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

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

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

```
data=pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv")
data # loaded a data set  ###
```

	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 I...
...	...	...	...	...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...

The dataset contains 8807 rows, 12 descriptons. After a quick view of the data frames, it looks like a typical movie/TVshows data frame . We can also see that there are NaN values in some columns.

```
data.columns
```

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

As wse can see there are 12 columns with different names. Names of the column are 'show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added','release_year', 'rating', 'duration', 'listed_in', 'description'.

```
data.ndim
```

```
2
```

It is a 2 dimentional data frame.

```
data.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
```

```
data.describe()
```

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

```
data.shape
```

```
(8807, 12)
```

The dataset provided to you consists of a list of all the TV shows/movies available on Netflix:

Show_id: Unique ID for every Movie / Tv Show Type: Identifier - A Movie or TV Show Title: Title of the Movie / Tv Show Director: Director of the Movie Cast: Actors involved in the movie/show Country: Country where the movie/show was produced Date_added: Date it was added on Netflix Release_year: Actual Release year of the movie/show Rating: TV Rating of the movie/show Duration: Total Duration - in minutes or number of seasons Listed_in: Genre

```
data=data.fillna("unknown")
data.head(10)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-4efb48bd18c9> in <cell line: 1>()
----> 1 data=data.fillna("unknown",inplace=True)
      2 data.head(10)

NameError: name 'data' is not defined
```

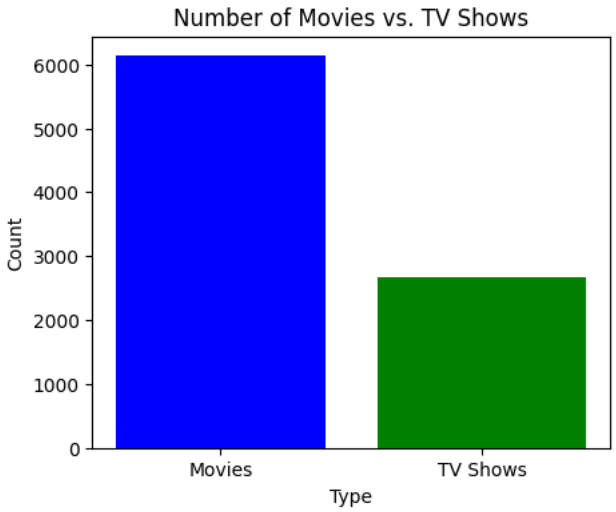
SEARCH STACK OVERFLOW

From the info, we know that there are 8807 entries and 12 columns to work with for this EDA. There are a few columns that contain null values, “director,” “cast,” “country,” “date_added,” “rating.”

```
no_of_movies=data[data.type=="Movie"].shape[0]
no_of_tv_shows=data[data.type=="TV Show"].shape[0]
print(no_of_movies,no_of_tv_shows)
```

```
6131 2676
```

```
plt.figure(figsize=(5, 4))
plt.bar(['Movies', 'TV Shows'], [no_of_movies, no_of_tv_shows], color=['blue', 'green'])
plt.xlabel('Type')
plt.ylabel('Count')
plt.title('Number of Movies vs. TV Shows')
plt.show()
```

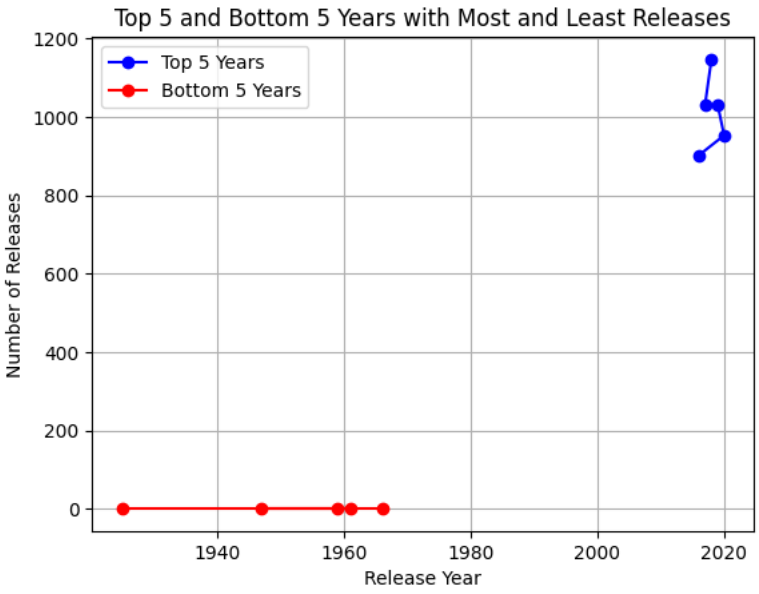


Based on the above anlysis, it appears that there are more movies in a dataset than TV shows. This suggests that the you may have a greater focus on movies compared to TV shows.

```
max_year=data["release_year"].value_counts()
last_5=max_year.sort_values(ascending=False).head(5)
first_5=max_year.sort_values(ascending=True).head(5)
print(last_5,first_5)
```

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

```
plt.plot(last_5.index, last_5.values, marker='o', linestyle='-', color='blue', label='Top 5 Years')
plt.plot(first_5.index, first_5.values, marker='o', linestyle='-', color='red', label='Bottom 5 Years')
plt.xlabel("Release Year")
plt.ylabel("Number of Releases")
plt.title("Top 5 and Bottom 5 Years with Most and Least Releases")
plt.legend()
plt.grid(True)
plt.show()
```



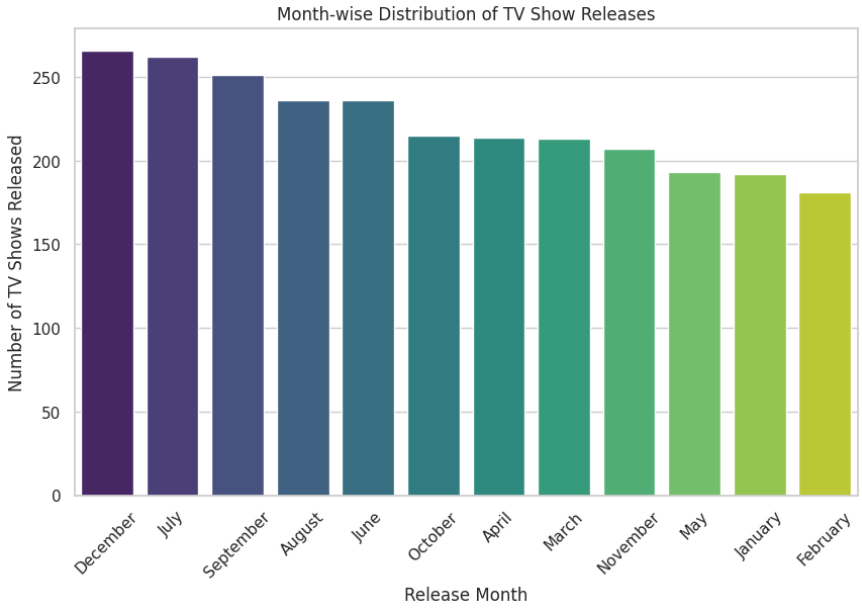
I assume that you have correctly calculated 'last_5' and 'first_5' using the top 5 and bottom 5 years with the most and least releases, respectively. We create a bivariate line plot using Matplotlib. We plot the top 5 years with a blue line and the bottom 5 years with a red line. We displayed the plot with gridlines for better readability. From the above code we can analyse the trend of movies and Tv shows are increasing very drastically as in years like 1940 and 1960, numbers were almost 1 or 2 and now in last few years numbers has been increased and keep on increasing. We will discuss the scenario in detail further.

```
Movies=data[data.type=="Movie"]
Movies["date_added"] = pd.to_datetime(Movies["date_added"])
release_month = Movies["date_added"].dt.month_name()
month_counts = release_month.value_counts()
month_counts
```

```
<ipython-input-27-15b3ea4949a9>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  Movies["date_added"] = pd.to_datetime(Movies["date_added"])
July         565
April        550
December     547
January      546
October      545
March        529
September    519
August       519
November     498
June         492
May          439
February     382
Name: date_added, dtype: int64
```

```
plt.figure(figsize=(10, 6))
sns.set(style="whitegrid")
sns.barplot(x=month_counts.index, y=month_counts.values, order=month_counts.index, palette="viridis")
plt.xlabel("Release Month")
plt.ylabel("Number of TV Shows Released")
plt.title("Month-wise Distribution of TV Show Releases")
plt.xticks(rotation=45)
plt.show()
```



We assume that you have correctly calculated 'month_counts' as the month-wise distribution of TV show releases. as we can see mximum number released in a month of july and decemeber, we recommend to relesae movies in a month of decemeber and july.January and ferbruary and not recommended to release much movie as people are less interested in these months.

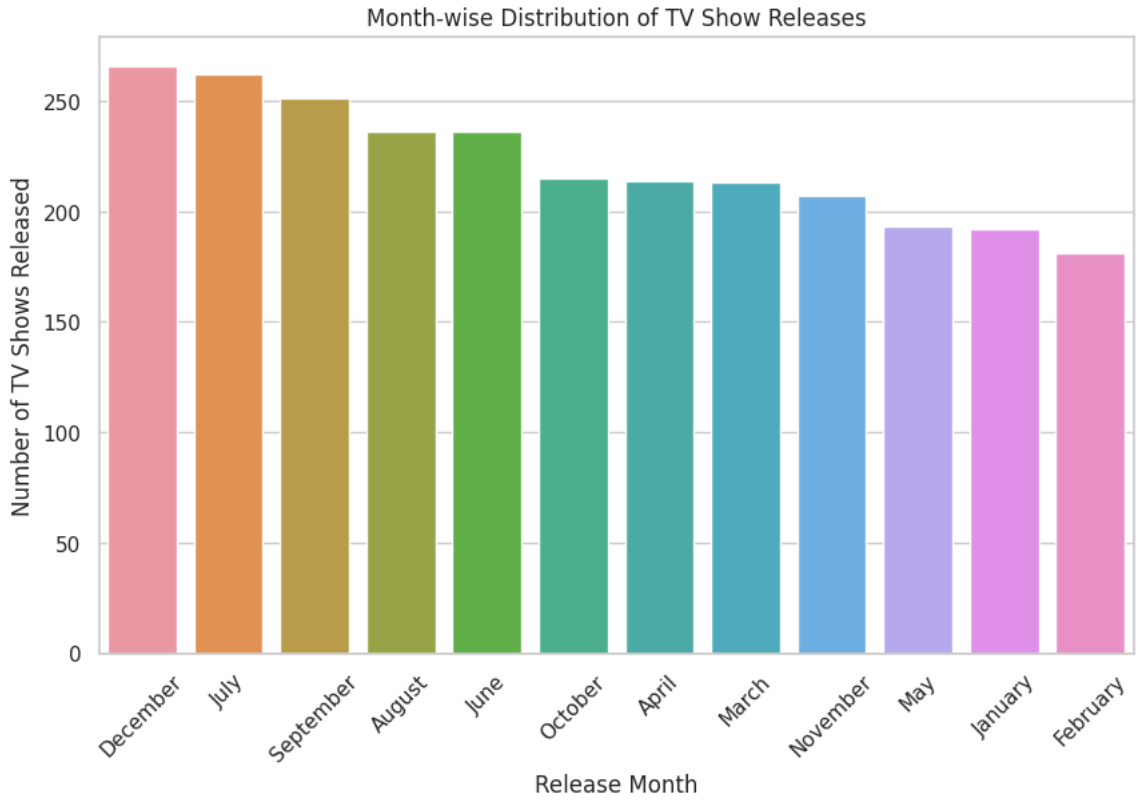
```
TVshows=data[data.type=="TV Show"]
TVshows["date_added"] = pd.to_datetime(TVshows["date_added"])
release_month = TVshows["date_added"].dt.month_name()
month_counts = release_month.value_counts()
month_counts
```

```
<ipython-input-37-86da5b47100a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  TVshows["date_added"] = pd.to_datetime(TVshows["date_added"])
December     266
July         262
September    251
August       236
June         236
October      215
April        214
March        213
November     207
May          193
January      192
February     181
Name: date_added, dtype: int64
```

Double-click (or enter) to edit

```
plt.figure(figsize=(10, 6))
sns.set(style="whitegrid")
sns.countplot(data=TVshows, x=release_month, order=release_month.value_counts().index)
plt.xlabel("Release Month")
plt.ylabel("Number of TV Shows Released")
plt.title("Month-wise Distribution of TV Show Releases")
plt.xticks(rotation=45)
plt.show()
```

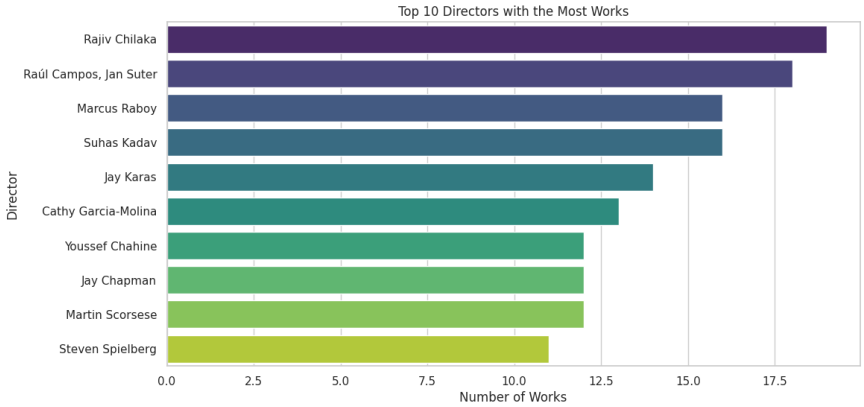


After calculating month_counts, you can further explore the data to gain insights. In above code movies are recommended more in month of december and july. Similar is the case with TV SHOWS. Best time to release is december and july.

```
no_of_directors=data["director"].value_counts().sort_values(ascending=False)
no_of_directors
```

```
Rajiv Chilaka              19
Raúl Campos, Jan Suter     18
Marcus Raboy               16
Suhas Kadav                16
Jay Karas                  14
                           ..
Austin Stark                1
Kristian Levring            1
Ajay Bhuyan, Kunal Kohli    1
Karan Lalit Butani          1
Mozez Singh                 1
Name: director, Length: 4528, dtype: int64
```

```
plt.figure(figsize=(12, 6))
sns.set(style="whitegrid")
sns.countplot(data=data, y='director', order=no_of_directors.index[:10], palette="viridis")
plt.xlabel("Number of Works")
plt.ylabel("Director")
plt.title("Top 10 Directors with the Most Works")
plt.show()
```

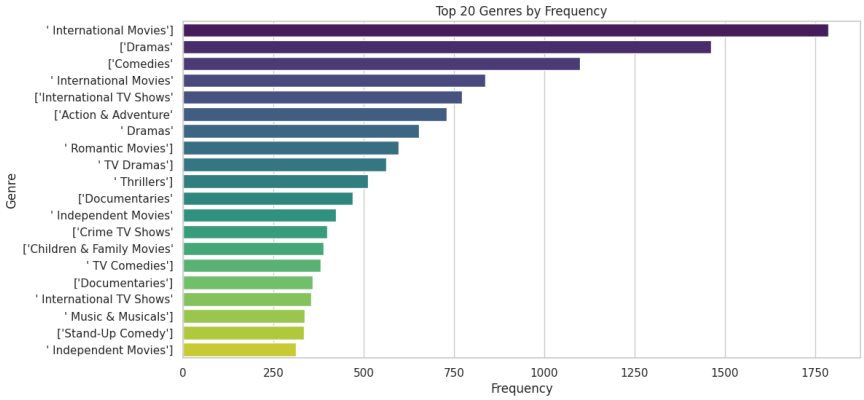


The resulting director_counts will provide you with information about which directors have the most works in your dataset, helping you identify prolific directors or those who have contributed to a significant number of movies or TV shows. A count plot showing the top 10 directors with the most works in an industry.

```
data['listed_in'] = data['listed_in'].astype(str)
data['listed_in'] = data['listed_in'].str.split(',')
diff_lists = data.explode('listed_in')
generes=diff_lists.listed_in.value_counts().sort_values(ascending=False)
generes.head(20)
```

```
 International Movies       2624
Dramas                      1600
Comedies                    1210
Action & Adventure           859
Documentaries                829
 Dramas                      827
International TV Shows       774
 Independent Movies          736
 TV Dramas                   696
 Romantic Movies             613
Children & Family Movies     605
 International TV Shows      577
 Thrillers                   512
 Comedies                    464
 TV Comedies                 461
Crime TV Shows               399
Kids' TV                     388
 Music & Musicals            357
 Romantic TV Shows           338
Stand-Up Comedy              334
Name: listed_in, dtype: int64
```

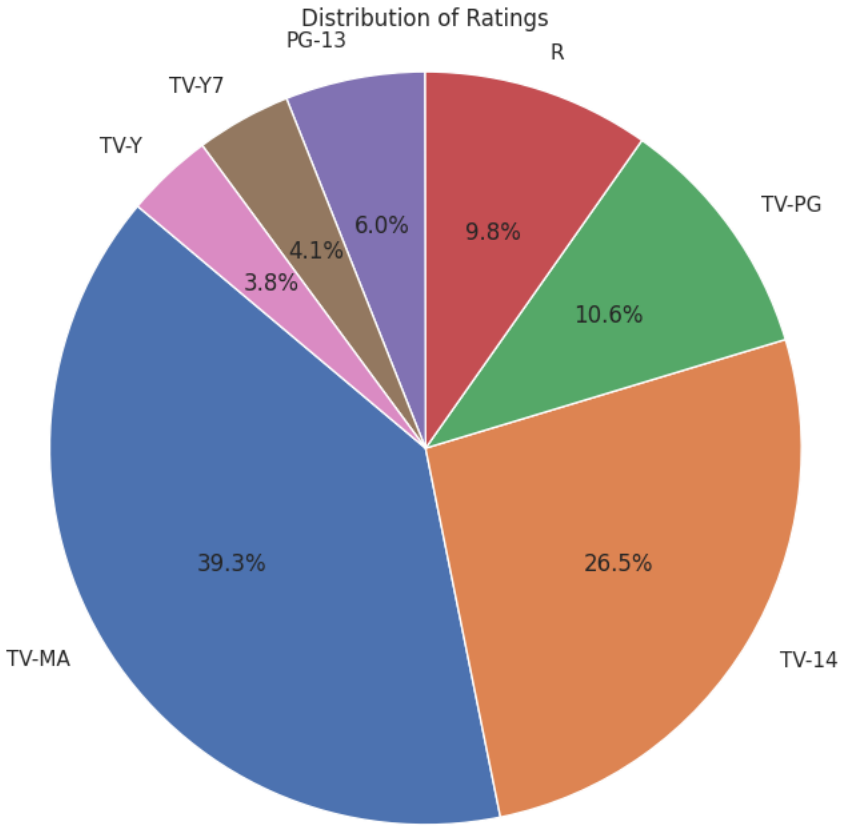
```
plt.figure(figsize=(12, 6))
sns.set(style="whitegrid")
sns.barplot(x=top_20_genres.values, y=top_20_genres.index, palette="viridis")
plt.xlabel("Frequency")
plt.ylabel("Genre")
plt.title("Top 20 Genres by Frequency")
plt.show()
```



generates a count plot of the top 20 genres based on their frequency in a dataset that includes a 'listed_in' column. Here are some insights that can be derived from the generated count plot: 1.Diversity of Genres: While the top genres are clearly visible, the presence of multiple genres indicates the diversity of content in the dataset. Some entries may belong to multiple genres, leading to a wider range of options for viewers. 2.Insights for Content Creators: Content creators and providers can gain insights into which genres are in demand among viewers. This information can help them make decisions about producing or acquiring content that aligns with popular genres. 3.Viewer Preferences: The count plot indirectly reflects viewer preferences and can help content providers understand which genres are resonating with their audience. This information can inform decisions about content acquisition and creation. 4.Data Quality: Anomalies or unusual genres that appear in the top 20 may warrant further investigation. They could be outliers or represent data quality issues that need to be addressed.

```
rating=data["rating"].value_counts()
rating.head(7)
```

```
plt.figure(figsize=(8, 8))
plt.pie(rating, labels=rating.index, autopct='%1.1f%%', startangle=140)
plt.title("Distribution of Ratings")
plt.axis('equal')
plt.show()
```




```
data['country'] = data['country'].astype(str)
data['country'] = data['country'].str.split(',')
diff_country = data.explode('country')
country=diff_country.country.value_counts().sort_values(ascending=False)
country
```

```
United States     3211
India             1008
nan                831
United Kingdom     628
 United States     479
                   ...
 Ghana               1
 Namibia             1
 Uganda              1
 East Germany        1
 Montenegro          1
Name: country, Length: 198, dtype: int64
```

Double-click (or enter) to edit

```
The code provided generates a desciption of the top 10 countries with the most works in this dataset.
1.Top Producing Countries: This  reveals the countries that have produced the most works in your dataset. These countries are likely to
2.Global vs. Local: Depending on your dataset, you can see whether global content (e.g., Hollywood productions) dominates or if there's
3.Content Strategy: Organizations can use this information to shape their content strategy. For example, they may choose to invest in a
```

```
data['cast'] = data['cast'].astype(str)
data['cast'] = data['cast'].str.split(',')
diff_cast = data.explode('cast')
diff_cast.cast.value_counts().sort_values(ascending=False)
```

```
nan                     825
 Anupam Kher             39
 Rupa Bhimani            31
 Takahiro Sakurai        30
 Julie Tejwani           28
                        ...
 Karen Dunbar             1
 John Macmillan           1
 Robert Lonsdale          1
 Danielle Walters         1
 Chittaranjan Tripathy    1
Name: cast, Length: 39297, dtype: int64
```

```
director_data1 = diff_dir[diff_dir['director'].apply(lambda x: "Rajiv Chilaka" in x)]
director_data2 = diff_dir[diff_dir['director'].apply(lambda x: "Jan Suter" in x)]
director_data3 = diff_dir[diff_dir['director'].apply(lambda x: "Raúl Campos" in x)]
combined_director_data = pd.concat([director_data1, director_data2, director_data3])
print(combined_director_data[["director", "country"]])
```

```
                      director country
406          ['Rajiv Chilaka']     NaN
407          ['Rajiv Chilaka']     NaN
408          ['Rajiv Chilaka']     NaN
409          ['Rajiv Chilaka']     NaN
410          ['Rajiv Chilaka']   India
```