Business Case Study

NETFLIX: Data Exploration and Visualizations

By: Sourav Abhangrao

Business Problem:

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

1.Defining problem statement and analyzing basic metrics:

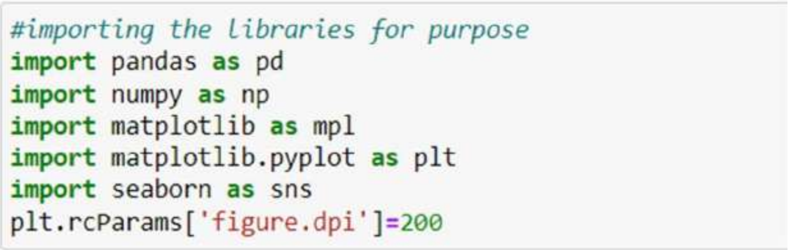
Problem statement:

Netflix is a multinational streaming company that produces movies and TV web series all around the year and all around the globe.

- Identify trends and patterns in the dataset.
- Use data-driven insights to guide content production and business expansion.
- explore genres, ratings, release years, countries, and other relevant attributes.
- Derive valuable insights to support decision-making processes.

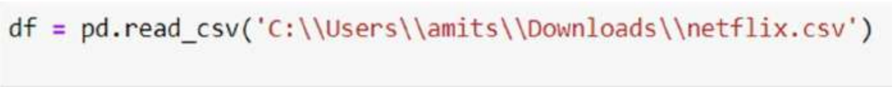
Basic metrics:

Importing Libraries:



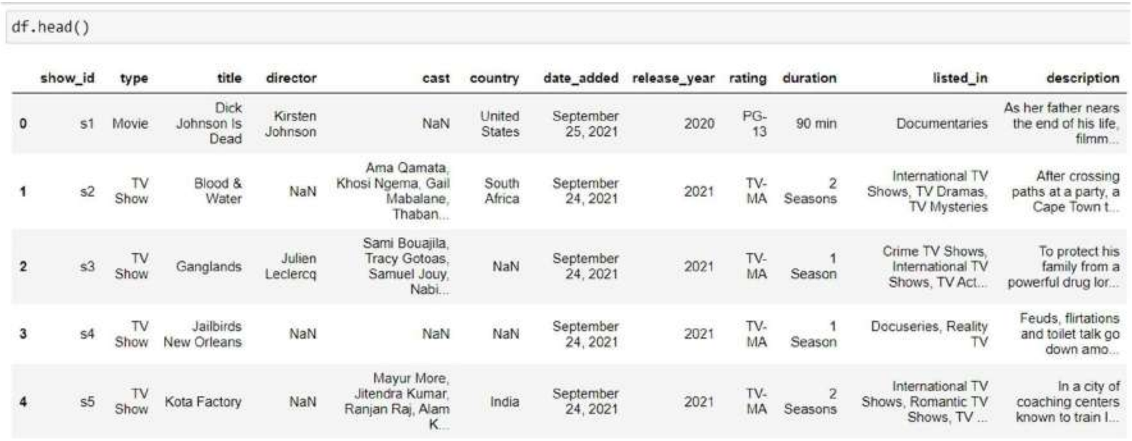
```python
#importing the libraries for purpose
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
plt.rcParams['figure.dpi']=200
```

Loading Datasets:



```python
df = pd.read_csv('C:\\Users\\amits\\Downloads\\netflix.csv')
```

Output:

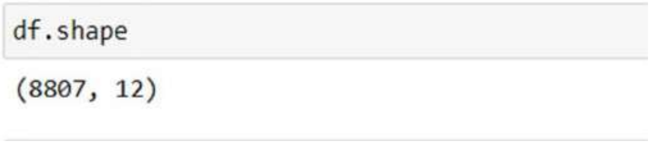


```
df.head()
```

	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	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...

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

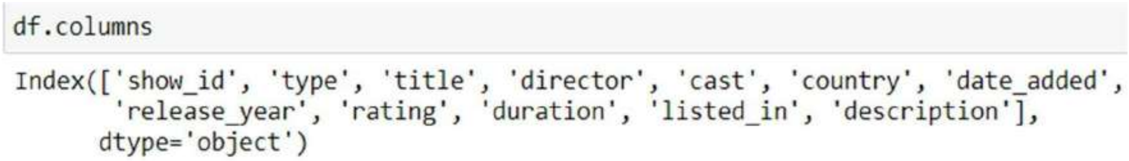
Shape of dataset:



```
df.shape
```

```
(8807, 12)
```

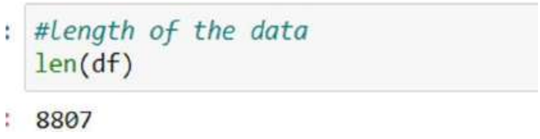
Column Names:



```
df.columns
```

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

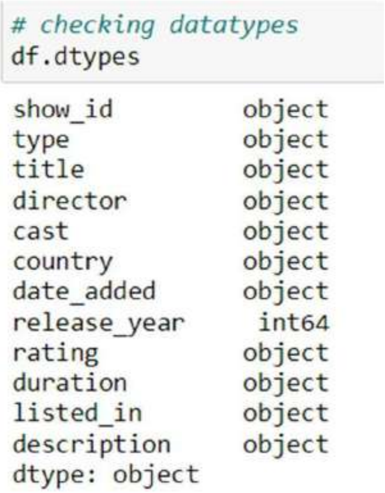
Length:



```
: #length of the data
  len(df)
```

```
: 8807
```

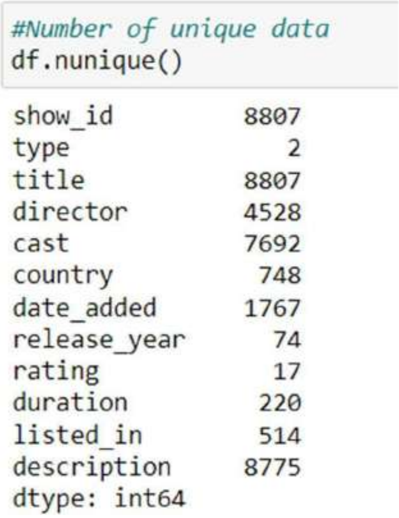
Data Types:



```
# checking datatypes
df.dtypes
```

```
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
```

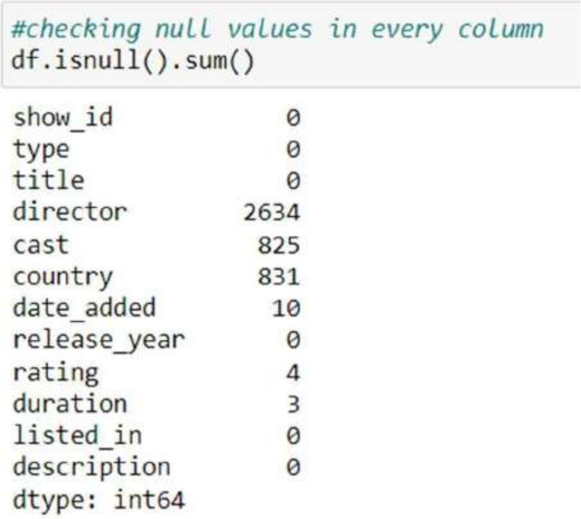
No. Unique Data:



```
#Number of unique data
df.nunique()
```

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

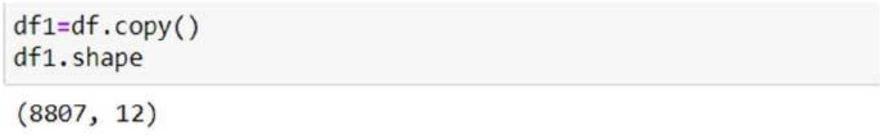
Checking Null values:



```
#checking null values in every column
df.isnull().sum()
```

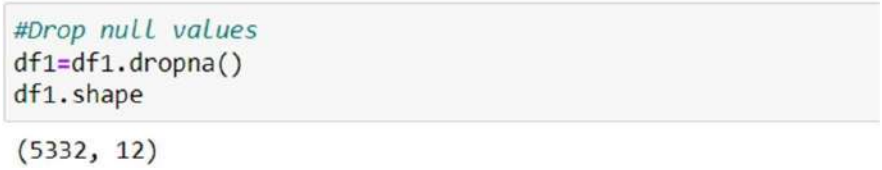
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

Copy of datasets:



```
df1=df.copy()
df1.shape
```

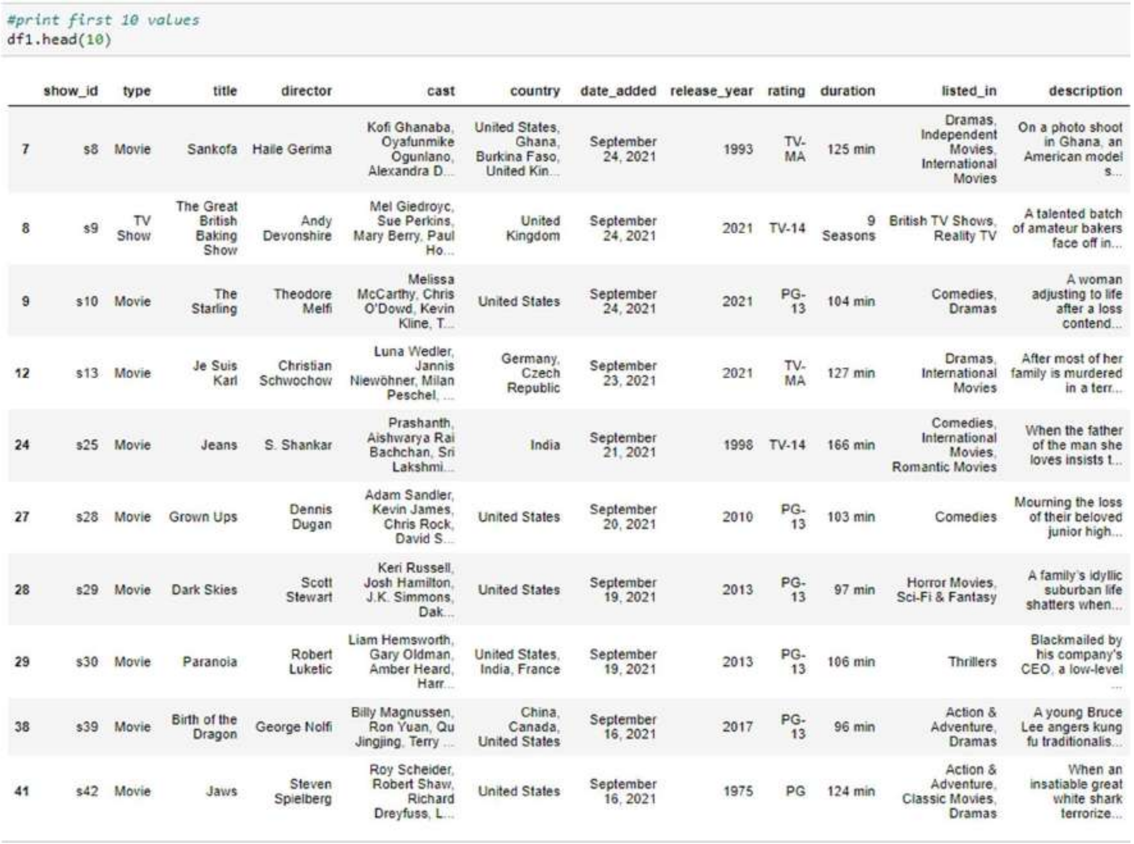
```
(8807, 12)
```



```
#Drop null values
df1=df1.dropna()
df1.shape
```

```
(5332, 12)
```

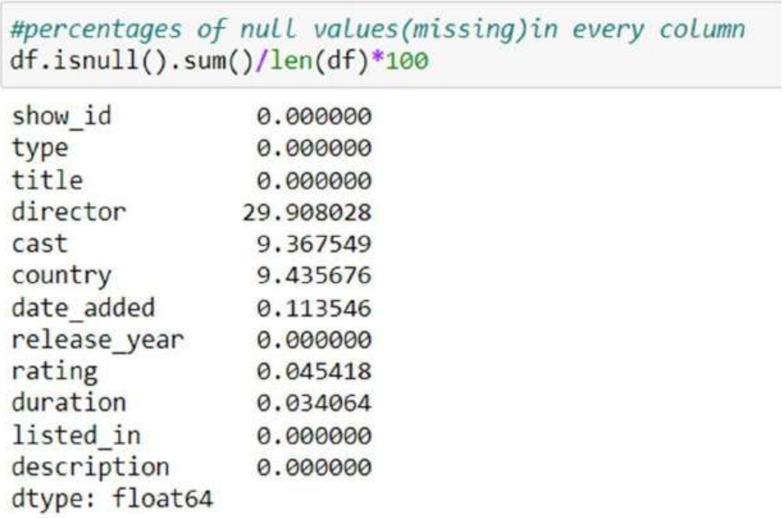
First 10 values in the data set:



```
#print first 10 values
df1.head(10)
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...
8	s9	TV Show	The Great British Baking Show	Andy Devonshire	Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho...	United Kingdom	September 24, 2021	2021	TV-14	9 Seasons	British TV Shows, Reality TV	A talented batch of amateur bakers face off in...
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	2021	PG-13	104 min	Comedies, Dramas	A woman adjusting to life after a loss contend...
12	s13	Movie	Je Suis Karl	Christian Schwochow	Luna Wedler, Jannis Niewöhner, Milan Peschel, ...	Germany, Czech Republic	September 23, 2021	2021	TV-MA	127 min	Dramas, International Movies	After most of her family is murdered in a terr...
24	s25	Movie	Jeans	S. Shankar	Prashanth, Aishwarya Rai Bachchan, Sri Lakshmi...	India	September 21, 2021	1998	TV-14	166 min	Comedies, International Movies, Romantic Movies	When the father of the man she loves insists t...
27	s28	Movie	Grown Ups	Dennis Dugan	Adam Sandler, Kevin James, Chris Rock, David S...	United States	September 20, 2021	2010	PG-13	103 min	Comedies	Mourning the loss of their beloved junior high...
28	s29	Movie	Dark Skies	Scott Stewart	Keri Russell, Josh Hamilton, J.K. Simmons, Dak...	United States	September 19, 2021	2013	PG-13	97 min	Horror Movies, Sci-Fi & Fantasy	A family's idyllic suburban life shatters when...
29	s30	Movie	Paranoia	Robert Luketic	Liam Hemsworth, Gary Oldman, Amber Heard, Harr...	United States, India, France	September 19, 2021	2013	PG-13	106 min	Thrillers	Blackmailed by his company's CEO, a low-level ...
38	s39	Movie	Birth of the Dragon	George Nolfi	Billy Magnussen, Ron Yuan, Qu Jingjing, Terry ...	China, Canada, United States	September 16, 2021	2017	PG-13	96 min	Action & Adventure, Dramas	A young Bruce Lee angers kung fu traditionalis...
41	s42	Movie	Jaws	Steven Spielberg	Roy Scheider, Robert Shaw, Richard Dreyfuss, L...	United States	September 16, 2021	1975	PG	124 min	Action & Adventure, Classic Movies, Dramas	When an insatiable great white shark terrorize...

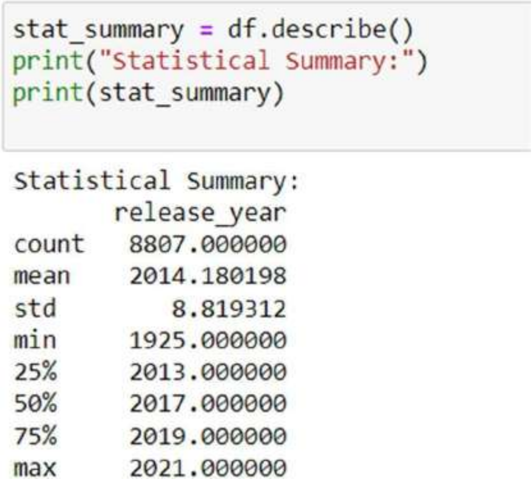
Missing Values:



```
#percentages of null values(missing)in every column
df.isnull().sum()/len(df)*100
```

```
show_id          0.000000
type             0.000000
title            0.000000
director        29.908028
cast             9.367549
country          9.435676
date_added       0.113546
release_year     0.000000
rating           0.045418
duration         0.034064
listed_in        0.000000
description      0.000000
dtype: float64
```

Summary:



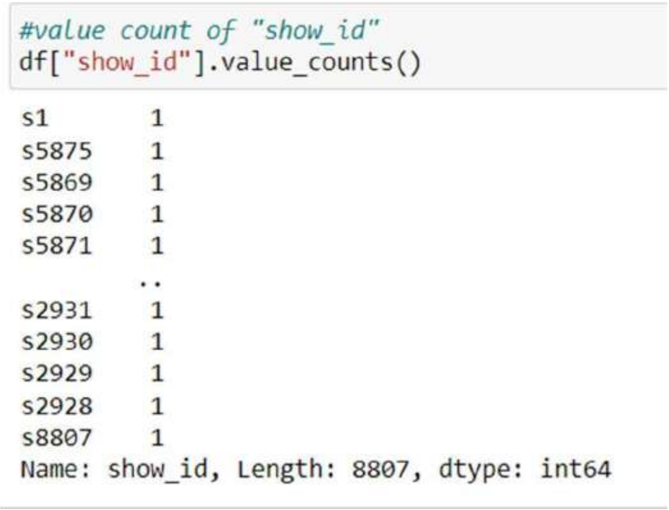
```
stat_summary = df.describe()
print("Statistical Summary:")
print(stat_summary)
```

```
Statistical Summary:
       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
```

2.Non-Graphical Analysis: Value counts and Unique Attributes:

Value counts:

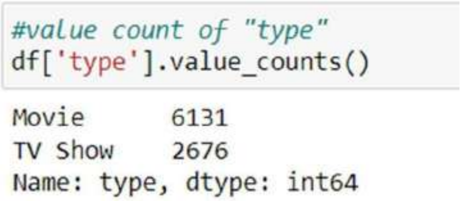
Count of “show_id”



```
#value count of "show_id"
df["show_id"].value_counts()
```

```
s1       1
s5875    1
s5869    1
s5870    1
s5871    1
        ..
s2931    1
s2930    1
s2929    1
s2928    1
s8807    1
Name: show_id, Length: 8807, dtype: int64
```

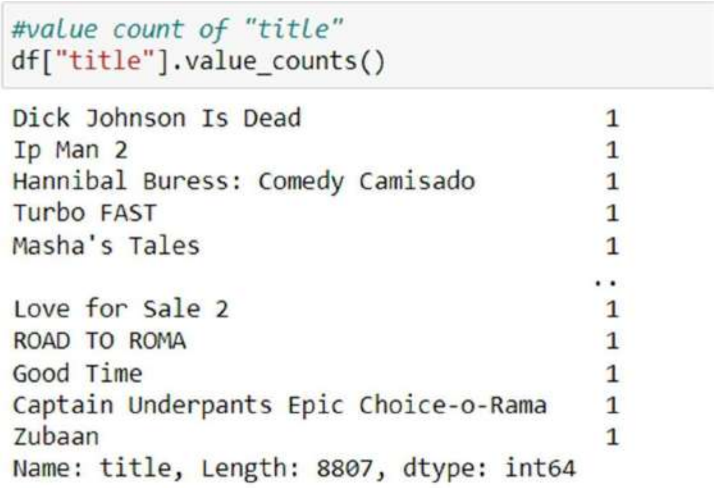
Count of “Type”



```
#value count of "type"
df['type'].value_counts()
```

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

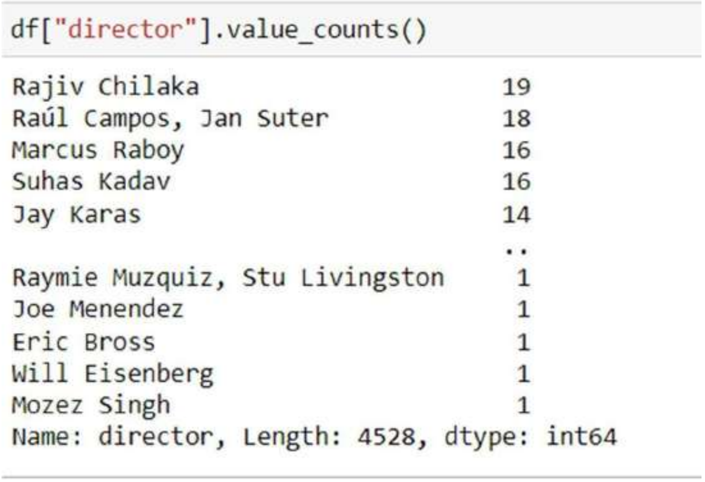
Count of “Title”



```
#value count of "title"
df["title"].value_counts()
```

```
Dick Johnson Is Dead                     1
Ip Man 2                                 1
Hannibal Buress: Comedy Camisado         1
Turbo FAST                               1
Masha's Tales                            1
                                        ..
Love for Sale 2                          1
ROAD TO ROMA                             1
Good Time                                1
Captain Underpants Epic Choice-o-Rama    1
Zubaan                                   1
Name: title, Length: 8807, dtype: int64
```

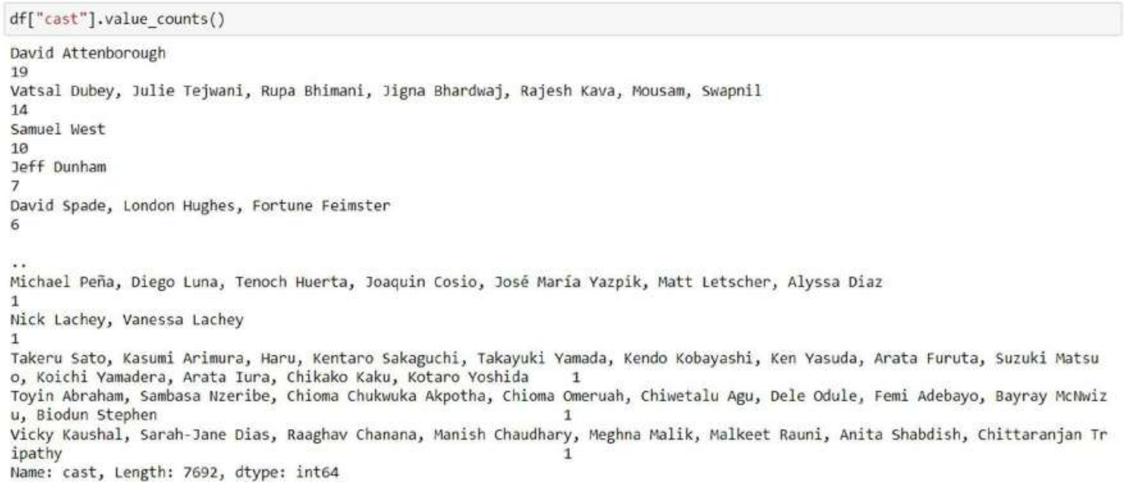
Count of “Director”



```
df["director"].value_counts()
```

```
Rajiv Chilaka                     19
Raúl Campos, Jan Suter            18
Marcus Raboy                      16
Suhas Kadav                       16
Jay Karas                         14
                                  ..
Raymie Muzquiz, Stu Livingston     1
Joe Menendez                       1
Eric Bross                         1
Will Eisenberg                     1
Mozez Singh                        1
Name: director, Length: 4528, dtype: int64
```

Count of “Cast”

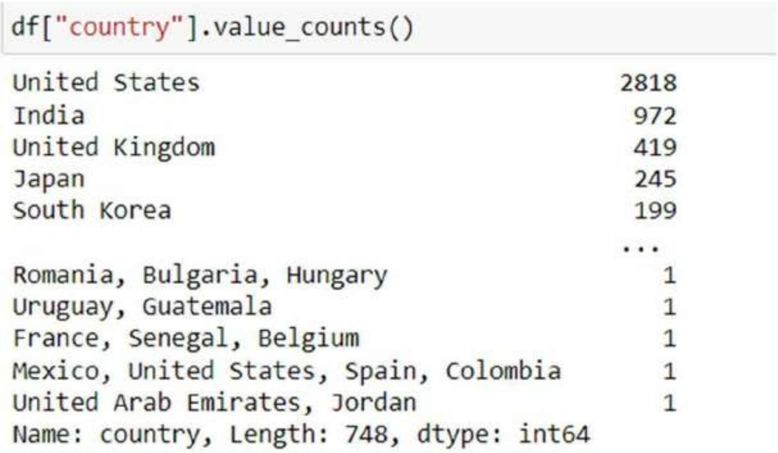


```
df["cast"].value_counts()
```

```
David Attenborough
19
Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mousam, Swapnil
14
Samuel West
10
Jeff Dunham
7
David Spade, London Hughes, Fortune Feimster
6

..
Michael Peña, Diego Luna, Tenoch Huerta, Joaquin Cosio, José María Yazpik, Matt Letscher, Alyssa Diaz
1
Nick Lachey, Vanessa Lachey
1
Takeru Sato, Kasumi Arimura, Haru, Kentaro Sakaguchi, Takayuki Yamada, Kendo Kobayashi, Ken Yasuda, Arata Furuta, Suzuki Matsu
o, Koichi Yamadera, Arata Iura, Chikako Kaku, Kotaro Yoshida     1
Toyin Abraham, Sambasa Nzeribe, Chioma Chukwuka Akpotha, Chioma Omeruah, Chiwetalu Agu, Dele Odule, Femi Adebayo, Bayray McNwiz
u, Biodun Stephen                                               1
Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna Malik, Malkeet Rauni, Anita Shabdish, Chittaranjan Tr
ipathy                                                          1
Name: cast, Length: 7692, dtype: int64
```

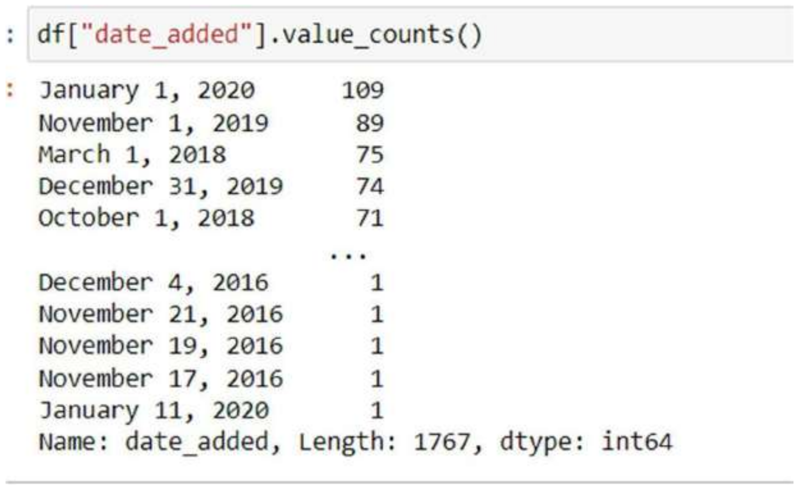
Count of “Country”:



```
df["country"].value_counts()
```

```
United States                             2818
India                                      972
United Kingdom                             419
Japan                                      245
South Korea                                199
                                          ...
Romania, Bulgaria, Hungary                   1
Uruguay, Guatemala                           1
France, Senegal, Belgium                     1
Mexico, United States, Spain, Colombia       1
United Arab Emirates, Jordan                 1
Name: country, Length: 748, dtype: int64
```

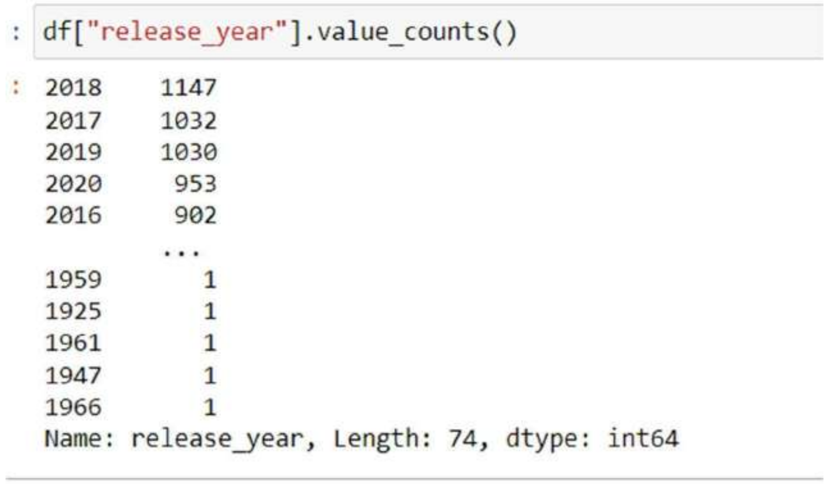
Count of “date_added”:



```
: df["date_added"].value_counts()

: January 1, 2020      109
  November 1, 2019      89
  March 1, 2018         75
  December 31, 2019     74
  October 1, 2018       71
                        ...
  December 4, 2016       1
  November 21, 2016      1
  November 19, 2016      1
  November 17, 2016      1
  January 11, 2020       1
  Name: date_added, Length: 1767, dtype: int64
```

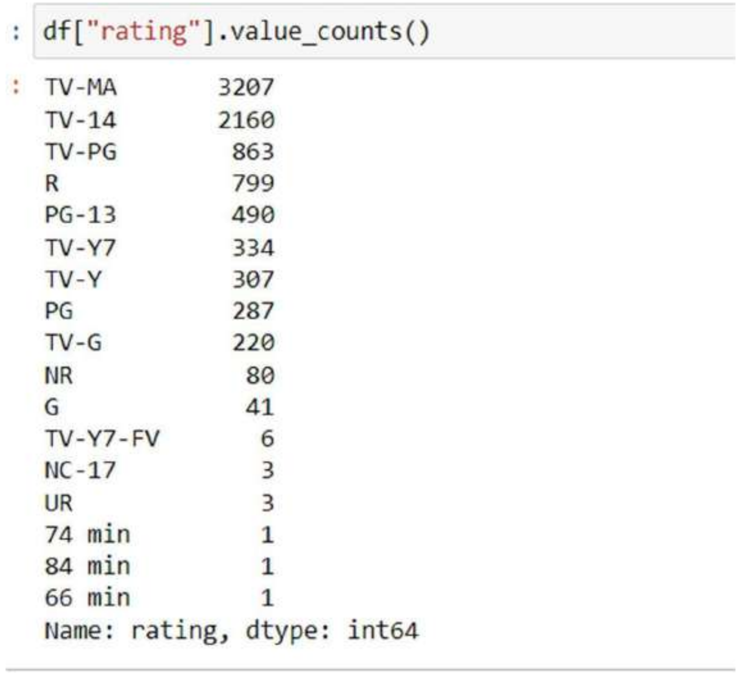
Count of “release_year”



```
: df["release_year"].value_counts()

: 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
```

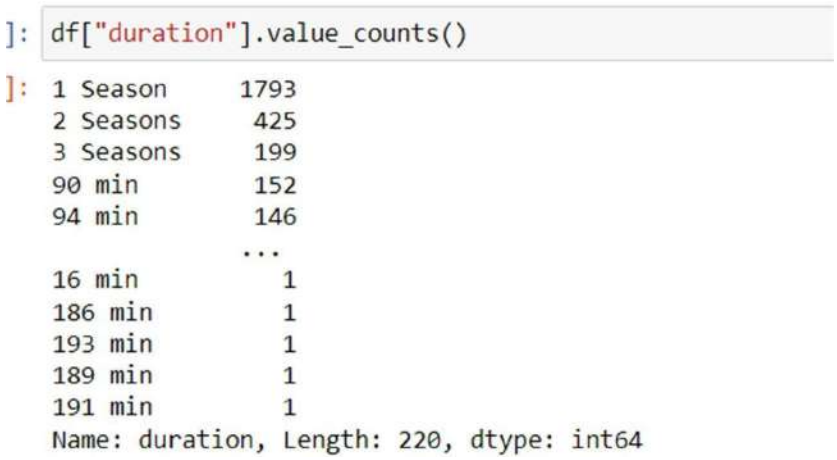
Count of “Rating”



```
: df["rating"].value_counts()

: 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
```

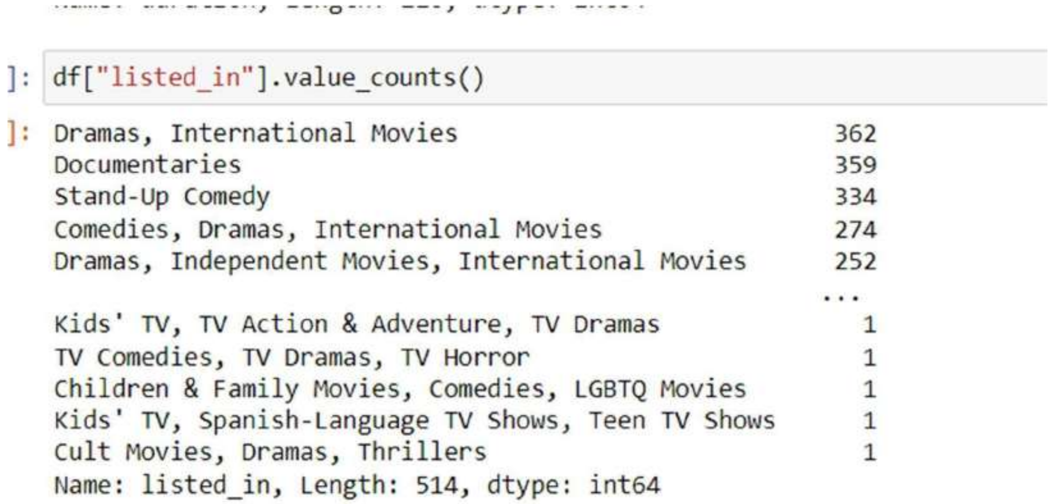
Count of “duration”



```
]: df["duration"].value_counts()

]: 1 Season     1793
   2 Seasons     425
   3 Seasons     199
   90 min        152
   94 min        146
                ...
   16 min          1
   186 min         1
   193 min         1
   189 min         1
   191 min         1
   Name: duration, Length: 220, dtype: int64
```

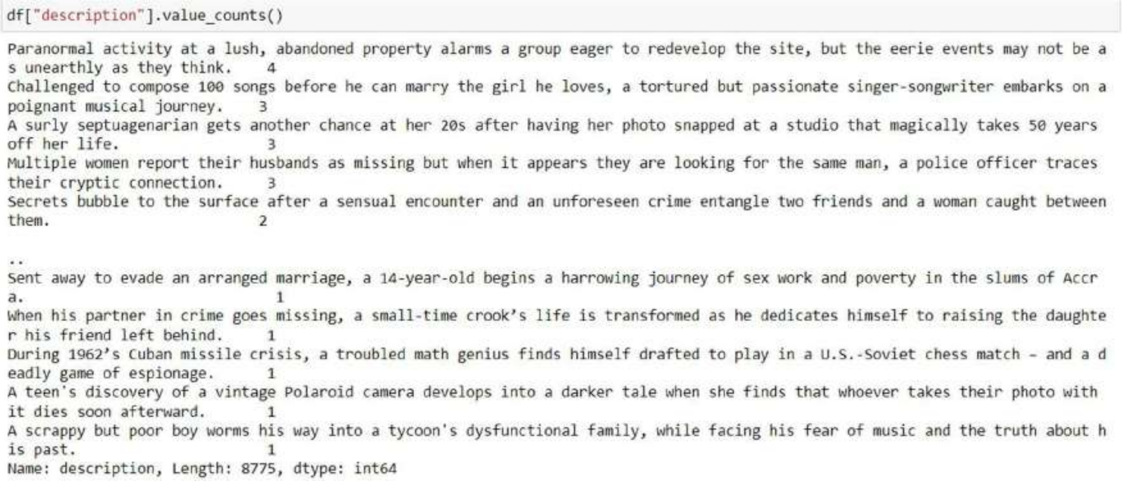
Count of “listed_in”:



```
]: df["listed_in"].value_counts()

]: Dramas, International Movies                          362
   Documentaries                                         359
   Stand-Up Comedy                                       334
   Comedies, Dramas, International Movies                274
   Dramas, Independent Movies, International Movies      252
                                                        ...
   Kids' TV, TV Action & Adventure, TV Dramas              1
   TV Comedies, TV Dramas, TV Horror                       1
   Children & Family Movies, Comedies, LGBTQ Movies        1
   Kids' TV, Spanish-Language TV Shows, Teen TV Shows      1
   Cult Movies, Dramas, Thrillers                          1
   Name: listed_in, Length: 514, dtype: int64
```

count of “Description”:

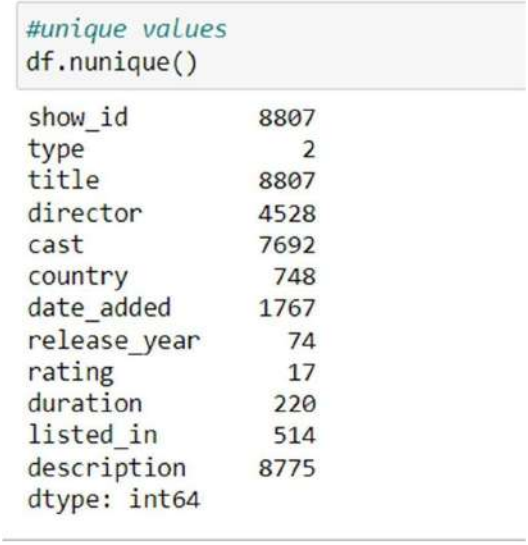


```
df["description"].value_counts()

Paranormal activity at a lush, abandoned property alarms a group eager to redevelop the site, but the eerie events may not be a
s unearthly as they think.    4
Challenged to compose 100 songs before he can marry the girl he loves, a tortured but passionate singer-songwriter embarks on a
poignant musical journey.    3
A surly septuagenarian gets another chance at her 20s after having her photo snapped at a studio that magically takes 50 years
off her life.                3
Multiple women report their husbands as missing but when it appears they are looking for the same man, a police officer traces
their cryptic connection.    3
Secrets bubble to the surface after a sensual encounter and an unforeseen crime entangle two friends and a woman caught between
them.                       2

..
Sent away to evade an arranged marriage, a 14-year-old begins a harrowing journey of sex work and poverty in the slums of Accr
a.                            1
When his partner in crime goes missing, a small-time crook’s life is transformed as he dedicates himself to raising the daughte
r his friend left behind.    1
During 1962’s Cuban missile crisis, a troubled math genius finds himself drafted to play in a U.S.-Soviet chess match – and a d
eadly game of espionage.     1
A teen's discovery of a vintage Polaroid camera develops into a darker tale when she finds that whoever takes their photo with
it dies soon afterward.      1
A scrappy but poor boy worms his way into a tycoon's dysfunctional family, while facing his fear of music and the truth about h
is past.                     1
Name: description, Length: 8775, dtype: int64
```

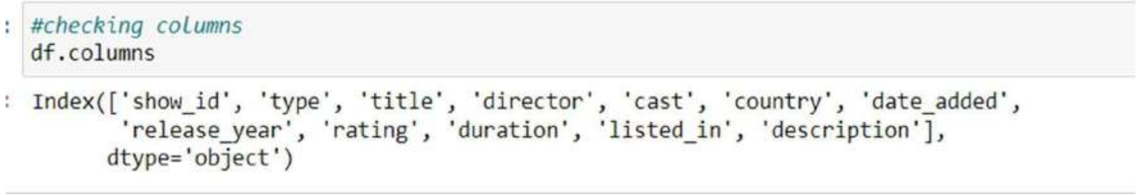
Unique Attributes:



```
#unique values
df.nunique()
```

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

Checking Columns:

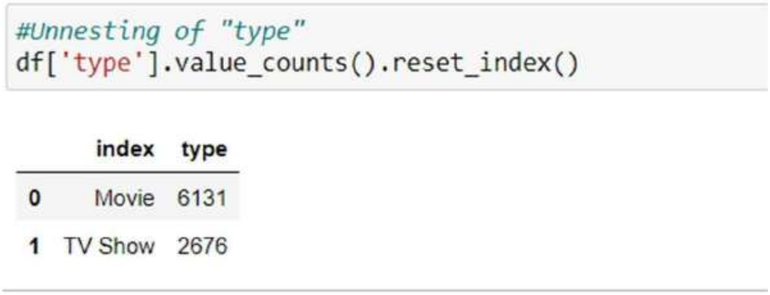


```
#checking columns
df.columns
```

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

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

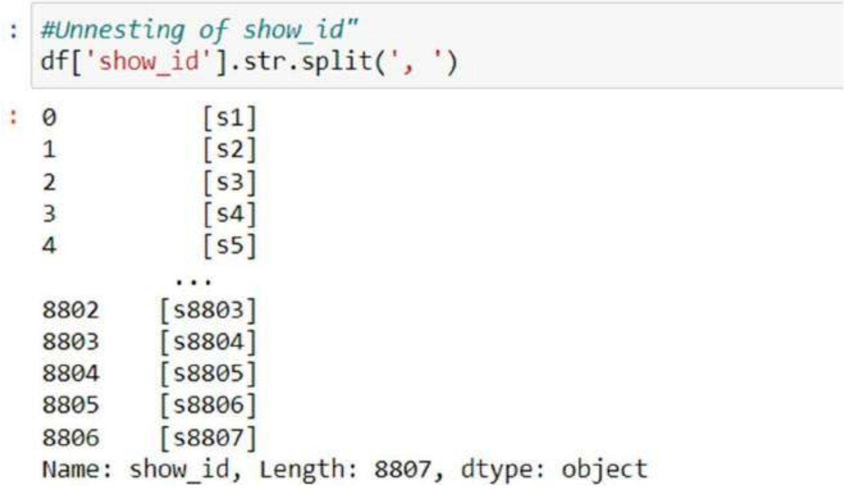
Unnest “Type”:



```
#Unnesting of "type"
df['type'].value_counts().reset_index()
```

	index	type
0	Movie	6131
1	TV Show	2676

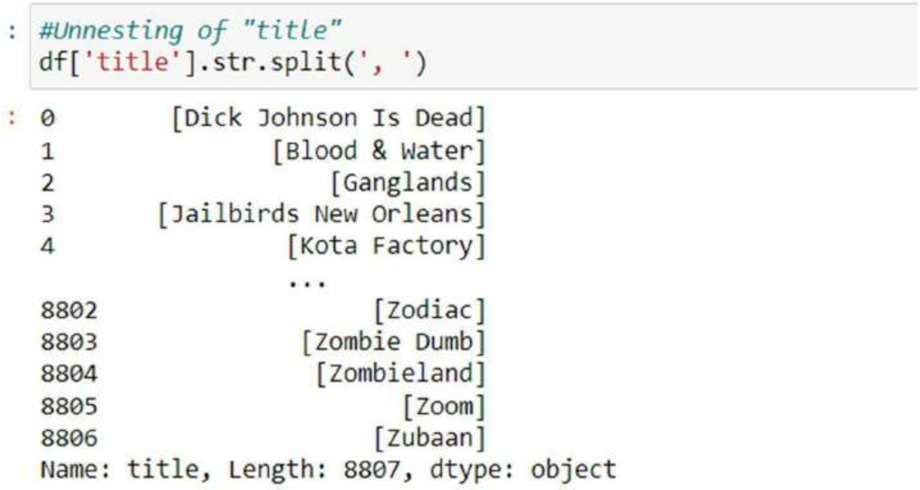
Unnest “Show_id”:



```
: #Unnesting of show_id"
  df['show_id'].str.split(', ')
```

```
: 0            [s1]
  1            [s2]
  2            [s3]
  3            [s4]
  4            [s5]
             ...
  8802     [s8803]
  8803     [s8804]
  8804     [s8805]
  8805     [s8806]
  8806     [s8807]
  Name: show_id, Length: 8807, dtype: object
```

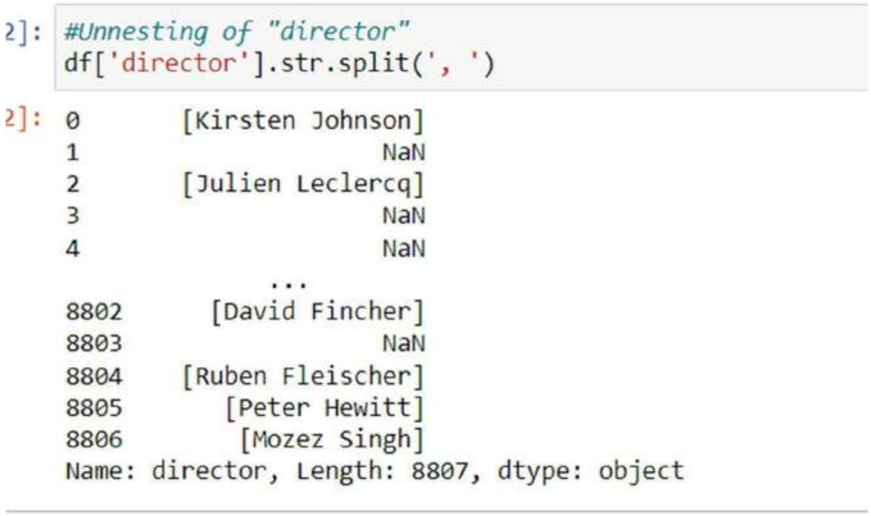
Unnest “Title”:



```
: #Unnesting of "title"
  df['title'].str.split(', ')
```

```
: 0         [Dick Johnson Is Dead]
  1                [Blood & Water]
  2                    [Ganglands]
  3        [Jailbirds New Orleans]
  4                 [Kota Factory]
                    ...
  8802                    [Zodiac]
  8803               [Zombie Dumb]
  8804                [Zombieland]
  8805                      [Zoom]
  8806                    [Zubaan]
  Name: title, Length: 8807, dtype: object
```

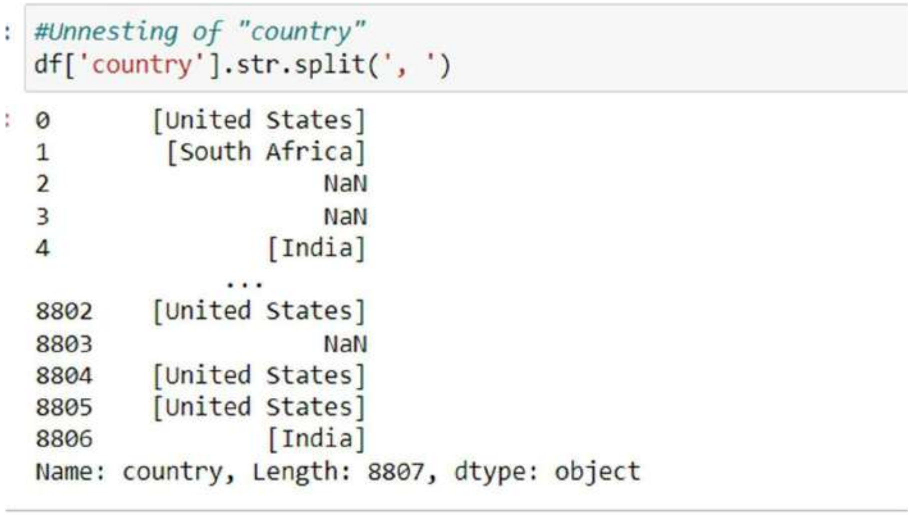
Unnest “director”:



```
2]: #Unnesting of "director"
    df['director'].str.split(', ')

2]: 0       [Kirsten Johnson]
    1                     NaN
    2       [Julien Leclercq]
    3                     NaN
    4                     NaN
                  ...
    8802      [David Fincher]
    8803                  NaN
    8804    [Ruben Fleischer]
    8805       [Peter Hewitt]
    8806        [Mozez Singh]
    Name: director, Length: 8807, dtype: object
```

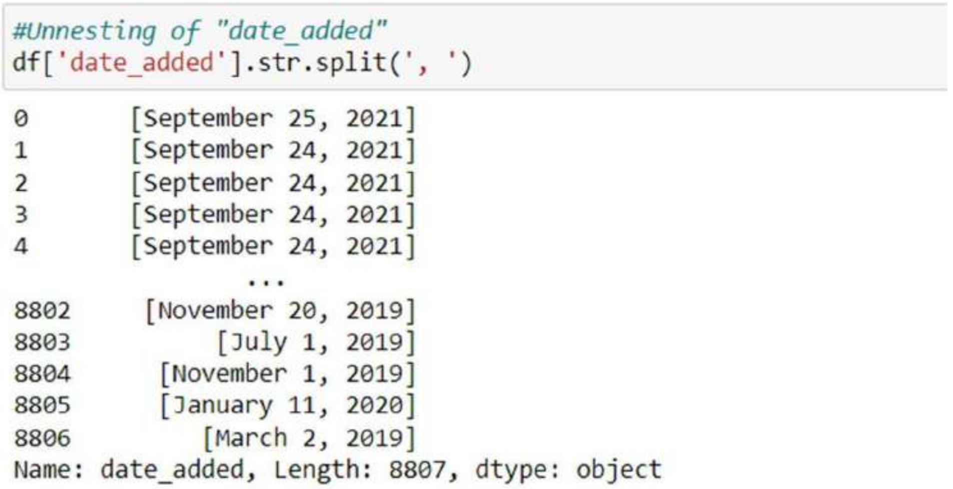
Unnest “country”:



```
: #Unnesting of "country"
  df['country'].str.split(', ')

: 0       [United States]
  1        [South Africa]
  2                   NaN
  3                   NaN
  4               [India]
                ...
  8802    [United States]
  8803                NaN
  8804    [United States]
  8805    [United States]
  8806            [India]
  Name: country, Length: 8807, dtype: object
```

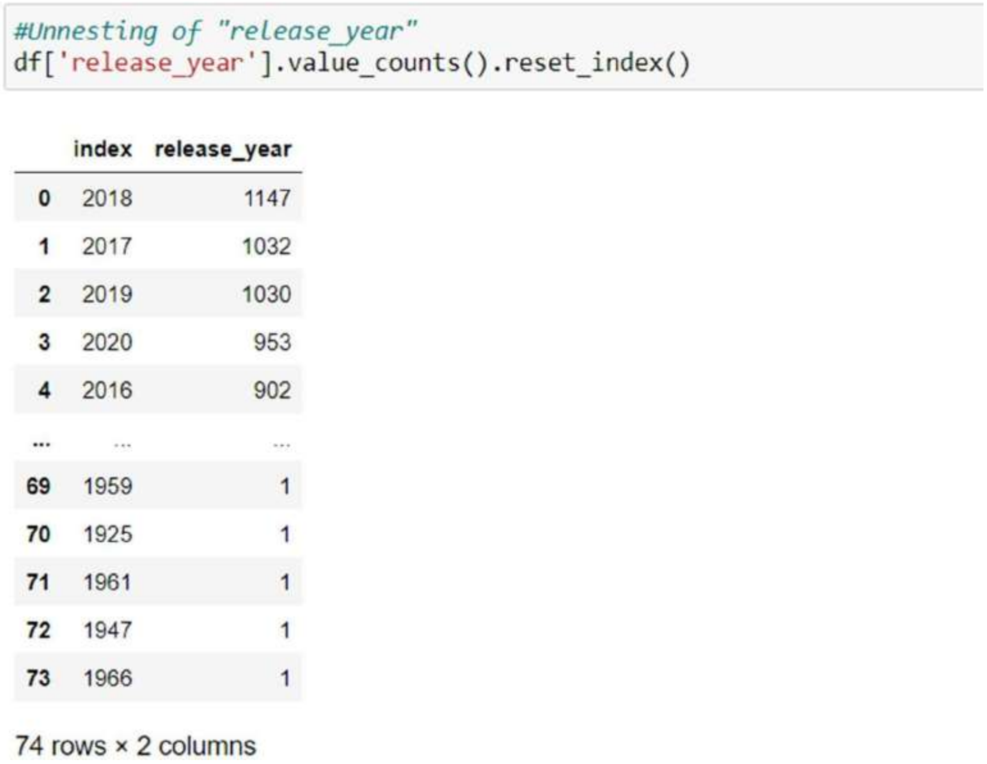
Unnest “date_added”:



```
#Unnesting of "date_added"
df['date_added'].str.split(', ')
```

```
0       [September 25, 2021]
1       [September 24, 2021]
2       [September 24, 2021]
3       [September 24, 2021]
4       [September 24, 2021]
                ...
8802     [November 20, 2019]
8803          [July 1, 2019]
8804      [November 1, 2019]
8805      [January 11, 2020]
8806         [March 2, 2019]
Name: date_added, Length: 8807, dtype: object
```

Unnest “release_year”:

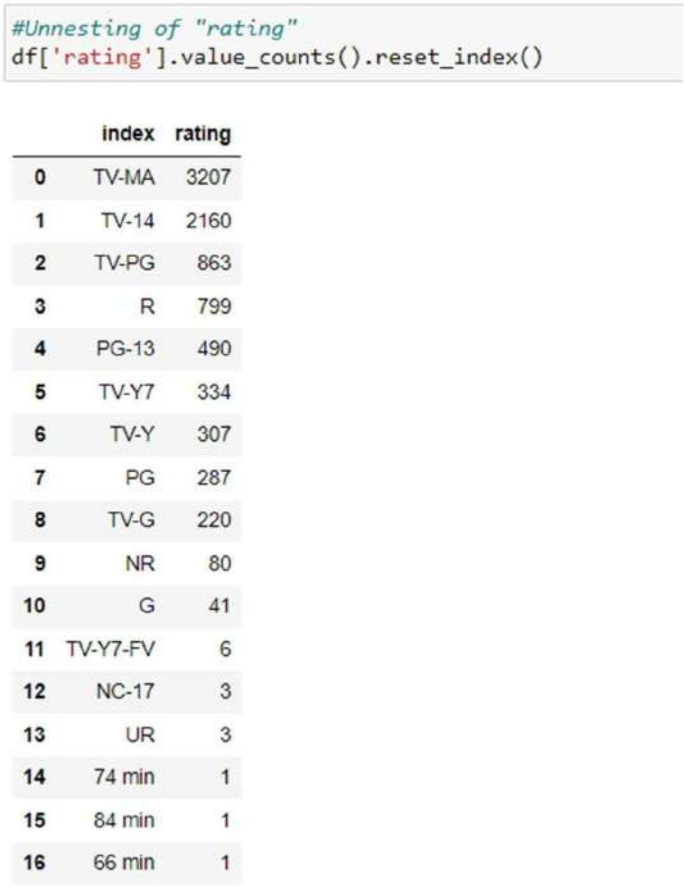


```
#Unnesting of "release_year"
df['release_year'].value_counts().reset_index()
```

	index	release_year
0	2018	1147
1	2017	1032
2	2019	1030
3	2020	953
4	2016	902
...	...	...
69	1959	1
70	1925	1
71	1961	1
72	1947	1
73	1966	1

74 rows × 2 columns

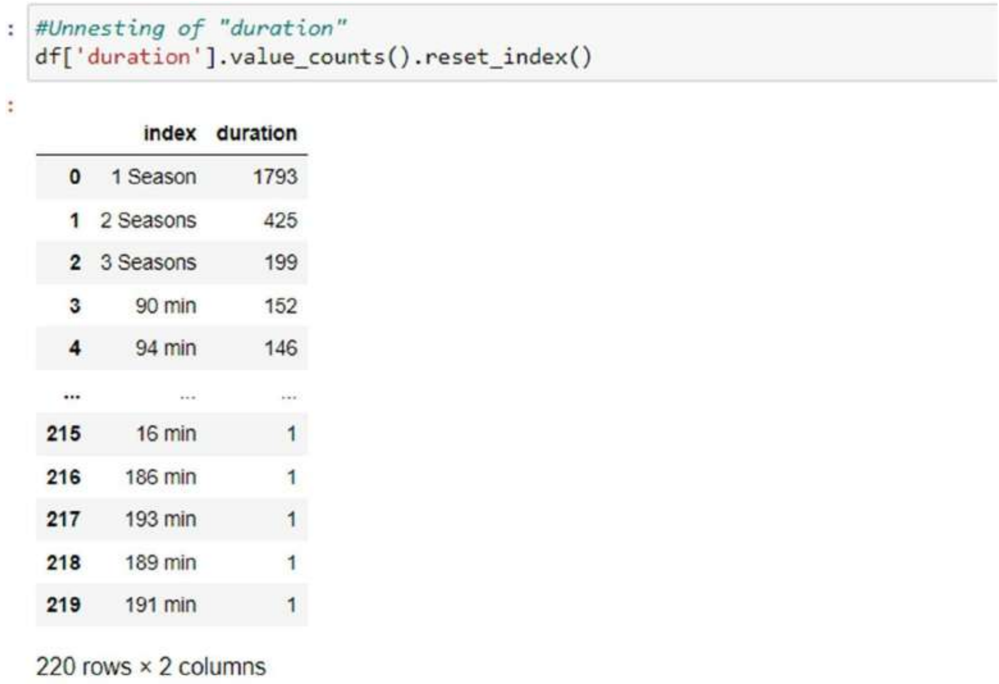
Unnest “rating”:



```
#Unnesting of "rating"
df['rating'].value_counts().reset_index()
```

	index	rating
0	TV-MA	3207
1	TV-14	2160
2	TV-PG	863
3	R	799
4	PG-13	490
5	TV-Y7	334
6	TV-Y	307
7	PG	287
8	TV-G	220
9	NR	80
10	G	41
11	TV-Y7-FV	6
12	NC-17	3
13	UR	3
14	74 min	1
15	84 min	1
16	66 min	1

Unnest “duration”:

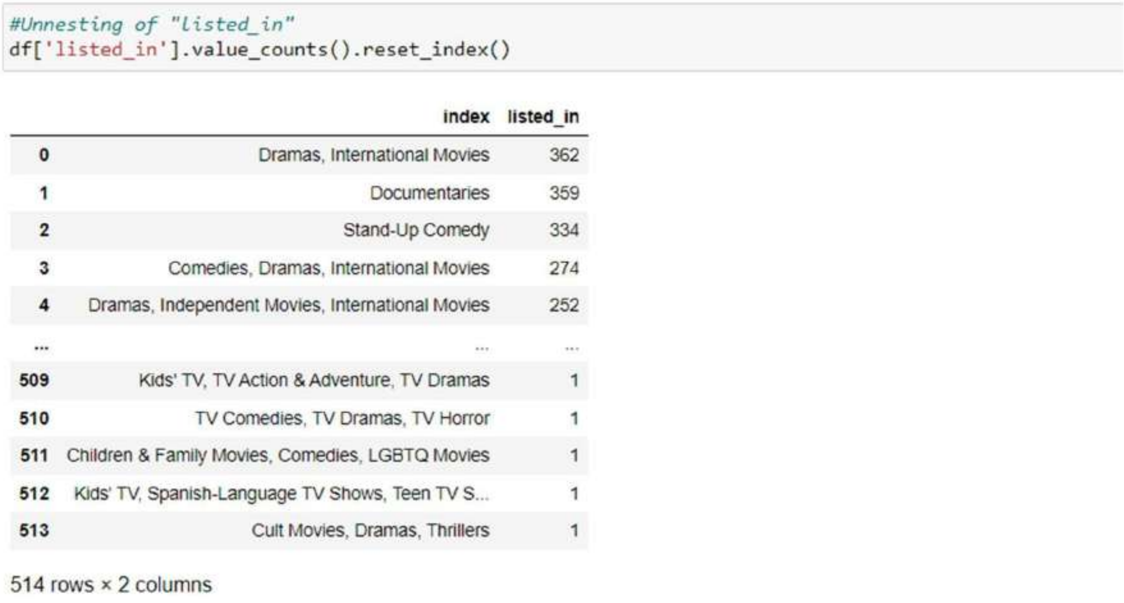


```
#Unnesting of "duration"
df['duration'].value_counts().reset_index()
```

	index	duration
0	1 Season	1793
1	2 Seasons	425
2	3 Seasons	199
3	90 min	152
4	94 min	146
...	...	...
215	16 min	1
216	186 min	1
217	193 min	1
218	189 min	1
219	191 min	1

220 rows × 2 columns

Unnest “listed_in column”:

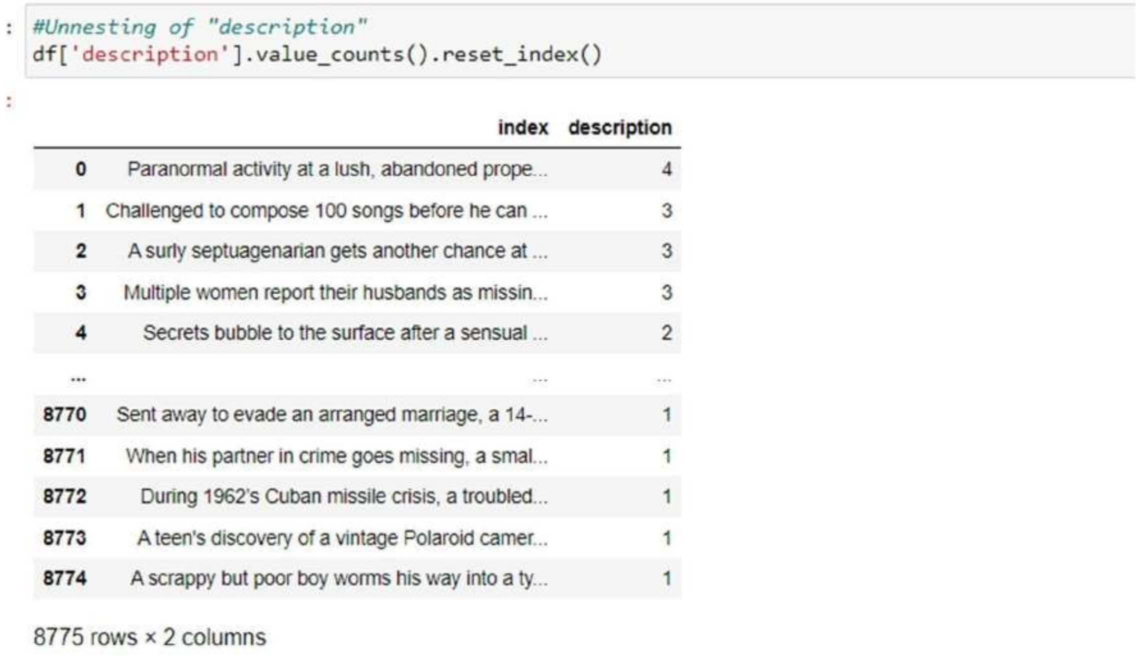


```
#Unnesting of "listed_in"
df['listed_in'].value_counts().reset_index()
```

	index	listed_in
0	Dramas, International Movies	362
1	Documentaries	359
2	Stand-Up Comedy	334
3	Comedies, Dramas, International Movies	274
4	Dramas, Independent Movies, International Movies	252
...	...	...
509	Kids' TV, TV Action & Adventure, TV Dramas	1
510	TV Comedies, TV Dramas, TV Horror	1
511	Children & Family Movies, Comedies, LGBTQ Movies	1
512	Kids' TV, Spanish-Language TV Shows, Teen TV S...	1
513	Cult Movies, Dramas, Thrillers	1

514 rows × 2 columns

Unnest “description”:



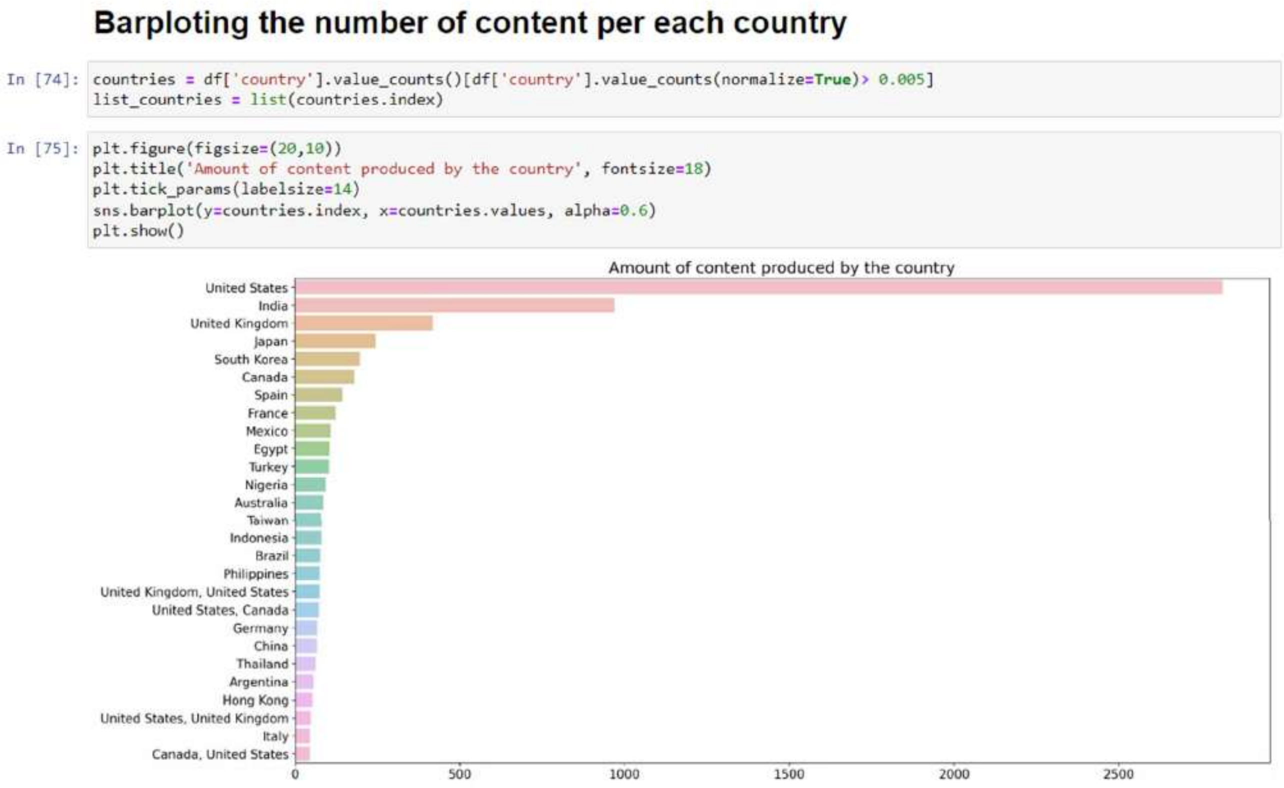
```
#Unnesting of "description"
df['description'].value_counts().reset_index()
```

	index	description
0	Paranormal activity at a lush, abandoned prope...	4
1	Challenged to compose 100 songs before he can ...	3
2	A surly septuagenarian gets another chance at ...	3
3	Multiple women report their husbands as missin...	3
4	Secrets bubble to the surface after a sensual ...	2
...	...	...
8770	Sent away to evade an arranged marriage, a 14-...	1
8771	When his partner in crime goes missing, a smal...	1
8772	During 1962's Cuban missile crisis, a troubled...	1
8773	A teen's discovery of a vintage Polaroid camer...	1
8774	A scrappy but poor boy worms his way into a ty...	1

8775 rows × 2 columns

4.1 For continuous variable: Distplot, countplot, histogram for univariate analysis

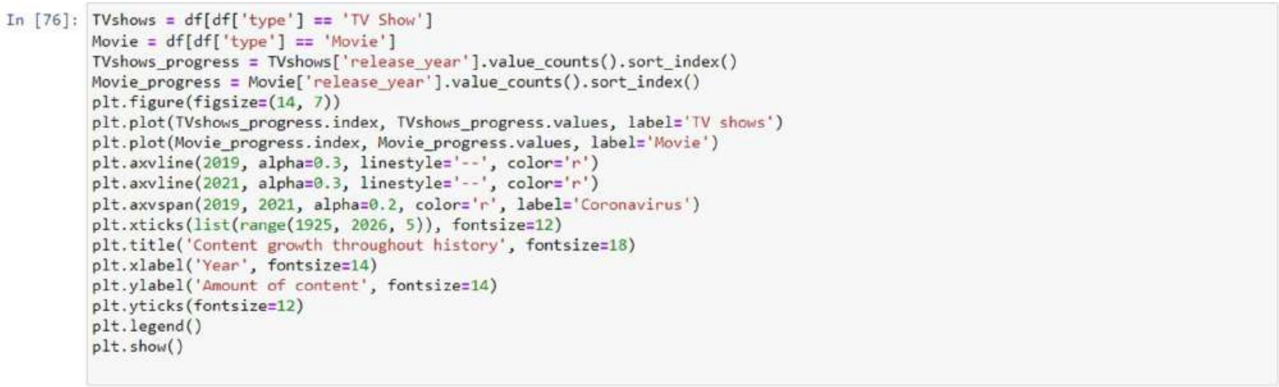
Amount of content produced by each country per year



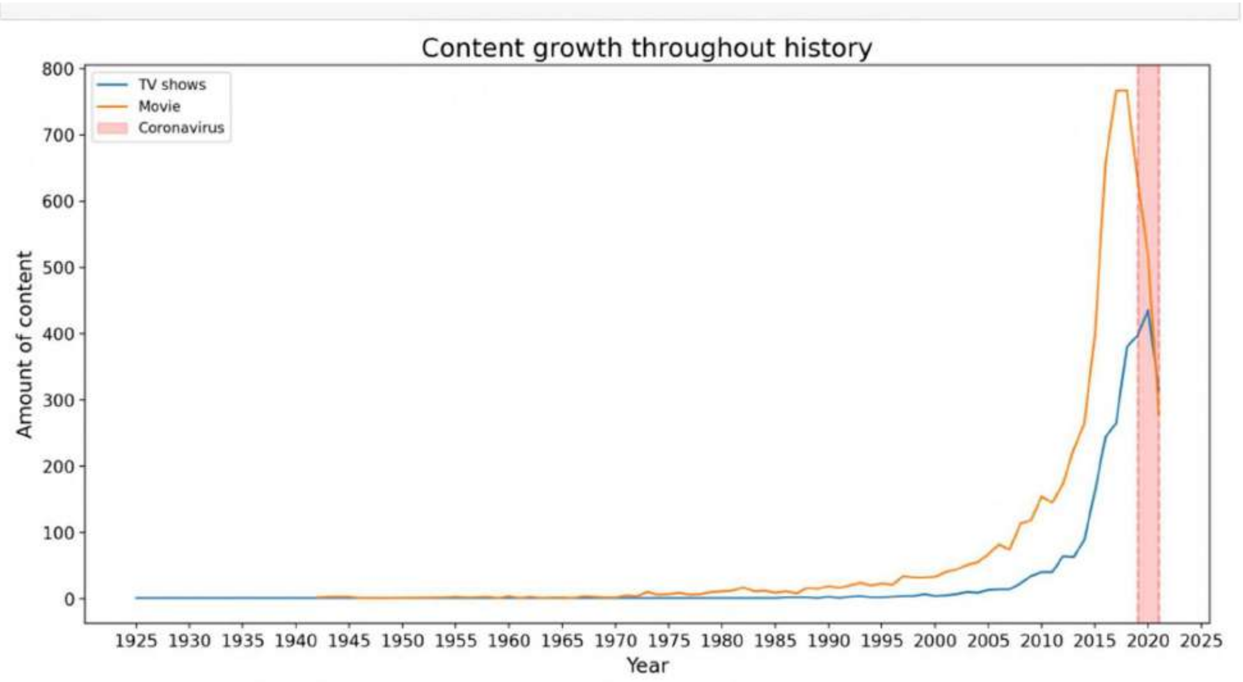
Barploting the number of content per each country

```
In [74]: countries = df['country'].value_counts()[df['country'].value_counts(normalize=True)> 0.005]
         list_countries = list(countries.index)
```

```
In [75]: plt.figure(figsize=(20,10))
         plt.title('Amount of content produced by the country', fontsize=18)
         plt.tick_params(labelsize=14)
         sns.barplot(y=countries.index, x=countries.values, alpha=0.6)
         plt.show()
```



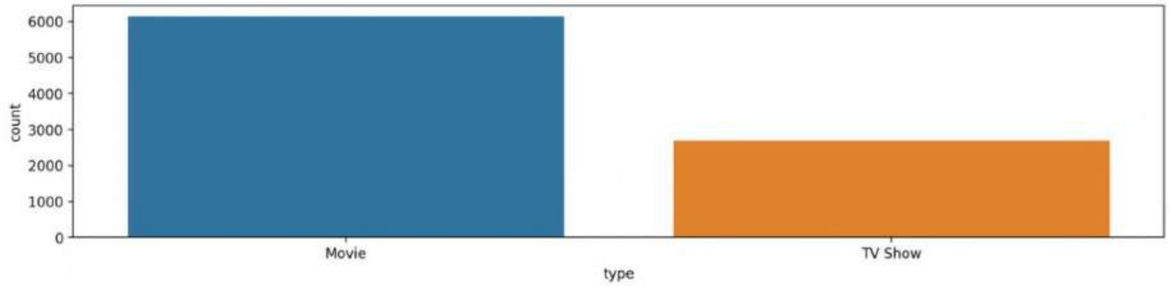
```
In [76]: TVshows = df[df['type'] == 'TV Show']
         Movie = df[df['type'] == 'Movie']
         TVshows_progress = TVshows['release_year'].value_counts().sort_index()
         Movie_progress = Movie['release_year'].value_counts().sort_index()
         plt.figure(figsize=(14, 7))
         plt.plot(TVshows_progress.index, TVshows_progress.values, label='TV shows')
         plt.plot(Movie_progress.index, Movie_progress.values, label='Movie')
         plt.axvline(2019, alpha=0.3, linestyle='--', color='r')
         plt.axvline(2021, alpha=0.3, linestyle='--', color='r')
         plt.axvspan(2019, 2021, alpha=0.2, color='r', label='Coronavirus')
         plt.xticks(list(range(1925, 2026, 5)), fontsize=12)
         plt.title('Content growth throughout history', fontsize=18)
         plt.xlabel('Year', fontsize=14)
         plt.ylabel('Amount of content', fontsize=14)
         plt.yticks(fontsize=12)
         plt.legend()
         plt.show()
```



countplot

```
In [79]: plt.figure(figsize=(14, 3))
         sns.countplot(x='type',data = df)

Out[79]: <Axes: xlabel='type', ylabel='count'>
```



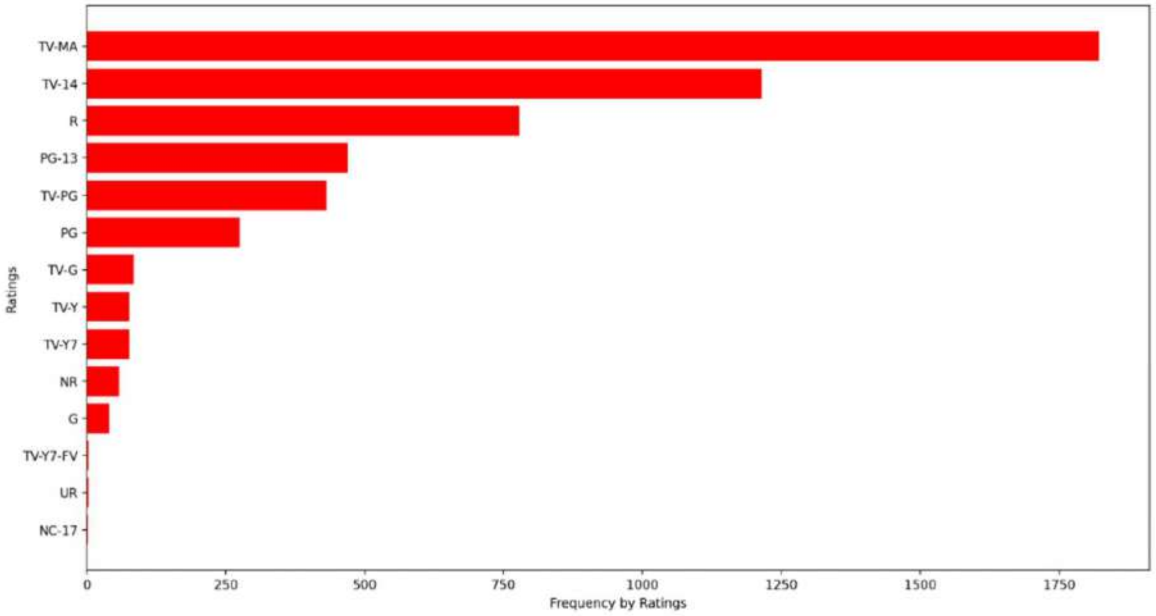
Barplot:

In [91]:
```
#number of distinct titles on the basis of rating
df.groupby(['rating']).agg({"title":"nunique"})
```

Out[91]:
```
          title
  rating
       G     40
   NC-17      2
      NR     58
      PG    275
   PG-13    470
       R    778
   TV-14   1214
    TV-G     84
   TV-MA   1822
   TV-PG    431
    TV-Y     76
   TV-Y7     76
TV-Y7-FV      3
      UR      3
```

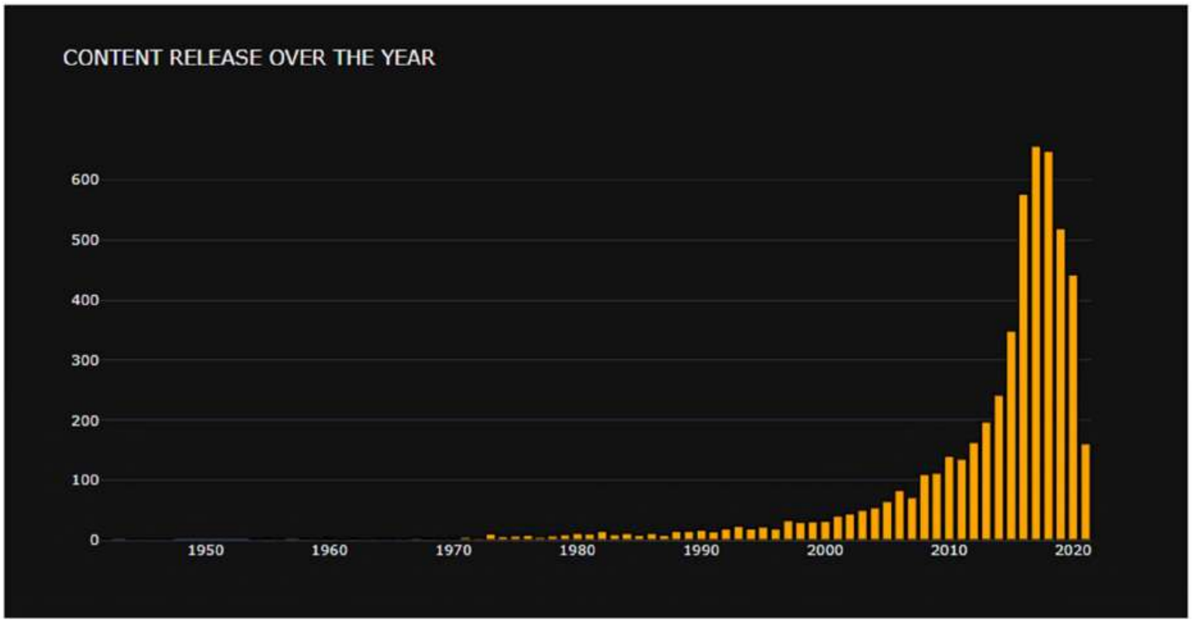
UR 3

In [89]:
```
rating=df.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)[:15]
plt.figure(figsize=(15,8))
plt.barh(rating[::-1]['rating'],rating[::-1]['title'],color=['red'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```

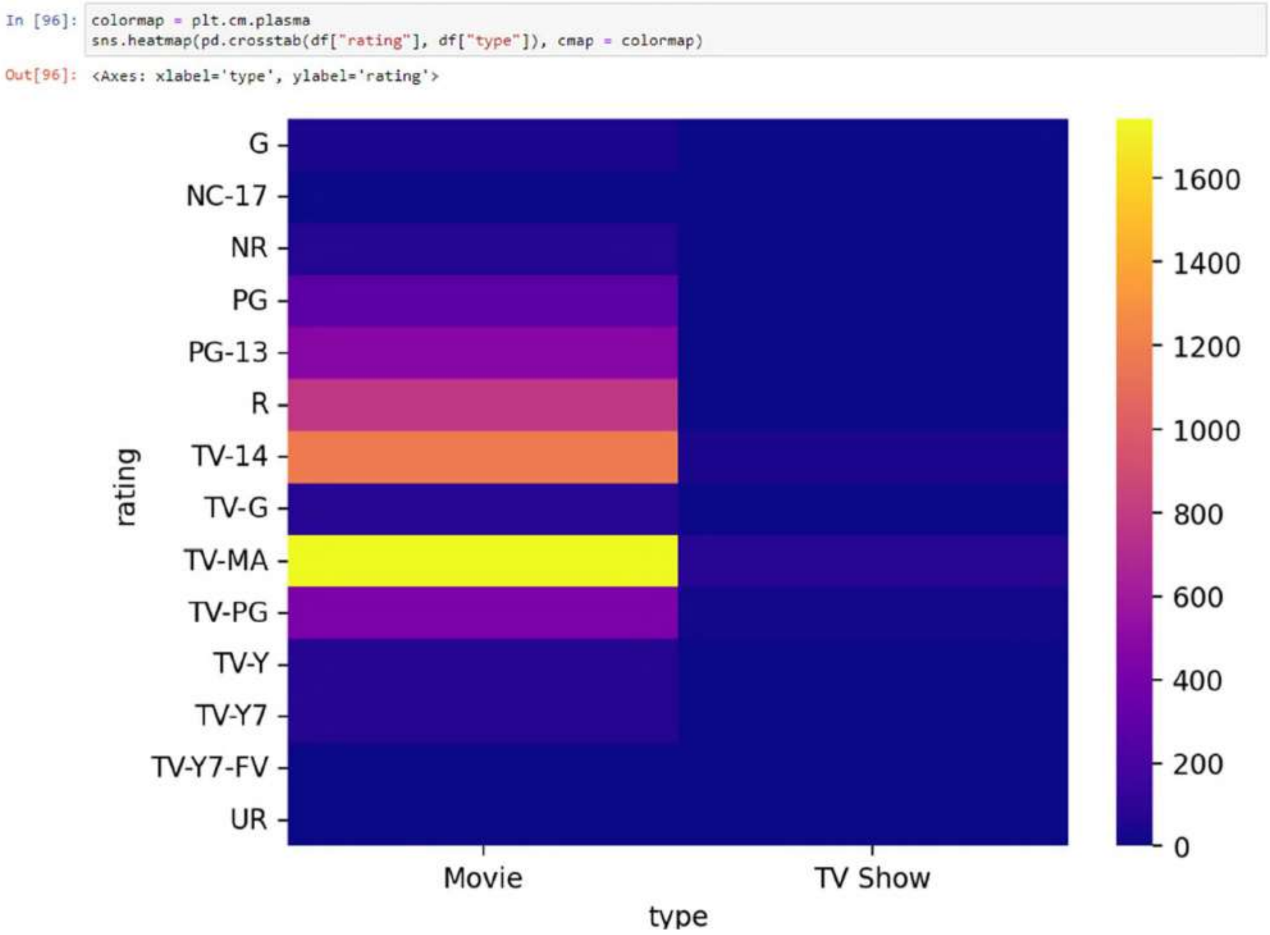


Content release over the year

In [92]:
```
temp_df1 = df['release_year'].value_counts().reset_index()
import plotly.graph_objects as go
trace1 = go.Bar(
 x = temp_df1['index'],
 y = temp_df1['release_year'],
 marker = dict(color = 'rgb(255,165,0)',
 line=dict(color='rgb(0,0,0)',width=1.5)))
layout = go.Layout(template= "plotly_dark",title = 'CONTENT RELEASE OVER THE YEAR')
fig = go.Figure(data = [trace1], layout = layout)
fig.show()
```

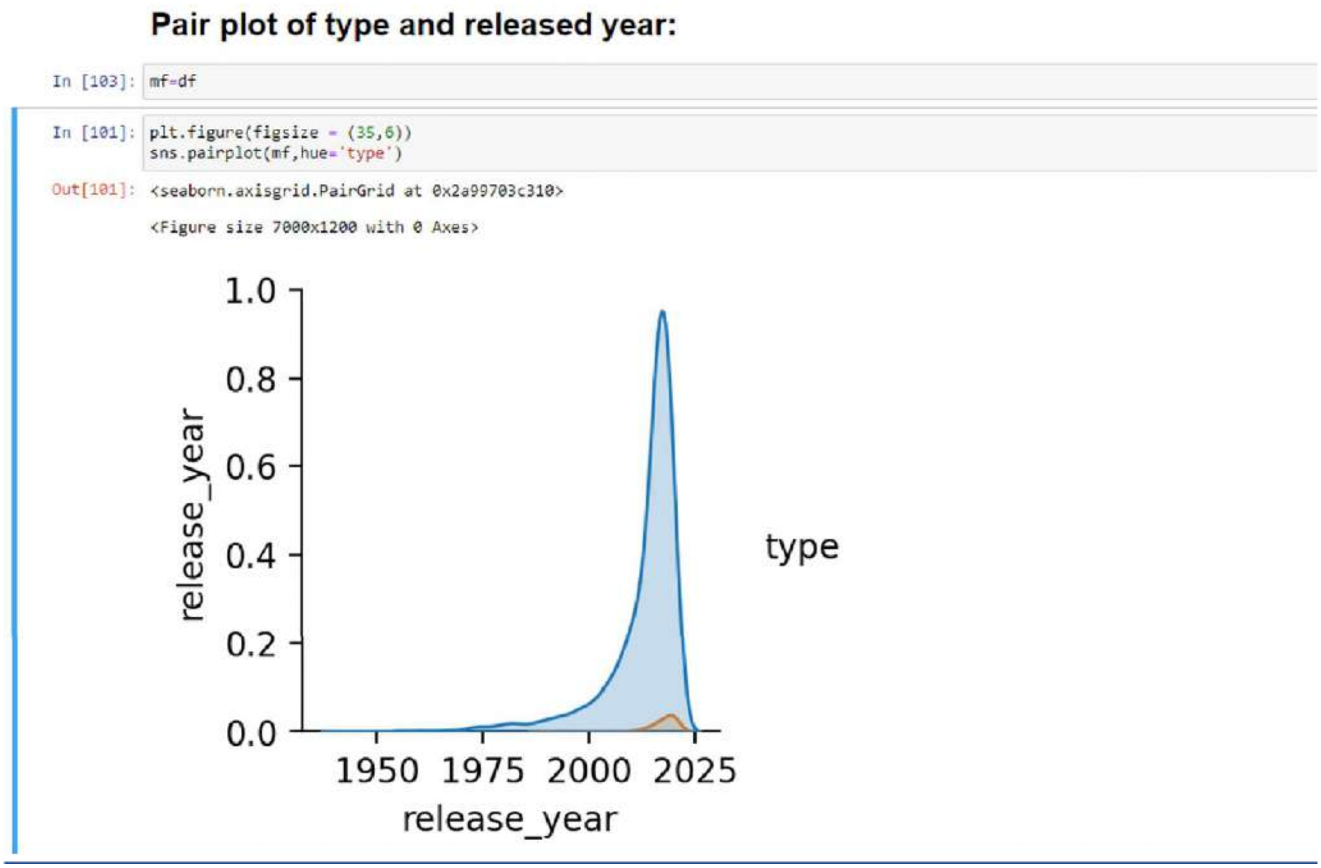


Heatmap:



```
In [96]: colormap = plt.cm.plasma
         sns.heatmap(pd.crosstab(df["rating"], df["type"]), cmap = colormap)

Out[96]: <Axes: xlabel='type', ylabel='rating'>
```



Pair plot of type and released year:

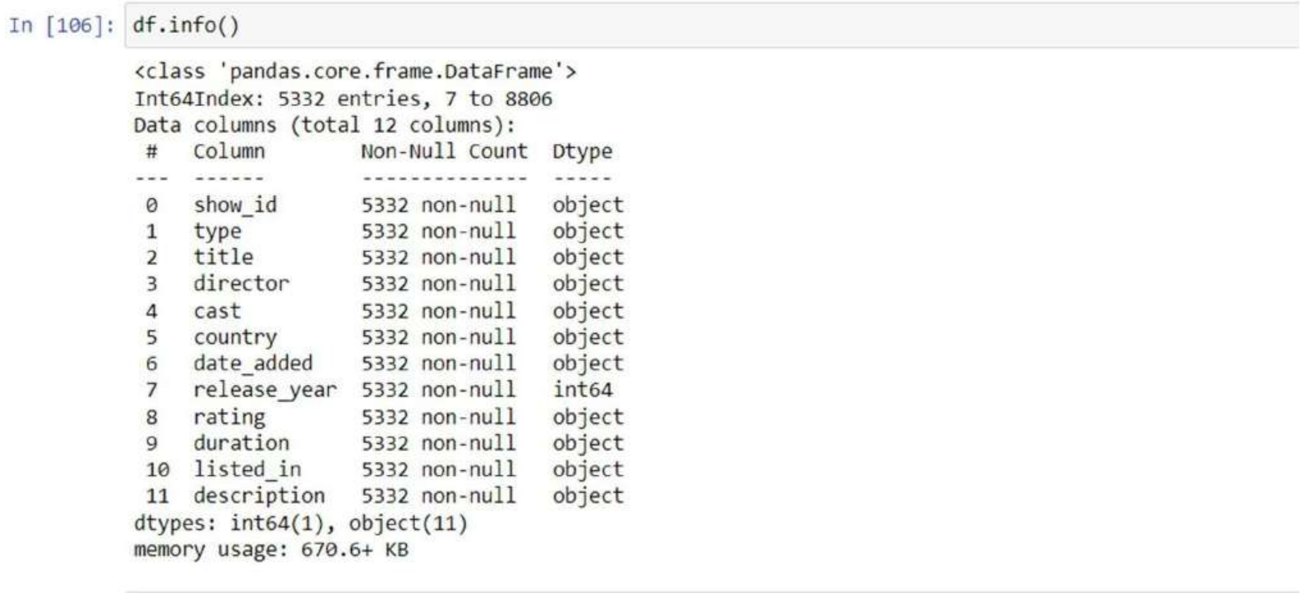
```
In [103]: mf=df
```

```
In [101]: plt.figure(figsize = (35,6))
          sns.pairplot(mf,hue='type')
```

```
Out[101]: <seaborn.axisgrid.PairGrid at 0x2a99703c310>

          <Figure size 7000x1200 with 0 Axes>
```

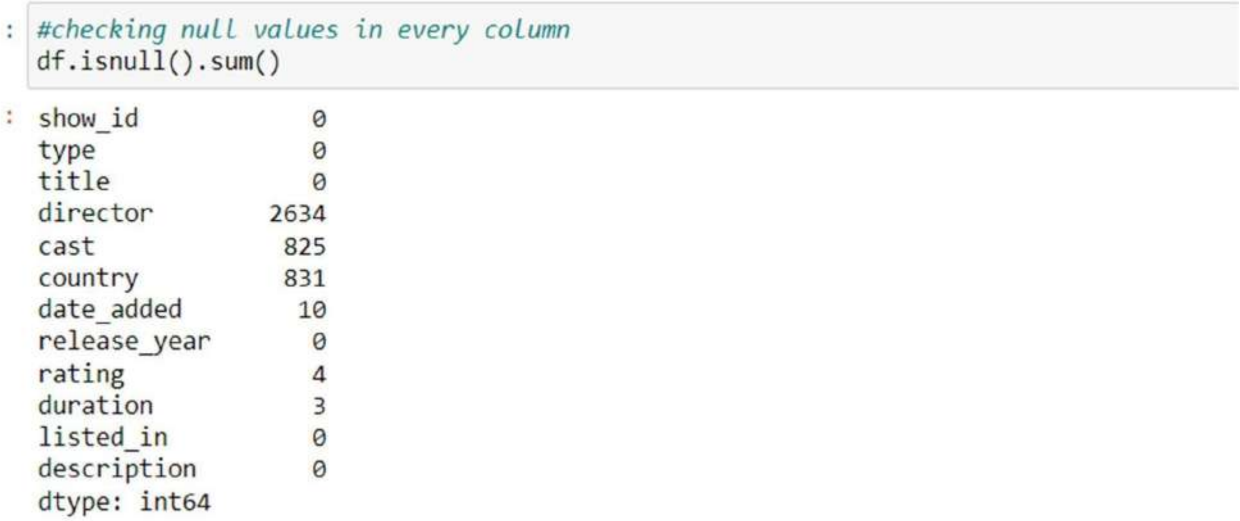
2. Missing Values and outlier check (Optional treatment):



```
In [106]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5332 entries, 7 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       5332 non-null   object
 1   type          5332 non-null   object
 2   title         5332 non-null   object
 3   director      5332 non-null   object
 4   cast          5332 non-null   object
 5   country       5332 non-null   object
 6   date_added    5332 non-null   object
 7   release_year  5332 non-null   int64 
 8   rating        5332 non-null   object
 9   duration      5332 non-null   object
 10  listed_in     5332 non-null   object
 11  description   5332 non-null   object
dtypes: int64(1), object(11)
memory usage: 670.6+ KB
```

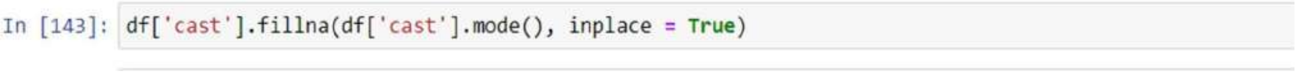
Check Null values:



```
#checking null values in every column
df.isnull().sum()
```

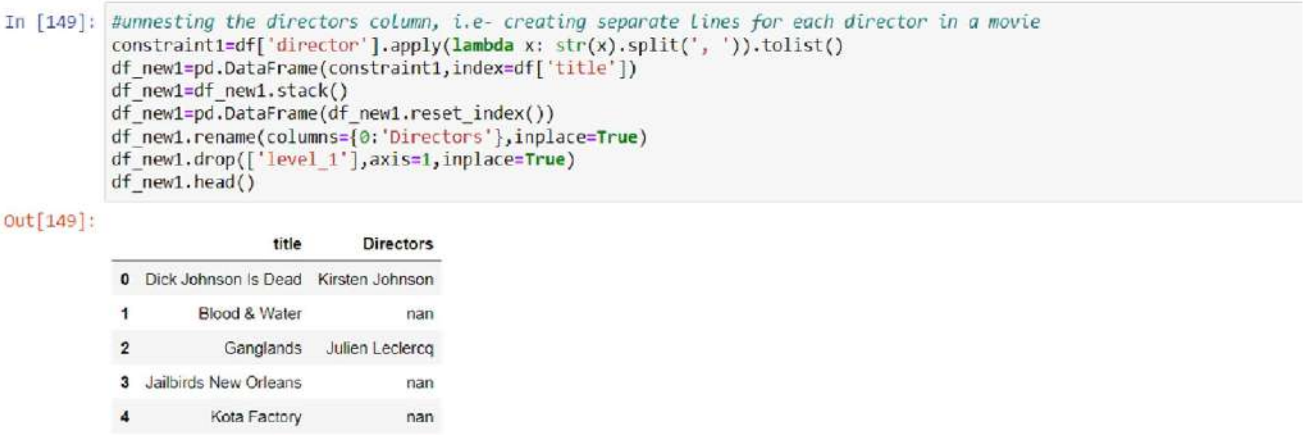
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

Replacing Null Values:



```
In [143]: df['cast'].fillna(df['cast'].mode(), inplace = True)
```

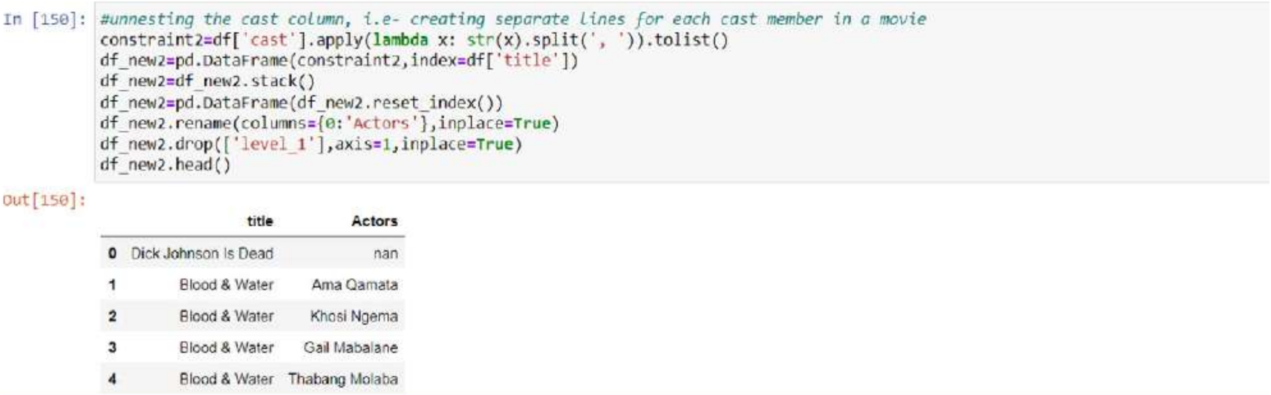
Treatment For Null Values:



```
In [149]: #unnesting the directors column, i.e- creating separate lines for each director in a movie
          constraint1=df['director'].apply(lambda x: str(x).split(', ')).tolist()
          df_new1=pd.DataFrame(constraint1,index=df['title'])
          df_new1=df_new1.stack()
          df_new1=pd.DataFrame(df_new1.reset_index())
          df_new1.rename(columns={0:'Directors'},inplace=True)
          df_new1.drop(['level_1'],axis=1,inplace=True)
          df_new1.head()
```

Out[149]:

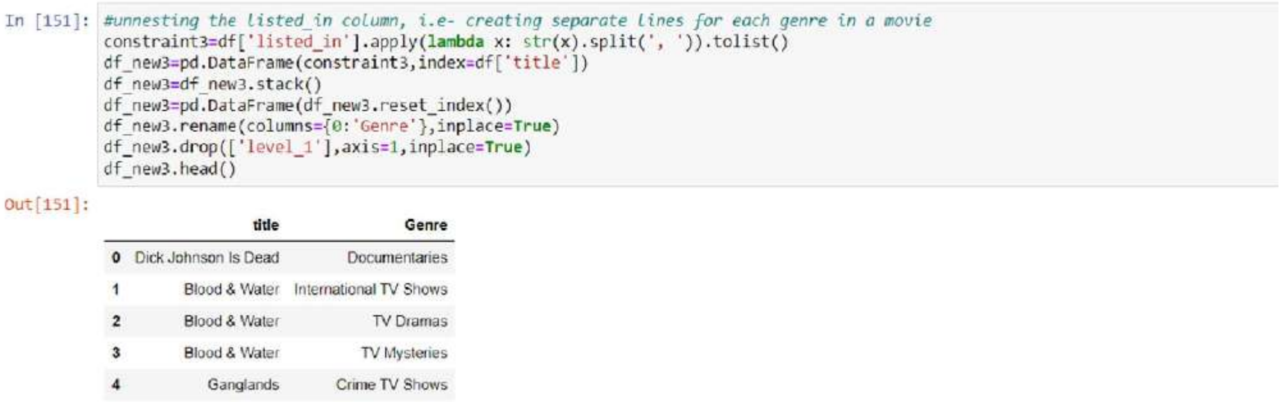
	title	Directors
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan



```
In [150]: #unnesting the cast column, i.e- creating separate lines for each cast member in a movie
          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=pd.DataFrame(df_new2.reset_index())
          df_new2.rename(columns={0:'Actors'},inplace=True)
          df_new2.drop(['level_1'],axis=1,inplace=True)
          df_new2.head()
```

Out[150]:

	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

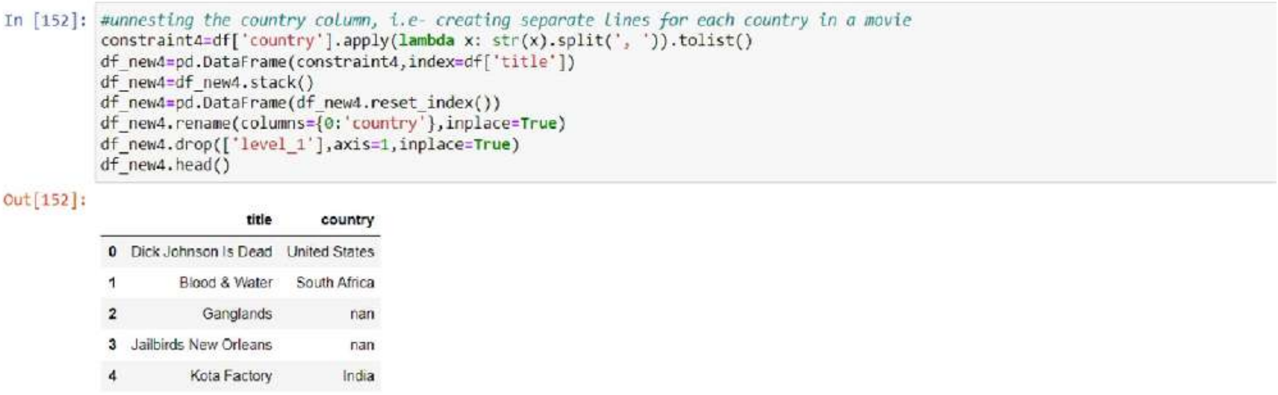


In [151]:

```python
#unnesting the listed_in column, i.e- creating separate lines for each genre in a movie
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.rename(columns={0:'Genre'},inplace=True)
df_new3.drop(['level_1'],axis=1,inplace=True)
df_new3.head()
```

Out[151]:

	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

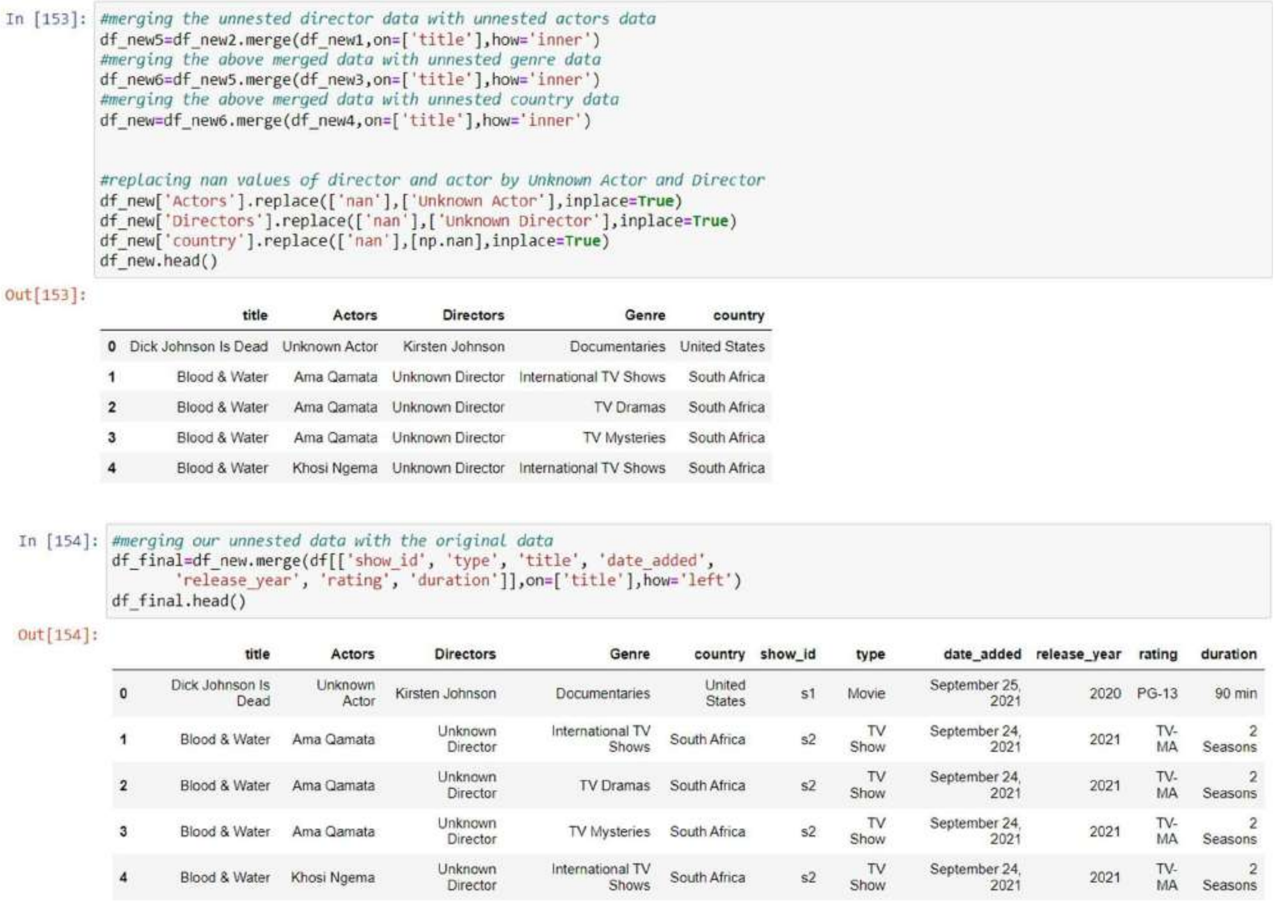


In [152]:

```python
#unnesting the country column, i.e- creating separate lines for each country in a movie
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.rename(columns={0:'country'},inplace=True)
df_new4.drop(['level_1'],axis=1,inplace=True)
df_new4.head()
```

Out[152]:

	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



In [153]:

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


#replacing nan values of director and actor by Unknown Actor and Director
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()
```

Out[153]:

	title	Actors	Directors	Genre	country
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa
3	Blood & Water	Ama Qamata	Unknown Director	TV Mysteries	South Africa
4	Blood & Water	Khosi Ngema	Unknown Director	International TV Shows	South Africa

In [154]:

```python
#merging our unnested data with the original data
df_final=df_new.merge(df[['show_id', 'type', 'title', 'date_added',
        'release_year', 'rating', 'duration']],on=['title'],how='left')
df_final.head()
```

Out[154]:

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

```
In [157]: #now checking nulls
          df_final.isnull().sum()
```

```
Out[157]: title               0
          Actors              0
          Directors           0
          Genre               0
          country         11897
          show_id             0
          type                0
          date_added        158
          release_year        0
          rating             67
          duration            3
          dtype: int64
```

```
In [158]: df_final.loc[df_final['duration'].isnull(),'duration']=df_final.loc[df_final['duration'].isnull(),'duration'].fillna(df_final['ra


          df_final.loc[df_final['rating'].str.contains('min', na=False),'rating']='NR'


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

```
Out[158]: title               0
          Actors              0
          Directors           0
          Genre               0
          country         11897
          show_id             0
          type                0
          date_added        158
          release_year        0
          rating             67
          duration            0
          dtype: int64
```

```
In [159]: #Ratings can't be in min, so it has been made NR(i.e- Non Rated)
          df_final.loc[df_final['rating'].str.contains('min', na=False),'rating']='NR'
          df_final['rating'].fillna('NR',inplace=True)
          pd.set_option('display.max_rows',None)
```

```
In [160]: #just an attempt to observe nulls in date_added column
          df_final[df_final['date_added'].isnull()].head()
```

Out[160]:

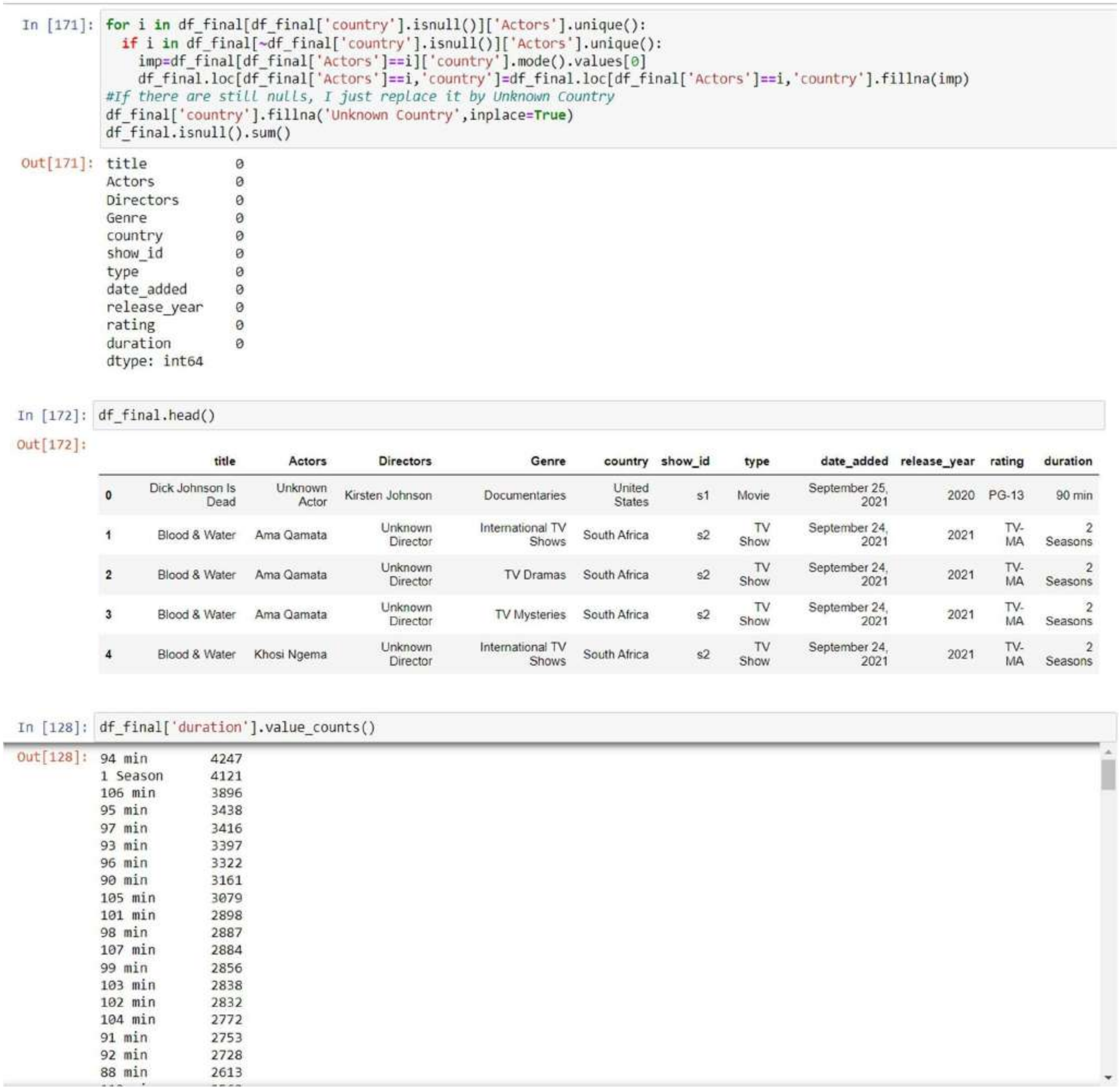
	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating	duration
136893	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136894	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	TV Comedies	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136895	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	TV Dramas	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136896	A Young Doctor's Notebook and Other Stories	Jon Hamm	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136897	A Young Doctor's Notebook and Other Stories	Jon Hamm	Unknown Director	TV Comedies	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons

```
In [161]: #date added column is imputed on the basis of release year,i.e- suppose there's a null for date_added
          #when release year was 2013.So below piece of code just checks the mode of date added for release year=2013
          # and imputes in place of nulls the corresponding mode

          for i in df_final[df_final['date_added'].isnull()]['release_year'].unique():
            imp=df_final[df_final['release_year']==i]['date_added'].mode().values[0]
            df_final.loc[df_final['release_year']==i,'date_added']=df_final.loc[df_final['release_year']==i,'date_added'].fillna(imp)
```

```
In [170]: #country column is imputed on the basis of director,i.e- suppose there's a null for country
          #when we have a director whose other movies have a country given.So below piece of code just checks the mode of
          #country for the director
          # and imputes in place of nulls the corresponding mode

          for i in df_final[df_final['country'].isnull()]['Directors'].unique():
            if i in df_final[~df_final['country'].isnull()]['Directors'].unique():
              imp=df_final[df_final['Directors']==i]['country'].mode().values[0]
              df_final.loc[df_final['Directors']==i,'country']=df_final.loc[df_final['Directors']==i,'country'].fillna(imp)
```



In [171]:
```python
for i in df_final[df_final['country'].isnull()]['Actors'].unique():
  if i in df_final[~df_final['country'].isnull()]['Actors'].unique():
    imp=df_final[df_final['Actors']==i]['country'].mode().values[0]
    df_final.loc[df_final['Actors']==i,'country']=df_final.loc[df_final['Actors']==i,'country'].fillna(imp)
#If there are still nulls, I just replace it by Unknown Country
df_final['country'].fillna('Unknown Country',inplace=True)
df_final.isnull().sum()
```

Out[171]:
```
title           0
Actors          0
Directors       0
Genre           0
country         0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
dtype: int64
```

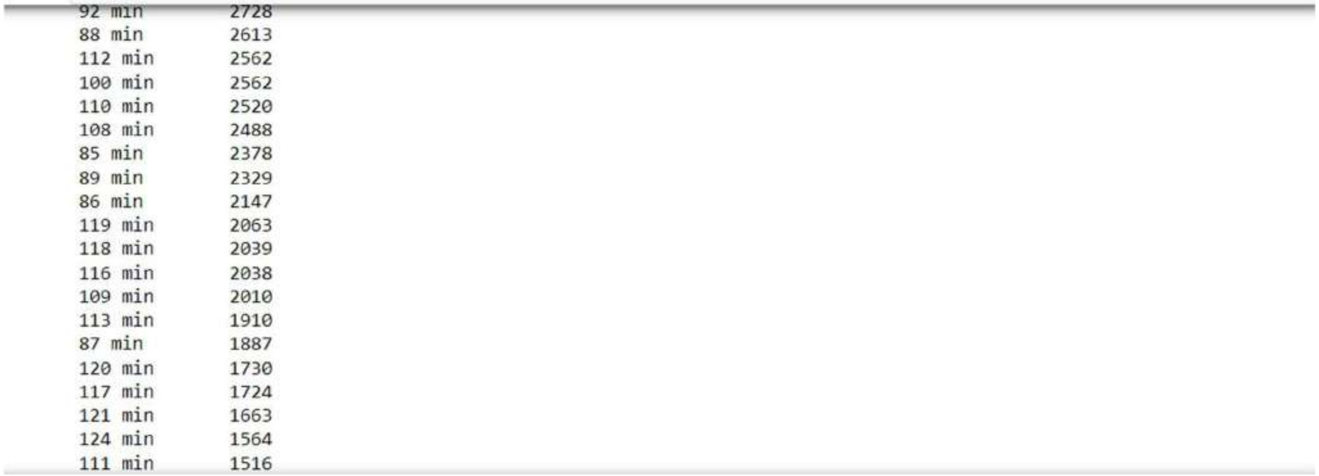
In [172]:
```python
df_final.head()
```

Out[172]:

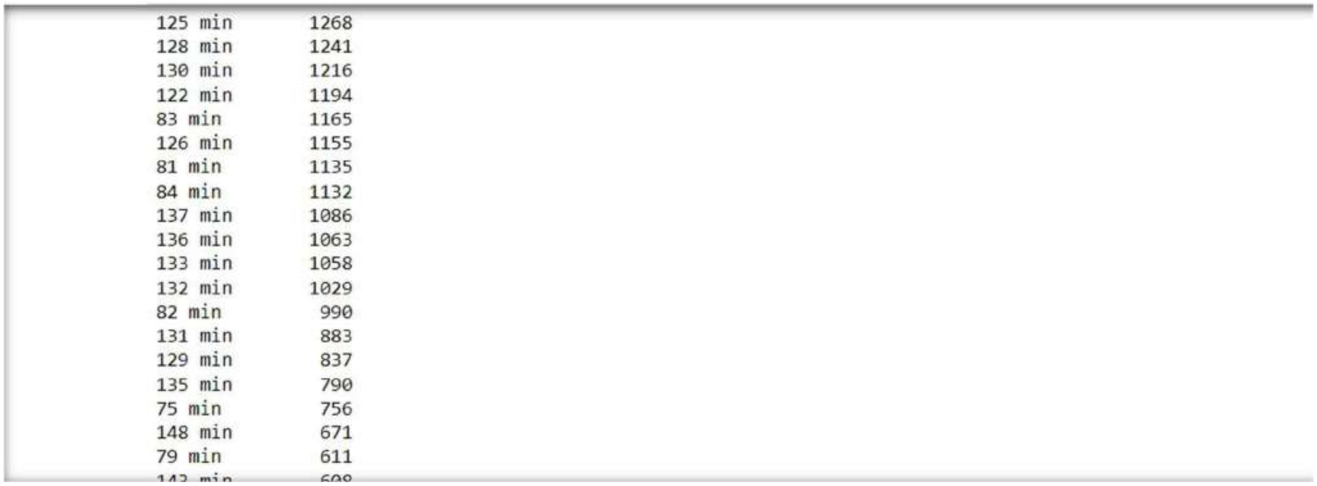
	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating	duration
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90 min
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV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

In [128]:
```python
df_final['duration'].value_counts()
```

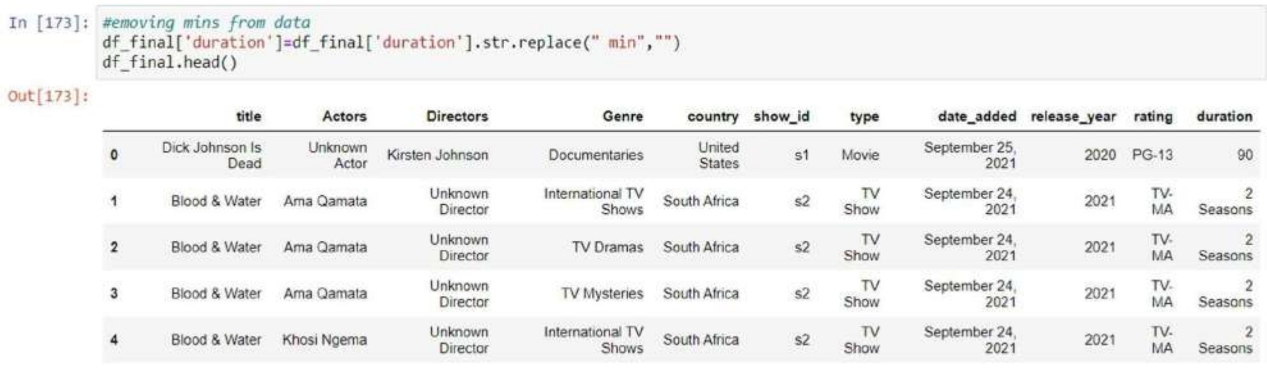
Out[128]:
```
94 min      4247
1 Season    4121
106 min     3896
95 min      3438
97 min      3416
93 min      3397
96 min      3322
90 min      3161
105 min     3079
101 min     2898
98 min      2887
107 min     2884
99 min      2856
103 min     2838
102 min     2832
104 min     2772
91 min      2753
92 min      2728
88 min      2613
```



```
92 min      2728
88 min      2613
112 min     2562
100 min     2562
110 min     2520
108 min     2488
85 min      2378
89 min      2329
86 min      2147
119 min     2063
118 min     2039
116 min     2038
109 min     2010
113 min     1910
87 min      1887
120 min     1730
117 min     1724
121 min     1663
124 min     1564
111 min     1516
```

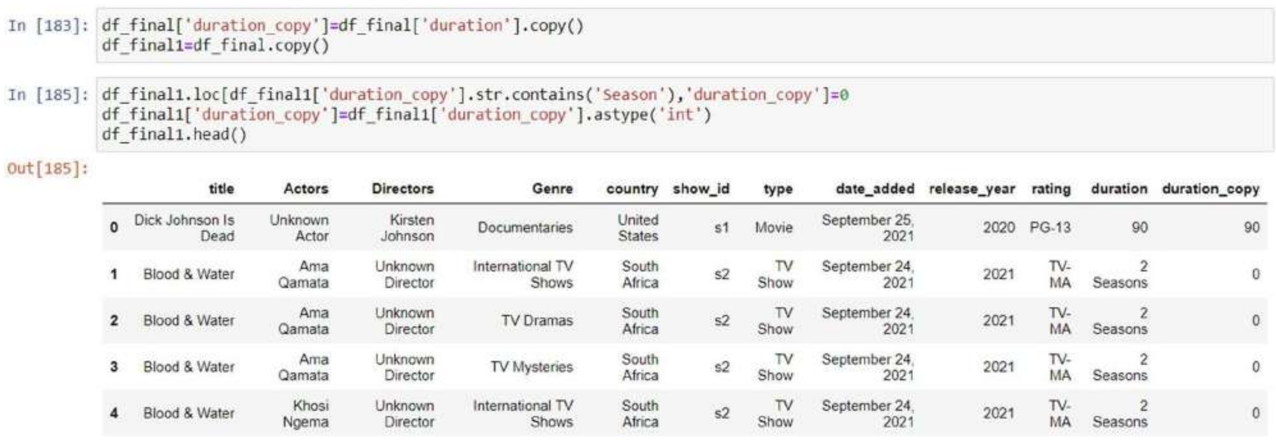


```
125 min     1268
128 min     1241
130 min     1216
122 min     1194
83 min      1165
126 min     1155
81 min      1135
84 min      1132
137 min     1086
136 min     1063
133 min     1058
132 min     1029
82 min       990
131 min      883
129 min      837
135 min      790
75 min       756
148 min      671
79 min       611
142 min      608
```



```
In [173]: #emoving mins from data
          df_final['duration']=df_final['duration'].str.replace(" min","")
          df_final.head()
Out[173]:
```

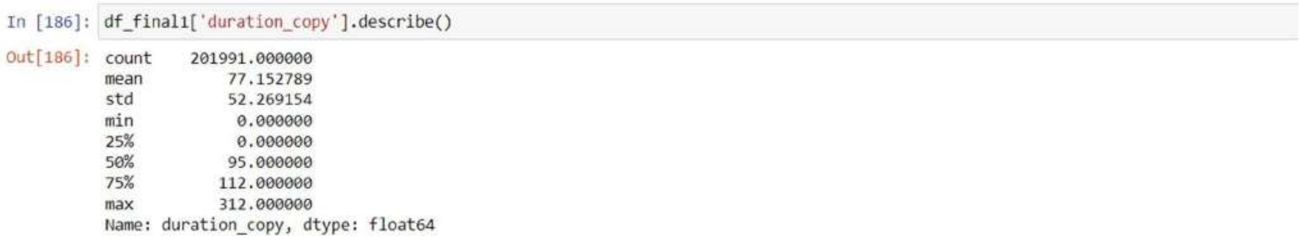
	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating	duration
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show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



```
In [183]: df_final['duration_copy']=df_final['duration'].copy()
          df_final1=df_final.copy()

In [185]: df_final1.loc[df_final1['duration_copy'].str.contains('Season'),'duration_copy']=0
          df_final1['duration_copy']=df_final1['duration_copy'].astype('int')
          df_final1.head()
Out[185]:
```

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



```
In [186]: df_final1['duration_copy'].describe()

Out[186]: count    201991.000000
          mean         77.152789
          std          52.269154
          min           0.000000
          25%           0.000000
          50%          95.000000
          75%         112.000000
          max         312.000000
          Name: duration_copy, dtype: float64
```

3. Insights based on Non-Graphical and Visual Analysis:

Insights on the range of attributes, Insights on the distribution of the variables and the relationship between them, and Insights for each univariate and bivariate plot:

- The most popular films are international movies, dramas, and comedies.
- The countries at the forefront of content creation on Netflix include the USA, India, UK, Canada, and France.
- The majority of top-rated content on Netflix is designed for mature viewers, with R-rated content intended for audiences aged 14 and above, as well as those who may need parental guidance.
- The most viewed content in our dataset typically ranges from 80 to 100 minutes in duration. This timeframe is primarily associated with movies and shows that consist of a single season.
- Our content distribution maintains a ratio of 70% movies to 30% TV shows..
- Anupam Kher, SRK (Shah Rukh Khan), Julie Tejwani, Naseeruddin Shah, and Takahiro Sakurai hold the highest positions in the list of Most Watched content.
- The number of original content releases that subsequently became available on Netflix saw an increase from 1980 to 2020, although this trend later experienced a decline, likely attributed to the impact of COVID-19.
- TV shows on Netflix featuring international content, dramas, and comedy genres enjoy widespread popularity.
- The United States holds a prominent position in both TV shows and movies, with the UK also delivering impressive content in both categories. Interestingly, India has a stronger presence in movies compared to TV shows.
- Furthermore, the volume of movies produced in India exceeds the combined count of TV shows and movies from the UK, as India secured the second position in the overall content sum on Netflix.
- Hence, it is reasonable to deduce that the popular ratings on Netflix encompass content suitable for mature audiences, including those aged over 14 or over 17.
- In the realm of movies, the most common durations fall within the ranges of 80-100, 100-120, and 120-150 minutes. This suggests that the optimal range for movie lengths could indeed be around 80 to 150 minutes.

- Bollywood actors like Anupam Kher, SRK (Shah Rukh Khan), and Naseeruddin Shah hold significant popularity within the realm of movies available on Netflix.
- Rajiv Chilka, Jan Suter, Raul Campos, and Suhas Kadav are esteemed directors who have garnered popularity in the world of movies.
- Until 2019, the content library on Netflix experienced a consistent growth trajectory. However, the emergence of the Covid-19 pandemic in 2020 impacted the movie category more significantly than TV shows. Subsequently, in 2021, there was a notable reduction in content across both TV shows and movies.
- A substantial influx of movies is observed on Netflix during the initial week and final month of the current year, as well as the opening month of the subsequent year.

- Dramas, Comedy, Kids 'TV Shows, International TV Shows, and Docuseries, Genres are popular in TV Series in the USA
- Dramas, Comedies, Documentaries, Family Movies, and Action Genres in Movies are popular in the USA.
- So, it seems plausible to conclude that the popular ratings across Netflix include Mature Audiences and those appropriate for over 14/over 17 ages in both Movies and TV Shows in the USA.
- Across movies, 80-100,100-120 is the range of minutes for which most movies lie. So quite possibly 80-120 minutes is the sweet spot we would be wanting for movies in the USA.
- Across movies, 80-100,100-120 is the range of minutes for which most movies lie. So quite possibly 80-120 minutes is the sweet spot we would be wanting for movies in the USA.
- Vincent Tong, Grey Griffin, and Kevin Richardson are the most popular actors across TV Shows in the USA
- TV Shows are added to Netflix by a tremendous amount in July and September in the USA
- Movies are added to Netflix in the USA by a tremendous amount in the first week/last month of the current year and the first month of next year.
- In the USA, though both Movies and Shows have reduced in 2021, the amount of decrease in the number of TV Shows is small as compared to Movies.

The Most Popular Actor Director Combination in Movies Across the USA are: -

'Smith Foreman and Stanley Moore',
'Marlon Wayans and Michael Tiddes',
'Adam Sandler and Steve Brill', 'Maisie
Benson and Stanley Moore', 'Ashleigh
Ball and Ishi Rudell',
'Tara Strong and Ishi Rudell',
'Rebecca Shoichet and Ishi Rudell',
'Kerry Gudjohnsen and Alex Woo',
'Kerry Gudjohnsen and Stanley Moore',
'Paul Killam and Alex Woo',
'Paul Killam and Stanley Moore',
'Andrea Libman and Ishi Rudell',
'Kevin Hart and Leslie Small',
'Maisie Benson and Alex Woo',
'Alexa PenaVega and Robert Rodriguez', 'Tabitha
St. Germain and Ishi Rudell'

The second Most Popular Actor Director Combination in Movies Across the USA are:

'Rory Markham and Mike Gunther',
'Erin Mathews and Steve Ball',
'Danny Trejo and Robert Rodriguez',
'Jeff Dunham and Michael Simon'

Popular Actors in TV Shows in India are: -

'Rajesh Kava',
'Nishka Raheja',
'Prakash Raj',
'Sabina Malik',
'Anjali',
'Aranya Kaur',
'Sonal Kaushal',
'Chandan Anand',
'Danish Husain'

Popular actors across Movies in India: -

'Anupam Kher',
'Shah Rukh Khan',
'Naseeruddin Shah',
'Akshay Kumar',
'Om Puri',
'Paresh Rawal',
'Julie Tejwani',
'Amitabh Bachchan',
'Boman Irani',
'Rupa Bhimani',
'Kareena Kapoor',
'Ajay Devgn',
'Rajesh Kava', 'Kay
Kay Menon'

Popular Directors Across Movies in India:-

'Gautham Vasudev Menon',
'Abhishek Chaubey',
'Sudha Kongara',
'Rathindran R Prasad',
'Sankalp Reddy',
'Sarjun',
'Soumendra Padhi',
'Srijit Mukherji',
'Tharun Bhascker Dhaassyam'

Popular directors across movies in India:-

'Rajiv Chilaka',
'Suhas Kadav',
'David Dhawan',
'Umesh Mehra',

'Anurag Kashyap',
'Ram Gopal Varma',
'Dibakar Banerjee',
'Zoya Akhtar',
'Tilak Shetty',
'Rajkumar Santoshi',
'Priyadarshan', 'Sooraj
R. Barjatya', 'Ashutosh
Gowariker', 'Milan
Luthria'

The Most Popular Actor Director Combination in Movies Across India are: -

'Rajesh Kava and Rajiv Chilaka',
'Julie Tejwani and Rajiv Chilaka',
'Rupa Bhimani and Rajiv Chilaka',
'Jigna Bhardwaj and Rajiv Chilaka',
'Vatsal Dubey and Rajiv Chilaka',
'Mousam and Rajiv Chilaka',
'Swapnil and Rajiv Chilaka',
'Saurav Chakraborty and Suhas Kadav',
'Smita Malhotra and Tilak Shetty',
'Anupam Kher and David Dhawan',
'Salman Khan and Sooraj R. Barjatya',

4. Business Insights:

1. There is a greater quantity of movies produced compared to TV shows.
2. The United States and India stand out as the top two countries contributing a substantial number of movies and TV shows.
3. India, South Korea, and the UK-USA share similar preferences when it comes to crafting movies.
4. The release of movies and TV shows was impacted by the COVID-19 pandemic.
5. There is a scarcity of child-oriented content in India.

5. Recommendations:

1. It is advisable to focus on content that aligns with the most popular genres across various countries and in both TV shows and movies, namely Drama, Comedy, and International TV Shows/Movies.
2. Consider adding TV shows during the months of July and August, and scheduling movie releases for the last week of the year or the first month of the following year.
3. For the US audience, it is advisable to target movies with a duration of 80-120 minutes. Additionally, Kids' TV Shows have gained popularity, along with the genres mentioned earlier, making them a recommended choice.
4. For the UK audience, it is recommended to adhere to the same movie length range as the USA, which is 80-120 minutes.
5. The intended audience for content in the USA and India is recommended to be 14 years and above, adhering to 14+ ratings. Conversely, for the UK audience, it is advisable to target a mature or R-rated audience.

6. The intended audience for content in the USA and India is recommended to be 14 years and above, adhering to 14+ ratings. Conversely, for the UK audience, it is advisable to target a mature or R-rated audience.The intended audience for content in the USA and India is recommended to be 14 years and above, adhering to 14+ ratings. Conversely, for the UK audience, it is advisable to target a mature or R-rated audience.
7. For the Japanese audience, focusing on the Anime genre is recommended, while for the South Korean audience, the Romantic genre in TV shows holds appeal.
8. When developing content, it's crucial to factor in the popularity of actors and directors within each specific country. Additionally, exploring director-actor combinations that have garnered high recommendations can significantly enhance the appeal of the content.