:
```
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   category
 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   datetime64[ns]
 7   release_year  8807 non-null   int64
 8   rating        8807 non-null   category
 9   duration      8807 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: category(2), datetime64[ns](1), int64(1), object(8)
memory usage: 706.1+ KB
```

2.4 Filling Null Values of date_added Column

[30]:
```
# Date_added column is imputed in the basis of release year
for year in df[df['date_added'].isnull()]['release_year'].unique() :
  imp=df[df['release_year']==year]['date_added'].mode().values[0]
  df.loc[df['release_year']==year,'date_added']=  df.
 ↪loc[df['release_year']==year,'date_added'].fillna(imp)
```

[31]:
```
df.isna().sum()
```

[31]:
```
show_id            0
type               0
title              0
director        2634
```

```
cast            825
country         831
date_added        0
release_year      0
rating            0
duration          0
listed_in         0
description       0
dtype: int64
```

2.5 Extracting month,week,year and day from added date

[32]:
```
df['month_added']=df['date_added'].dt.month
df['day_added']=df['date_added'].dt.day
df['week_added']=df['date_added'].dt.week
df['year_added']=df['date_added'].dt.year
```

```
<ipython-input-32-c23715942ca9>:3: FutureWarning: Series.dt.weekofyear and
Series.dt.week have been deprecated. Please use Series.dt.isocalendar().week
instead.
  df['week_added']=df['date_added'].dt.week
```

[33]:
```
df.head()
```

[33]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Documentaries	
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…	South Africa	2021-09-24	2021	TV-MA	2 Seasons		
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…	NaN	2021-09-24	2021	TV-MA	1 Season		
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	1 Season		
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…	India	2021-09-24	2021	TV-MA	2 Seasons		

```
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  month_added  day_added  \
0  As her father nears the end of his life, filmm…            9         25
1  After crossing paths at a party, a Cape Town t…            9         24
2  To protect his family from a powerful drug lor…            9         24
3  Feuds, flirtations and toilet talk go down amo…            9         24
4  In a city of coaching centers known to train I…            9         24

   week_added  year_added
0          38        2021
1          38        2021
2          38        2021
3          38        2021
4          38        2021
```

2.6 unnesting

Here we have four constraints i.e. values in directors ,country,listed_in and cast column

unnesting director column

[34]:
```
constraint1=df['director'].apply(lambda x :str(x).split(', ')).tolist()
df_new1=pd.DataFrame(constraint1,index=df['title'])
```

[35]:
```
df_new1.head(7)
```

[35]:
```
                                          0              1     2     3   \
title
Dick Johnson Is Dead              Kirsten Johnson           None  None  None
Blood & Water                             nan           None  None  None
Ganglands                       Julien Leclercq           None  None  None
Jailbirds New Orleans                     nan           None  None  None
Kota Factory                              nan           None  None  None
Midnight Mass                   Mike Flanagan             None  None  None
My Little Pony: A New Generation  Robert Cullen  José Luis Ucha  None  None

                                    4     5     6     7     8     9     10  \
title
Dick Johnson Is Dead              None  None  None  None  None  None  None
Blood & Water                     None  None  None  None  None  None  None
Ganglands                         None  None  None  None  None  None  None
Jailbirds New Orleans             None  None  None  None  None  None  None
Kota Factory                      None  None  None  None  None  None  None
Midnight Mass                     None  None  None  None  None  None  None
```

```
My Little Pony: A New Generation  None  None  None  None  None  None  None

                                    11    12
title
Dick Johnson Is Dead               None  None
Blood & Water                      None  None
Ganglands                          None  None
Jailbirds New Orleans              None  None
Kota Factory                       None  None
Midnight Mass                      None  None
My Little Pony: A New Generation   None  None
```

[36]:
```
df_new1=df_new1.stack()
```

[37]:
```
df_new1.head(8)
```

[37]:
```
title
Dick Johnson Is Dead               0    Kirsten Johnson
Blood & Water                      0                nan
Ganglands                          0    Julien Leclercq
Jailbirds New Orleans              0                nan
Kota Factory                       0                nan
Midnight Mass                      0     Mike Flanagan
My Little Pony: A New Generation   0     Robert Cullen
                                   1     José Luis Ucha
dtype: object
```

[38]:
```
df_new1=pd.DataFrame(df_new1.reset_index())
```

[39]:
```
df_new1.head(10)
```

[39]:

	title	level_1	0
0	Dick Johnson Is Dead	0	Kirsten Johnson
1	Blood & Water	0	nan
2	Ganglands	0	Julien Leclercq
3	Jailbirds New Orleans	0	nan
4	Kota Factory	0	nan
5	Midnight Mass	0	Mike Flanagan
6	My Little Pony: A New Generation	0	Robert Cullen
7	My Little Pony: A New Generation	1	José Luis Ucha
8	Sankofa	0	Haile Gerima
9	The Great British Baking Show	0	Andy Devonshire

[40]:
```
#renaming the director column and dropping the level column
df_new1.rename(columns={0:'Directors'},inplace=True)
df_new1.drop(['level_1'],axis=1,inplace=True)
```

[41]:
```
df_new1.head()
```

[41]:

	title	Directors
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan

[42]:
```
constraint2=df['cast'].apply(lambda x :str(x).split(', ')).tolist()
df_new2=pd.DataFrame(constraint2,index=df['title'])
df_new2=df_new2.stack()
df_new2=df_new2.reset_index()
```

[43]:
```
df_new2.head()
```

[43]:

	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

[44]:
```
df_new2.rename(columns={0:'Actors'},inplace=True)
df_new2.drop(['level_1'],axis=1,inplace=True)
df_new2.head()
```

[44]:

	title	Actors
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

[45]:
```
constraint3=df['listed_in'].apply(lambda x :str(x).split(', ')).tolist()
df_new3=pd.DataFrame(constraint3,index=df['title'])
df_new3=df_new3.stack()
df_new3=pd.DataFrame(df_new3.reset_index())
df_new3
```

[45]:

	title	level_1	0
0	Dick Johnson Is Dead	0	Documentaries
1	Blood & Water	0	International TV Shows
2	Blood & Water	1	TV Dramas
3	Blood & Water	2	TV Mysteries
4	Ganglands	0	Crime TV Shows
…	…	…	…

```
19318                  Zoom        0  Children & Family Movies
19319                  Zoom        1                  Comedies
19320                Zubaan        0                    Dramas
19321                Zubaan        1       International Movies
19322                Zubaan        2           Music & Musicals

[19323 rows x 3 columns]
```

[46]:
```
df_new3.rename(columns={0:'Genre'},inplace=True)
df_new3.drop(['level_1'],axis=1,inplace=True)
df_new3.head()
```

[46]:
```
                   title                    Genre
0  Dick Johnson Is Dead              Documentaries
1          Blood & Water  International TV Shows
2          Blood & Water                TV Dramas
3          Blood & Water             TV Mysteries
4              Ganglands           Crime TV Shows
```

[47]:
```
constraint4=df['country'].apply(lambda x :str(x).split(', ')).tolist()
df_new4=pd.DataFrame(constraint4,index=df['title'])
df_new4=df_new4.stack()
df_new4=pd.DataFrame(df_new4.reset_index())
df_new4.head()
```

[47]:
```
                    title  level_1              0
0   Dick Johnson Is Dead        0  United States
1           Blood & Water        0   South Africa
2               Ganglands        0            nan
3  Jailbirds New Orleans        0            nan
4            Kota Factory        0          India
```

[48]:
```
df_new4.rename(columns={0:'country'},inplace=True)
df_new4.drop(['level_1'],axis=1,inplace=True)
df_new4.head()
```

[48]:
```
                    title        country
0   Dick Johnson Is Dead  United States
1           Blood & Water   South Africa
2               Ganglands            nan
3  Jailbirds New Orleans            nan
4            Kota Factory          India
```

2.7 Merging dataframes with original

[49]:
```
#merging director data with actors data
df_new5=df_new1.merge(df_new2,on=['title'],how='inner')
#merging above merged data with genre data
df_new6=df_new5.merge(df_new3,on=['title'],how='inner')
#merging above merged data with country data
df_new=df_new6.merge(df_new4,on=['title'],how='inner')

df_new.head()
```

[49]:
```
                  title        Directors       Actors                    Genre  \
0  Dick Johnson Is Dead  Kirsten Johnson          nan           Documentaries
1         Blood & Water              nan   Ama Qamata  International TV Shows
2         Blood & Water              nan   Ama Qamata               TV Dramas
3         Blood & Water              nan   Ama Qamata            TV Mysteries
4         Blood & Water              nan  Khosi Ngema  International TV Shows

         country
0  United States
1   South Africa
2   South Africa
3   South Africa
4   South Africa
```

[50]:
```
# filling null values of above merged data
df_new['Actors'].replace(['nan'],['unknown Actor'],inplace=True)
df_new['Directors'].replace(['nan'],['unknown Director'],inplace=True)
df_new['country'].replace(['nan'],[np.nan],inplace=True)
df_new.head()
```

[50]:
```
                  title         Directors         Actors  \
0  Dick Johnson Is Dead   Kirsten Johnson  unknown Actor
1         Blood & Water  unknown Director     Ama Qamata
2         Blood & Water  unknown Director     Ama Qamata
3         Blood & Water  unknown Director     Ama Qamata
4         Blood & Water  unknown Director    Khosi Ngema

                    Genre        country
0           Documentaries  United States
1  International TV Shows   South Africa
2               TV Dramas   South Africa
3            TV Mysteries   South Africa
4  International TV Shows   South Africa
```

[51]:
```
# merging unnest data with original
```

```
df_final=df_new.
  ↪merge(df[['show_id','type','date_added','title','release_year','rating','duration','day_add
df_final.head()
```

[51]:

	title	Directors	Actors	Genre	country	show_id	type	date_added	release_year	rating	duration	day_added	month_added	week_added
0	Dick Johnson Is Dead	Kirsten Johnson	unknown Actor	Documentaries	United States	s1	Movie	2021-09-25	2020	PG-13	90 min	25	9	38
1	Blood & Water	unknown Director	Ama Qamata	International TV Shows	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	2 Seasons	24	9	38
2	Blood & Water	unknown Director	Ama Qamata	TV Dramas	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	2 Seasons	24	9	38
3	Blood & Water	unknown Director	Ama Qamata	TV Mysteries	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	2 Seasons	24	9	38
4	Blood & Water	unknown Director	Khosi Ngema	International TV Shows	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	2 Seasons	24	9	38

[52]:
```
len(df_final)
```

[52]:
```
201991
```

[53]:
```
df_final.isnull().sum()
```

[53]:
```
title              0
Directors          0
Actors             0
Genre              0
country        11897
show_id            0
type               0
date_added         0
release_year       0
rating             0
duration           0
day_added          0
month_added        0
week_added         0
dtype: int64
```

2.8 Filling Null Values of Country column

[54]:
```
# country column is imputed on the basis of director column
for director in df_final[df_final['country'].isnull()]['Directors'].unique():
  if director in  df_final[~df_final['country'].isnull()]['Directors'].unique():
    imp=df_final[df_final['Directors']==director]['country'].mode().values[0]
    df_final.loc[df_final['Directors']==director,'country']=df_final.
 ↪loc[df_final['Directors']==director,'country'].fillna(imp)
```

[55]:
```
df_final.isna().sum()
```

[55]:
```
title              0
Directors          0
Actors             0
Genre              0
country         4276
show_id            0
type               0
date_added         0
release_year       0
rating             0
duration           0
day_added          0
month_added        0
week_added         0
dtype: int64
```

[56]:
```
# filling country column remaining rows using actors column
for actor in df_final[df_final['country'].isnull()]['Actors'].unique():
  if actor in  df_final[~df_final['country'].isnull()]['Actors'].unique():
    imp=df_final[df_final['Actors']==actor]['country'].mode().values[0]
    df_final.loc[df_final['Actors']==actor,'country']=df_final.
 ↪loc[df_final['Actors']==actor,'country'].fillna(imp)
```

[57]:
```
df_final.isna().sum()
```

[57]:
```
title              0
Directors          0
Actors             0
Genre              0
country         2069
show_id            0
type               0
date_added         0
release_year       0
rating             0
duration           0
```

```
day_added          0
month_added        0
week_added         0
dtype: int64
```

[58]:
```
# filling remaining values as  unknown country
df_final['country'].fillna('unknown country',inplace=True)
df_final.isna().sum()
```

[58]:
```
title           0
Directors       0
Actors          0
Genre           0
country         0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
day_added       0
month_added     0
week_added      0
dtype: int64
```

[59]:
```
df_final.head()
```

[59]:

	title	Directors	Actors	Genre	country	show_id	type	date_added	release_year	rating	duration	day_added	month_added	week_added
0	Dick Johnson Is Dead	Kirsten Johnson	unknown Actor	Documentaries	United States	s1	Movie	2021-09-25	2020	PG-13	90 min	25	9	38
1	Blood & Water	unknown Director	Ama Qamata	International TV Shows	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	2 Seasons	24	9	38
2	Blood & Water	unknown Director	Ama Qamata	TV Dramas	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	2 Seasons	24	9	38
3	Blood & Water	unknown Director	Ama Qamata	TV Mysteries	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	2 Seasons	24	9	38
4	Blood & Water	unknown Director	Khosi Ngema	International TV Shows	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	2 Seasons	24	9	38

[60]:
```
# removing brackets in titles
df_final['title']=df_final['title'].str.replace(r'\(.*\)','')
```

```
<ipython-input-60-f9fc89cb5518>:2: FutureWarning: The default value of regex
will change from True to False in a future version.
  df_final['title']=df_final['title'].str.replace(r'\(.*\)','')
```

2.9 Dropping Duplicates

[61]:
```
df_final.drop_duplicates(inplace=True,ignore_index=True)
df_final.reset_index(drop=True)
```

[61]:

	title	Directors	Actors	\
0	Dick Johnson Is Dead	Kirsten Johnson	unknown Actor	
1	Blood & Water	unknown Director	Ama Qamata	
2	Blood & Water	unknown Director	Ama Qamata	
3	Blood & Water	unknown Director	Ama Qamata	
4	Blood & Water	unknown Director	Khosi Ngema	
…	…	…	…	
201931	Zubaan	Mozez Singh	Anita Shabdish	
201932	Zubaan	Mozez Singh	Anita Shabdish	
201933	Zubaan	Mozez Singh	Chittaranjan Tripathy	
201934	Zubaan	Mozez Singh	Chittaranjan Tripathy	
201935	Zubaan	Mozez Singh	Chittaranjan Tripathy	

	Genre	country	show_id	type	date_added	\
0	Documentaries	United States	s1	Movie	2021-09-25	
1	International TV Shows	South Africa	s2	TV Show	2021-09-24	
2	TV Dramas	South Africa	s2	TV Show	2021-09-24	
3	TV Mysteries	South Africa	s2	TV Show	2021-09-24	
4	International TV Shows	South Africa	s2	TV Show	2021-09-24	
…	…	…	…	…	…	
201931	International Movies	India	s8807	Movie	2019-03-02	
201932	Music & Musicals	India	s8807	Movie	2019-03-02	
201933	Dramas	India	s8807	Movie	2019-03-02	
201934	International Movies	India	s8807	Movie	2019-03-02	
201935	Music & Musicals	India	s8807	Movie	2019-03-02	

	release_year	rating	duration	day_added	month_added	week_added
0	2020	PG-13	90 min	25	9	38
1	2021	TV-MA	2 Seasons	24	9	38
2	2021	TV-MA	2 Seasons	24	9	38
3	2021	TV-MA	2 Seasons	24	9	38
4	2021	TV-MA	2 Seasons	24	9	38
…	…	…	…	…	…	
201931	2015	TV-14	111 min	2	3	9
201932	2015	TV-14	111 min	2	3	9

```
201933          2015  TV-14    111 min          2           3           9
201934          2015  TV-14    111 min          2           3           9
201935          2015  TV-14    111 min          2           3           9

[201936 rows x 14 columns]
```

2.10 Statistical Analysis

[62]:
```
df.head()
```

[62]:
```
  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

                                                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

  date_added  release_year rating   duration  \
0 2021-09-25          2020  PG-13     90 min
1 2021-09-24          2021  TV-MA  2 Seasons
2 2021-09-24          2021  TV-MA   1 Season
3 2021-09-24          2021  TV-MA   1 Season
4 2021-09-24          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  month_added  day_added  \
0  As her father nears the end of his life, filmm…            9         25
1  After crossing paths at a party, a Cape Town t…            9         24
2  To protect his family from a powerful drug lor…            9         24
3  Feuds, flirtations and toilet talk go down amo…            9         24
4  In a city of coaching centers known to train I…            9         24

   week_added  year_added
0          38        2021
```

```
1          38        2021
2          38        2021
3          38        2021
4          38        2021
```

[63]:
```
df['movie_duration']=df['duration']
df['movie_duration']=df['movie_duration'].str.replace('min','')
df.loc[df['movie_duration'].str.contains('Season'),'movie_duration']=0
df['movie_duration']=df['movie_duration'].astype(int)
```

[64]:
```
bins1=[-1,1,50,80,100,120,150,200,315]
 labels1=['Tv␣
 ↪series','1-50','50-80','80-100','100-120','120-150','150-200','200-315']
 df['duration_range']=pd.cut(df['movie_duration'],bins=bins1,labels=labels1)
 df.head()
```

[64]:
```
  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

                                                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

  date_added  release_year rating   duration  \
0 2021-09-25          2020  PG-13     90 min
1 2021-09-24          2021  TV-MA  2 Seasons
2 2021-09-24          2021  TV-MA   1 Season
3 2021-09-24          2021  TV-MA   1 Season
4 2021-09-24          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  month_added  day_added  \
0  As her father nears the end of his life, filmm…            9         25
1  After crossing paths at a party, a Cape Town t…            9         24
```

```
2  To protect his family from a powerful drug lor…           9         24
3  Feuds, flirtations and toilet talk go down amo…           9         24
4  In a city of coaching centers known to train I…           9         24

   week_added  year_added  movie_duration duration_range
0          38        2021              90         80-100
1          38        2021               0      Tv series
2          38        2021               0      Tv series
3          38        2021               0      Tv series
4          38        2021               0      Tv series
```

[65]:
```
df['seasons']=df['duration']
df.loc[df['duration'].str.contains('min'),'seasons']=np.nan
```

[66]:
```
df.describe()
```

[66]:
```
       release_year  month_added    day_added   week_added   year_added  \
count   8807.000000  8807.000000  8807.000000  8807.000000  8807.000000
mean    2014.180198     6.658113    12.490746    26.717611  2018.870785
std        8.819312     3.436811     9.889079    15.037763     1.574804
min     1925.000000     1.000000     1.000000     1.000000  2008.000000
25%     2013.000000     4.000000     1.000000    14.000000  2018.000000
50%     2017.000000     7.000000    13.000000    27.000000  2019.000000
75%     2019.000000    10.000000    20.000000    40.000000  2020.000000
max     2021.000000    12.000000    31.000000    53.000000  2021.000000

       movie_duration
count     8807.000000
mean        69.312252
std         51.519154
min          0.000000
25%          0.000000
50%         88.000000
75%        106.000000
max        312.000000
```

[67]:
```
df_final.nunique()
```

[67]:
```
title           8791
Directors       4994
Actors         36440
Genre             42
country          128
show_id         8807
type               2
date_added      1714
release_year      74
```

```
rating             14
duration          220
day_added          31
month_added        12
week_added         53
dtype: int64
```

[68]:
```
df_final['Genre'].unique()
```

[68]:
```
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)
```

[69]:
```
df_final['country'].unique()
```

[69]:
```
array(['United States', 'South Africa', 'France', 'India', 'Ghana',
       'Burkina Faso', 'United Kingdom', 'Germany', 'Ethiopia',
       'unknown country', 'Thailand', 'Czech Republic', 'Brazil',
       'Mexico', 'Turkey', 'Australia', 'Belgium', 'Finland', 'China',
       'Canada', 'Japan', 'Nigeria', 'Spain', 'Sweden', 'South Korea',
       'Singapore', 'Philippines', 'Italy', 'Romania', 'Argentina',
       'Venezuela', 'Angola', 'Mauritius', 'Hong Kong', 'Russia', '',
       'Ireland', 'Egypt', 'Taiwan', 'Nepal', 'New Zealand', 'Greece',
       'Jordan', 'Colombia', 'Switzerland', 'Israel', 'Bulgaria',
       'Algeria', 'Poland', 'Denmark', 'Saudi Arabia', 'Indonesia',
       'Kuwait', 'Cameroon', 'Netherlands', 'Malaysia', 'Vietnam',
       'Hungary', 'Lebanon', 'Syria', 'Iceland', 'United Arab Emirates',
       'Norway', 'Qatar', 'Austria', 'Palestine', 'Uruguay', 'Cuba',
       'United Kingdom,', 'Kenya', 'Chile', 'Luxembourg', 'Cambodia',
       'Bangladesh', 'Portugal', 'Cayman Islands', 'Senegal', 'Serbia',
       'Malta', 'Namibia', 'Peru', 'Mozambique', 'Cambodia,', 'Belarus',
       'Zimbabwe', 'Puerto Rico', 'Pakistan', 'Cyprus', 'Guatemala',
       'Iraq', 'Malawi', 'Paraguay', 'Croatia', 'Iran', 'West Germany',
       'United States,', 'Albania', 'Georgia', 'Soviet Union', 'Morocco',
       'Slovakia', 'Ukraine', 'Bermuda', 'Ecuador', 'Armenia', 'Mongolia',
       'Bahamas', 'Sri Lanka', 'Latvia', 'Liechtenstein', 'Nicaragua',
```

```
'Poland,', 'Slovenia', 'Dominican Republic', 'Samoa', 'Azerbaijan',
'Botswana', 'Vatican City', 'Jamaica', 'Kazakhstan', 'Lithuania',
'Afghanistan', 'Somalia', 'Sudan', 'Panama', 'Uganda',
'East Germany', 'Montenegro'], dtype=object)
```

[70]:
```
df_final.nunique()
```

[70]:
```
title           8791
Directors       4994
Actors         36440
Genre             42
country          128
show_id         8807
type               2
date_added      1714
release_year      74
rating            14
duration         220
day_added         31
month_added       12
week_added        53
dtype: int64
```

[71]:
```
df_final['rating'].unique()
```

[71]:
```
['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', …, 'G', 'NC-17', 'NR', 'TV-Y7-FV',
'UR']
Length: 14
Categories (14, object): ['G', 'NC-17', 'NR', 'PG', …, 'TV-Y', 'TV-Y7',
'TV-Y7-FV', 'UR']
```

[72]:
```
# finding the duration time of movies
df_final['movie_duration']=df_final['duration']
df_final['movie_duration']=df_final['movie_duration'].str.replace('min','')
df_final1=df_final.copy()
```

[73]:
```
df_final1.loc[df_final1['movie_duration'].str.
  ↪contains('Season'),'movie_duration']=0
df_final1['movie_duration']=df_final1['movie_duration'].astype(int)
```

[74]:
```
df[df['movie_duration']!=0]['movie_duration'].describe()
# Shows the analysis of movies streaming in netflix
```

[74]:
```
count    6131.000000
mean       99.564998
std        28.289504
min         3.000000
```

```
25%        87.000000
50%        98.000000
75%       114.000000
max       312.000000
Name: movie_duration, dtype: float64
```

[75]:
```
# Calculate the difference between 'Date_added' and 'Release_year' columns in days
df['Days_to_Netflix'] = (df['date_added'] - pd.to_datetime(df['release_year'], format='%Y')).dt.days

# Get the mode of 'Days_to_Netflix'
mode_days_to_netflix = df['Days_to_Netflix'].mode()[0]

# Print the mode
print("Mode of days to Netflix:", mode_days_to_netflix, "days")
df[df['Days_to_Netflix']>0]['Days_to_Netflix'].describe()
```

```
Mode of days to Netflix: 334 days
```

[75]:
```
count     8779.000000
mean      1902.604283
std       3213.351176
min          1.000000
25%        271.000000
50%        582.000000
75%       2076.500000
max      34331.000000
Name: Days_to_Netflix, dtype: float64
```

3 3. Non-Graphical Analysis: Value counts and unique attributes

[76]:
```
bins1=[-1,1,50,80,100,120,150,200,315]
 labels1=['Tv series','1-50','50-80','80-100','100-120','120-150','150-200','200-315']
 df_final1['duration_range']=pd.
 cut(df_final1['movie_duration'],bins=bins1,labels=labels1)
 df_final1.head()
```

[76]:

	title	Directors	Actors	\
0	Dick Johnson Is Dead	Kirsten Johnson	unknown Actor	
1	Blood & Water	unknown Director	Ama Qamata	
2	Blood & Water	unknown Director	Ama Qamata	
3	Blood & Water	unknown Director	Ama Qamata	
4	Blood & Water	unknown Director	Khosi Ngema	

```
                    Genre        country show_id     type date_added  \
0           Documentaries  United States      s1    Movie 2021-09-25
1  International TV Shows   South Africa      s2  TV Show 2021-09-24
2              TV Dramas   South Africa      s2  TV Show 2021-09-24
3            TV Mysteries   South Africa      s2  TV Show 2021-09-24
4  International TV Shows   South Africa      s2  TV Show 2021-09-24

   release_year rating   duration  day_added  month_added  week_added  \
0          2020  PG-13     90 min         25            9          38
1          2021  TV-MA  2 Seasons         24            9          38
2          2021  TV-MA  2 Seasons         24            9          38
3          2021  TV-MA  2 Seasons         24            9          38
4          2021  TV-MA  2 Seasons         24            9          38

   movie_duration duration_range
0              90         80-100
1               0      Tv series
2               0      Tv series
3               0      Tv series
4               0      Tv series
```

[77]:
```
df_final1.loc[~df_final1['duration'].str.
 ↪contains('Season'),'duration']=df_final1.loc[~df_final1['duration'].str.
 ↪contains('Season'),'duration_range']
df_final1.head()
```

[77]:
```
                 title         Directors         Actors  \
0  Dick Johnson Is Dead   Kirsten Johnson  unknown Actor
1         Blood & Water  unknown Director     Ama Qamata
2         Blood & Water  unknown Director     Ama Qamata
3         Blood & Water  unknown Director     Ama Qamata
4         Blood & Water  unknown Director    Khosi Ngema

                    Genre        country show_id     type date_added  \
0           Documentaries  United States      s1    Movie 2021-09-25
1  International TV Shows   South Africa      s2  TV Show 2021-09-24
2              TV Dramas   South Africa      s2  TV Show 2021-09-24
3            TV Mysteries   South Africa      s2  TV Show 2021-09-24
4  International TV Shows   South Africa      s2  TV Show 2021-09-24

   release_year rating   duration  day_added  month_added  week_added  \
0          2020  PG-13     80-100         25            9          38
1          2021  TV-MA  2 Seasons         24            9          38
2          2021  TV-MA  2 Seasons         24            9          38
3          2021  TV-MA  2 Seasons         24            9          38
4          2021  TV-MA  2 Seasons         24            9          38
```

```
   movie_duration duration_range
0              90         80-100
1               0      Tv series
2               0      Tv series
3               0      Tv series
4               0      Tv series
```

[78]:
```
df_final1['duration'].value_counts() # duration of movie or Tv series
```

[78]:
```
80-100        52931
100-120       48675
1 Season      35035
120-150       26691
2 Seasons      9559
50-80          7700
150-200        6737
3 Seasons      5084
1-50           2530
4 Seasons      2134
5 Seasons      1698
7 Seasons       843
6 Seasons       633
200-315         524
8 Seasons       286
9 Seasons       257
10 Seasons      220
13 Seasons      132
12 Seasons      111
15 Seasons       96
17 Seasons       30
11 Seasons       30
Name: duration, dtype: int64
```

[79]:
```
df_final1['duration_range'].value_counts()
```

[79]:
```
Tv series    56148
80-100       52931
100-120      48675
120-150      26691
50-80         7700
150-200       6737
1-50          2530
200-315        524
Name: duration_range, dtype: int64
```

[80]:
```
df_final['type'].value_counts()
```

[80]:
```
Movie      145788
TV Show     56148
Name: type, dtype: int64
```

[81]:
```
df_final['Genre'].value_counts()
```

[81]:
```
Dramas                          29756
International Movies            28192
Comedies                        20829
International TV Shows          12845
Action & Adventure              12216
Independent Movies               9818
Children & Family Movies         9771
TV Dramas                        8942
Thrillers                        7106
Romantic Movies                  6412
TV Comedies                      4963
Crime TV Shows                   4733
Horror Movies                    4571
Kids' TV                         4568
Sci-Fi & Fantasy                 4037
Music & Musicals                 3077
Romantic TV Shows                3049
Documentaries                    2407
Anime Series                     2313
TV Action & Adventure            2288
Spanish-Language TV Shows        2126
British TV Shows                 1808
Sports Movies                    1531
Classic Movies                   1434
TV Mysteries                     1281
Korean TV Shows                  1122
Cult Movies                      1077
TV Sci-Fi & Fantasy              1045
Anime Features                   1045
TV Horror                         941
Docuseries                        845
LGBTQ Movies                      838
TV Thrillers                      768
Teen TV Shows                     742
Reality TV                        735
Faith & Spirituality              719
Stand-Up Comedy                   540
Movies                            412
TV Shows                          337
Classic & Cult TV                 272
Stand-Up Comedy & Talk Shows      268
```

```
Science & Nature TV               157
Name: Genre, dtype: int64
```

[82]:
```
df_final['country'].value_counts()
```

[82]:
```
United States     65225
India             24121
United Kingdom    13023
Japan              9053
France             8369
                  …
Samoa                 2
Nicaragua             1
United States,        1
Kazakhstan            1
Uganda                1
Name: country, Length: 128, dtype: int64
```

[83]:
```
df['date_added'].value_counts()
```

[83]:
```
2020-01-01    110
2019-11-01     91
2018-03-01     75
2019-12-31     74
2018-10-01     71
              …
2017-02-21      1
2017-02-07      1
2017-01-29      1
2017-01-25      1
2020-01-11      1
Name: date_added, Length: 1714, dtype: int64
```

[84]:
```
df['release_year'].value_counts()
```

[84]:
```
2018    1147
2017    1032
2019    1030
2020     953
2016     902
        …
1959       1
1925       1
1961       1
1947       1
1966       1
Name: release_year, Length: 74, dtype: int64
```

[85]:
```
df['week_added'].value_counts()
```

[85]:
```
1     372
44    320
40    287
31    269
26    268
35    265
9     254
13    250
27    240
18    234
5     208
22    206
48    200
50    190
37    183
14    173
39    168
24    164
11    164
16    160
30    160
17    154
33    153
15    152
23    151
7     147
25    143
34    143
36    142
49    140
29    140
38    139
51    137
10    135
42    135
46    134
52    132
20    131
28    130
32    122
47    120
21    117
41    116
19    116
43    116
```

```
3     113
8     110
12    109
2     108
53    104
45     98
6      97
4      88
Name: week_added, dtype: int64
```

[86]:
```
df_dateadded=df.groupby(['date_added']).agg({'title':'nunique'}).reset_index().
  ↪sort_values(by=['title'],ascending=False).head(15)
df_dateadded
```

[86]:
```
      date_added  title
1147 2020-01-01    110
1092 2019-11-01     91
564  2018-03-01     75
1146 2019-12-31     74
737  2018-10-01     71
766  2018-11-01     62
1063 2019-10-01     62
1639 2021-07-01     60
1692 2021-09-01     58
518  2018-01-01     55
987  2019-07-01     52
1611 2021-06-02     51
1478 2021-01-01     49
448  2017-10-01     47
590  2018-04-01     44
```

[87]:
```
df['day_added'].value_counts()
```

[87]:
```
1     2219
15     689
2      325
16     289
31     274
20     249
19     243
5      231
22     230
10     214
30     211
6      210
18     207
26     206
```

```
8      201
14     198
25     197
27     195
7      194
21     193
28     190
23     184
12     181
17     180
4      175
13     175
24     159
3      151
11     149
9      147
29     141
Name: day_added, dtype: int64
```

[88]:
```
df['month_added'].value_counts()
```

[88]:
```
7     827
12    814
9     772
4     764
10    762
8     756
3     743
1     738
6     728
11    708
5     632
2     563
Name: month_added, dtype: int64
```

[89]:
```
df['seasons'].value_counts()
```

[89]:
```
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
```

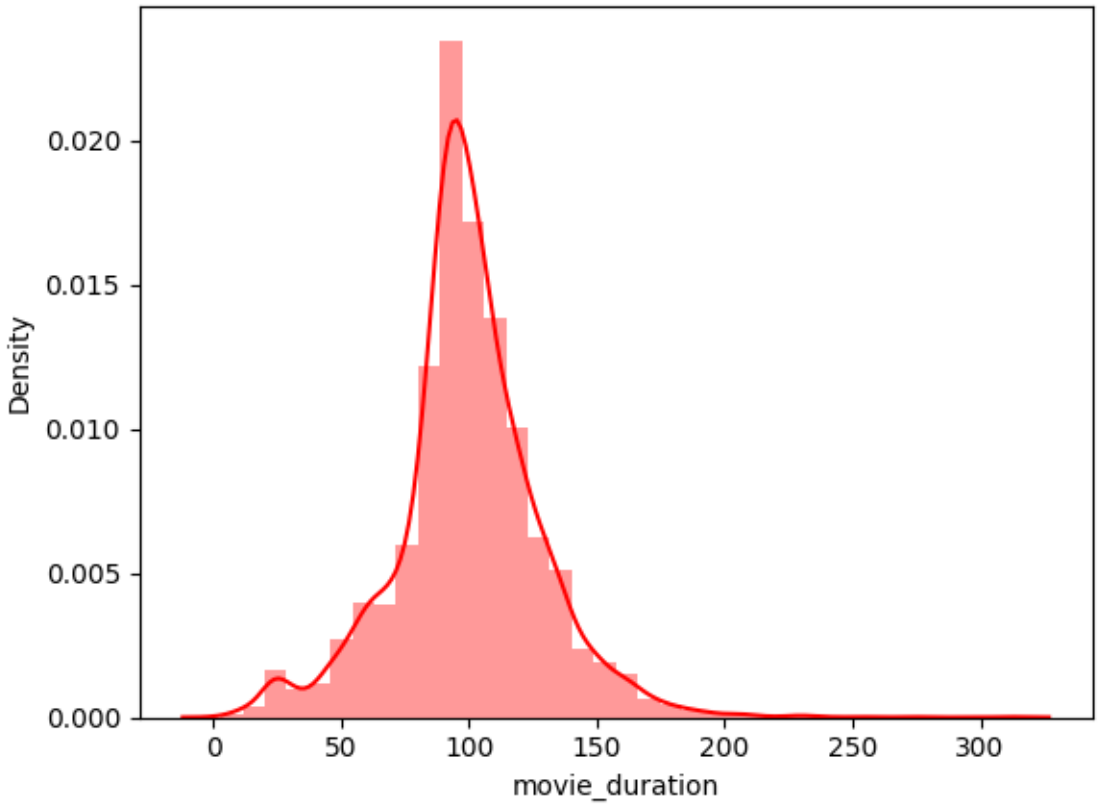
```
13 Seasons       3
15 Seasons       2
12 Seasons       2
11 Seasons       2
17 Seasons       1
Name: seasons, dtype: int64
```

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

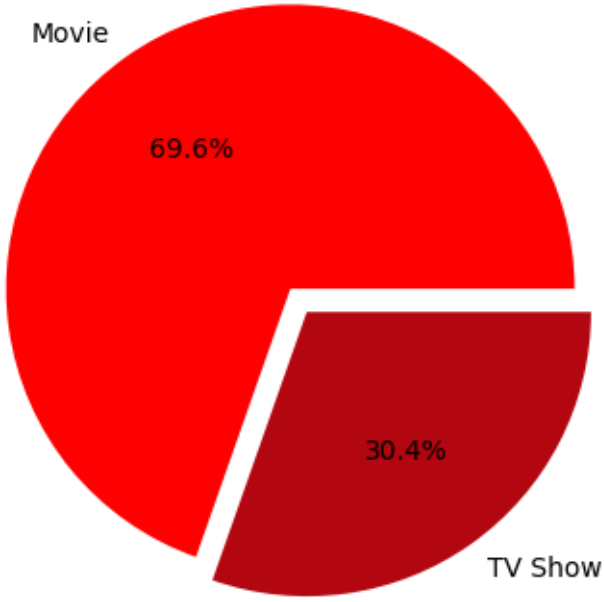
4.1 4.1 Univariate

[90]:
```
df_movie=df[df['movie_duration']!=0]
sns.
  ↪distplot(df_movie['movie_duration'],hist=True,kde=True,bins=int(36),color='red')
plt.show()
```

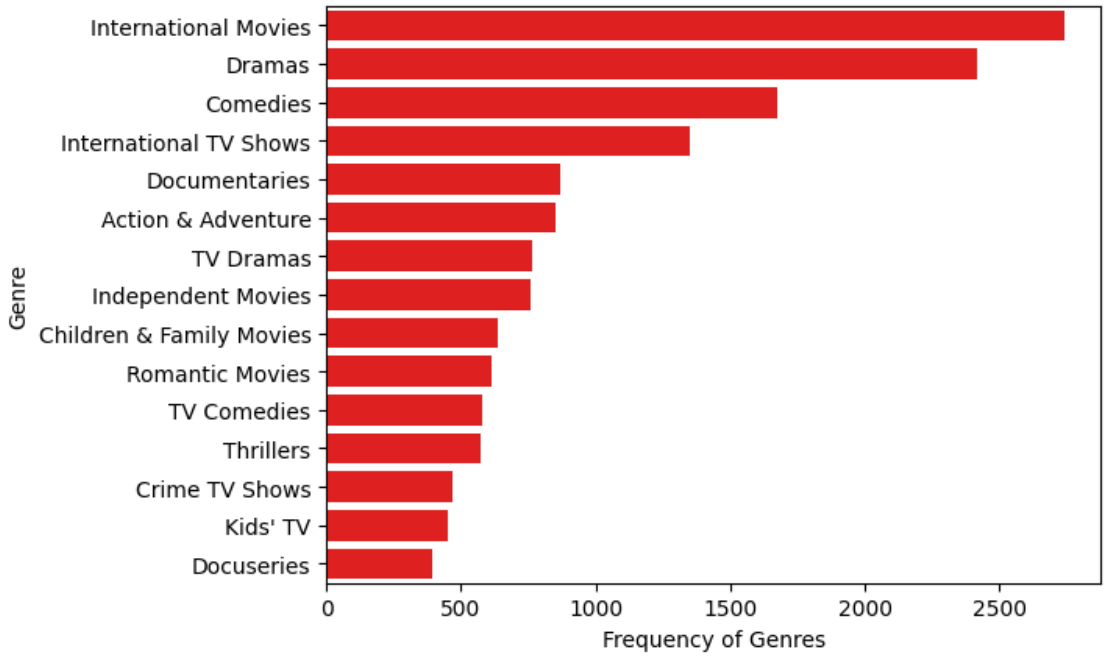
```
<ipython-input-90-ea04428cb4dc>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_movie['movie_duration'],hist=True,kde=True,bins=int(36),color=
'red')
```



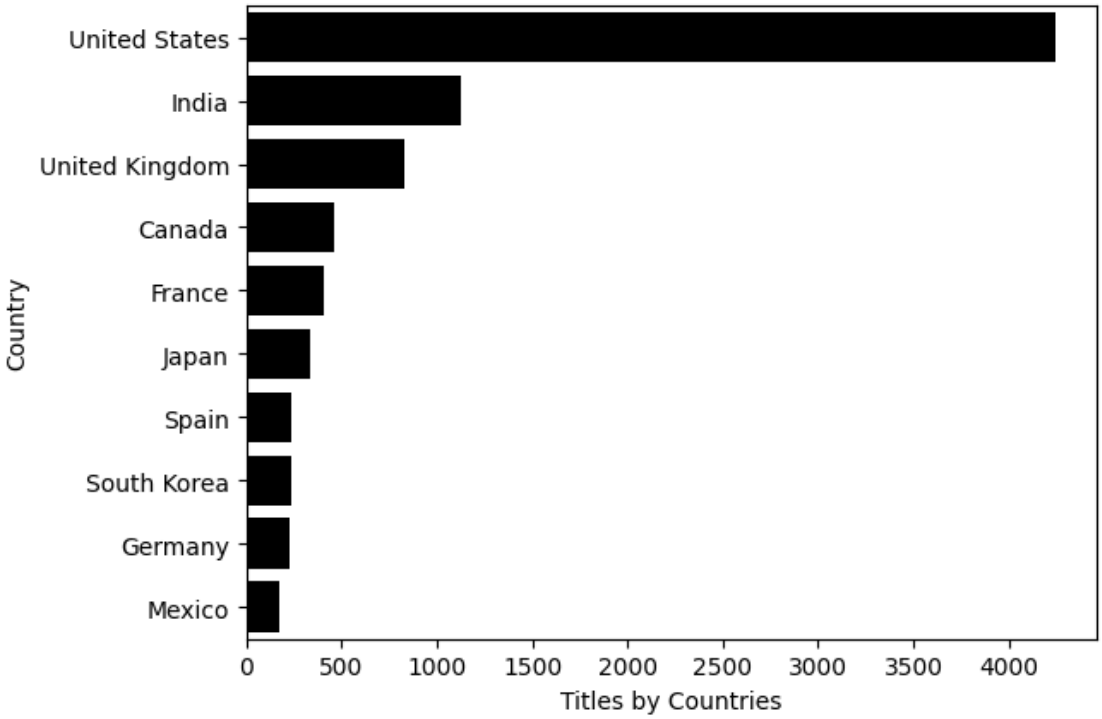
[91]:
```
df_type=df.groupby(['type']).agg({'title':'nunique'}).reset_index()
plt.pie(df_type['title'],explode=(0.05,0.05),labels=df_type['type'],autopct='%.
  ↪1f%%',colors=['red','#b20710'])
plt.show()
```



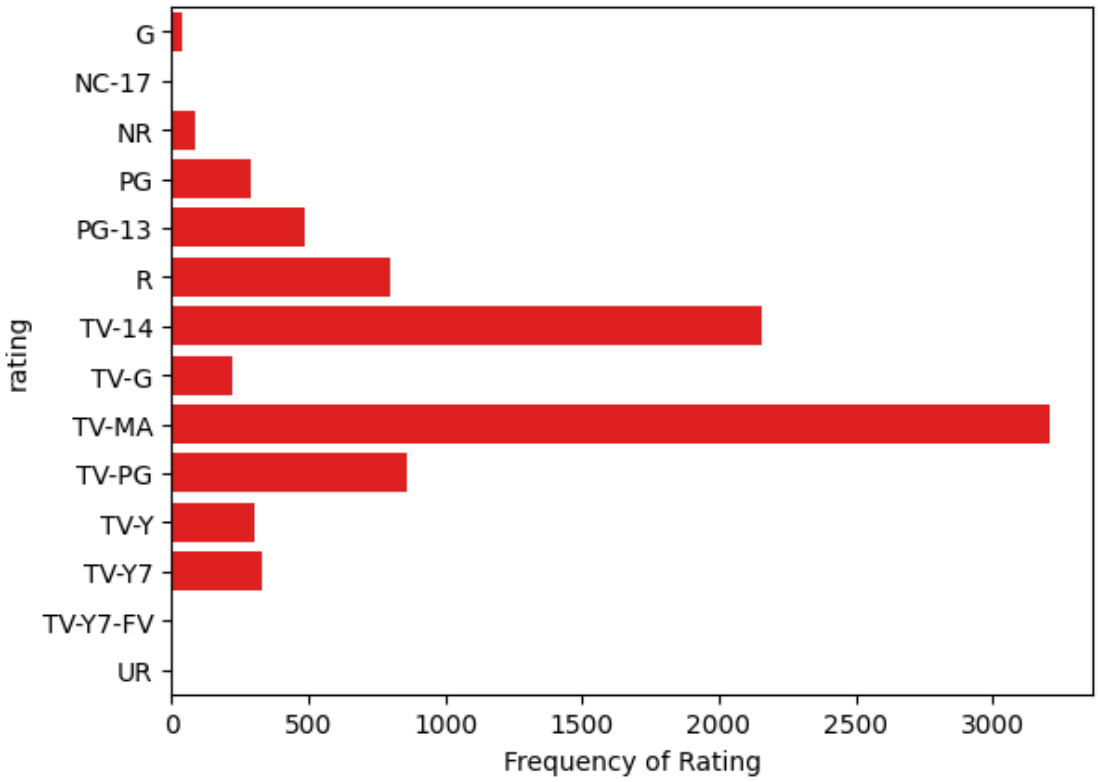
[92]:
```
df_genre=df_final.groupby(['Genre']).agg({'title':'nunique'}).reset_index().
  ↪sort_values(by=['title'],ascending=False).head(15)
sns.barplot(data=df_genre,x='title',y='Genre',color='red')
plt.ylabel('Genre')
plt.xlabel('Frequency of Genres')
plt.show()
```



[93]:
```
df_country=df_final.groupby(['country']).agg({'title':'nunique'}).reset_index().
  ↪sort_values(['title'],ascending=False).head(10)
sns.barplot(data=df_country,x='title',y='country',color='black')
plt.ylabel('Country')
plt.xlabel('Titles by Countries')
plt.show()
```

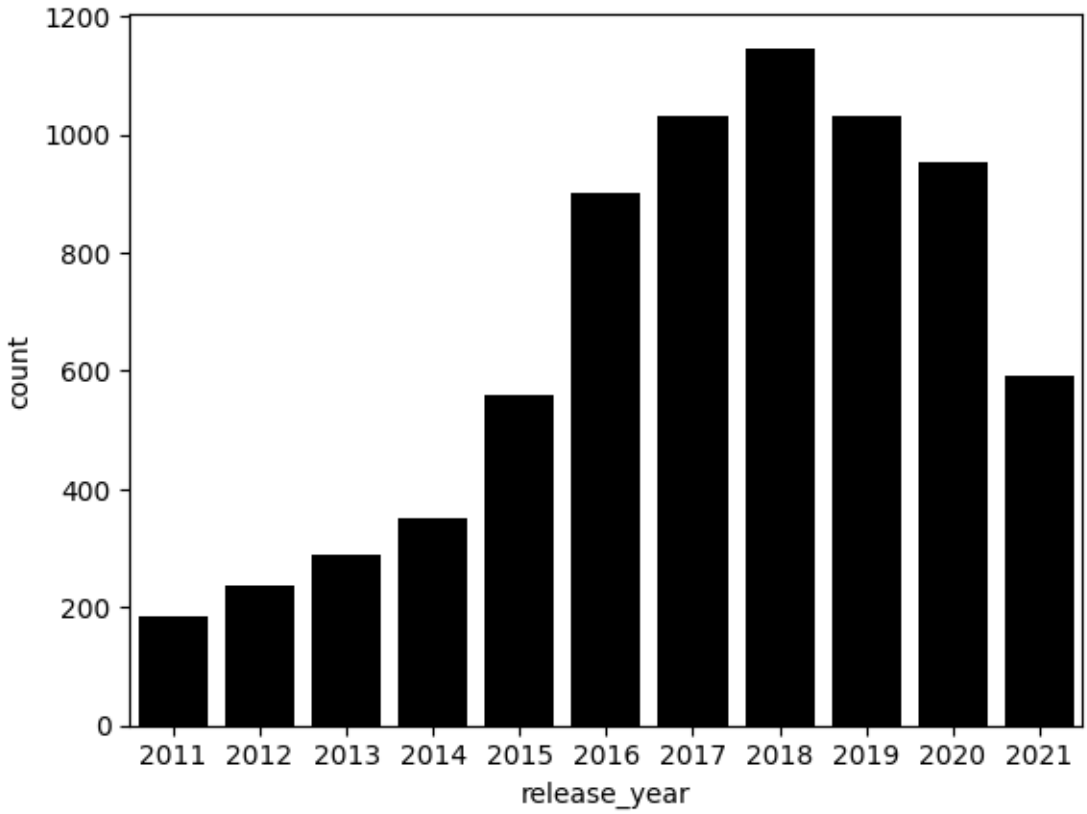


[94]:
```
sns.countplot(data=df,y='rating',color='red')
plt.xlabel('Frequency of Rating')
plt.show()
```



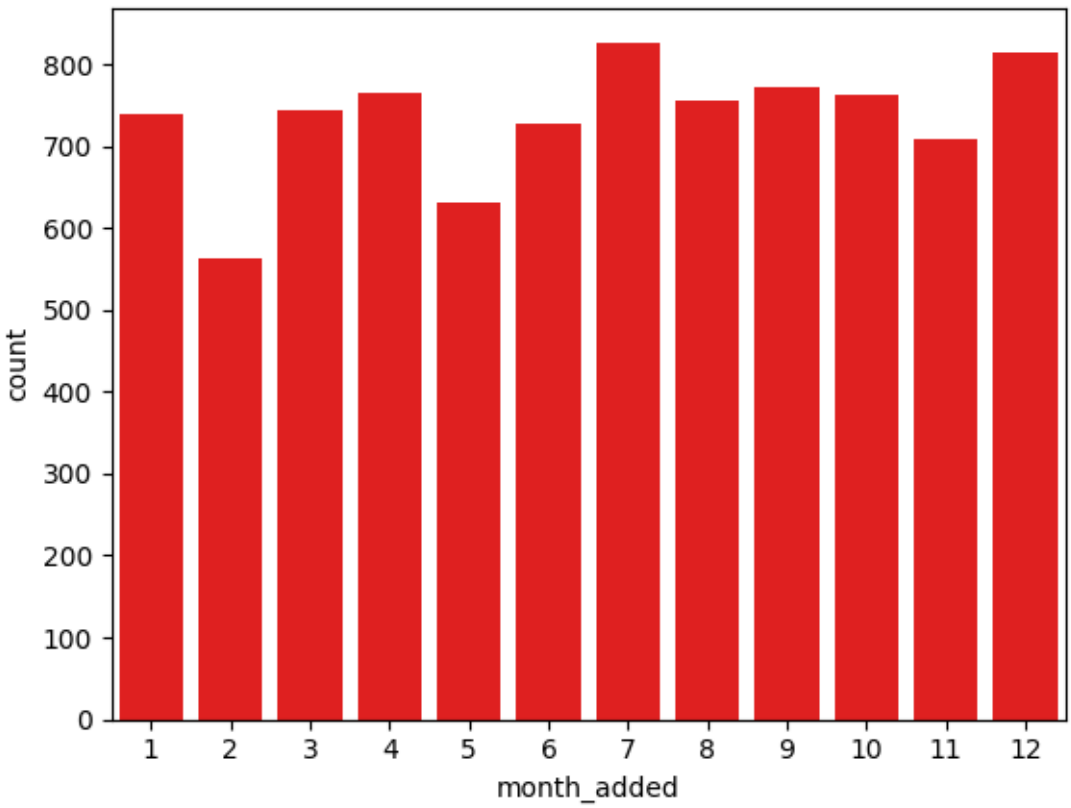
[95]:
```
df_year=df[df['release_year']>2010]
sns.countplot(data=df_year,x='release_year',color='black')
```

[95]:
```
<Axes: xlabel='release_year', ylabel='count'>
```



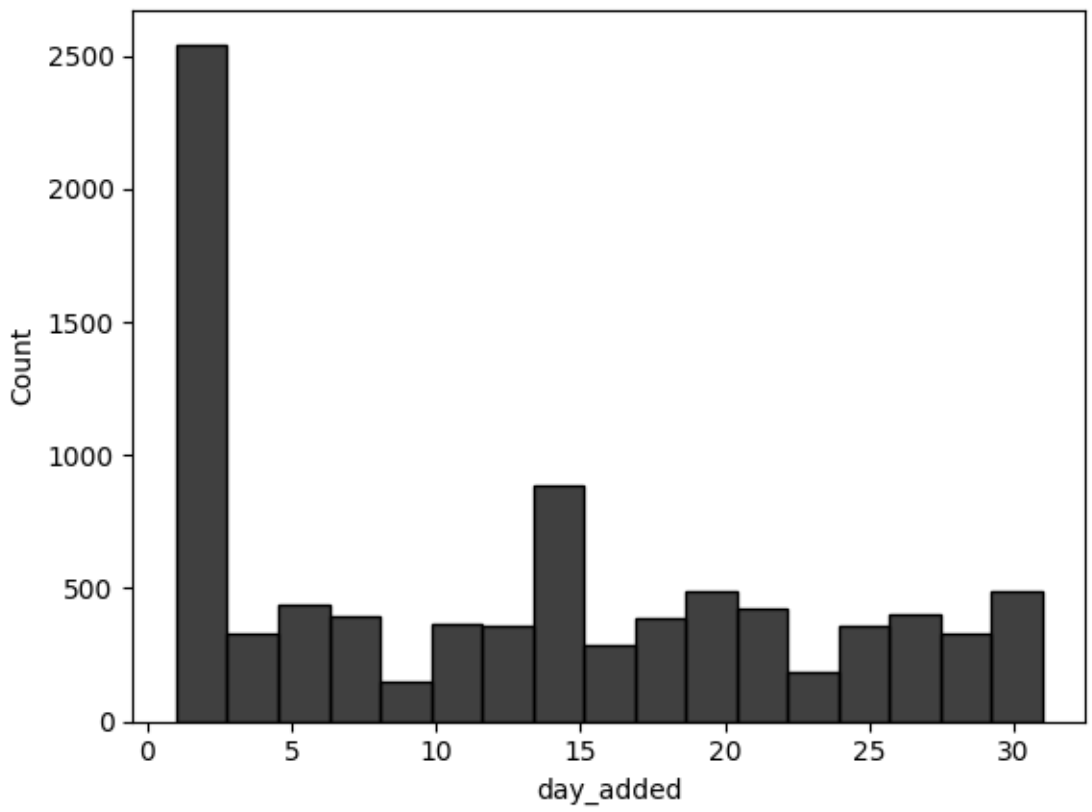
[96]:
```
sns.countplot(data=df,x='month_added',color='red')
```

[96]: <Axes: xlabel='month_added', ylabel='count'>



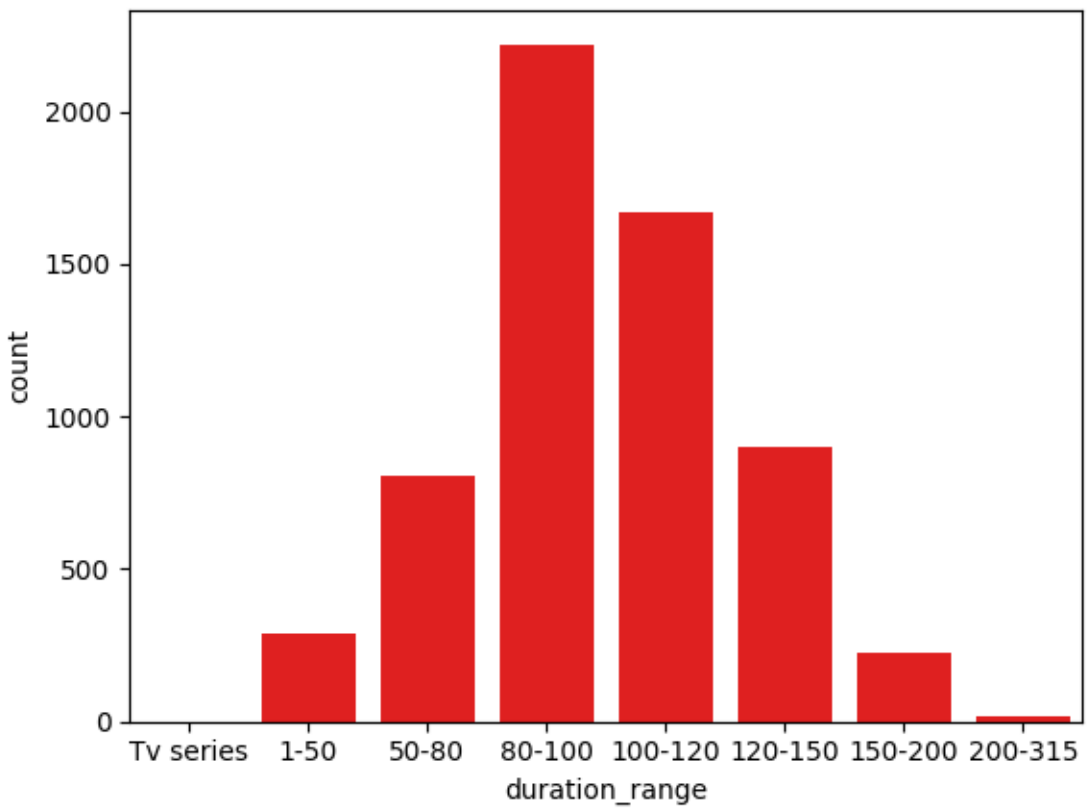
[97]:
```
sns.histplot(data=df,x='day_added',color='black')
```

[97]:
```
<Axes: xlabel='day_added', ylabel='Count'>
```



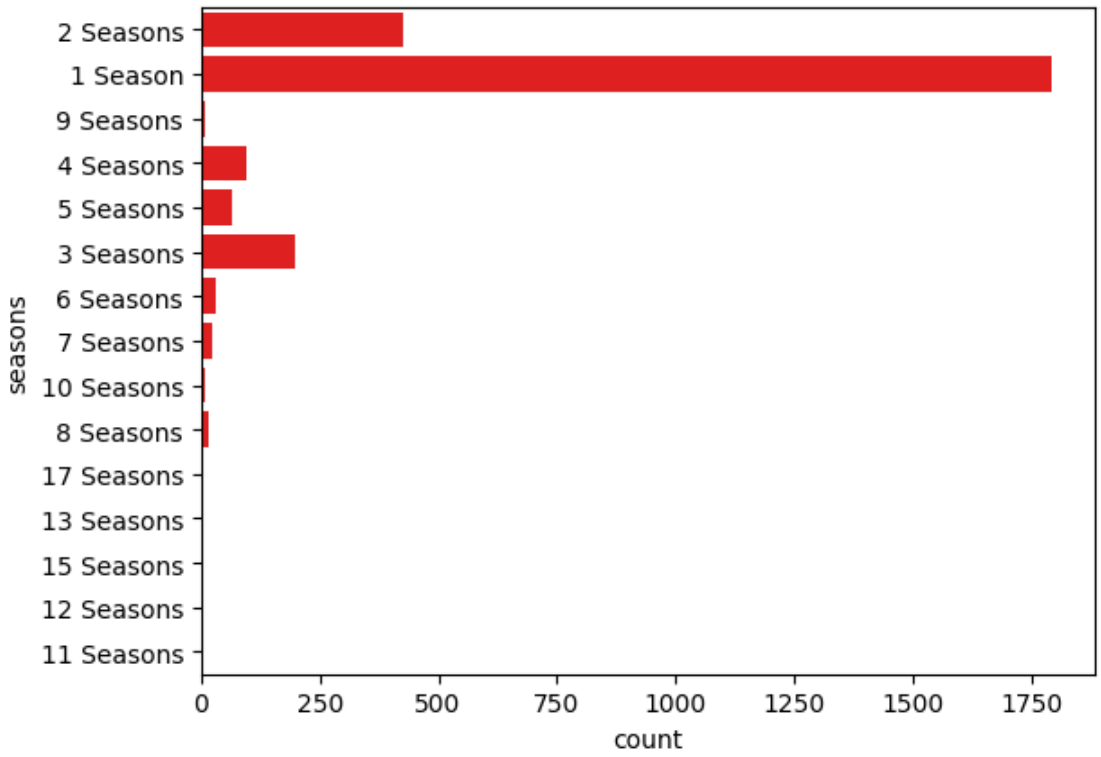
[98]:
```
df_duration=df[df['duration_range']!='Tv series']
sns.countplot(data=df_duration,x='duration_range',color='red')
```

[98]: <Axes: xlabel='duration_range', ylabel='count'>

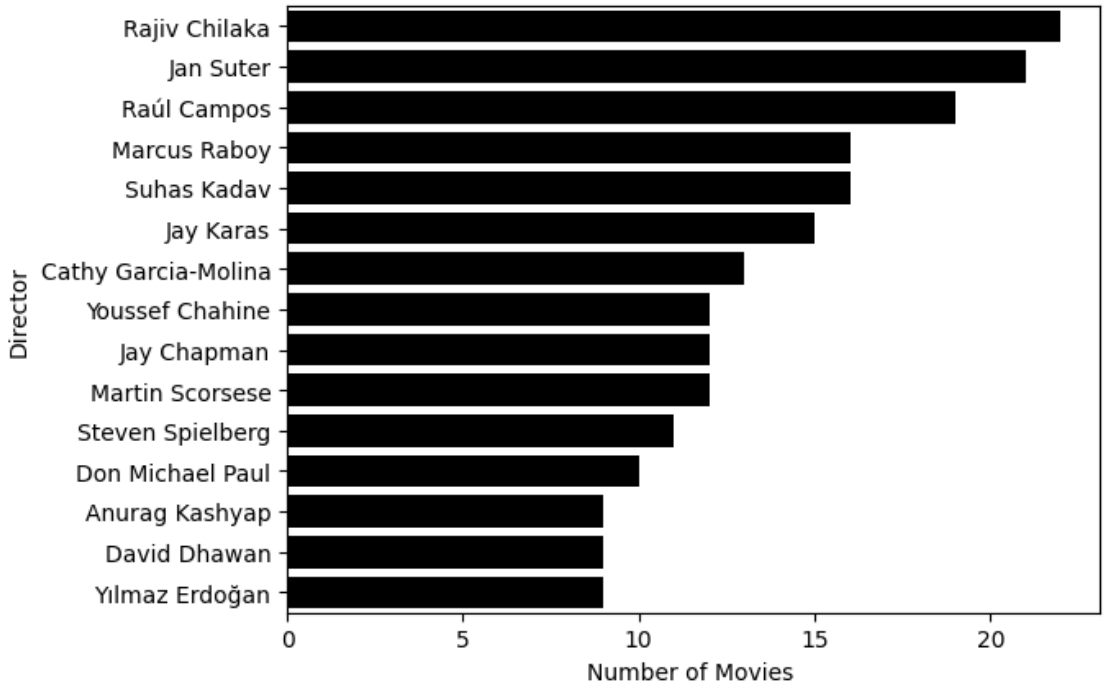


[99]:
```
sns.countplot(data=df,y='seasons',color='red')
```

[99]:
```
<Axes: xlabel='count', ylabel='seasons'>
```



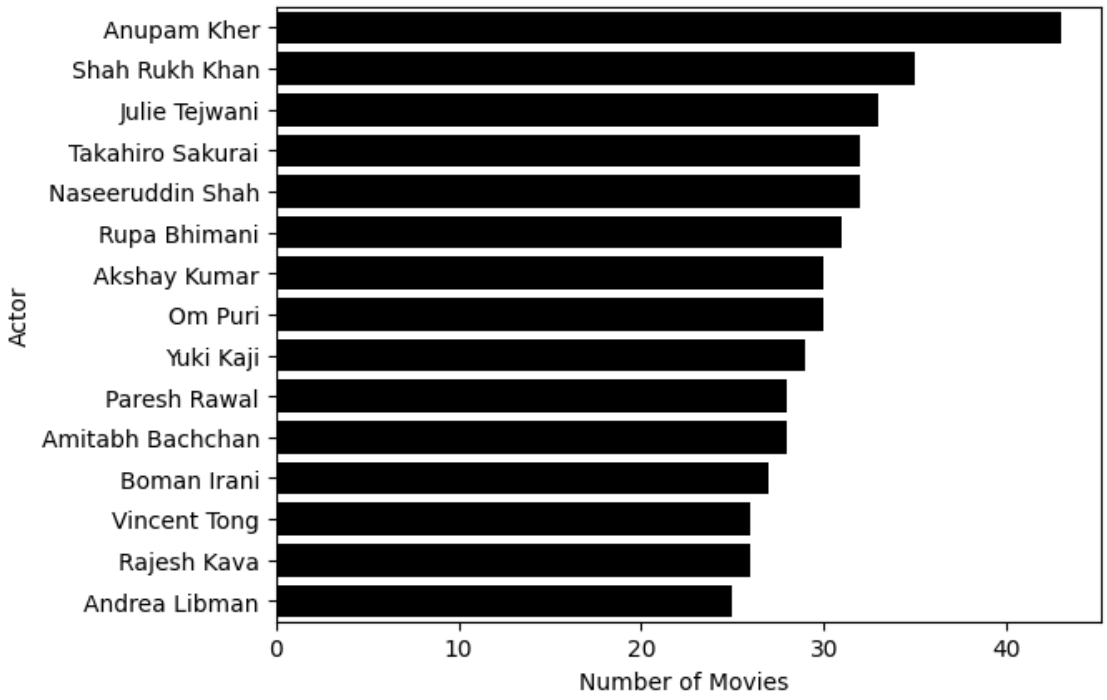
[100]:
```
df_dir=df_final.groupby(['Directors']).agg({'title':'nunique'}).reset_index()
df_dir=df_dir[df_dir['Directors']!='unknown Director']
df_dir=df_dir.sort_values(['title'],ascending=False)
sns.barplot(data=df_dir.head(15),x='title',y='Directors',color='black')
plt.xlabel('Number of Movies ')
plt.ylabel('Director')
plt.show()
```



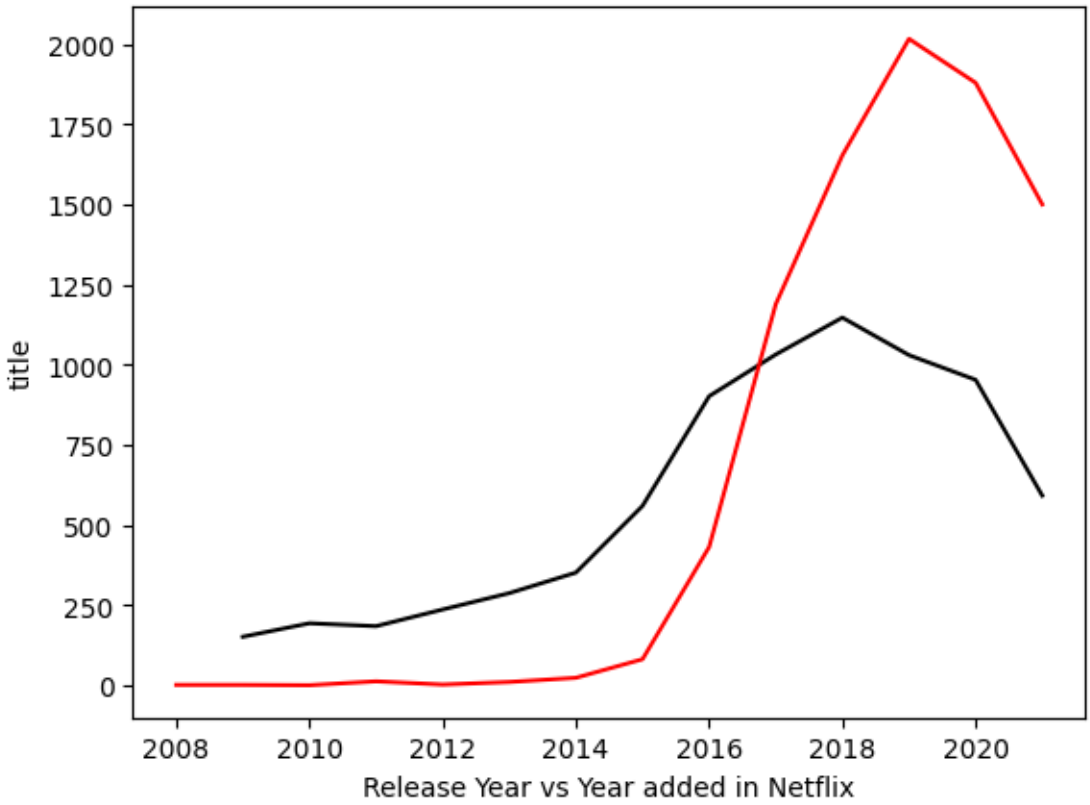
[101]:
```
df_final['Actors'].value_counts()
```

[101]:
```
unknown Actor     2146
Liam Neeson        161
Alfred Molina      160
John Krasinski     139
Salma Hayek        130
                   ...
Dario Yazbek         1
Corinne Foxx         1
Jacob Craner         1
Laila Berzins        1
Wendy McColm         1
Name: Actors, Length: 36440, dtype: int64
```

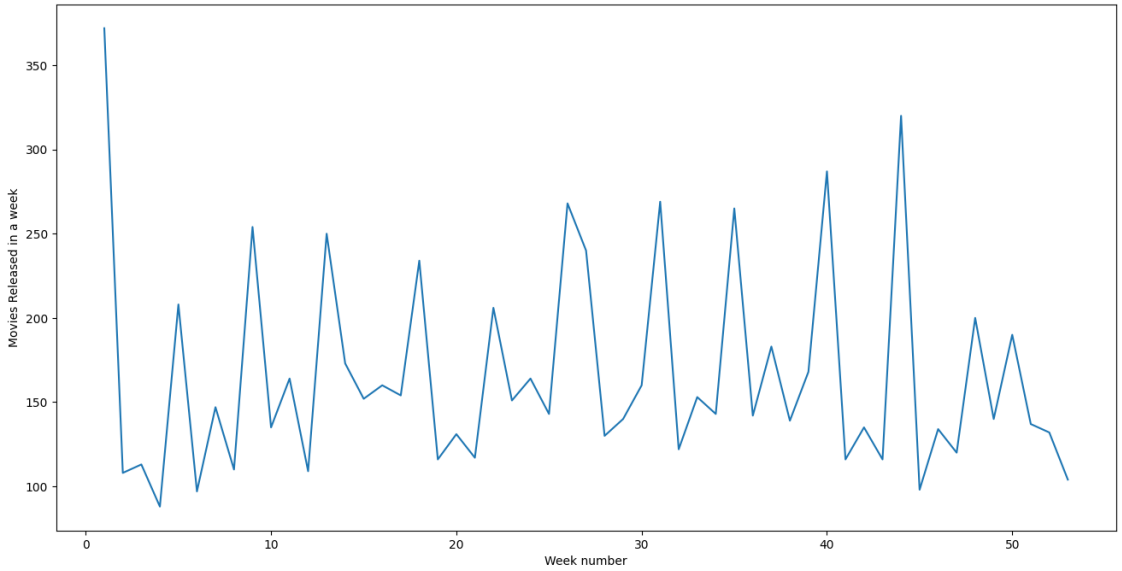
[102]:
```
df_act=df_final.groupby(['Actors']).agg({'title':'nunique'}).reset_index()
df_act=df_act[df_act['Actors']!='unknown Actor']
df_act=df_act.sort_values(['title'],ascending=False)
sns.barplot(data=df_act.head(15),x='title',y='Actors',color='black')
plt.xlabel('Number of Movies ')
plt.ylabel('Actor')
plt.show()
```



[103]:
```
df_year=df.groupby(['release_year']).agg({'title':'nunique'})
df_year=df_year.reset_index().sort_values(['release_year'])
df_year=df_year[df_year['release_year']>2008]
sns.lineplot(data=df_year,x='release_year',y='title',color='black')

df_year1=df.groupby(['year_added']).agg({'title':'nunique'})
df_year1=df_year1.reset_index().sort_values(['year_added'])
sns.lineplot(data=df_year1,x='year_added',y='title',color='red')
plt.xlabel('Release Year vs Year added in Netflix')
plt.show()
```



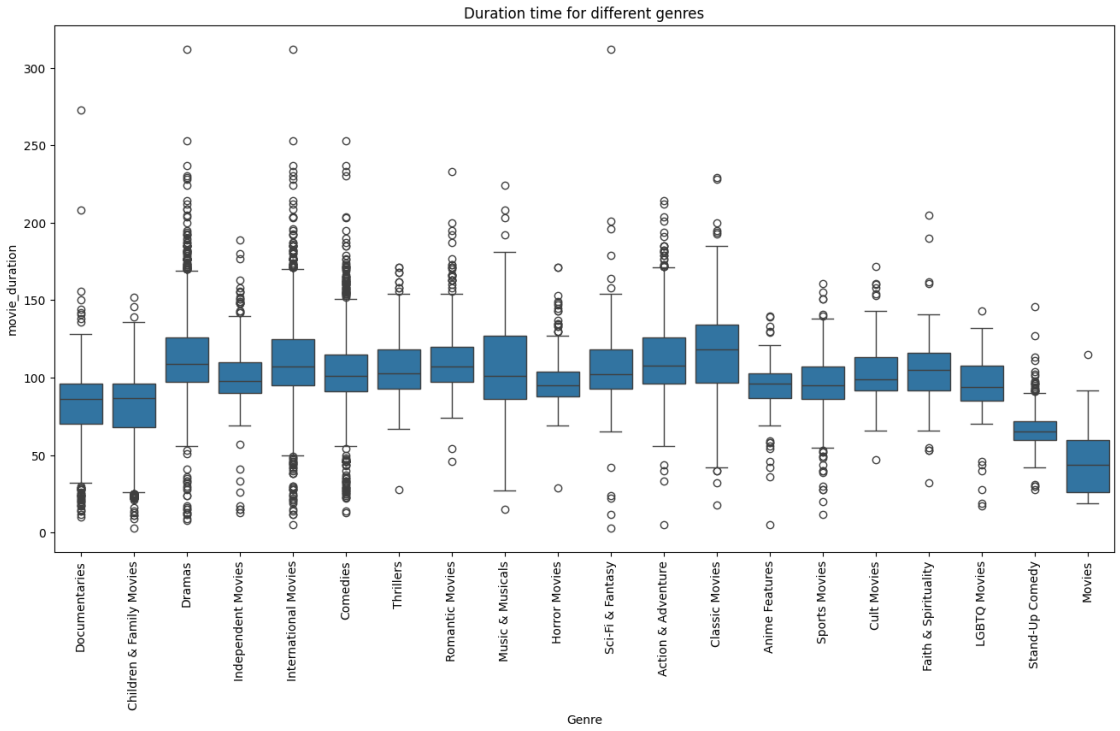
[104]:
```
df_week=df.groupby(['week_added']).agg({'title':'nunique'}).reset_index()
plt.figure(figsize=(16,8))
sns.lineplot(data=df_week,x='week_added',y='title')
plt.xlabel('Week number')
plt.ylabel('Movies Released in a week')
plt.show()
```



4.2 4.2. Categorical Data

[105]:
```
#Duration time for different genres of Movies
plt.figure(figsize=(16, 8))
data_ = df_final1.loc[df_final1["type"]=="Movie", ["title", "Genre",␣
 ↪"movie_duration"]].drop_duplicates()
plt.xticks(rotation=90)
sns.boxplot(data = data_, x = "Genre", y = "movie_duration")
plt.title("Duration time for different genres")
```

[105]:
```
Text(0.5, 1.0, 'Duration time for different genres')
```



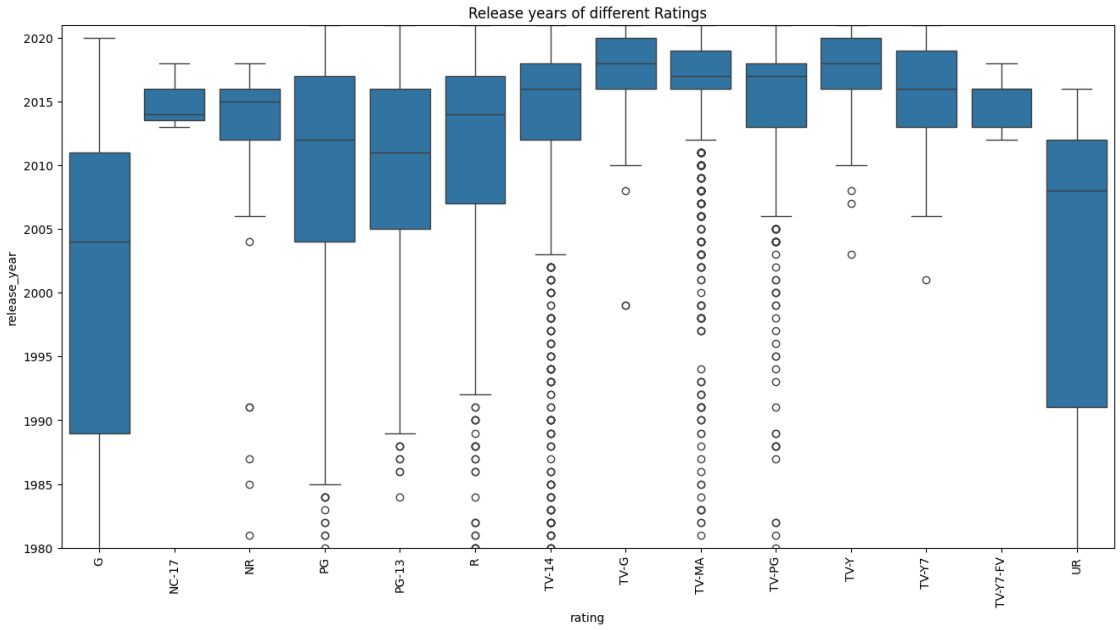
We observe median duration of classical movies is the highest.

The genre of 'Movies' has the least median duration.These genre of movies are mainly short movies which is of 1 min

The genre 'International Movies' and 'Drama' have the biggest no. of outliers.

[106]:
```
#Release years of different Ratings
plt.figure(figsize=(16, 8))
data_ = df_final.loc[df_final["type"]=="Movie", ["title", "rating",␣
 ↪"release_year"]].drop_duplicates()
plt.xticks(rotation=90)
plt.ylim([1980,2021])
sns.boxplot(data = data_, x = "rating", y = "release_year")
plt.title("Release years of different Ratings")
```

[106]:
```
Text(0.5, 1.0, 'Release years of different Ratings')
```



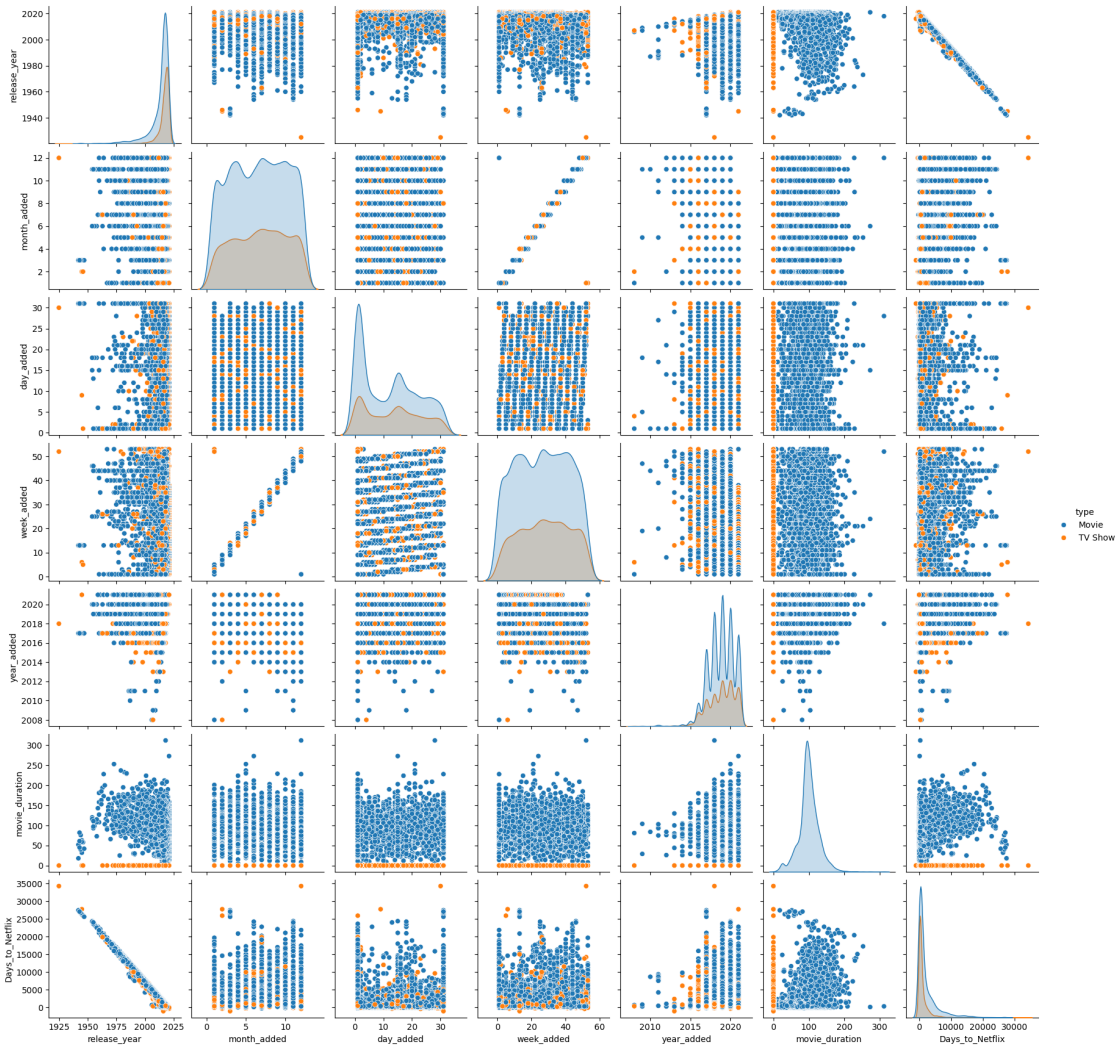
We observe that rating category 'G' and 'UR' are mostly for old movies/shows.

The rating category 'TV-Y' and 'TV-G' are mostly for newer movies/shows.

4.3 4.3. Heatmaps and Pairplots

[107]:
```
plt.figure(figsize = (18,12))
sns.pairplot(df, hue = "type")
```

[107]:
```
<seaborn.axisgrid.PairGrid at 0x782c50cfe470>

<Figure size 1800x1200 with 0 Axes>
```



We see that TV shows duration mostly appear at 1, and movies mainly appear around 100.

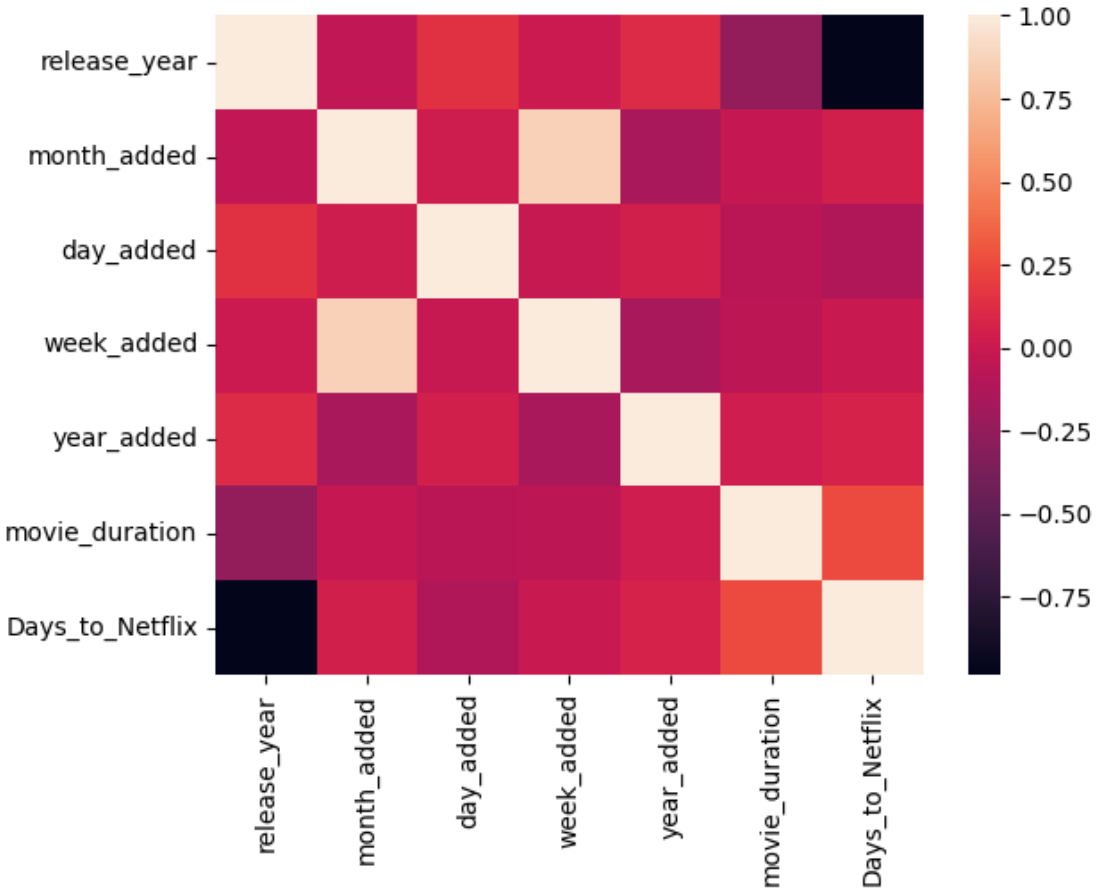
Most of the movies/shows have been added recently.

The release years have been sparse before the year 2000, but after that it seems the number per year is uniform.

[108]:
```
sns.heatmap(data=df.corr())
```

```
<ipython-input-108-afb2b4e09bbc>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(data=df.corr())
```

[108]: <Axes: >



5 5. Missing values and outlier check

5.1 5.1 Missing values have already been addressed in the Preprocessing of the Data set

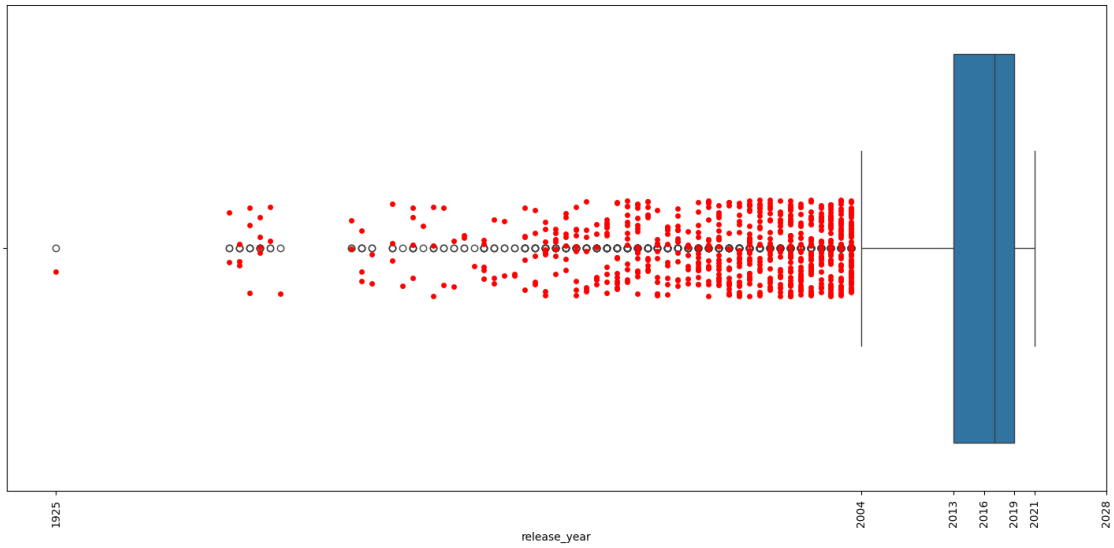
[109]:
```
df_final.isna().sum().sum()
```

[109]: 0

5.2 5.2 Outlier Check

[110]:
```
# Checking for outliers in the release_year column
df_year = df.loc[:, ["title", "release_year"]].drop_duplicates()
outl = df_year["release_year"].describe()
Q1 = outl.loc["25%"]
Q3 = outl.loc["75%"]
iqr = Q3 - Q1
low = Q1 - 1.5*iqr
```

```
upp = Q3 + 1.5*iqr
outliers = df_year[(df_year["release_year"]<low) |␣
 ↪(df_year["release_year"]>upp)]
plt.figure(figsize = (18,8))
plt.xticks(rotation=90)
sns.boxplot(x = df_year["release_year"])
sns.stripplot(x = outliers["release_year"], color = "red")
plt.xticks([df_final["release_year"].min(), low, Q1,df_final["release_year"].
 ↪median(), Q3, upp, df_final["release_year"].max() ])
plt.show()
```

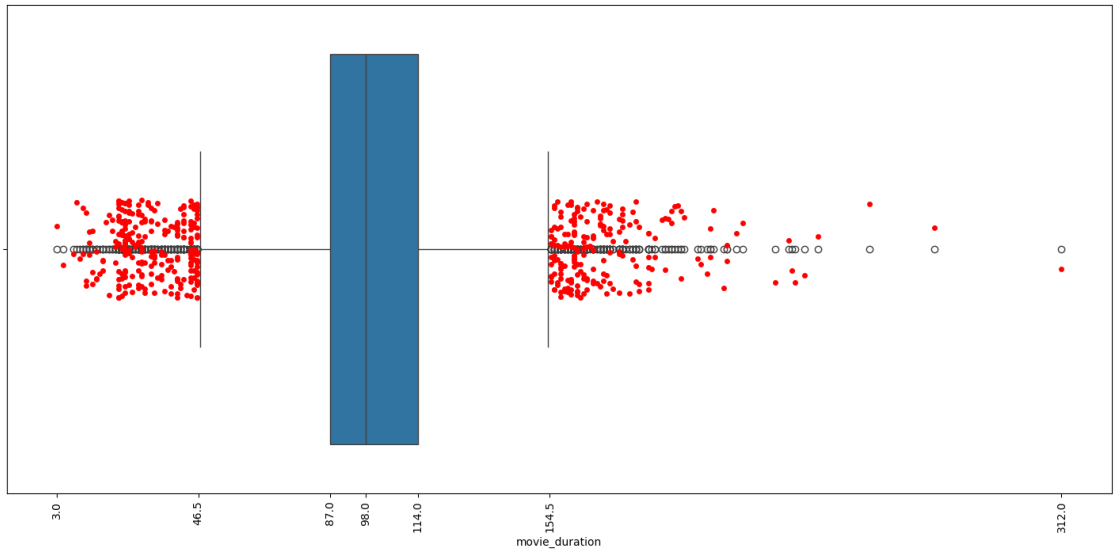


Since most of the movies/shows have been added recently, there are no outliers above the upper whisker

All the shows/movies in the outliers are from the year 1942 to 2004.

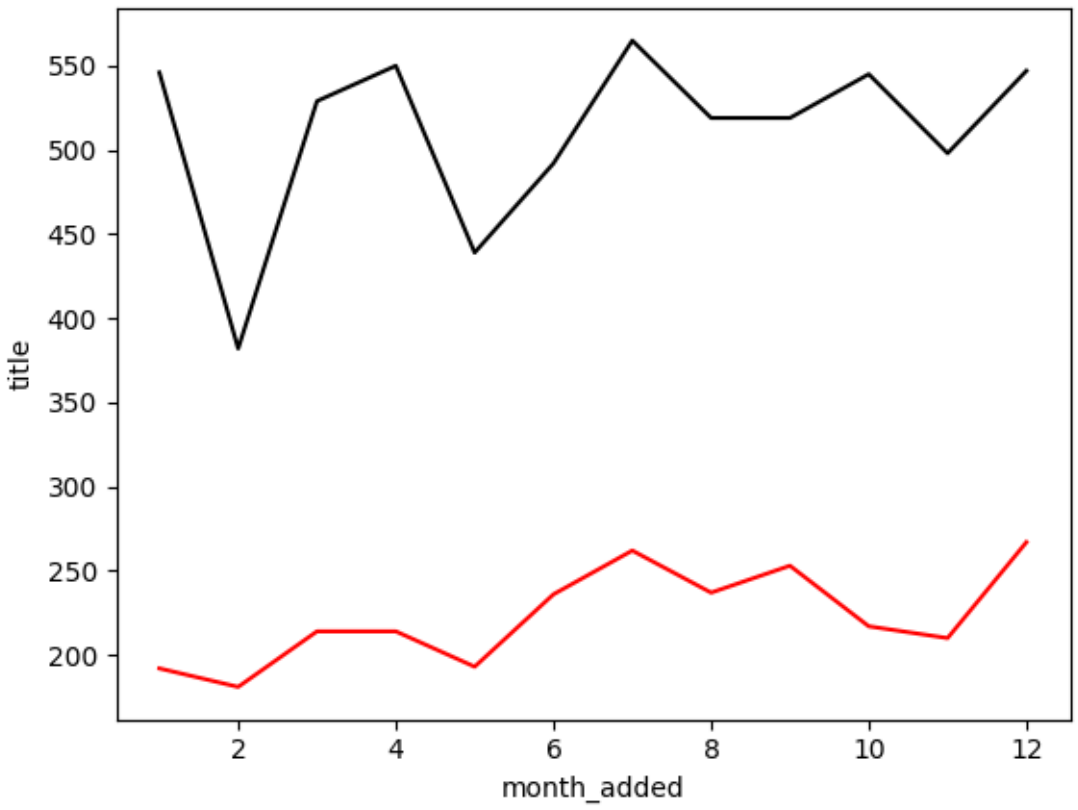
[111]:
```
# Checking for outliers in the movies duration column
df_movie= df.loc[df["type"] =="Movie", ["title", "movie_duration"]].
 ↪drop_duplicates()
df_movie=df_movie[df_movie['movie_duration']!=0]
outl = df_movie["movie_duration"].describe()
Q1 = outl.loc["25%"]
Q3 = outl.loc["75%"]
iqr = Q3 - Q1
low = Q1 - 1.5*iqr
upp = Q3 + 1.5*iqr
outliers = df_movie[(df_movie["movie_duration"]<low) |␣
 ↪(df_movie["movie_duration"]>upp)]
plt.figure(figsize = (18,8))
```

```
plt.xticks(rotation=90)
sns.boxplot(x = df_movie["movie_duration"])
sns.stripplot(x = outliers["movie_duration"], color = "red")
plt.xticks([df_movie["movie_duration"].min(), low,␣
 ↪Q1,df_movie["movie_duration"].median(), Q3, upp, df_movie["movie_duration"].
 ↪max()])
plt.show()
```

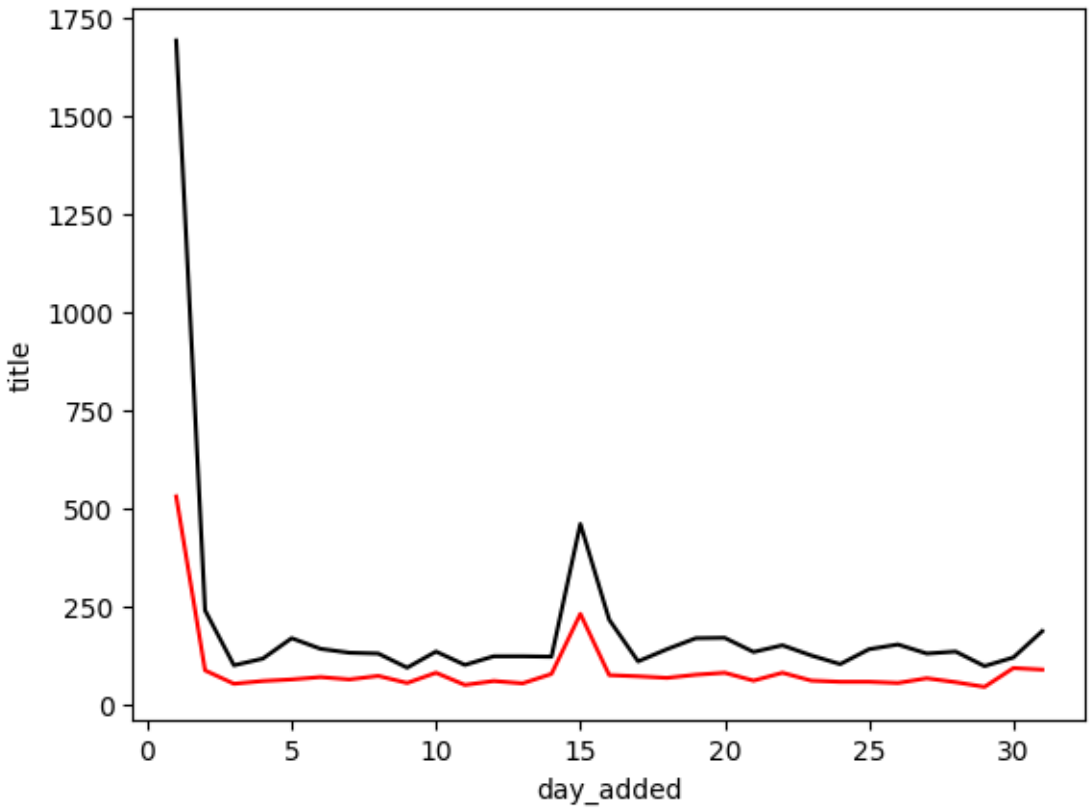


6 Analysis

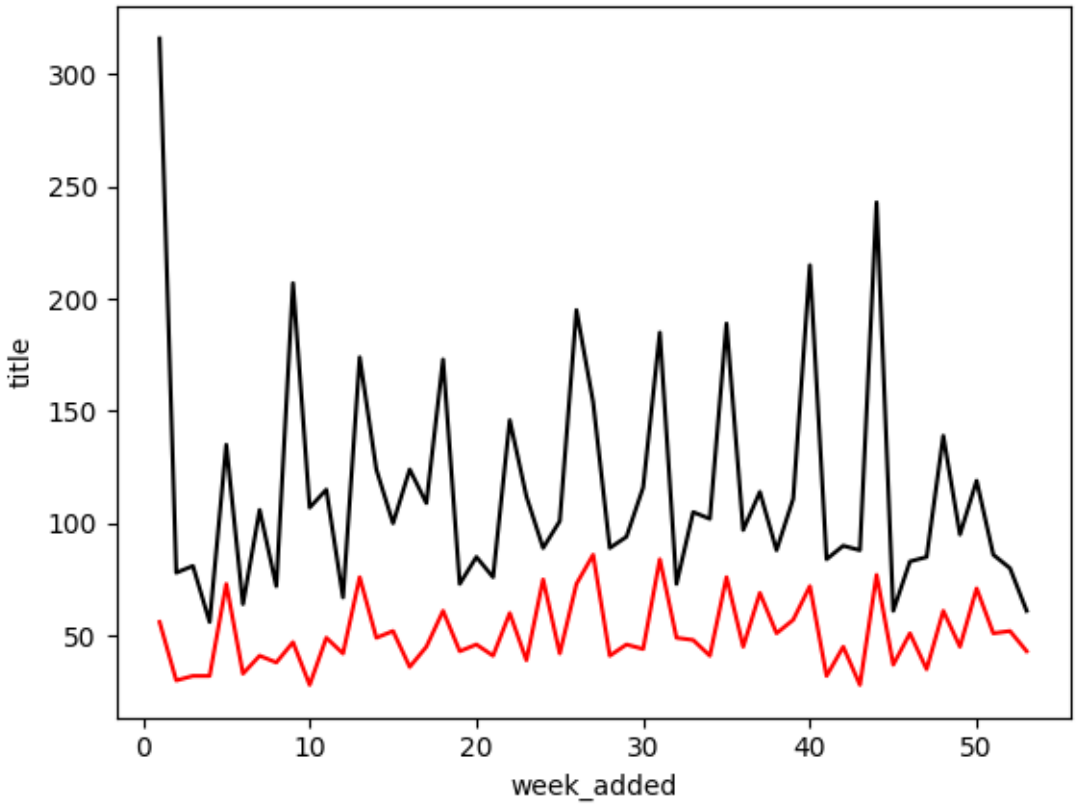
[112]:
```
df_movie=df[df['type']=='Movie']
df_series=df[df['type']=='TV Show']
df_movie_group=df_movie.groupby(['month_added']).agg({'title':'nunique'}).
 ↪reset_index().sort_values(['title'])
df_series_group=df_series.groupby(['month_added']).agg({'title':'nunique'}).
 ↪reset_index().sort_values(['title'])
sns.lineplot(data=df_movie_group,x='month_added',y='title',color='black')
sns.lineplot(data=df_series_group,x='month_added',y='title',color='red')
plt.show()
```



[113]:
```
df_movie=df[df['type']=='Movie']
df_series=df[df['type']=='TV Show']
df_movie_group=df_movie.groupby(['day_added']).agg({'title':'nunique'}).
 ↪reset_index().sort_values(['title'])
df_series_group=df_series.groupby(['day_added']).agg({'title':'nunique'}).
 ↪reset_index().sort_values(['title'])
sns.lineplot(data=df_movie_group,x='day_added',y='title',color='black')
sns.lineplot(data=df_series_group,x='day_added',y='title',color='red')
plt.show()
```



[114]:
```
df_movie=df[df['type']=='Movie']
df_series=df[df['type']=='TV Show']
df_movie_group=df_movie.groupby(['week_added']).agg({'title':'nunique'}).
 ↪reset_index().sort_values(['title'])
df_series_group=df_series.groupby(['week_added']).agg({'title':'nunique'}).
 ↪reset_index().sort_values(['title'])
sns.lineplot(data=df_movie_group,x='week_added',y='title',color='black')
sns.lineplot(data=df_series_group,x='week_added',y='title',color='red')
plt.show()
```



By observing above three graphs Tv Shows and Movies are adding in the same ratio in the netflix

[115]:
```
df_most_genre=df_final.groupby(['country','Genre'])['title'].count().
↪reset_index().sort_values('title',ascending=False)
```

[116]:
```
df_most_genre.head(20)
```

[116]:

	country	Genre	title
1366	United States	Comedies	8550
1371	United States	Dramas	8219
536	India	International Movies	7300
532	India	Dramas	5731
1363	United States	Children & Family Movies	5195
1359	United States	Action & Adventure	4548
1374	United States	Independent Movies	3898
1393	United States	TV Dramas	3275
527	India	Comedies	2799
1400	United States	Thrillers	2726
1392	United States	TV Comedies	2580
1383	United States	Romantic Movies	2281
1377	United States	Kids' TV	2256

```
1373   United States             Horror Movies   2129
1385   United States          Sci-Fi & Fantasy   2099
1376   United States    International TV Shows   1974
1329  United Kingdom                   Dramas   1944
674            Japan    International TV Shows   1809
661            Japan             Anime Series   1785
398           France      International Movies   1768
```

[117]:
```
# Group the DataFrame by 'Country' and 'Genre' and count the occurrences
country_genre_counts = df_final.groupby(['country', 'Genre']).size().
 ↪reset_index(name='count')

# Sort the data within each country group by count in descending order
country_genre_counts['rank'] = country_genre_counts.groupby('country')['count'].
 ↪rank(ascending=False, method='dense')

# Filter the data to keep only the top 3 genres for each country
top_3_genres = country_genre_counts[country_genre_counts['rank'] <= 3]

top_3_genres.sort_values(['count'],ascending=False).head(30).
 ↪sort_values(['rank'])
```

[117]:
```
             country                     Genre  count  rank
1366   United States                  Comedies   8550   1.0
1281          Turkey      International Movies    721   1.0
569        Indonesia      International Movies    757   1.0
432          Germany                   Dramas    765   1.0
355            Egypt      International Movies    901   1.0
183           Canada                  Comedies   1029   1.0
885          Nigeria      International Movies   1032   1.0
1126     South Korea    International TV Shows   1185   1.0
398           France      International Movies   1768   1.0
1164           Spain      International Movies   1147   1.0
952      Philippines      International Movies    690   1.0
674            Japan    International TV Shows   1809   1.0
536            India      International Movies   7300   1.0
1329  United Kingdom                   Dramas   1944   1.0
436          Germany      International Movies    716   2.0
1371   United States                   Dramas   8219   2.0
532            India                   Dramas   5731   2.0
881          Nigeria                   Dramas    787   2.0
661            Japan             Anime Series   1785   2.0
394           France                   Dramas   1726   2.0
1128     South Korea           Korean TV Shows    972   2.0
1160           Spain                   Dramas    702   2.0
1320  United Kingdom          British TV Shows   1339   2.0
181           Canada  Children & Family Movies    925   2.0
```

```
1363   United States  Children & Family Movies   5195   3.0
188           Canada                    Dramas    814   3.0
527            India                  Comedies   2799   3.0
1333  United Kingdom      International Movies   1199   3.0
397           France        Independent Movies    716   3.0
659            Japan        Action & Adventure    937   3.0
```

[118]:
```
df_movie=df_final[df_final['type']=='Movie']
df_movie.groupby(['country']).agg({'title':'nunique'}).reset_index().
 ↪sort_values(['title'],ascending=False).head(10)
```

[118]:
```
             country  title
115    United States   2937
44             India   1040
113   United Kingdom    556
20            Canada    334
35            France    318
37           Germany    187
101            Spain    176
123  unknown country    156
52             Japan    138
76           Nigeria    129
```

[119]:
```
df_series=df_final[df_final['type']=='TV Show']
df_series.groupby(['country']).agg({'title':'nunique'}).reset_index().
 ↪sort_values(['title'],ascending=False).head(10)
```

[119]:
```
            country  title
64   United States   1308
63  United Kingdom    273
31           Japan    200
53     South Korea    171
9           Canada    126
20          France     91
26           India     86
58          Taiwan     72
3        Australia     66
54           Spain     63
```

[120]:
```
df_combo= df_final.loc[:, ["Actors", "title", "Directors"]].drop_duplicates()
df_combo=df_combo[df_combo['Directors']!='unknown Director']
df_combo=df_combo[df_final['Actors']!='unknown Actor']
df_combo=df_combo.groupby(['Directors','Actors']).agg({'title':'nunique'}).
 ↪reset_index().sort_values(['title'],ascending=False).head(20)
df_combo
```

```
<ipython-input-120-73bb6ab46984>:3: UserWarning: Boolean Series key will be
```

```
reindexed to match DataFrame index.
  df_combo=df_combo[df_final['Actors']!='unknown Actor']
```

[120]:

	Directors	Actors	title
35331	Rajiv Chilaka	Julie Tejwani	19
35337	Rajiv Chilaka	Rajesh Kava	19
35330	Rajiv Chilaka	Jigna Bhardwaj	18
35338	Rajiv Chilaka	Rupa Bhimani	18
35345	Rajiv Chilaka	Vatsal Dubey	16
35334	Rajiv Chilaka	Mousam	13
35343	Rajiv Chilaka	Swapnil	13
43028	Suhas Kadav	Saurav Chakraborty	8
45181	Toshiya Shinohara	Houko Kuwashima	7
45195	Toshiya Shinohara	Satsuki Yukino	7
45187	Toshiya Shinohara	Koji Tsujitani	7
45183	Toshiya Shinohara	Kappei Yamaguchi	7
45188	Toshiya Shinohara	Kumiko Watanabe	7
47785	Yılmaz Erdoğan	Yılmaz Erdoğan	7
19771	Joey So	Joseph May	6
32030	Omoni Oboli	Omoni Oboli	6
13082	Fernando Ayllón	Ricardo Quevedo	6
15306	Hakan Algül	Ata Demirer	6
7037	Cathy Garcia-Molina	Joross Gamboa	6
10018	David Dhawan	Anupam Kher	6

Geographical Distribution of Combined Cast and Director Movies

[121]:
```
df_combo= df_final.loc[:, ["Actors", "title", "Directors",'country']].
  ↪drop_duplicates()
df_combo=df_combo[df_combo['Directors']!='unknown Director']
df_combo=df_combo[df_final['Actors']!='unknown Actor']
df_combo=df_combo[df_final['country']!='unknown country']
df_combo=df_combo.groupby(['Directors','Actors','country']).agg({'title':
  ↪'nunique'}).reset_index().sort_values(['title'],ascending=False).head(20)
df_combo
```

```
<ipython-input-121-113d06f68d86>:3: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df_combo=df_combo[df_final['Actors']!='unknown Actor']
<ipython-input-121-113d06f68d86>:4: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df_combo=df_combo[df_final['country']!='unknown country']
```

[121]:

	Directors	Actors	country	title
46100	Rajiv Chilaka	Julie Tejwani	India	19
46106	Rajiv Chilaka	Rajesh Kava	India	19
46099	Rajiv Chilaka	Jigna Bhardwaj	India	18
46107	Rajiv Chilaka	Rupa Bhimani	India	18

46114	Rajiv Chilaka	Vatsal Dubey	India	16
46112	Rajiv Chilaka	Swapnil	India	13
46103	Rajiv Chilaka	Mousam	India	13
55924	Suhas Kadav	Saurav Chakraborty	India	8
58864	Toshiya Shinohara	Houko Kuwashima	Japan	7
62033	Yılmaz Erdoğan	Yılmaz Erdoğan	Turkey	7
58866	Toshiya Shinohara	Kappei Yamaguchi	Japan	7
58871	Toshiya Shinohara	Kumiko Watanabe	Japan	7
58870	Toshiya Shinohara	Koji Tsujitani	Japan	7
58878	Toshiya Shinohara	Satsuki Yukino	Japan	7
41790	Omoni Oboli	Omoni Oboli	Nigeria	6
8891	Cathy Garcia-Molina	Joross Gamboa	Philippines	6
57642	Tilak Shetty	Smita Malhotra	India	6
19800	Hakan Algül	Ata Demirer	Turkey	6
12625	David Dhawan	Anupam Kher	India	6
38524	Mike Smith	Mike Smith	Canada	5

Director-Actor Dual Roles:

[122]:
```
# Here Director and Actor are same
df_same=df_final[df_final['Directors']==df_final['Actors']]
df_same=df_same.loc[:,['title','Directors','Actors','country']].
 ↪drop_duplicates()
df_same=df_same.groupby(['Directors','country']).agg({'title':'nunique'}).
 ↪reset_index().sort_values(['title'],ascending=False).head(10)
df_same
```

[122]:

	Directors	country	title
349	Yılmaz Erdoğan	Turkey	7
233	Omoni Oboli	Nigeria	6
210	Mike Smith	Canada	5
153	John Paul Tremblay	Canada	5
273	Robb Wells	Canada	5
60	Clint Eastwood	United States	4
176	Kunle Afolayan	Nigeria	3
36	Bo Burnham	United States	3
180	Louis C.K.	United States	3
227	Note Chern-Yim	Thailand	3

#6.1 Insights on range of attributes

Release year: From the above boxplot to find the outliers in the release_year column, we see that the range of movie/show release year is from 1942 to 2021. The older movies/shows are less compared to recently released ones.

Movie duration: From the outlier boxplot mentioned above, we see that it ranges from as low as 8 mins to 312 mins!. However the ideal time duration for a movie is 100 mins(median).

TV show duration: From the above mentioned boxplots, we see that the number of seasons of TV

show ranges from 1 to 17. Majority of them are 1 season shows. The number of shows which is aired for 4 or more seasons is very less.

Rating: The number of movies/shows for each rating range from 3 (NC-17, UR) to 2884 (TV-MA). Which means the succefull shows on Netflix are usually from the rating of TV-MA and TV-14.

Genre: The number of movies/shows for each genre is mapped. It is found that 'International Movies' genre has 2574(highest) count and 'TV Shows' genre has 11(least) count.

7 Insights from Data

1. **Recent Releases Dominating**: A notable observation reveals that a significant portion of movies available on Netflix were released recently. Approximately only 25% of the movies on the platform were released before 2013, indicating a preference for newer content.
2. **Consistent Monthly Additions**: The addition of titles to Netflix appears to be evenly distributed across months, suggesting a consistent approach to content acquisition throughout the year.
3. **Common Addition Dates**: Most movies and TV shows are added to Netflix either on the 1st or 15th day of the month, reflecting a pattern in content release scheduling.
4. **Content Addition Trends**: Between 2018 and 2021, a substantial proportion (approximately 75%) of the movies available on Netflix were added, indicating a concentrated effort in expanding the platform's library during this period.
5. **Duration Preference**: Movies with a duration of less than 2 hours are prevalent on Netflix, with a significant portion falling within the 80-120 minutes range, aligning with viewers' preferences for shorter content.
6. **Time Gap from Release to Addition**: On average, movies are added to Netflix approximately 2 years after their release date, indicating a lag between theatrical release and availability on the streaming platform.
7. **Content Distribution**: The majority (around 70%) of content available on Netflix consists of movies, with the remainder comprising TV series, demonstrating a slight preference for cinematic content.
8. **Top Countries for Content**: The top three countries contributing movies and TV shows to Netflix are the United States, India, and the United Kingdom, with a notable concentration of content originating from the United States.
9. **TV Show Season Distribution**: Most TV shows available on Netflix have only one season, suggesting a preference for standalone or limited-series content.
10. **Common Content Ratings**: The majority of titles on Netflix carry ratings of 'TV-MA' and 'TV-14', indicating a focus on mature and adolescent audiences.
11. **Frequent Directors**: Directors such as Rajiv Chilaka, Jan Suter, and Raul Campos are prominent contributors to the Netflix library, with several titles attributed to their directorial credits.

12. **Content Growth Trends**: The addition of movies to Netflix showed a gradual increase up to 2019, followed by a decline post-2019, indicating fluctuations in content acquisition strategies over time.

13. **Genre Preferences by Country**: The most common genres across countries include International Movies, Dramas, and Comedies. However, specific preferences emerge, such as International TV Shows in South Korea and Japan, Korean TV Shows in South Korea, Anime Series in Japan, and British TV Shows in the United Kingdom.

14. **Frequently Featured Actors**: Anupam Kher, Shah Rukh Khan, and Julie Tejaswani emerge as frequently featured actors across the Netflix catalog, indicating their popularity and recurring presence in streamed content.

15. **Rating Distribution Across Time Periods**: Further analysis of the rating categories reveals interesting patterns regarding their distribution across different time periods. It's observed that the 'G' (General Audience) and 'UR' (Unrated) rating categories are predominantly associated with older movies and shows, suggesting a historical preference for family-friendly content without age restrictions. Conversely, newer movies and shows tend to be categorized under 'TV-Y' (All Children) and 'TV-G' (General Audience) ratings, indicating a shift towards content suitable for younger audiences. This observation aligns with evolving content standards and preferences, reflecting a trend towards more inclusive and age-appropriate programming in recent years.

16. **Repetitive Director-Actor Pairings**: Upon examining director-actor combinations, it becomes evident that certain pairs exhibit repetitive collaborations, suggesting a strong working relationship or shared artistic vision. Notably, Director Rajiv Chilaka emerges as the most frequent collaborator with specific actors, indicating a consistent partnership that has resulted in multiple projects together. Following closely, Director Toshiya Shinohara also displays notable recurring pairings with certain actors, highlighting a pattern of consistent collaboration and possibly shared creative synergy between the director and these actors. Such observations shed light on the dynamics of collaborative relationships within the filmmaking industry, where directors and actors often develop enduring partnerships that contribute to the creation of compelling and memorable cinematic experiences.

17. **Geographical Distribution of Combined Cast and Director Movies**: An interesting observation emerges when examining the geographical distribution of movies featuring combined cast and director collaborations. Specifically, it is noted that such movies are predominantly released in India when directed by Rajiv Chilaka. This suggests a strong association between the director's work and the Indian film industry, indicating a significant presence and influence within this regional cinema landscape. Furthermore, Japan emerges as the next prominent location for movies featuring combined cast and director collaborations, particularly when directed by Toshiya Shinohara. This highlights a similar trend of geographical concentration, where the director's work is closely tied to the Japanese film industry, reflecting a significant contribution to the country's cinematic landscape. These observations underscore the impact of regional cinema dynamics on collaborative efforts between directors and cast members, showcasing how specific filmmakers may have stronger associations with particular geographical regions, thereby influencing the production and distribution of combined cast and director movies.

18. **Director-Actor Dual Roles**: A noteworthy trend is observed in movies where Director Yılmaz Erdoğan also takes on an acting role within the same film, particularly prevalent in

movies originating from Turkey. This suggests a significant involvement of Yılmaz Erdoğan in both creative and performance aspects of Turkish cinema, showcasing versatility and multifaceted contributions to the filmmaking process. Furthermore, a similar pattern is identified with three directors from Canada, each appearing as actors in the same movie for five movies. This highlights a distinct trend within Canadian cinema, where directors actively participate in on-screen roles, contributing to the unique narrative and aesthetic qualities of Canadian films. Additionally, comparable observations are noted in movies from Nigeria and the United States, where directors similarly engage in dual roles as actors in the same movie, each for five movies. This indicates a shared phenomenon across different film industries, reflecting a common practice among directors to explore their talents both behind and in front of the camera, thereby enriching the cinematic experience and narrative depth of their respective films.

8 8. Recommendations

Based on the insights derived from the dataset, here are some recommendations:

1. **Content Acquisition Strategy**:
   - Netflix should continue prioritizing recent releases, as the data suggests a preference for newer content among viewers. This aligns with evolving audience preferences and ensures a fresh and engaging content library.
2. **Release Scheduling**:
   - Netflix should maintain its consistent approach to adding titles every month. This ensures a steady stream of new content for subscribers and prevents fluctuations in viewer engagement.
3. **Optimal Addition Dates**:
   - Leveraging insights on common addition dates, Netflix can strategically plan content releases around the 1st and 15th of each month to maximize viewer engagement and retention.
4. **Library Expansion**:
   - The concentration of content additions between 2018 and 2021 indicates a period of significant growth for Netflix. To sustain this momentum, the platform should continue investing in content acquisition across diverse genres and regions.
5. **Content Duration**:
   - Given the preference for shorter movies, Netflix should prioritize acquiring and producing content with durations of around 80-120 minutes. This caters to viewers' preferences for concise and engaging storytelling experiences.
6. **Time Gap Analysis**:
   - Understanding the average time gap between release and addition to Netflix, the platform can optimize its content acquisition strategy to ensure timely availability of popular movies post-theatrical release.
7. **Genre Diversity**:
   - While movies constitute the majority of Netflix's content, the platform should continue diversifying its library by acquiring a balanced mix of movies and TV series across various genres to cater to diverse viewer preferences.
8. **Geographical Focus**:
   - Netflix should prioritize content acquisition from top contributing countries such as the United States, India, and the United Kingdom, while also exploring opportunities to

expand its global footprint by investing in content from emerging markets.

9. **TV Show Seasoning**:
    - Recognizing the prevalence of single-season TV shows, Netflix can capitalize on the popularity of limited series formats and invest in producing high-quality standalone seasons to appeal to binge-watching audiences.
10. **Content Ratings**:
    - With 'TV-MA' and 'TV-14' ratings being prevalent, Netflix should continue curating a diverse range of content suitable for mature and adolescent audiences while ensuring adherence to content guidelines and viewer preferences.
11. **Director Collaboration**:
    - Netflix can explore strategic partnerships with frequent directors such as Rajiv Chilaka, Jan Suter, and Raul Campos to develop exclusive content tailored to subscriber preferences and strengthen the platform's original content portfolio.
12. **Content Growth Strategies**:
    - While the addition of movies showed a decline post-2019, Netflix should adopt agile content acquisition strategies to adapt to evolving market dynamics and maintain a competitive edge in the streaming landscape.
13. **Regional Content Preferences**:
    - Netflix should leverage insights on genre preferences by country to tailor its content offerings and localization strategies, ensuring relevance and appeal to diverse global audiences.
14. **Actor Collaborations**:
    - Identifying frequently featured actors such as Anupam Kher, Shah Rukh Khan, and Julie Tejaswani, Netflix can explore opportunities for exclusive collaborations and talent partnerships to create compelling and engaging content experiences.

These recommendations aim to capitalize on the insights derived from the dataset to inform strategic decisions and optimize content acquisition, production, and distribution efforts for Netflix. By leveraging data-driven insights, Netflix can enhance its content offerings, drive subscriber engagement, and maintain its position as a leading global streaming platform.