Netflix_Case_Study

May 30, 2024

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

[ ]:
```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/
  ↪original/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, 70.5MB/s]
```

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

Data Exploration

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

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

df.shape shows that initially DataFrame had 8807 rows and 12 columns.

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      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
```

0.1 1. Find the counts of each categorical variable both using graphical and non-graphical analysis

Here we can observe,the dataframe consists of 12 columns. Each column's data type is object, except for the release_year column. The count of non-null values is provided for each 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
```

The df.isnull().sum() method returns the number of null values in each column. The results are as follows:

1. The director column has 2,634 null values.
2. The cast column has 825 null values.
3. The date_added column has 10 null values.
4. The rating and duration columns have 4 and 3 null values, respectively.

[ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

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

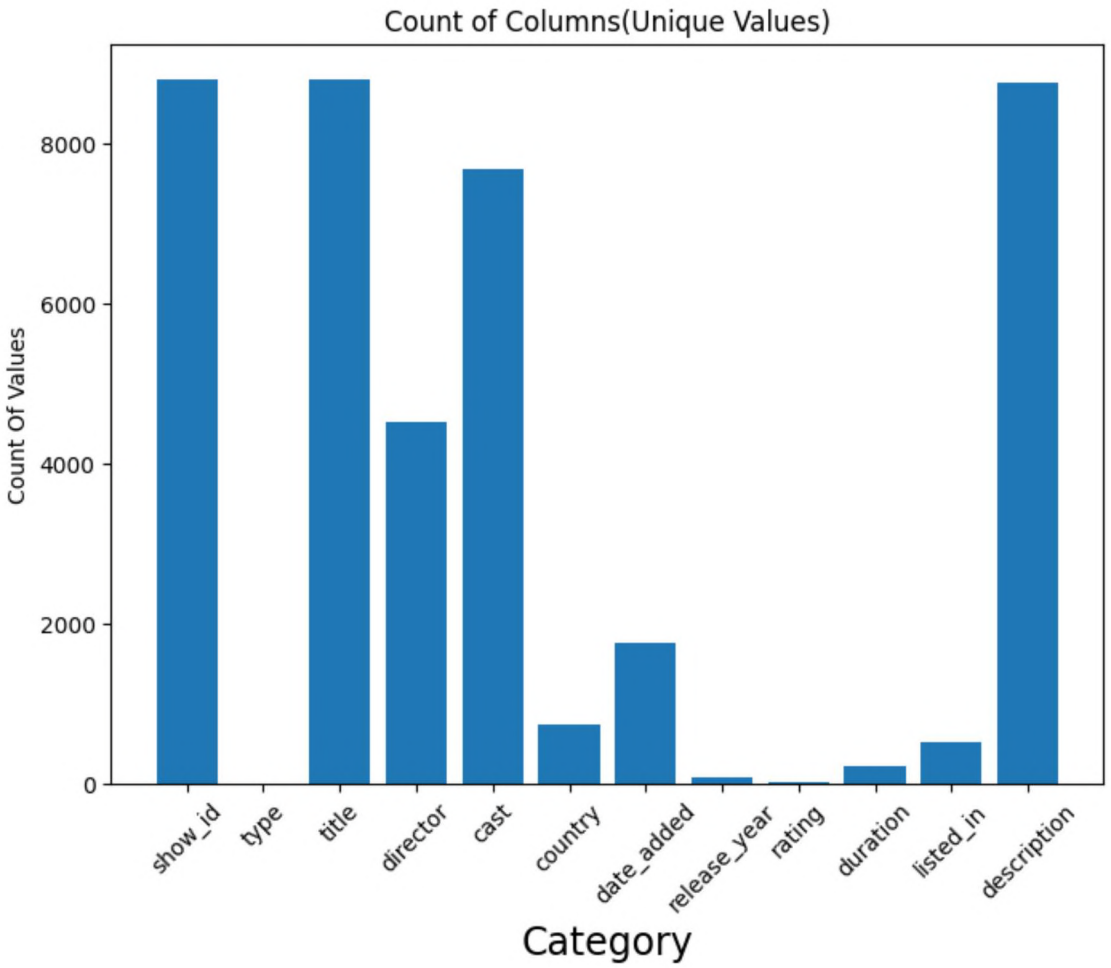
[ ]:
```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1767
```

```
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
```

[ ]:
```
df_dict=dict(df.nunique())
df_dict
```

[ ]:
```
{'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}
```

[ ]:
```
plt.figure(figsize=(8,6))
plt.bar(df_dict.keys(), df_dict.values())
plt.title('Count of Columns(Unique Values)')
plt.xlabel('Category',size=17)
plt.ylabel('Count Of Values',size=10)
plt.xticks(rotation=45)
plt.show()
```



[ ]:
```
df_info=df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      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
```

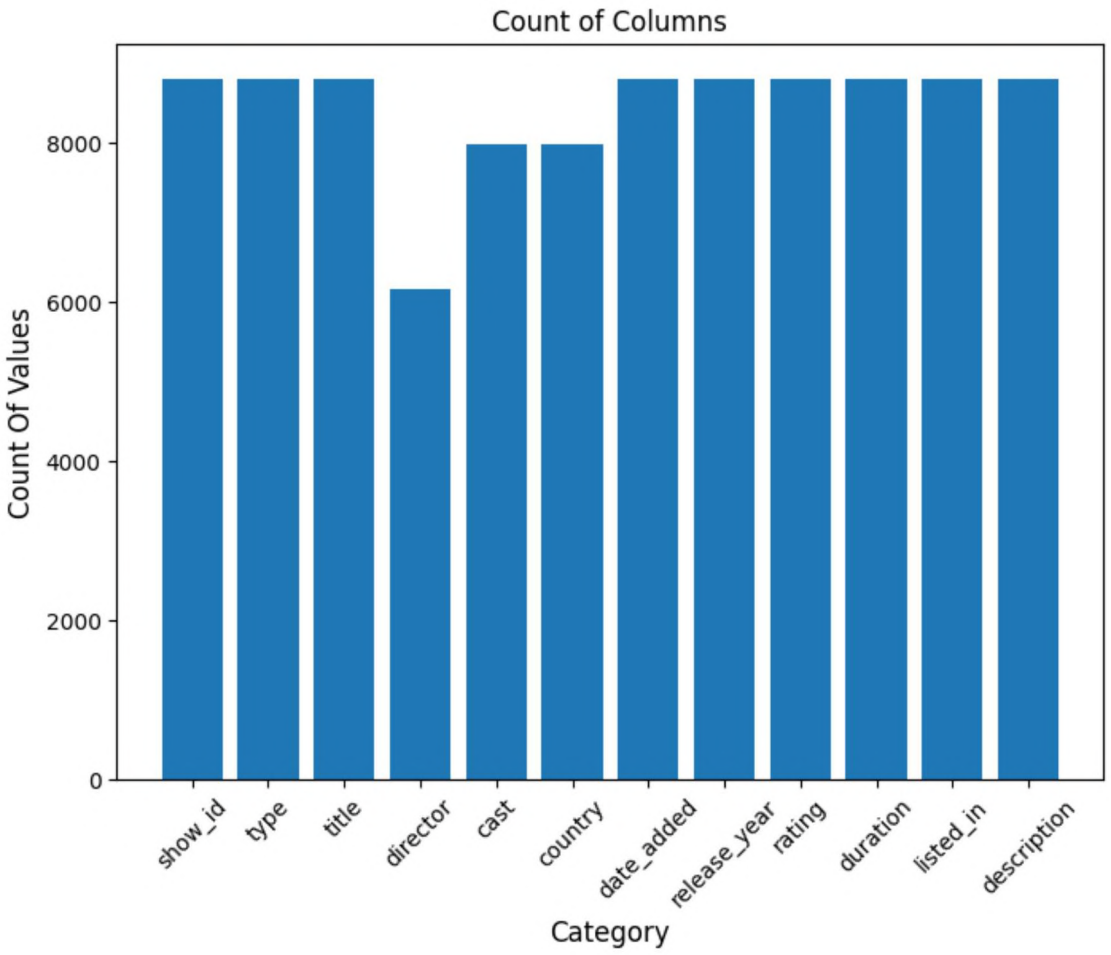
```
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

[ ]:
```
dt= {
    'Column': ['show_id', 'type', 'title', 'director', 'cast', 'country',␣
 ↪'date_added', 'release_year', 'rating', 'duration', 'listed_in',␣
 ↪'description'],
    'Value_count': [8807, 8807, 8807, 6173, 7982, 7976, 8797, 8807, 8803, 8804,␣
 ↪8807, 8807]
}
pd.DataFrame(dt).head(5)
```

[ ]:

	Column	Value_count
0	show_id	8807
1	type	8807
2	title	8807
3	director	6173
4	cast	7982

[ ]:
```
plt.figure(figsize=(8,6))
plt.bar(dt["Column"], dt["Value_count"])
plt.title('Count of Columns')
plt.xlabel('Category',size=12)
plt.ylabel('Count Of Values',size=12)
plt.xticks(rotation=45)
plt.show()
```



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

[ ]:
```
     show_id     type                          title  \
3280   s3281    Movie            The Garden of Words
3777   s3778    Movie                         Silent
5362   s5363  TV Show                     The In-Laws
3186   s3187    Movie  Tiffany Haddish: Black Mitzvah
7103   s7104    Movie                     It Takes Two

                             director  \
3280                     Makoto Shinkai
3777  Limbert Fabian, Brandon Oldenburg
5362                               NaN
3186                     Linda Mendoza
7103                      Andy Tennant

                                                cast        country  \
```

```
3280  Miyu Irino, Kana Hanazawa, Fumi Hirano, Takesh…          Japan
3777                                                    NaN  United States
5362  Louise Lee, Rui En, Pierre Png, Darren Lim, Cy…      Singapore
3186                                    Tiffany Haddish  United States
7103  Kirstie Alley, Steve Guttenberg, Mary-Kate Ols…  United States

            date_added  release_year rating  duration  \
3280  November 15, 2019          2013  TV-14    46 min
3777       June 4, 2019          2014   TV-Y     3 min
5362     August 1, 2017          2011  TV-14  1 Season
3186   December 3, 2019          2019  TV-MA    56 min
7103  November 20, 2019          1995     PG   101 min

                                          listed_in  \
3280  Anime Features, International Movies, Romantic…
3777         Children & Family Movies, Sci-Fi & Fantasy
5362                   International TV Shows, TV Dramas
3186                                 Stand-Up Comedy
7103                Children & Family Movies, Comedies

                                         description
3280  When a lonely teenager skips his morning class…
3777  "Silent" is an animated short film created by …
5362  She swore not to meddle in her kids' lives lik…
3186  On her 40th birthday, Tiffany Haddish drops a …
7103  The Olsen twins put a cute new spin on Mark Tw…
```

By examining the dataframe, we can see that the director, cast, country, and listed_in columns contain multiple entries within a single row

0.2 Data Cleaning

1. Splitting Multiple Entries into Separate Rows:

[ ]:
```
df_director_r=pd.DataFrame(df["director"].apply( lambda x: str(x).split(",")).
 ↪tolist(),index=df["title"])
df_director=df_director_r.stack().reset_index()
df_director.drop("level_1", axis=1, inplace=True)
df_director.rename(columns={0:"director"}, inplace=True)
```

[ ]:
```
df_director.sample(5)
```

[ ]:
```
                                    title      director
5001                    If I were an Animal           nan
8881  Teenage Mutant Ninja Turtles: The Movie  Steve Barron
5919                    Simplemente Manu NNa     Jan Suter
```

```
3211                              37 Seconds        Hikari
5181                        To Each, Her Own  Myriam Aziza
```

2. Unnesting column cast.

[ ]:
```
df_director["director"].nunique()
```

[ ]:
```
5121
```

I'm removing extra spaces,to prevent duplicate values caused by extra spaces in director column.

[ ]:
```
df_director["director"] = df_director["director"].str.strip().str.
↪replace(r'\s+', ' ', regex=True)
```

[ ]:
```
df_director["director"].nunique()
```

[ ]:
```
4994
```

Splitting Multiple Entries into Separate Rows:

[ ]:
```
df_cast_r=pd.DataFrame(df["cast"].apply(lambda x:str(x).split(",")).tolist(),␣
↪index=df["title"])
df_cast=cast=df_cast_r.stack().reset_index()
df_cast.drop("level_1", axis=1, inplace=True)
df_cast.rename(columns={0:"cast"},inplace=True)
df_cast
```

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

[64951 rows x 2 columns]
```

[ ]:
```
df_cast["cast"].nunique()
```

[ ]:
```
39297
```

[ ]:
```
df_cast["cast"].unique()
```

```
array(['nan', 'Ama Qamata', ' Khosi Ngema', …, ' Malkeet Rauni',
       ' Anita Shabdish', ' Chittaranjan Tripathy'], dtype=object)
```

```
df_cast["cast"] = df_cast["cast"].str.strip().str.replace(r'\s+', ' ',␣
↪regex=True)
```

```
df_cast["cast"].nunique()
```

```
36440
```

2. Creating new dataframe new_df by merging datafram director and dataframe cast.

```
new_df=df_director.merge(df_cast, on="title", how="left")
```

```
new_df.head(5)
```

```
                  title         director            cast
0  Dick Johnson Is Dead  Kirsten Johnson             nan
1         Blood & Water              nan      Ama Qamata
2         Blood & Water              nan     Khosi Ngema
3         Blood & Water              nan   Gail Mabalane
4         Blood & Water              nan  Thabang Molaba
```

Finally, merging columns of datafran new_df director and cast into our final DataFrame, df_final.

```
df_final=new_df.merge(df[["title","show_id", "type","country", "date_added",␣
↪"release_year",        "rating","duration",        "listed_in",        "description"]],␣
↪on="title", how="left")
```

```
df_final.sample(5)
```

```
                                                   title        director  \
10818                                       Paper Lives       Can Ulkay
4567                                    Rock the Kasbah  Barry Levinson
54667  GLOW: The Story of the Gorgeous Ladies of Wres…  Brett Whitcomb
28384                      Hey Arnold! The Jungle Movie  Raymie Muzquiz
31520                                             Beats  Chris Robinson

                     cast show_id   type                          country  \
10818      Emir Ali Doğrul   s1213  Movie                           Turkey
4567          Bruce Willis    s495  Movie                    United States
54667                  nan   s6857  Movie                    United States
28384  Benjamin Flores Jr.   s3320  Movie  United States, South Korea, Japan
31520  Emayatzy Corinealdi   s3732  Movie                    United States

            date_added  release_year rating duration  \
10818    March 12, 2021          2021  TV-MA   97 min
4567       July 8, 2021          2015      R  106 min
```

```
54667    March 31, 2017          2012     NR   77 min
28384  November 2, 2019          2017  TV-PG   81 min
31520     June 19, 2019          2019  TV-MA  110 min

                                        listed_in  \
10818                  Dramas, International Movies
4567                     Comedies, Music & Musicals
54667                  Documentaries, Sports Movies
28384            Children & Family Movies, Comedies
31520  Dramas, Independent Movies, Music & Musicals

                                              description
10818  In the streets of Istanbul, ailing waste wareh…
4567   When a has-been music producer gets stuck in A…
54667  This engaging documentary chronicles the 1980s…
28384  When Arnold and his crew win a trip to San Lor…
31520  On Chicago's South Side, hip-hop prodigy Augus…
```

[ ]:
```
df_final.shape
```

[ ]: (70812, 12)

Similary Unnesting rows of column Country and listed_in.

[ ]:
```
df_country_r=pd.DataFrame(df_final["country"].apply(lambda x:str(x).split(",")).
 ↪tolist(), index=df_final["title"])
df_country=df_country_r.stack().reset_index()
df_country.drop("level_1",axis=1,inplace=True)
df_country.rename(columns={0:"country"},inplace=True)
```

[ ]:
```
df_country.nunique()
```

[ ]:
```
title      8807
country     198
dtype: int64
```

[ ]:
```
df_country['country'] = df_country['country'].str.strip().str.replace(r'\s+', '␣
 ↪', regex=True)
```

[ ]:
```
df_country.nunique()
```

[ ]:
```
title      8807
country     124
dtype: int64
```

Merging columns of datafram df_country into datafram df_final

[ ]:
```
df_final=df_country.merge(df_final[["title",        "director",        "cast",␣
↪"show_id",        "type",␣
↪"date_added",        "release_year",        "rating",        "duration",        "listed_in"
```

[ ]:
```
df_final.shape
```

[ ]:
```
(1453377, 12)
```

Deleting duplicate rowf from df_final.

[ ]:
```
df_final[df_final.duplicated()]
```

[ ]:
```
                 title       country     director                  cast  \
20       Blood & Water  South Africa          nan             Ama Qamata
21       Blood & Water  South Africa          nan            Khosi Ngema
22       Blood & Water  South Africa          nan          Gail Mabalane
23       Blood & Water  South Africa          nan         Thabang Molaba
24       Blood & Water  South Africa          nan       Dillon Windvogel
…                  …             …           …                      …
1453372         Zubaan         India  Mozez Singh       Manish Chaudhary
1453373         Zubaan         India  Mozez Singh           Meghna Malik
1453374         Zubaan         India  Mozez Singh          Malkeet Rauni
1453375         Zubaan         India  Mozez Singh         Anita Shabdish
1453376         Zubaan         India  Mozez Singh  Chittaranjan Tripathy

        show_id     type          date_added  release_year rating   duration  \
20           s2  TV Show  September 24, 2021          2021  TV-MA  2 Seasons
21           s2  TV Show  September 24, 2021          2021  TV-MA  2 Seasons
22           s2  TV Show  September 24, 2021          2021  TV-MA  2 Seasons
23           s2  TV Show  September 24, 2021          2021  TV-MA  2 Seasons
24           s2  TV Show  September 24, 2021          2021  TV-MA  2 Seasons
…          …       …                 …           …    …        …
1453372   s8807    Movie       March 2, 2019          2015  TV-14    111 min
1453373   s8807    Movie       March 2, 2019          2015  TV-14    111 min
1453374   s8807    Movie       March 2, 2019          2015  TV-14    111 min
1453375   s8807    Movie       March 2, 2019          2015  TV-14    111 min
1453376   s8807    Movie       March 2, 2019          2015  TV-14    111 min

                                                 listed_in  \
20       International TV Shows, TV Dramas, TV Mysteries
21       International TV Shows, TV Dramas, TV Mysteries
22       International TV Shows, TV Dramas, TV Mysteries
23       International TV Shows, TV Dramas, TV Mysteries
24       International TV Shows, TV Dramas, TV Mysteries
…                                                    …
1453372   Dramas, International Movies, Music & Musicals
1453373   Dramas, International Movies, Music & Musicals
```

```
1453374   Dramas, International Movies, Music & Musicals
1453375   Dramas, International Movies, Music & Musicals
1453376   Dramas, International Movies, Music & Musicals

                                                 description
20       After crossing paths at a party, a Cape Town t…
21       After crossing paths at a party, a Cape Town t…
22       After crossing paths at a party, a Cape Town t…
23       After crossing paths at a party, a Cape Town t…
24       After crossing paths at a party, a Cape Town t…
…                                                      …
1453372  A scrappy but poor boy worms his way into a ty…
1453373  A scrappy but poor boy worms his way into a ty…
1453374  A scrappy but poor boy worms his way into a ty…
1453375  A scrappy but poor boy worms his way into a ty…
1453376  A scrappy but poor boy worms his way into a ty…

[1363982 rows x 12 columns]
```

[ ]:
```
df_final.drop_duplicates(inplace=True)
```

[ ]:
```
df_final.shape
```

[ ]:
```
(89395, 12)
```

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

[ ]:
```
title           8807
country          124
director        4994
cast           36440
show_id         8807
type               2
date_added      1767
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
```

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

[ ]:
```
title           0
country         0
director        0
cast            0
```

```
show_id          0
type             0
date_added      69
release_year     0
rating          38
duration         3
listed_in        0
description      0
dtype: int64
```

Spliting nested rows.

[ ]:
```
df_lst_r=pd.DataFrame(df_final["listed_in"].apply(lambda x:str(x).split(",")).
 ↪tolist(), index=df_final["title"])
df_lst=df_lst_r.stack().reset_index()
df_lst.drop("level_1",axis=1,inplace=True)
df_lst.rename(columns={0:"listed_in"},inplace=True)
df_lst.sample(10)
```

[ ]:
```
                         title                   listed_in
152314            Ex-Boyfriend      International TV Shows
48488                    Cargo                    Comedies
15062            Seven Pounds             Romantic Movies
35808            Tarung Sarung          Action & Adventure
90593          Rosario Tijeras   Spanish-Language TV Shows
92637            Rainbow Jelly                     Dramas
191775  The Invention of Lying             Romantic Movies
178617                Remember                     Dramas
382           Bangkok Breaking       TV Action & Adventure
26388           House of Cards                   TV Dramas
```

Mrging column of df_lst to final_df.

[ ]:
```
df_lst['listed_in'] = df_lst['listed_in'].str.strip().str.replace(r'\s+', ' ',␣
 ↪regex=True)
```

[ ]:
```
df_final=df_lst.
 ↪merge(df_final[["title",        "country","director",        "cast",        "show_id",
 ↪"date_added",        "release_year",        "rating",        "duration",        "descriptio
```

[ ]:
```
df_final.sample()
```

[ ]:
```
                               title                listed_in        country  \
3409529  Kahlil Gibran's The Prophet  Children & Family Movies  United States

            director            cast show_id   type       date_added  \
3409529  Paul Brizzi  John Krasinski   s7165  Movie  October 1, 2017
```

```
         release_year rating duration  \
3409529          2014     PG   85 min

                                                    description
3409529  A troubled young girl and her mother find sola…
```

[ ]:
```
df_final.shape
```

[ ]:
```
(4656562, 12)
```

[ ]:
```
df_final.drop_duplicates(inplace=True)
```

[ ]:
```
df_final.shape
```

[ ]:
```
(202010, 12)
```

After splitting all nested rows, merging them together, and deleting duplicate rows, df_final has 202010 rows and 12 columns.**bold text**

0.3 Handling null values and correcting data types

Date column "date_added" have datatype object, so here converting it to datetime.

[ ]:
```
df_final["date_added"] = pd.to_datetime(df_final["date_added"], errors='coerce')
```

[ ]:
```
df_final.sample()
```

[ ]:
```
                                                    title  \
3022078  Don Quixote: The Ingenious Gentleman of La Mancha

                  listed_in        country     director          cast show_id  \
3022078  Independent Movies  United States  Will Lowell  Luis Guzman   s6625

          type date_added  release_year rating duration  \
3022078  Movie 2018-01-05          2015  TV-14   83 min

                                                    description
3022078  In this modern adaptation of a Spanish classic…
```

[ ]:
```
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 202010 entries, 0 to 4656393
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   title         202010 non-null  object
 1   listed_in     202010 non-null  object
```

```
 2   country       202010 non-null  object
 3   director      202010 non-null  object
 4   cast          202010 non-null  object
 5   show_id       202010 non-null  object
 6   type          202010 non-null  object
 7   date_added    200264 non-null  datetime64[ns]
 8   release_year  202010 non-null  int64
 9   rating        202010 non-null  object
 10  duration      202010 non-null  object
 11  description   202010 non-null  object
dtypes: datetime64[ns](1), int64(1), object(10)
memory usage: 28.1+ MB
```

[ ]:
```
df_final.shape
```

[ ]:
```
(202010, 12)
```

Looking for null values.

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

[ ]:
```
title              0
listed_in          0
country            0
director           0
cast               0
show_id            0
type               0
date_added      1746
release_year       0
rating            67
duration           3
description        0
dtype: int64
```

The DataFrame df_final.isna() indicates that only the columns 'date_added', 'rating', and 'duration' have null values. Let's further confirm this by using value_counts().

[ ]:
```
df_final["country"].value_counts()
```

[ ]:
```
country
United States     59325
India             22814
United Kingdom    12965
nan               11897
Japan              8679
                  …
Panama                2
```

```
Mongolia                2
Kazakhstan              1
Nicaragua               1
Uganda                  1
Name: count, Length: 124, dtype: int64
```

Here,we can see column country have 11897 nan values.

Steps for replacing null values:

1. Replace string 'nan' with NaN.
2. Define a function to fill null values with the country of the director.
3. If a director has multiple countries, take the mode and return it. otherwise, return "Not Available"

[ ]:
```
df_final["country"].replace("nan",np.nan, inplace=True)
```

[ ]:
```
def solve(director_name):
  director_country = df_final[df_final['director'] == director_name]["country"].
 ↪mode()
  if not director_country.empty:
    return director_country.iloc[0]

  return "Not Available"
```

[ ]:
```
df_final["country"]=df_final.apply(lambda x:solve(x["director"]) if pd.
 ↪isna(x["country"]) else x["country"],axis=1)
```

Similarly checking values of columns director and cast.

[ ]:
```
df_final["director"].value_counts()
```

[ ]:
```
director
nan                     50643
Martin Scorsese           419
Youssef Chahine           409
Cathy Garcia-Molina       356
Steven Spielberg          355
                        …
Brendon Marotta             1
Charlie Siskel              1
Adam Bolt                   1
Anthony Palmer              1
Kirsten Johnson             1
Name: count, Length: 4994, dtype: int64
```

[ ]:
```
df_final["cast"].value_counts()
```

[ ]:
```
cast
nan                2149
Liam Neeson         161
Alfred Molina       160
John Krasinski      139
Salma Hayek         130
                   …
Peter Dunning         1
Drew Ray Tanner       1
Rena Strober          1
Valerie Arem          1
Jenelle Evans         1
Name: count, Length: 36440, dtype: int64
```

Replacing nan value of column director and cast.

[ ]:
```
df_final["director"].replace(['nan'],['Unknown director'], inplace=True)
df_final["cast"].replace(['nan'],['Unknow actor'], inplace=True)
```

[ ]:
```
df_final["director"].value_counts()
```

[ ]:
```
director
Unknown director       50643
Martin Scorsese          419
Youssef Chahine          409
Cathy Garcia-Molina      356
Steven Spielberg         355
                        …
Brendon Marotta            1
Charlie Siskel             1
Adam Bolt                  1
Anthony Palmer             1
Kirsten Johnson            1
Name: count, Length: 4994, dtype: int64
```

[ ]:
```
df_final["cast"].value_counts()
```

[ ]:
```
cast
Unknow actor       2149
Liam Neeson         161
Alfred Molina       160
John Krasinski      139
Salma Hayek         130
                   …
Peter Dunning         1
Drew Ray Tanner       1
Rena Strober          1
```

```
Valerie Arem          1
Jenelle Evans         1
Name: count, Length: 36440, dtype: int64
```

Column duration have 3 null values.

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

[ ]:
```
duration
1 Season     35035
2 Seasons     9559
3 Seasons     5084
94 min        4343
106 min       4040
             …
3 min            4
5 min            3
11 min           2
8 min            2
9 min            2
Name: count, Length: 220, dtype: int64
```

[ ]:
```
df_final[df_final["duration"].isna()]
```

[ ]:
```
                                        title listed_in        country  \
2438885                      Louis C.K. 2017    Movies  United States
2527139                 Louis C.K.: Hilarious    Movies  United States
2528281  Louis C.K.: Live at the Comedy Store    Movies  United States

           director        cast show_id   type date_added  release_year  \
2438885  Louis C.K.  Louis C.K.    s5542  Movie 2017-04-04          2017
2527139  Louis C.K.  Louis C.K.    s5795  Movie 2016-09-16          2010
2528281  Louis C.K.  Louis C.K.    s5814  Movie 2016-08-15          2015

         rating duration                                          description
2438885  74 min      NaN  Louis C.K. muses on religion, eternal love, gi…
2527139  84 min      NaN  Emmy-winning comedy writer Louis C.K. brings h…
2528281  66 min      NaN  The comic puts his trademark hilarious/thought…
```

[ ]:
```
df_final.loc[[2438885,2527139,2528281],"duration"]=['74 min','84 min','66 min']
```

[ ]:
```
df_final["duration"].isna().sum()
```

[ ]:
```
0
```

[ ]:
```
df_final.loc[[2438885,2527139,2528281]]
```

[ ]:
```
                                      title listed_in        country  \
2438885                     Louis C.K. 2017    Movies  United States
2527139                 Louis C.K.: Hilarious    Movies  United States
2528281  Louis C.K.: Live at the Comedy Store    Movies  United States

           director        cast show_id   type date_added  release_year  \
2438885  Louis C.K.  Louis C.K.    s5542  Movie 2017-04-04          2017
2527139  Louis C.K.  Louis C.K.    s5795  Movie 2016-09-16          2010
2528281  Louis C.K.  Louis C.K.    s5814  Movie 2016-08-15          2015

         rating duration                                          description
2438885  74 min   74 min  Louis C.K. muses on religion, eternal love, gi…
2527139  84 min   84 min  Emmy-winning comedy writer Louis C.K. brings h…
2528281  66 min   66 min  The comic puts his trademark hilarious/thought…
```

We can observe that the "rating" column has incorrect values at indices [2438885, 2527139, 2528281]. Hence, I will remove those values.

[ ]:
```
df_final.loc[[2438885,2527139,2528281],"rating"]=np.nan
```

[ ]:
```
df_final["rating"].isna().sum()
```

[ ]:
```
70
```

[ ]:
```
df_final["rating"].fillna('Not Available', inplace=True)
```

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

[ ]:
```
rating
TV-MA            73867
TV-14            43951
R                25859
PG-13            16246
TV-PG            14926
PG               10919
TV-Y7             6304
TV-Y              3665
TV-G              2779
NR                1573
G                 1530
NC-17              149
TV-Y7-FV            86
UR                  86
Not Available       70
Name: count, dtype: int64
```

[ ]:
```
df_final.sample(10)
```

[ ]:

	title	listed_in	country	director	cast	show_id	type	date_added	release_year	rating	duration	description
157150	'76	International Movies	Nigeria	Izu Ojukwu	Daniel K. Daniel	s307	Movie	2021-08-04	2016	TV-PG	118 min	When her husband is accused of taking part in …
920876	Dad Wanted	Children & Family Movies	Mexico	Javier Colinas	Omar Fierro	s2008	Movie	2020-09-11	2020	TV-14	103 min	What does a thrill-seeker tween girl do when h…
1108186	Mirai	Action & Adventure	United States	Mamoru Hosoda	Haru Kuroki	s2452	Movie	2020-06-01	2018	PG	98 min	Unhappy after his new baby sister displaces hi…
2808955	Bala Loca	Spanish-Language TV Shows	Chile	Unknown director	Daniel Candia	s6220	TV Show	2017-04-01	2016	TV-MA	1 Season	A veteran journalist starting a news site abou…
2074673	Ingobernable	International TV Shows	Mexico	Unknown director	Kate del Castillo	s4639	TV Show	2018-09-14	2018	TV-MA	2 Seasons	The first lady of Mexico is a woman of convict…
2263497	Fullmetal Alchemist: Brotherhood	International TV Shows	Japan	Yasuhiro Irie	Yuji Ueda	s5097	TV Show	2018-01-01	2010	TV-14	5 Seasons	After both suffer physical damage - brothers E…
418438	Ragnarok	International TV Shows	Denmark	Unknown director	Jonas Strand Gravli	s838	TV Show	2021-05-27	2021	TV-MA	2 Seasons	In a Norwegian town poisoned by pollution and …
2433404	Mar de Plástico	Spanish-Language TV Shows	Spain	Unknown director	Pedro Casablanc	s5515	TV Show	2017-04-27	2017	TV-MA	2 Seasons	In a town in southern Spain where racial tensi…
284862	Austin Powers: International Man of Mystery	Comedies	United States	Jay Roach	Mimi Rogers	s563	Movie	2021-07-01	1997	PG-13	90 min	A swingin' fashion photographer by day and a g…
1924521	Animas	Horror Movies	Belgium	José Ortuño	Chacha Huang	s4166	Movie	2019-01-25	2018	TV-MA	88 min	A teen's eerie visions become increasingly fre…

Column “date_added have 158 null values. Replacing ithe null values with”Not Available”.

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

[ ]:
```
title              0
listed_in          0
country            0
director           0
cast               0
show_id            0
type               0
date_added      1746
release_year       0
rating             0
duration           0
description        0
dtype: int64
```

[ ]:
```
df_final["date_added"].fillna('Not Available', inplace=True)
```

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

[ ]:
```
title           0
listed_in       0
country         0
director        0
cast            0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
description     0
dtype: int64
```

Now, I have the DataFrame as desired for further analysis.

0.4 Question 2: Comparison of tv shows vs. movies.

[ ]:
```
df_final.sample()
```

[ ]:
```
                  title                 listed_in country      director  \
1067440  A Whisker Away  Children & Family Movies   Japan  Junichi Sato

                  cast show_id   type           date_added  release_year  \
1067440  Sayaka Ohara   s2365  Movie  2020-06-18 00:00:00          2020
```

```
        rating duration                                          description
1067440  TV-PG  104 min  A peculiar girl transforms into a cat to catch…
```

```
[ ]: df_movies=df_final[df_final["type"]=="Movie"]
     df_movies.head(5)
```

```
[ ]:                                 title                listed_in  \
     0                  Dick Johnson Is Dead             Documentaries
     2289  My Little Pony: A New Generation  Children & Family Movies
     2290  My Little Pony: A New Generation  Children & Family Movies
     2291  My Little Pony: A New Generation  Children & Family Movies
     2292  My Little Pony: A New Generation  Children & Family Movies

                country         director             cast show_id   type  \
     0        United States  Kirsten Johnson     Unknow actor      s1  Movie
     2289  Not Available    Robert Cullen  Vanessa Hudgens      s7  Movie
     2290  Not Available    Robert Cullen     Kimiko Glenn      s7  Movie
     2291  Not Available    Robert Cullen    James Marsden      s7  Movie
     2292  Not Available    Robert Cullen     Sofia Carson      s7  Movie

                   date_added  release_year rating duration  \
     0     2021-09-25 00:00:00          2020  PG-13   90 min
     2289  2021-09-24 00:00:00          2021     PG   91 min
     2290  2021-09-24 00:00:00          2021     PG   91 min
     2291  2021-09-24 00:00:00          2021     PG   91 min
     2292  2021-09-24 00:00:00          2021     PG   91 min

                                              description
     0     As her father nears the end of his life, filmm…
     2289  Equestria's divided. But a bright-eyed hero be…
     2290  Equestria's divided. But a bright-eyed hero be…
     2291  Equestria's divided. But a bright-eyed hero be…
     2292  Equestria's divided. But a bright-eyed hero be…
```

Top 10 Countries, who produce most movies.

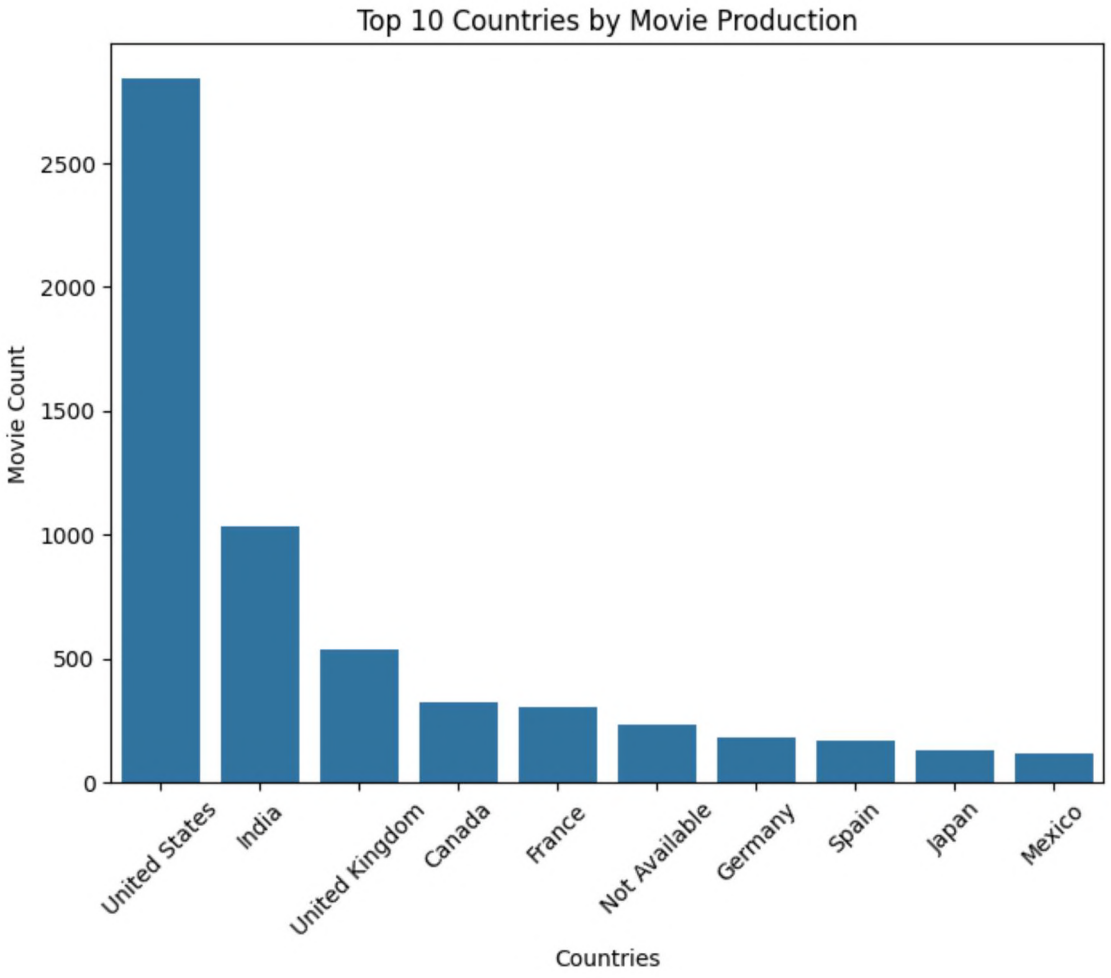
```
[ ]: top_10_contry_movies=df_movies.groupby("country")["title"].nunique().
     ↪sort_values(ascending=False).reset_index()[:10]
```

```
[ ]: top_10_contry_movies.rename(columns={"title":"movie_count"}, inplace=True)
     top_10_contry_movies
```

```
[ ]:           country  movie_count
     0   United States         2840
     1           India         1031
     2  United Kingdom          538
     3          Canada          322
```

```
4          France          305
5   Not Available          230
6         Germany          183
7           Spain          171
8           Japan          127
9          Mexico          115
```

```
[ ]: fig = plt.figure(figsize=(8,6))

sns.barplot(data=top_10_contry_movies, x="country", y="movie_count")
plt.xlabel("Countries")
plt.ylabel("Movie Count")
plt.title("Top 10 Countries by Movie Production")
plt.xticks(rotation=45)
plt.show()
```



Obervation: * The graph illustrates that the United States produces the most movies, with 2840

films, followed by India with 1031, and the United Kingdom with 538 movies.

- It can be observed that India and the United Kingdom combined produce approximately half the number of movies as the United States.
- It can also be observed that india is the olny Asian country in this Top 10 list.

Recommendations:

- United States tops the list because English-language movies are watched around the world and have a large audience.
- India is also one of the biggest entertainment markets. If Indian movies can be dubbed and produced in different languages, they can reach a larger audience within the country and also internationally.

[ ]:
```
df_tv=df_final[df_final["type"]=="TV Show"]
df_tv.head(5)
```

[ ]:
```
           title               listed_in       country          director  \
1  Blood & Water  International TV Shows  South Africa  Unknown director
2  Blood & Water  International TV Shows  South Africa  Unknown director
3  Blood & Water  International TV Shows  South Africa  Unknown director
4  Blood & Water  International TV Shows  South Africa  Unknown director
5  Blood & Water  International TV Shows  South Africa  Unknown director

                cast show_id     type           date_added  release_year  \
1        Ama Qamata      s2  TV Show  2021-09-24 00:00:00          2021
2       Khosi Ngema      s2  TV Show  2021-09-24 00:00:00          2021
3     Gail Mabalane      s2  TV Show  2021-09-24 00:00:00          2021
4    Thabang Molaba      s2  TV Show  2021-09-24 00:00:00          2021
5  Dillon Windvogel      s2  TV Show  2021-09-24 00:00:00          2021

  rating   duration                                          description
1  TV-MA  2 Seasons  After crossing paths at a party, a Cape Town t…
2  TV-MA  2 Seasons  After crossing paths at a party, a Cape Town t…
3  TV-MA  2 Seasons  After crossing paths at a party, a Cape Town t…
4  TV-MA  2 Seasons  After crossing paths at a party, a Cape Town t…
5  TV-MA  2 Seasons  After crossing paths at a party, a Cape Town t…
```

Top 10 Countries, who produce most TV Shows.

[ ]:
```
top_10_contry_tv=df_tv.groupby("country")["title"].nunique().
↪sort_values(ascending=False).reset_index()[:10]
```

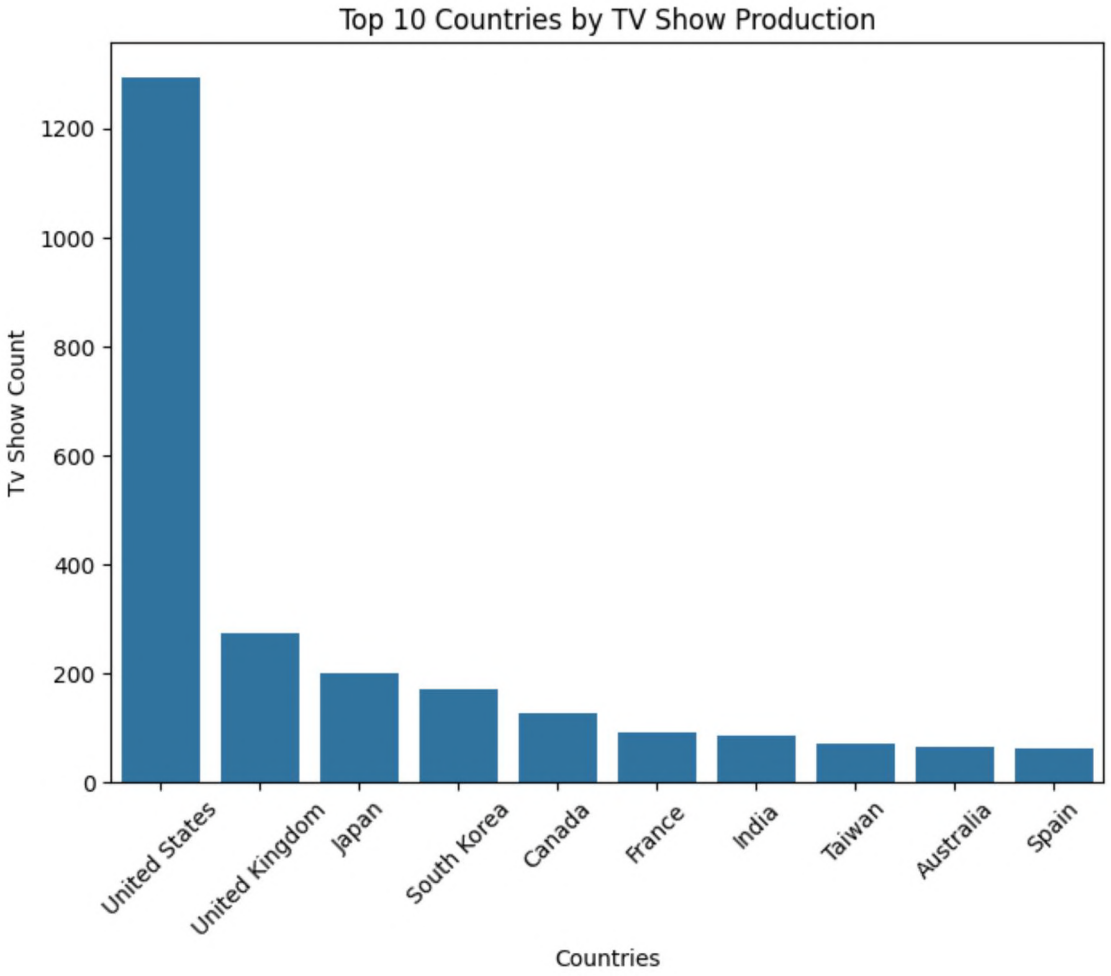
[ ]:
```
top_10_contry_tv.rename(columns={"title":"tv_show_count"}, inplace=True)
top_10_contry_tv
```

[ ]:
```
         country  tv_show_count
0  United States           1293
```

```
1  United Kingdom            273
2           Japan            199
3     South Korea            170
4          Canada            126
5          France             91
6           India             85
7          Taiwan             71
8       Australia             66
9           Spain             63
```

This shows that united states produce most movies.

[ ]:
```
fig = plt.figure(figsize=(8,6))

sns.barplot(data=top_10_contry_tv, x="country", y="tv_show_count")
plt.xlabel("Countries")
plt.ylabel("Tv Show Count")
plt.title("Top 10 Countries by TV Show Production")
plt.xticks(rotation=45)
plt.show()
```



Observation:

- The United States tops the list with a TV show production count of 1293, followed by the United Kingdom with 273 shows and Japan with 199.
- It's noteworthy that the United States produced a significantly greater number of shows than the other top 10 countries.
- In this list, India ranks sixth.
- Both Japan and South Korea produce more TV shows than India.

Recommendations * The United States has the largest TV show production because they cater to a global audience. One of the reasons for this is the high quality of content they produce, which can be consumed by a larger audience. * Indian audiences have historically enjoyed movies more than TV serials, but in recent years, the trend has changed. People are now enjoying good TV shows and series. * If Indian producers can produce high-quality content showcasing the diverse culture and different stories of India, they can not only captivate domestic but also international audiences. In doing so, Indian shows can compete with Japanese and Korean TV shows.

0.5 3. What is the best time to launch a TV show?

[ ]:
```
df_final["month"]=df_final["date_added"].dt.month
df_final["month"] = df_final["month"].fillna(0).astype(int)
```

[ ]:
```
df_final['week_number'] = df_final['date_added'].dt.isocalendar().week
```

[ ]:
```
df_final.sample()
```

[ ]:
```
                          title listed_in country       director       cast  \
2556124  Special Correspondents  Comedies  Canada  Ricky Gervais  Eric Bana

         show_id   type date_added  release_year rating duration  \
2556124   s5849  Movie 2016-04-29          2016  TV-MA  101 min

                                                description  month  week_number
2556124  When they lose their passports, a bickering ra…      4           17
```

Best week for movies.

[ ]:
```
df_movies=df_final[df_final["type"]=="Movie"]
df_movies.sample(5)
```

[ ]:
```
                   title             listed_in    country  \
2483351           Stereo             Thrillers    Germany
437323             Ferry  International Movies    Belgium
2830998          Bewafaa       Romantic Movies      India
2268795  Seoul Searching              Comedies      China
1982290     Break Up 100  International Movies  Hong Kong

                     director                  cast show_id   type  \
2483351  Maximilian Erlenwein       Georg Friedrich   s5635  Movie
437323      Cecilia Verheyden      Maarten Heijmans    s879  Movie
2830998      Dharmesh Darshan          Sushmita Sen   s6290  Movie
2268795            Benson Lee        Rosalina Leigh   s5125  Movie
1982290        Lawrence Cheng  Miriam Chin Wah Yeung   s4335  Movie

         date_added  release_year rating duration  \
2483351 2017-01-15          2014  TV-MA   91 min
437323  2021-05-14          2021  TV-MA  107 min
2830998 2019-07-21          2005  TV-PG  149 min
2268795 2017-12-15          2015  TV-MA  109 min
1982290 2018-12-01          2014  TV-14  105 min

                                                description  month  week_number
2483351  Erik's peaceful rural family life is shaken by…      1            2
437323   Before he built a drug empire, Ferry Bouman re…      5           19
2830998  A young woman faces pressure to marry her brot…      7           29
```

```
2268795  Teens of Korean descent who were born and rais…     12           50
1982290  After their 99th breakup, a career woman and h…     12           48
```

[ ]:
```
best_week=df_movies.groupby("week_number")["title"].nunique().reset_index()
best_week.rename(columns={"title":"movie_count"},inplace=True)
```

[ ]:
```
best_week.sort_values(by="movie_count",ascending=False).head(10)
```

[ ]:
```
    week_number  movie_count
0             1          316
43           44          243
39           40          215
8             9          207
25           26          195
34           35          189
30           31          185
12           13          174
17           18          173
26           27          154
```

Best week for Tv Shows.

[ ]:
```
df_tvShow=df_final[df_final["type"]=="TV Show"]
df_tvShow.sample(5)
```

[ ]:
```
                         title                     listed_in        country  \
4559429        ThirTEEN Terrors                     TV Horror       Thailand
2305723  Beyond Stranger Things  Stand-Up Comedy & Talk Shows  United States
1850735           KO One Return                   TV Comedies  United States
870617           Emily in Paris                   TV Comedies  United States
2324135           Gurren Lagann        International TV Shows          Japan

                director                    cast show_id     type date_added  \
4559429  Unknown director  Narupornkamol Chaisang   s8575  TV Show 2019-03-01
2305723  Unknown director           David Harbour   s5201  TV Show 2017-10-27
1850735  Unknown director             Sylvia Wang   s3883  TV Show 2019-04-30
870617   Unknown director            Bruno Gouery   s1895  TV Show 2020-10-02
2324135  Unknown director       Katsuyuki Konishi   s5299  TV Show 2017-09-01

         release_year rating  duration  \
4559429          2014  TV-14  1 Season
2305723          2017  TV-14  1 Season
1850735          2012  TV-MA  1 Season
870617           2020  TV-MA  1 Season
2324135          2007  TV-14  1 Season

                                                 description  month  week_number
```

```
4559429  A group of teens searches for the dark truth b…      3           9
2305723  Secrets from the "Stranger Things 2" universe …     10          43
1850735  A time quake propels Wang Dadong into the futu…      4          18
870617   After landing her dream job in Paris, Chicago …     10          40
2324135  When a young laborer escapes to the world abov…      9          35
```

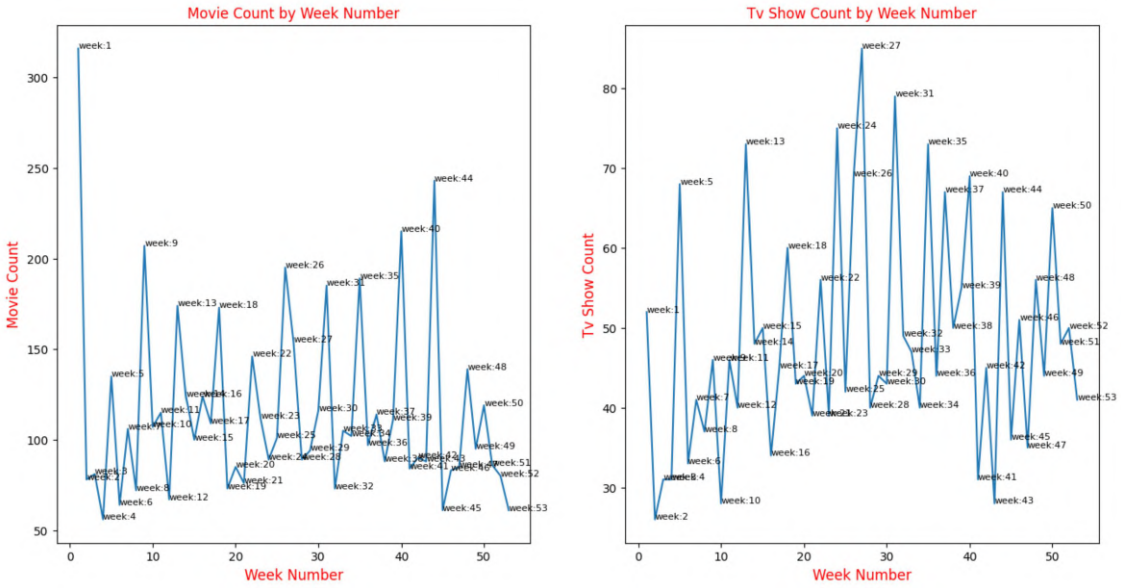
[ ]:
```
best_week_tv=df_tvShow.groupby("week_number")["title"].nunique().reset_index()
best_week_tv.rename(columns=({"title":"tvShow_count"}),inplace=True)
```

[ ]:
```
best_week_tv.sort_values(by="tvShow_count",ascending=False).head(10)
```

[ ]:
```
    week_number  tvShow_count
26           27            85
30           31            79
23           24            75
34           35            73
12           13            73
39           40            69
25           26            69
4             5            68
43           44            67
36           37            67
```

[ ]:
```
plt.figure(figsize=(16,8))

plt.subplot(1, 2, 1)

sns.lineplot(data=best_week, x="week_number", y="movie_count")
plt.xlabel('Week Number', fontsize=12,color="red")
plt.ylabel('Movie Count', fontsize=12,color="red")

xp=best_week["week_number"]
yp=best_week["movie_count"]
for i in range(len(yp)):
  plt.annotate(f"week:{xp[i]}",
               xy=(xp[i],yp[i]),
                horizontalalignment="left",fontsize=8)
plt.title('Movie Count by Week Number', color="red")

plt.subplot(1,2,2)
sns.lineplot(data=best_week_tv, x="week_number", y="tvShow_count")
plt.xlabel('Week Number',fontsize=12, color="red")
plt.ylabel('Tv Show Count',fontsize=12, color="red")
xp=best_week_tv["week_number"]
yp=best_week_tv["tvShow_count"]
for i in range(len(yp)):
  plt.annotate(f"week:{xp[i]}",
```

```
                xy=(xp[i],yp[i]),
                 horizontalalignment="left",fontsize=8)

plt.title('Tv Show Count by Week Number', color="red")
plt.show()
```



Observation:

- The graph indicates that weeks 1,9, 13, 18, 26, 31, 35, 40, and 44 are the optimal weeks to release movies. Considering that this year's weeks coincide with the last weeks of the month, we can conclude that the last week of the month is the best time to release a movie.
- The graph indicates that weeks 27, 31, 24, 35, 13, 40, and 26 are the optimal weeks to release TV shows.
- Week numbers 37, 31, 24, and 35 fall within June, while weeks 35, 40, and 44 correspond to September and October. Therefore, we can conclude that for TV shows, summer and festive times are the best periods for release.

Best month to release the Tv-show or the movie

[ ]:
```
# Best month to release movies
best_month=df_movies.groupby("month")["title"].nunique().reset_index()
best_month.rename(columns=({"title":"movie_count"}),inplace=True)
```

[ ]:
```
best_month.sort_values(by="movie_count",ascending=False)
```

[ ]:

	month	movie_count
6	7	565
3	4	550

```
11     12          547
0       1          546
9      10          545
2       3          529
7       8          519
8       9          519
10     11          498
5       6          492
4       5          439
1       2          382
```

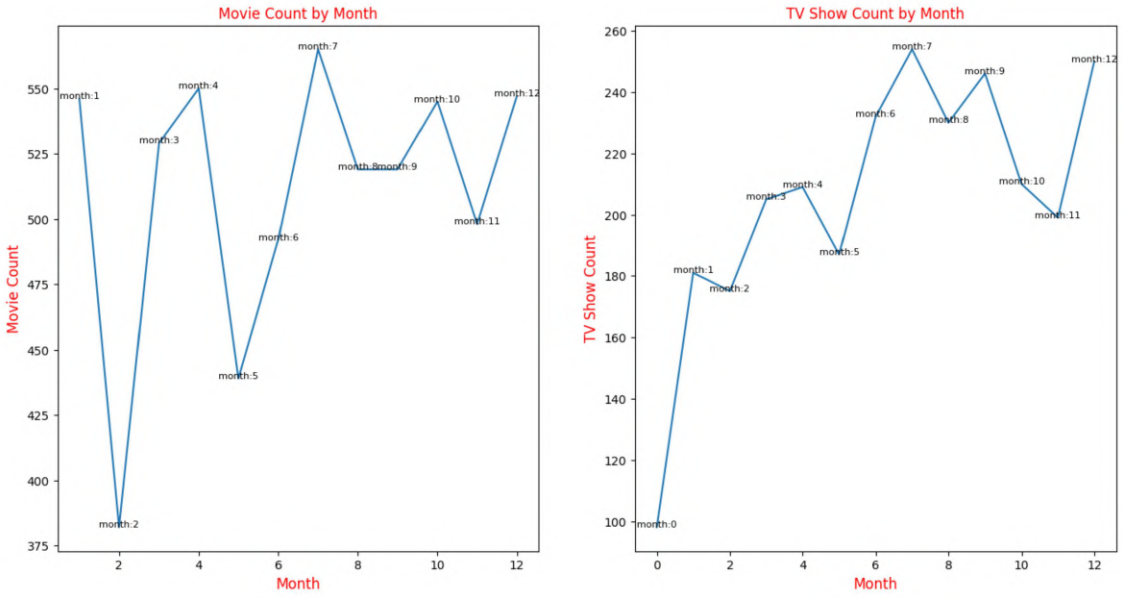
[ ]:
```
# Best month to release Tv Show
best_month_tv=df_tvShow.groupby("month")["title"].nunique().reset_index()
best_month_tv.rename(columns={"title":"tvShow_count"},inplace=True)
```

[ ]:
```
best_month_tv.sort_values(by="tvShow_count",ascending=False)
```

[ ]:
```
    month  tvShow_count
7       7           254
12     12           250
9       9           246
6       6           232
8       8           230
10     10           210
4       4           209
3       3           205
11     11           199
5       5           187
1       1           181
2       2           175
0       0            98
```

[ ]:
```
plt.figure(figsize=(16,8))

plt.subplot(1, 2, 1)

sns.lineplot(data=best_month, x="month", y="movie_count")
plt.xlabel('Month', fontsize=12,color="red")
plt.ylabel('Movie Count', fontsize=12,color="red")

yp=best_month["movie_count"]
for i in range(len(yp)):
  plt.annotate(f"month:{xp[i]}",
               xy=(xp[i],yp[i]),
                horizontalalignment="center",fontsize=8)
plt.title('Movie Count by Month', color="red")
```

```
plt.subplot(1,2,2)
sns.lineplot(data=best_month_tv, x="month", y="tvShow_count")
plt.xlabel('Month',fontsize=12, color="red")
plt.ylabel('TV Show Count',fontsize=12, color="red")
xp=best_month_tv["month"]
yp=best_month_tv["tvShow_count"]
for i in range(len(yp)):
  plt.annotate(f"month:{xp[i]}",
               xy=(xp[i],yp[i]),
                horizontalalignment="center",fontsize=8)

plt.title('TV Show Count by Month', color="red")
plt.show()
```



Observation:

- The graph indicates that, except for the months of February, May, and June, all other months receive a good number of movie releases. Therefore, we can infer that movies can be released throughout the year.
- On the other hand, we observe that months such as June, July, September, and December receive the highest number of TV show releases. This suggests that the best time to release TV shows is during the summer and holiday seasons.

Recommendation: * Movies have a good market throughout the year, so high-quality content can always be released. However, we do observe a higher number of releases during holidays. Therefore, with strategic planning and effective promotion, profits can be increased. * As observed, summer and holidays are the best times to release movies, offering opportunities for increased content releases and profitability. * Additionally, other seasons can be utilized for TV show releases. We can assume that kids and students are big portion of the audience during summer and holidays, so

other seasons can cater to different age groups.

0.6 4. Analysis of actors/directors of different types of shows/movies.

The top 10 directors who have appeared in most movies or TV shows

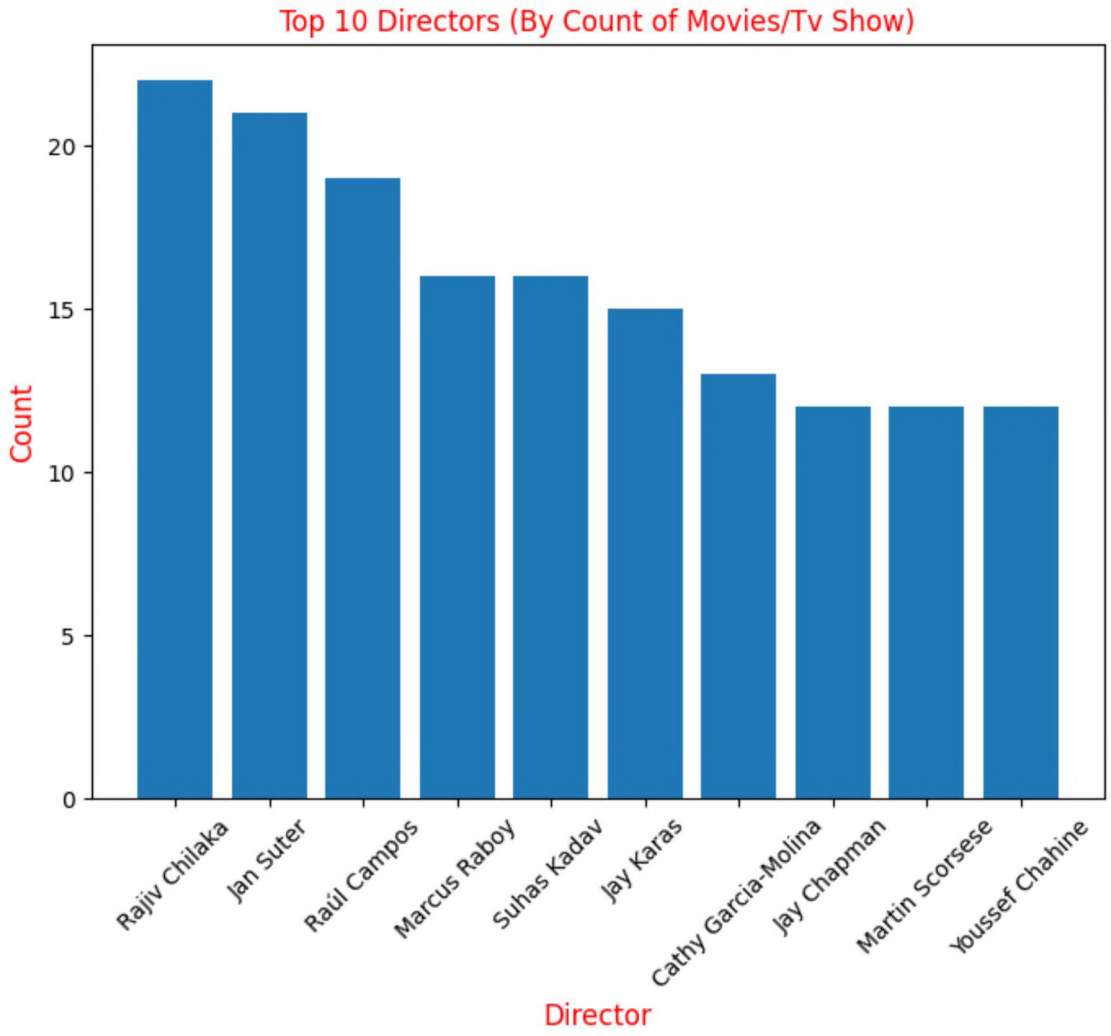
[ ]:
```
df_final.sample()
```

[ ]:
```
                              title listed_in         country   director  \
3343055  Houston, We Have a Problem!  Comedies  Czech Republic  Ziga Virc

                 cast show_id   type date_added  release_year rating duration  \
3343055  Slavoj Zizek   s7012  Movie 2017-07-20          2016  TV-14   84 min

                                                 description  month  week_number
3343055  Blending fact with myth, this conspiracy-minde…      7           29
```

[ ]:
```
top10_director = df_final.groupby("director")["title"].nunique().reset_index()
top10_director.rename(columns={"title": "count"}, inplace=True)
# Ignoring the 2634 rows where the "director" column contains "unknown"␣
 ↪director.
top10_director_sorted = top10_director.sort_values(by="count",␣
 ↪ascending=False)[1:11]
top10_director_sorted
```

[ ]:
```
                director  count
3749        Rajiv Chilaka     22
1906            Jan Suter     21
3800          Raúl Campos     19
2866         Marcus Raboy     16
4457          Suhas Kadav     16
1954            Jay Karas     15
755   Cathy Garcia-Molina     13
1951          Jay Chapman     12
2945      Martin Scorsese     12
4942      Youssef Chahine     12
```

[ ]:
```
plt.figure(figsize=(8, 6))
x=top10_director_sorted["director"]
y=top10_director_sorted["count"]
plt.bar(x,y)
plt.xticks(rotation=45)
plt.xlabel("Director", fontsize=12,color="red")
plt.ylabel("Count", fontsize=12, color="red")
plt.title("Top 10 Directors (By Count of Movies/Tv Show)", color="red")
plt.show()
```



Observation:

- The graph displays the top 10 directors ranked by the number of movies and TV shows they have produced.
- The graph illustrates that Rajiv Chilaka has produced the highest number of movies and shows, followed by Jan Suter and Rahul Campos. Interestingly, these directors have produced a similar number of releases.

Recommendations: * These top directors have the highest number of movies on Netflix. This means that subscribers enjoy the movies by these directors. Netflix can add other projects by these directors to its database, and movies by these directors can also be provided in other languages to reach a larger audience. * Older movies by these directors can also be added, promoted, and recommended to the audience who enjoy movies by these directors

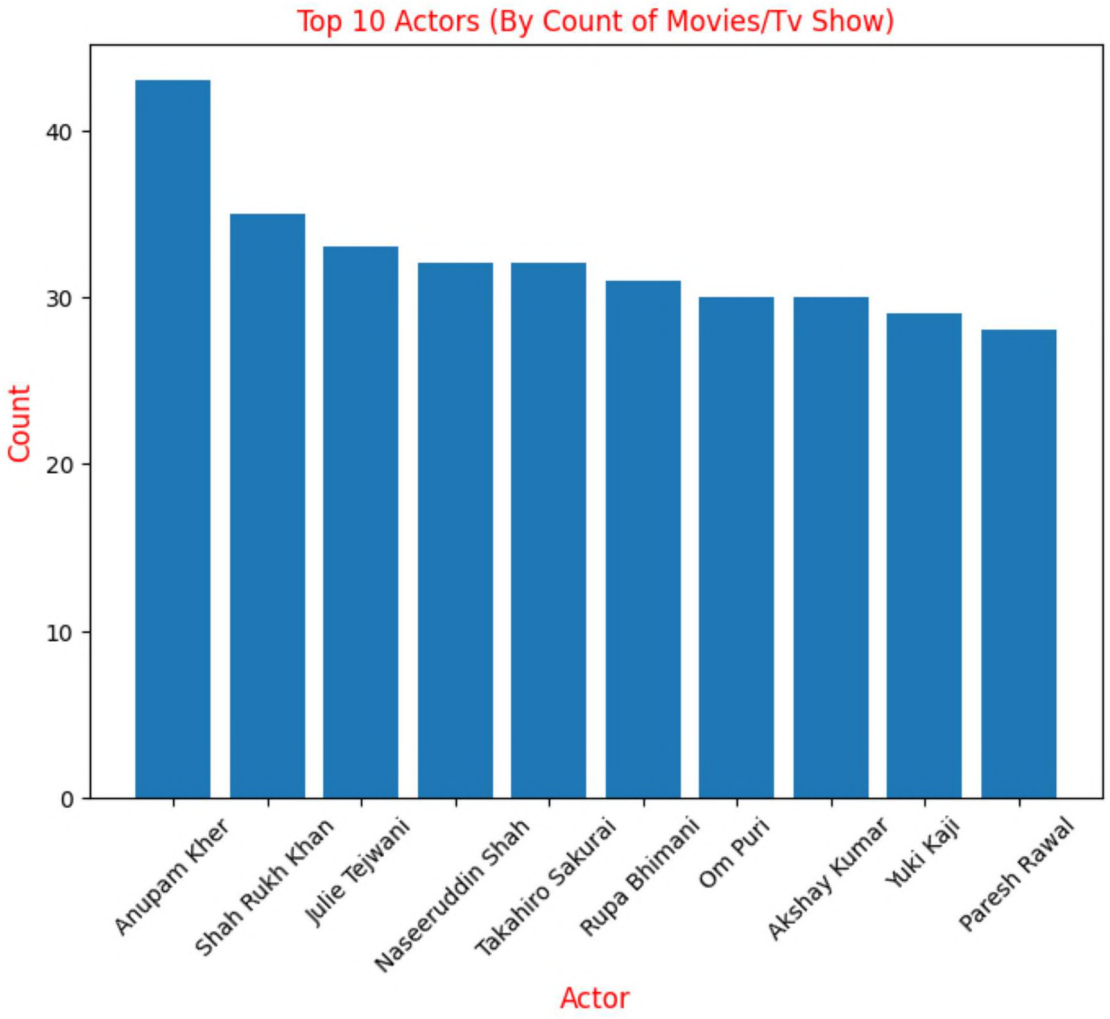
The top 10 actors who have appeared in most movies or TV shows.

[ ]:
```
top10_actor=df_final.groupby("cast")["title"].nunique().reset_index()
top10_actor.rename(columns={"title":"count"},inplace=True)
# Ignoring the 825 rows where the "cast" column contains "unknown" actors.
top10_actor_sorted=top10_actor.sort_values(by="count",ascending=False)[1:11]
top10_actor_sorted
```

[ ]:

	cast	count
2832	Anupam Kher	43
30489	Shah Rukh Khan	35
16697	Julie Tejwani	33
24215	Naseeruddin Shah	32
32591	Takahiro Sakurai	32
28974	Rupa Bhimani	31
25424	Om Puri	30
845	Akshay Kumar	30
35881	Yuki Kaji	29
25782	Paresh Rawal	28

[ ]:
```
plt.figure(figsize=(8, 6))
x=top10_actor_sorted["cast"]
y=top10_actor_sorted["count"]
plt.bar(x,y)
plt.xticks(rotation=45)
plt.xlabel("Actor", fontsize=12,color="red")
plt.ylabel("Count", fontsize=12, color="red")
plt.title("Top 10 Actors (By Count of Movies/Tv Show)", color="red")
plt.show()
```



Observation:

- The graph indicates the top 10 actors based on the number of movies they have on Netflix.
- Anupam Kher leads the list, followed by Shah Rukh Khan and Ujwala Tejwani. It can be assumed that these actors are among the most watched on Netflix.
- We can observe that most of the most watched actors are Indian. Therefore, it's reasonable to assume that India has a large number of Netflix users.

Recommendations : * We know Anupam Kher is a renowned actor worldwide, having appeared in numerous Bollywood and Hollywood movies. This could be the reason he tops the list. * Similarly, Netflix can produce movies with directors and actors who are known worldwide to reach a larger audience. Actors and directors from different industries can create high-quality content that resonates with audiences across multiple nations. * Promoting and recommending movies featuring these top actors can increase viewership on Netflix.

0.7 5. Which genre movies are more popular or produced more

[ ]:
```
from wordcloud import WordCloud
import seaborn as sns
import matplotlib.pyplot as plt
```

[ ]:
```
df_final.sample()
```

[ ]:
```
                        title       listed_in       country          director  \
2267076  The Indian Detective  Crime TV Shows  South Africa  Unknown director

                    cast show_id     type date_added  release_year rating  \
2267076  Christina Cole    s5119  TV Show 2017-12-19          2017  TV-14

         duration                                          description  month  \
2267076  1 Season  In this crime dramedy, a suspended Canadian co…     12

         week_number
2267076           51
```

[ ]:
```
df_genre=df_final.groupby("listed_in")["listed_in"].count()
```

[ ]:
```
df_genre=pd.DataFrame(df_genre)
df_genre.rename(columns={"listed_in":"count"}, inplace=True)
df_genre.reset_index().sort_values(by="count",ascending=False).head(10)
```

[ ]:
```
                   listed_in  count
12                    Dramas  29787
16      International Movies  28224
7                    Comedies  20829
17    International TV Shows  12845
0         Action & Adventure  12216
15        Independent Movies   9818
4   Children & Family Movies   9771
34                 TV Dramas   8942
41                 Thrillers   7106
24           Romantic Movies   6412
```

[ ]:
```
#converting series to text
genre=" ".join(df_final["listed_in"])
```

[ ]:
```
# word cloud
plt.figure(figsize=(10, 8))
wordcloud = WordCloud(width=800, height=400, background_color='Black').
 ↪generate(genre)

plt.imshow(wordcloud, interpolation='bilinear')
```

```
plt.axis('off')
plt.title('Popular Genre', fontsize=14, color="red")
plt.show()
```

Observation

- The graph indicates that dramas top the list, followed by international movies and comedies.
- Additionally, international TV shows, action & adventure, independent movies, and children & family movies are among the other top contents on Netflix.

Recommendations:

- As we can observe, there are numerous different genres that are liked by various kinds of audiences. Netflix can increase the content of these top categories, which are loved by the audience.
- Drama is the most-watched genre around the world, so Netflix can increase its movie content by adding more movies of a similar genre. Older drama movies can be added to the database, as well as drama movies of foreign languages.
- Data shows that audiences enjoy international content, whether it be movies or TV shows. The more of this content that is available, the more audiences consume it. Therefore, adding top foreign movies and TV shows can increase profits.

0.8 6. Find After how many days the movie will be added to Netflix after the release of the movie bold text

[ ]:
```
# Defining a function to find out the most recent date a movie was added to␣
 ↪Netflix.
def mode_date(x):
  dates=x.mode()
  if not dates.empty:
    return dates.iloc[0]
  return None
```

[ ]:
```
mode_dates = df_final.groupby("title")["date_added"].apply(mode_date)
```

[ ]:
```
recent_date=mode_dates.reset_index()
recent_date.rename(columns={"date_added":"recent_added_date"},inplace=True)
recent_date["recent_added_date"]=pd.
 ↪to_datetime(recent_date["recent_added_date"])
```

[ ]:
```
# adding colum recent date to original dataframe.
df_final=recent_date.merge(df_final, on="title", how="inner")
```

[ ]:
```
df_final.sample()
```

[ ]:
```
         title recent_added_date       listed_in country      director  \
58713  GANTZ:0        2021-04-15  Anime Features   Japan  Keiichi Sato

                    cast show_id   type date_added  release_year rating  \
58713  Masane Tsukayama   s1042  Movie 2021-04-15          2016  TV-MA

      duration                                          description  month  \
58713   96 min  Teams of recently deceased people who've been …      4

       week_number
58713           15
```

[ ]:
```
df_final["recent_added_year"]=df_final["recent_added_date"].dt.year
```

[ ]:
```
df_final["time_diff"]=df_final["recent_added_year"]-df_final["release_year"]
```

[ ]:
```
df_final.sample(5)
```

[ ]:
```
                                  title recent_added_date  \
195347                     West Beirut        2020-10-19
143604             Sniper: Ghost Shooter        2021-04-01
155515                       The Borgias        2014-02-01
119359                 Pagpag: Nine Lives        2020-10-29
50592   Elf Pets: Santa's Reindeer Rescue        2020-11-01
```

```
                      listed_in        country          director  \
195347      International Movies         France      Ziad Doueiri
143604        Action & Adventure  United States  Don Michael Paul
155515                 TV Dramas        Hungary  Unknown director
119359      International Movies    Philippines     Frasco Mortiz
50592   Children & Family Movies  United States       Chanda Bell

                     cast show_id     type date_added  release_year rating  \
195347     Mohamad Chamas   s1830    Movie 2020-10-19          1999  TV-MA
143604        Enoch Frost   s1134    Movie 2021-04-01          2016      R
155515  Holliday Grainger   s5934  TV Show 2014-02-01          2013  TV-MA
119359      Michelle Vito   s1785    Movie 2020-10-29          2013  TV-14
50592         Manny Mahen   s1753    Movie 2020-11-01          2020   TV-Y

         duration                                          description  month  \
195347    106 min  Three intrepid teens roam the streets of Beiru…     10
143604     99 min  Snipers ordered to protect a gas pipeline from…      4
155515  3 Seasons  Follow the lives of the notorious Borgia famil…      2
119359    106 min  After ignoring superstitions, a group of teena…     10
50592      27 min  Determined to help Santa get ready for his mer…     11

        week_number  recent_added_year  time_diff
195347           43             2020.0       21.0
143604           13             2021.0        5.0
155515            5             2014.0        1.0
119359           44             2020.0        7.0
50592            44             2020.0        0.0
```

[ ]:
```
time_diff=df_final.groupby("time_diff")["time_diff"].count()
time_diff=pd.DataFrame(time_diff)
time_diff=time_diff.rename(columns={"time_diff":"count"}).reset_index()
```

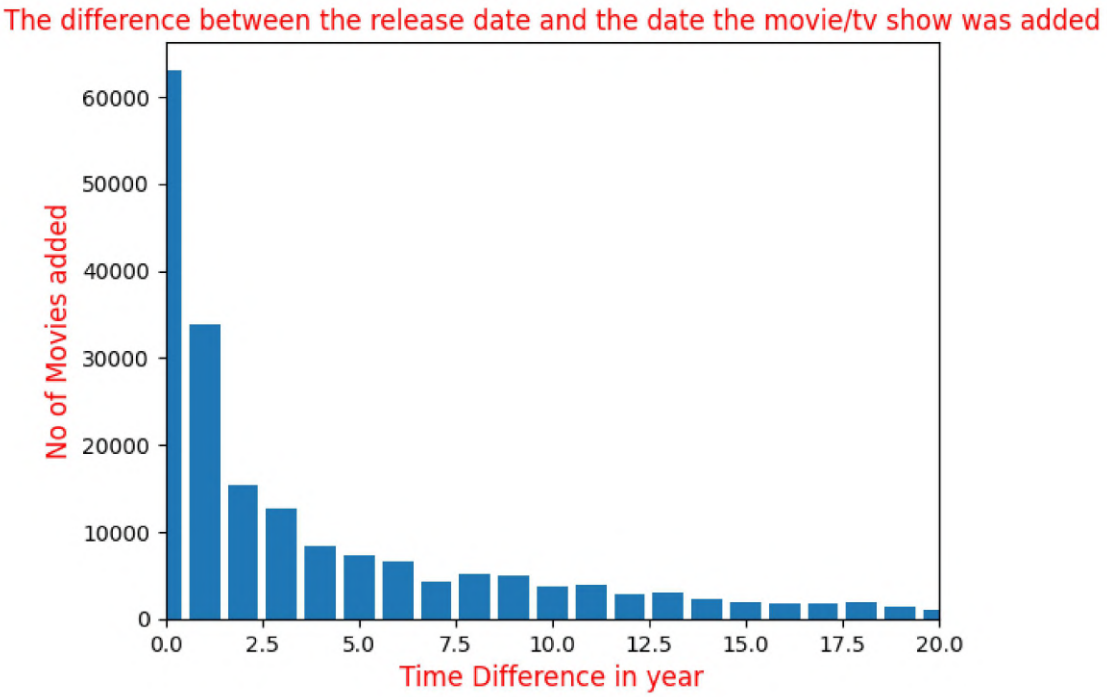
[ ]:
```
time_diff=time_diff.sort_values(by="count", ascending=False)
```

[ ]:
```
time_diff
```

[ ]:
```
    time_diff  count
3         0.0  63086
4         1.0  33802
5         2.0  15409
6         3.0  12744
7         4.0   8399
..        …    …
68       71.0      6
72       75.0      6
71       74.0      5
```

```
73       76.0      1
74       93.0      1

[75 rows x 2 columns]
```

[ ]:
```
x=time_diff["time_diff"]
y=time_diff["count"]
plt.bar(x, y)

plt.xlabel('Time Difference in year',fontsize=12, color="red")
plt.xlim(left=0,right=20)
plt.ylabel('No of Movies added', fontsize=12, color="red")
plt.title('The difference between the release date and the date the movie/tv␣
 ↪show was added', color="red")

plt.show()
```



Observation:

- The bar graph illustrates that Netflix has added a majority of movies and shows within a year.
- The data indicates a decreasing difference between the release date and the date when movies are added to the platform.
- This trend suggests that Netflix is increasingly adding older movies to its collection over time.

Recommendations: * As it can be observed, Netflix is adding more and more older movies to its database. It is recommended to add diverse movies to ensure a varied selection for viewers. This includes adding popular movies from that time period. Regular promotion and recommendation of these older movies are also essential to ensure they reach and engage the audience effectively.

[8]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[9]:
```
!sudo apt-get install texlive-xetex texlive-fonts-recommended␣
↪texlive-plain-generic
```

```
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
texlive-fonts-recommended is already the newest version (2021.20220204-1).
texlive-plain-generic is already the newest version (2021.20220204-1).
texlive-xetex is already the newest version (2021.20220204-1).
0 upgraded, 0 newly installed, 0 to remove and 45 not upgraded.
```

[10]:
```
!jupyter nbconvert --to pdf /content/drive/MyDrive/Colab Notebooks/Another copy␣
↪of Netflix case study.ipynb
```

```
[NbConvertApp] WARNING | pattern '/content/drive/MyDrive/Colab' matched no files
[NbConvertApp] WARNING | pattern 'Notebooks/Another' matched no files
[NbConvertApp] WARNING | pattern 'copy' matched no files
[NbConvertApp] WARNING | pattern 'of' matched no files
[NbConvertApp] WARNING | pattern 'Netflix' matched no files
[NbConvertApp] WARNING | pattern 'case' matched no files
[NbConvertApp] WARNING | pattern 'study.ipynb' matched no files
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
=======
The options below are convenience aliases to configurable class-options,
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
```