business-casestudy-final-netflix

April 15, 2024

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[4]:
```
df = pd.read_csv("netflix_titles.csv")
df
```

[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…	NaN	NaN	NaN	NaN	NaN
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…	India	NaN	NaN	NaN	NaN
…	…	…	…	…	…	…	…	…	…	…
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J…	United States	NaN	NaN	NaN	NaN
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, …	United States	NaN	NaN	NaN	NaN
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma…	United States	NaN	NaN	NaN	NaN
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan…	India	NaN	NaN	NaN	NaN

```
2     September 24, 2021          2021  TV-MA   1 Season
3     September 24, 2021          2021  TV-MA   1 Season
4     September 24, 2021          2021  TV-MA  2 Seasons
…                   …           …     …        …
8802   November 20, 2019          2007      R    158 min
8803        July 1, 2019          2018  TV-Y7  2 Seasons
8804    November 1, 2019          2009      R     88 min
8805    January 11, 2020          2006     PG     88 min
8806       March 2, 2019          2015  TV-14    111 min

                                                listed_in  \
0                                          Documentaries
1        International TV Shows, TV Dramas, TV Mysteries
2     Crime TV Shows, International TV Shows, TV Act…
3                                Docuseries, Reality TV
4     International TV Shows, Romantic TV Shows, TV …
…                                                     …
8802                    Cult Movies, Dramas, Thrillers
8803            Kids' TV, Korean TV Shows, TV Comedies
8804                           Comedies, Horror Movies
8805                Children & Family Movies, Comedies
8806    Dramas, International Movies, Music & Musicals

                                               description
0     As her father nears the end of his life, filmm…
1     After crossing paths at a party, a Cape Town t…
2     To protect his family from a powerful drug lor…
3     Feuds, flirtations and toilet talk go down amo…
4     In a city of coaching centers known to train I…
…                                                   …
8802  A political cartoonist, a crime reporter and a…
8803  While living alone in a spooky town, a young g…
8804  Looking to survive in a world taken over by zo…
8805  Dragged from civilian life, a former superhero…
8806  A scrappy but poor boy worms his way into a ty…

[8807 rows x 12 columns]
```

[5]:
```
print("Dimension of Netflix Dataframe:")
df.ndim
```

```
Dimension of Netflix Dataframe:
```

[5]: 2

[6]:
```
print("Rows and Columns in Netflix Dataframe:")
df.shape
```

Rows and Columns in Netflix Dataframe:

[6]: (8807, 12)

[7]:
```
print("Columns Names in Netflix Dataframe:")
df.columns
```

Columns Names in Netflix Dataframe:

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

[46]:
```
df.head()
```

[46]:

	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 …	

```
4  In a city of coaching centers known to train I…
```

[47]:
```
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
```

[12]:
```
df.describe()
```

[12]:

	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

[50]:
```
print('\nChecking for Column with missing value:')
print(df.isnull().any())
```

```
Checking for Column with missing value:
show_id         False
type            False
title           False
director         True
cast             True
country          True
date_added       True
```

```
release_year    False
rating           True
duration         True
listed_in       False
description     False
dtype: bool
```

[8]:
```
print("Count of Missing/Null Values per Column:")
print(df.isnull().sum())
```

```
Count of Missing/Null Values per Column:
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
```

[52]:
```
print("Sum of Null values Present:")
df.isnull().sum().sum()
```

```
Sum of Null values Present:
```

[52]:
```
4307
```

[9]:
```
print("After replacement of Null values:")
df.director.fillna("No director",inplace=True)
df.cast.fillna("No cast",inplace = True)
df.country.fillna("No country",inplace=True)
df.dropna(subset=["date_added","rating","duration"],inplace=True)
df
```

```
After replacement of Null values:
```

[9]:

	show_id	type	title	director	\
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	
1	s2	TV Show	Blood & Water	No director	
2	s3	TV Show	Ganglands	Julien Leclercq	
3	s4	TV Show	Jailbirds New Orleans	No director	
4	s5	TV Show	Kota Factory	No director	
…	…	…	…	…	

```
8802   s8803    Movie                Zodiac    David Fincher
8803   s8804  TV Show           Zombie Dumb      No director
8804   s8805    Movie            Zombieland  Ruben Fleischer
8805   s8806    Movie                 Zoom     Peter Hewitt
8806   s8807    Movie               Zubaan      Mozez Singh

                                                    cast        country  \
0                                                 No cast  United States
1     Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…   South Africa
2     Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…     No country
3                                                 No cast     No country
4     Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…           India
…                                                    …           …
8802  Mark Ruffalo, Jake Gyllenhaal, Robert Downey J…  United States
8803                                              No cast     No country
8804  Jesse Eisenberg, Woody Harrelson, Emma Stone, …  United States
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma…  United States
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan…           India

             date_added  release_year rating   duration  \
0     September 25, 2021          2020  PG-13     90 min
1     September 24, 2021          2021  TV-MA  2 Seasons
2     September 24, 2021          2021  TV-MA   1 Season
3     September 24, 2021          2021  TV-MA   1 Season
4     September 24, 2021          2021  TV-MA  2 Seasons
…                    …           …    …        …
8802   November 20, 2019          2007      R    158 min
8803        July 1, 2019          2018  TV-Y7  2 Seasons
8804    November 1, 2019          2009      R     88 min
8805    January 11, 2020          2006     PG     88 min
8806       March 2, 2019          2015  TV-14    111 min

                                              listed_in  \
0                                         Documentaries
1       International TV Shows, TV Dramas, TV Mysteries
2     Crime TV Shows, International TV Shows, TV Act…
3                               Docuseries, Reality TV
4     International TV Shows, Romantic TV Shows, TV …
…                                                    …
8802                    Cult Movies, Dramas, Thrillers
8803            Kids' TV, Korean TV Shows, TV Comedies
8804                          Comedies, Horror Movies
8805                Children & Family Movies, Comedies
8806    Dramas, International Movies, Music & Musicals

                                           description
0     As her father nears the end of his life, filmm…
```

```
1     After crossing paths at a party, a Cape Town t…
2     To protect his family from a powerful drug lor…
3     Feuds, flirtations and toilet talk go down amo…
4     In a city of coaching centers known to train I…
…                                                   …
8802  A political cartoonist, a crime reporter and a…
8803  While living alone in a spooky town, a young g…
8804  Looking to survive in a world taken over by zo…
8805  Dragged from civilian life, a former superhero…
8806  A scrappy but poor boy worms his way into a ty…

[8790 rows x 12 columns]
```

[10]:
```
print("Sum of Null values Present:")
df.isnull().sum().sum()
```

```
Sum of Null values Present:
```

[10]:
```
0
```

[11]:
```
print("Values in Listed_in column seperated by Genre:")
# Split the "listed_in" column by commas and explode into separate rows
df_exploded_listed_in = df.assign(listed_in=df['listed_in'].str.split(',')).
 ↪explode('listed_in')
# Update the DataFrame
df = df_exploded_listed_in
df
```

```
Values in Listed_in column seperated by Genre:
```

[11]:

	show_id	type	title	director	cast	country
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No cast	United States
1	s2	TV Show	Blood & Water	No director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…	South Africa
1	s2	TV Show	Blood & Water	No director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…	South Africa
1	s2	TV Show	Blood & Water	No director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…	South Africa
2	s3	TV Show	Ganglands	Julien Leclercq	…	…
…	…	…	…	…	…	…
8805	s8806	Movie	Zoom	Peter Hewitt	…	…
8805	s8806	Movie	Zoom	Peter Hewitt	…	…
8806	s8807	Movie	Zubaan	Mozez Singh	…	…
8806	s8807	Movie	Zubaan	Mozez Singh	…	…
8806	s8807	Movie	Zubaan	Mozez Singh	…	…

```
1     Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…   South Africa
2     Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…     No country
…                                                   …             …
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma…  United States
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma…  United States
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan…          India
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan…          India
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan…          India

              date_added  release_year rating   duration  \
0     September 25, 2021          2020  PG-13     90 min
1     September 24, 2021          2021  TV-MA  2 Seasons
1     September 24, 2021          2021  TV-MA  2 Seasons
1     September 24, 2021          2021  TV-MA  2 Seasons
2     September 24, 2021          2021  TV-MA   1 Season
…                    …           …    …        …
8805    January 11, 2020          2006     PG     88 min
8805    January 11, 2020          2006     PG     88 min
8806       March 2, 2019          2015  TV-14    111 min
8806       March 2, 2019          2015  TV-14    111 min
8806       March 2, 2019          2015  TV-14    111 min

                     listed_in  \
0                Documentaries
1       International TV Shows
1                    TV Dramas
1                 TV Mysteries
2                Crime TV Shows
…                          …
8805  Children & Family Movies
8805                 Comedies
8806                   Dramas
8806      International Movies
8806          Music & Musicals

                                              description
0     As her father nears the end of his life, filmm…
1     After crossing paths at a party, a Cape Town t…
1     After crossing paths at a party, a Cape Town t…
1     After crossing paths at a party, a Cape Town t…
2     To protect his family from a powerful drug lor…
…                                                   …
8805  Dragged from civilian life, a former superhero…
8805  Dragged from civilian life, a former superhero…
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…
```

[19294 rows x 12 columns]

[24]:
```
print("Values in cast column seperated by Actors:")
# Split the "listed_in" column by commas and explode into separate rows
df_exploded_cast = df.assign(cast=df['cast'].str.split(',')).explode('cast')
# Update the DataFrame
df = df_exploded_cast
df
```

Values in cast column seperated by Actors:

[24]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No cast	United States	September 25, 2021	2020	PG-13	90 min	Documentaries
1	s2	TV Show	Blood & Water	No director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows
1	s2	TV Show	Blood & Water	No director	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows
1	s2	TV Show	Blood & Water	No director	Gail Mabalane	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows
1	s2	TV Show	Blood & Water	No director	Thabang Molaba	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows
…	…	…	…	…	…	…	…	…	…	…	…
8806	s8807	Movie	Zubaan	Mozez Singh	Manish Chaudhary	India	March 2, 2019	2015	TV-14	111 min	Music & Musicals
8806	s8807	Movie	Zubaan	Mozez Singh	Meghna Malik	India	March 2, 2019	2015	TV-14	111 min	Music & Musicals
8806	s8807	Movie	Zubaan	Mozez Singh	Malkeet Rauni	India	March 2, 2019	2015	TV-14	111 min	Music & Musicals
8806	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	March 2, 2019	2015	TV-14	111 min	Music & Musicals
8806	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	March 2, 2019	2015	TV-14	111 min	Music & Musicals

```
8806  TV-14    111 min        Music & Musicals
8806  TV-14    111 min        Music & Musicals
8806  TV-14    111 min        Music & Musicals

                                              description
0     As her father nears the end of his life, filmm…
1     After crossing paths at a party, a Cape Town t…
1     After crossing paths at a party, a Cape Town t…
1     After crossing paths at a party, a Cape Town t…
1     After crossing paths at a party, a Cape Town t…
…                                                   …
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…

[149284 rows x 12 columns]
```

[25]:
```
print("Values in cast column seperated by country:")
# Split the "listed_in" column by commas and explode into separate rows
df_exploded_country = df.assign(country=df['country'].str.split(',')).
 ↪explode('country')
# Update the DataFrame
df = df_exploded_country
df
```

```
Values in cast column seperated by country:
```

[25]:
```
     show_id     type                  title         director  \
0         s1    Movie  Dick Johnson Is Dead  Kirsten Johnson
1         s2  TV Show         Blood & Water      No director
1         s2  TV Show         Blood & Water      No director
1         s2  TV Show         Blood & Water      No director
1         s2  TV Show         Blood & Water      No director
…       …       …                     …              …
8806   s8807    Movie                Zubaan      Mozez Singh
8806   s8807    Movie                Zubaan      Mozez Singh
8806   s8807    Movie                Zubaan      Mozez Singh
8806   s8807    Movie                Zubaan      Mozez Singh
8806   s8807    Movie                Zubaan      Mozez Singh

                        cast        country          date_added  release_year  \
0                  No cast  United States  September 25, 2021          2020
1                Ama Qamata   South Africa  September 24, 2021          2021
1                Khosi Ngema  South Africa  September 24, 2021          2021
1              Gail Mabalane  South Africa  September 24, 2021          2021
```

```
1              Thabang Molaba   South Africa  September 24, 2021          2021
…                         …              …                   …           …
8806         Manish Chaudhary          India       March 2, 2019          2015
8806             Meghna Malik          India       March 2, 2019          2015
8806            Malkeet Rauni          India       March 2, 2019          2015
8806           Anita Shabdish          India       March 2, 2019          2015
8806    Chittaranjan Tripathy          India       March 2, 2019          2015

     rating   duration               listed_in  \
0     PG-13     90 min           Documentaries
1     TV-MA  2 Seasons  International TV Shows
1     TV-MA  2 Seasons  International TV Shows
1     TV-MA  2 Seasons  International TV Shows
1     TV-MA  2 Seasons  International TV Shows
…     …         …                        …
8806  TV-14    111 min        Music & Musicals
8806  TV-14    111 min        Music & Musicals
8806  TV-14    111 min        Music & Musicals
8806  TV-14    111 min        Music & Musicals
8806  TV-14    111 min        Music & Musicals

                                              description
0     As her father nears the end of his life, filmm…
1     After crossing paths at a party, a Cape Town t…
1     After crossing paths at a party, a Cape Town t…
1     After crossing paths at a party, a Cape Town t…
1     After crossing paths at a party, a Cape Town t…
…                                                   …
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…
8806  A scrappy but poor boy worms his way into a ty…

[186171 rows x 12 columns]
```

[26]:
```
print("Checking for any empty strings in each column:")
df.eq('').any()
```

```
Checking for any empty strings in each column:
```

[26]:
```
show_id         False
type            False
title           False
director        False
cast            False
country          True
```

```
date_added      False
release_year    False
rating          False
duration        False
listed_in       False
description     False
dtype: bool
```

[30]:
```
df['country'] = df['country'].replace('', 'Unknown')
df.eq('').any()
```

[30]:
```
show_id         False
type            False
title           False
director        False
cast            False
country         False
date_added      False
release_year    False
rating          False
duration        False
listed_in       False
description     False
dtype: bool
```

[31]:
```
print("Updated Rows and Columns after removing Null values:")
df.shape
```

```
Updated Rows and Columns after removing Null values:
```

[31]:
```
(186171, 12)
```

[ ]:
```
#1.How has the number of movies released per year changed over the last 20-30␣
 ↪years?
```

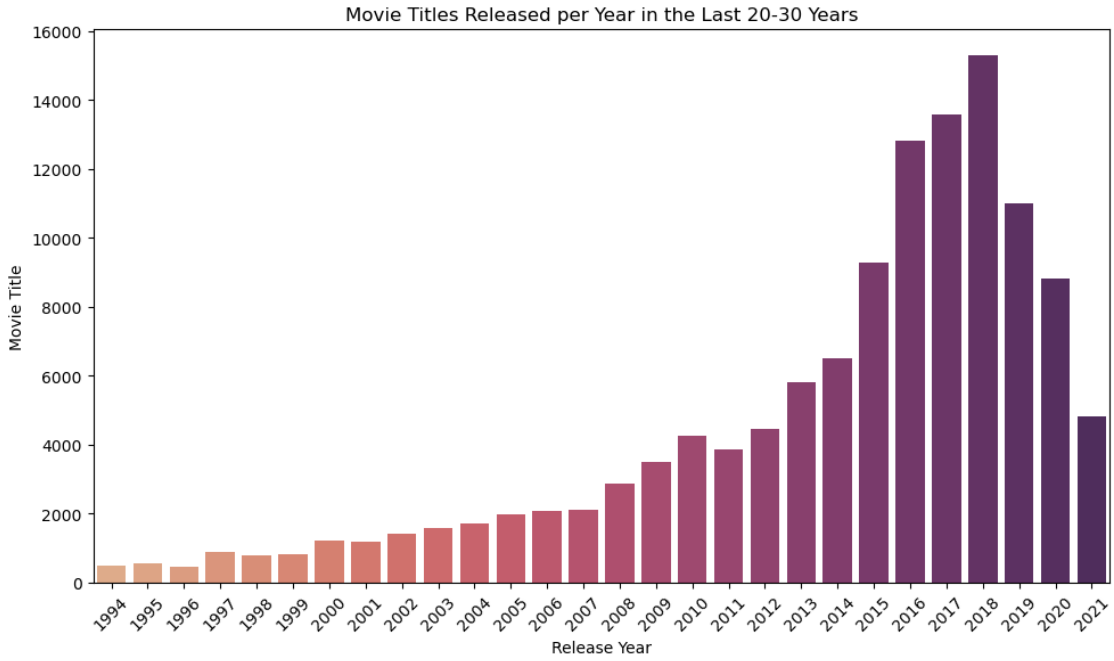
[62]:
```
#movies_per_year = df[df["type"]=="Movie"].groupby("release_year")["title"].
 ↪nunique()
print("Count of Movie Titles Released per Year in the Last 20-30 Years:")
movies_df = df[df['type'] == 'Movie']
current_year= pd.Timestamp.now().year
last_20_years_data = movies_df[(movies_df['release_year'] >= current_year - 30)]

# Group by release year and count the number of movies released each year
movies_per_year = last_20_years_data.groupby('release_year')["title"].size()
movies_per_year
```

```
Count of Movie Titles Released per Year in the Last 20-30 Years:
```

[62]:
```
release_year
1994      480
1995      539
1996      447
1997      883
1998      797
1999      825
2000     1218
2001     1181
2002     1403
2003     1586
2004     1724
2005     1976
2006     2065
2007     2111
2008     2877
2009     3509
2010     4249
2011     3867
2012     4459
2013     5802
2014     6485
2015     9282
2016    12798
2017    13584
2018    15277
2019    10985
2020     8807
2021     4830
Name: title, dtype: int64
```

[33]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x='release_year', y='title',data=movies_per_year.reset_index(),␣
 ↪palette="flare")
plt.xlabel('Release Year')
plt.ylabel('Movie Title')
plt.title('Movie Titles Released per Year in the Last 20-30 Years')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



[105]:
```
# 2. Find the counts of each categorical variable both using graphical and␣
 ↪non-graphical analysis.
```

[149]:
```
print("Type and unique count of Categorical variables:")
df[["type", "director", "country", "rating", "listed_in"]].nunique()
```

```
Type and unique count of Categorical variables:
```

[149]:
```
type            2
director     4527
country       198
rating         14
listed_in      73
dtype: int64
```

[153]:
```
#Non-graphical
print("Non Graphical representation of value counts for each Categorical␣
 ↪Variables:")
# Value counts for 'type' column
type_counts = df['type'].value_counts()
print("Count of Each Type:")
print(type_counts)
print()

# Value counts for 'rating' column
```

```
rating_counts = df['rating'].value_counts()
print("Count of Each Rating:")
print(rating_counts)
print()

# Value counts for 'country' column (showing top 10 countries)
top_countries = df['country'].value_counts().head(10)
print("Top 10 Countries:")
print(top_countries)
print()

# Value counts for 'listed_in' column (showing top 10 genres)
top_genres = df['listed_in'].value_counts().head(10)
print("Top 10 Genres:")
print(top_genres)
print()

# Value counts for 'release_year' column (showing top 10 years)
top_years = df['release_year'].value_counts().head(10)
print("Top 10 Release Years:")
print(top_years)
```

```
Non Graphical representation of value counts for each Categorical Variables:
Count of Each Type:
Movie      131919
TV Show     54252
Name: type, dtype: int64

Count of Each Rating:
TV-MA       67644
TV-14       41982
R           23990
PG-13       15233
TV-PG       13765
PG           9011
TV-Y7        5782
TV-Y         3151
TV-G         2650
NR           1491
G            1151
NC-17         149
TV-Y7-FV       86
UR             86
Name: rating, dtype: int64

Top 10 Countries:
United States     45365
```

```
India              20526
No country         11142
United Kingdom      9324
 United States      8768
Japan               6774
South Korea         4556
Spain               4087
Canada              4072
France              3856
Name: country, dtype: int64

Top 10 Genres:
 International Movies       25097
Dramas                      18650
Comedies                    12262
Action & Adventure          11124
 Dramas                      9142
 Independent Movies          8577
 TV Dramas                   7480
Children & Family Movies     7267
International TV Shows       6589
 Romantic Movies             6139
Name: listed_in, dtype: int64

Top 10 Release Years:
2018-01-01    22645
2019-01-01    20571
2017-01-01    18990
2020-01-01    18159
2016-01-01    17183
2015-01-01    12744
2021-01-01    10834
2014-01-01     7935
2013-01-01     6954
2012-01-01     5512
Name: release_year, dtype: int64
```

[156]:
```
# b. For graphical analysis:

# Create a figure object
fig, axs = plt.subplots(3, 2, figsize=(15, 15))

# Plot count plot for 'type' column
sns.countplot(data=df, x="type", ax=axs[0, 0])
axs[0, 0].set_xlabel("Type")
axs[0, 0].set_ylabel("Count")
axs[0, 0].set_title("Count of Each Type")
```

```
# Plot count plot for 'rating' column
sns.countplot(data=df, x="rating", ax=axs[0, 1])
axs[0, 1].set_xlabel("Rating")
axs[0, 1].set_ylabel("Count")
axs[0, 1].set_title("Count of Each Rating")

# Plot count plot for 'country' column (showing top 10 countries)
top_countries = df['country'].value_counts().head(10)
sns.barplot(x=top_countries.values, y=top_countries.index, ax=axs[1, 0])
axs[1, 0].set_xlabel("Count")
axs[1, 0].set_ylabel("Country")
axs[1, 0].set_title("Top 10 Countries")

# Plot count plot for 'listed_in' column (showing top 10 genres)
top_genres = df['listed_in'].value_counts().head(10)
sns.barplot(x=top_genres.values, y=top_genres.index, ax=axs[1, 1])
axs[1, 1].set_xlabel("Count")
axs[1, 1].set_ylabel("Genre")
axs[1, 1].set_title("Top 10 Genres")

# Plot count plot for 'release_year' column (showing top 10 years)
top_years = df['release_year'].value_counts().head(10)
sns.barplot(x=top_years.values, y=top_years.index, ax=axs[2, 0])
axs[2, 0].set_xlabel("Count")
axs[2, 0].set_ylabel("Year")
axs[2, 0].set_title("Top 10 Release Years")

# Remove empty subplot
fig.delaxes(axs[2,1])

# Adjust layout
plt.tight_layout()
print("Graphical Representation of counts of each categorical variable:")
# Show plot
plt.show()
```

```
Graphical Representation of counts of each categorical variable:
```

[144]:
```
#3.Comparison of tv shows vs. movies
print("Comparing the number of Movies and TV Shows present in Netflix:")
df.groupby("type").size()
```

```
Comparing the number of Movies and TV Shows present in Netflix:
```

[144]:
```
type
Movie      131919
TV Show     54252
dtype: int64
```

[64]:
```
# a.Find the number of movies produced in each country and pick the top 10
# countries.
```

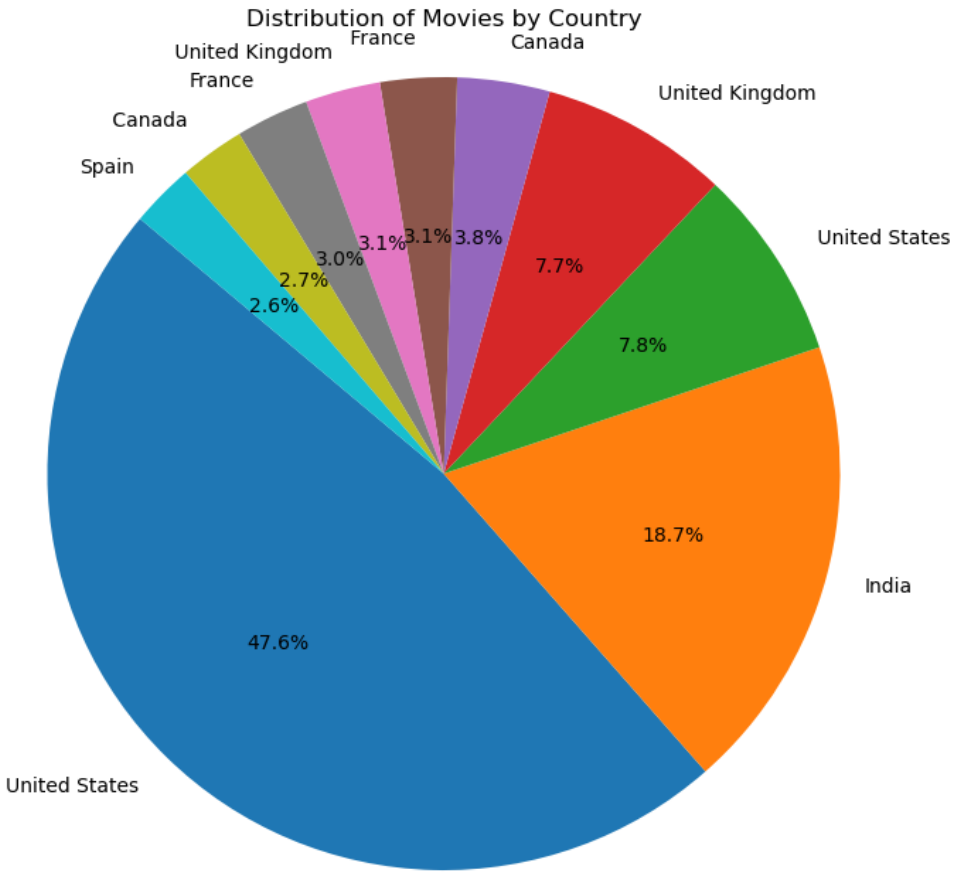
```
print("Top 10 Countries by Number of Movies Produced:")
df_filtered = df[df['country'] != 'No country']
df_movies_in_country = df_filtered[df_filtered["type"]=="Movie"].
 ↪groupby("country")["title"].nunique().sort_values(ascending=False).head(10)
df_movies_in_country
```

```
Top 10 Countries by Number of Movies Produced:
```

[64]:
```
country
United States      2361
India               927
 United States      388
United Kingdom      382
Canada              187
 France             155
 United Kingdom     152
France              148
 Canada             132
Spain               129
Name: title, dtype: int64
```

[70]:
```
# Prepare data
labels = df_movies_in_country.reset_index()['country']
sizes = df_movies_in_country.reset_index()['title']

# Create a pie chart
plt.figure(figsize=(10, 8))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Distribution of Movies by Country')
plt.show()
```



[73]:
```
# b. Find the number of Tv-Shows produced in each country and pick the top 10
# countries.
# Hint : We want you to apply group by each country and find the count of unique
# titles of Tv-shows
print("Top 10 Countries by Number of TV Shows Produced:")
df_filtered = df[df['country'] != 'No country']

# Group by country and count the number of unique TV show titles
df_TVShow_in_country = df_filtered[df_filtered["type"] == "TV Show"].
 ↪groupby("country")["title"].nunique().sort_values(ascending=False).head(10)
df_TVShow_in_country
```

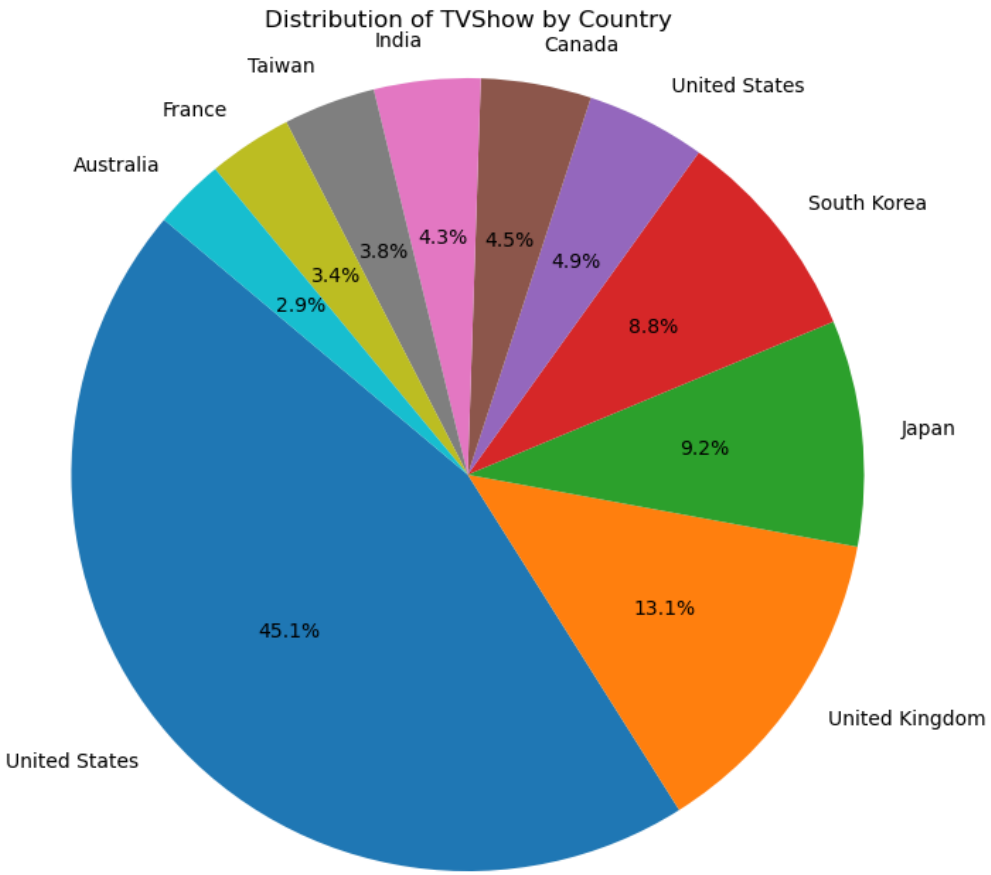
```
Top 10 Countries by Number of TV Shows Produced:
```

[73]:
```
country
United States     841
United Kingdom    245
Japan             172
```

```
South Korea       164
 United States     91
Canada             84
India              81
Taiwan             70
France             64
Australia          54
Name: title, dtype: int64
```

[74]:
```python
# Prepare data
labels = df_TVShow_in_country.reset_index()['country']
sizes = df_TVShow_in_country.reset_index()['title']

# Create a pie chart
plt.figure(figsize=(10, 8))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Distribution of TVShow by Country')
plt.show()
```



[78]:
```
# 4.a. What is the best time to launch a TV show?
#Best week to release the Tv-show or the movie.
print("Best week to release the Tv-show:")
df["date_added"] = pd.to_datetime(df["date_added"])
df["week_added"] = df["date_added"].dt.week
tv_shows_per_week  = df[df["type"]=="TV Show"].groupby("week_added").size().
 ↪sort_values(ascending=False)
tv_shows_per_week.head(10)
```

```
Best week to release the Tv-show:

C:\Users\Harini\AppData\Local\Temp\ipykernel_1868\1956008584.py:5:
FutureWarning: Series.dt.weekofyear and Series.dt.week have been deprecated.
Please use Series.dt.isocalendar().week instead.
  df["week_added"] = df["date_added"].dt.week
```

[78]:
```
week_added
27    1977
35    1942
24    1702
13    1554
26    1530
48    1469
31    1461
40    1362
44    1347
5     1344
dtype: int64
```

[79]:
```
print("Best week to release the Movie:")
movies_per_week = df[df['type'] == 'Movie'].groupby('week_added').size().
 ↪sort_values(ascending=False)
movies_per_week.head(10)
```

```
Best week to release the Movie:
```

[79]:
```
week_added
1     7838
44    5275
9     4664
40    4600
35    4589
26    4457
31    3891
27    3397
48    3394
```

```
13    3289
dtype: int64
```

[145]:
```
# b. Find which is the best month to release the Tv-show or the movie. Do the
# analysis separately for Tv-shows and Movies
print("Best month to release the TV Show:")
df["month_added"] = df["date_added"].dt.month
tv_shows_per_month  = df[df["type"]=="TV Show"].groupby("month_added").size().
 ↪sort_values(ascending=False)
tv_shows_per_month
```

```
Best month to release the TV Show:
```

[145]:
```
month_added
12    5313
7     5089
9     4855
6     4846
8     4800
11    4531
4     4415
3     4324
1     4296
10    4220
5     3886
2     3677
dtype: int64
```

[90]:
```
print("Best month to release the Movie:")
df["month_added"] = df["date_added"].dt.month
movies_per_month = df[df['type'] == 'Movie'].groupby('month_added').size().
 ↪sort_values(ascending=False)
movies_per_month
```

```
Best month to release the Movie:
```

[90]:
```
month_added
7     12939
1     12687
9     11970
10    11941
12    11576
4     11057
8     11047
3     10897
6     10567
11    10336
```

```
5      8728
2      8174
dtype: int64
```

[95]:
```
# 5. Analysis of actors/directors of different types of shows/movies.
# a. Identify the top 10 directors who have appeared in most movies or TV shows.

print("Top 10 Directors with the Highest Number of Unique TV Shows:")
filtered_df = df[df["director"] != "No director"]
filtered_df[filtered_df["type"]=="TV Show"].groupby('director')['title'].
 ↪nunique().sort_values(ascending=False).head(10)
```

```
Top 10 Directors with the Highest Number of Unique TV Shows:
```

[95]:
```
director
Alastair Fothergill                                                 3
Stan Lathan                                                         2
Iginio Straffi                                                      2
Rob Seidenglanz                                                     2
Ken Burns                                                           2
Shin Won-ho                                                         2
Hsu Fu-chun                                                         2
Miguel Conde                                                        1
Mike Flanagan                                                       1
Norm Hiscock, Gary Howsam, Mike Smith, John Paul Tremblay, Robb Wells    1
Name: title, dtype: int64
```

[96]:
```
print("Top 10 Directors with the Highest Number of Unique Movies:")
filtered_df = df[df["director"] != "No director"]
filtered_df[filtered_df["type"]=="Movie"].groupby('director')['title'].
 ↪nunique().sort_values(ascending=False).head(10)
```

```
Top 10 Directors with the Highest Number of Unique Movies:
```

[96]:
```
director
Rajiv Chilaka                      19
Raúl Campos, Jan Suter             18
Suhas Kadav                        16
Marcus Raboy                       15
Jay Karas                          14
                                   ..
Jose Gomez                          1
Jose Javier Reyes                   1
Bilal Lashari                       1
Joseduardo Giordano, Sergio Goyri Jr.    1
Kief Davidson, Pedro Kos            1
Name: title, Length: 4352, dtype: int64
```

[100]:
```
#b. Identify the top 10 actors who have appeared in most movies or TV shows.
print("Top 10 Actors with the Highest Number of Unique TV Show:")
filtered_df = df[df["cast"] != "No cast"]
filtered_df[filtered_df["type"]=="TV Show"].groupby('cast')['title'].nunique().
 ↪sort_values(ascending=False).head(10)
```

```
Top 10 Actors with the Highest Number of Unique TV Show:
```

[100]:
```
cast
 Takahiro Sakurai     24
 Yuki Kaji            17
 Junichi Suwabe       17
 Ai Kayano            16
David Attenborough    14
 Daisuke Ono          14
 Yoshimasa Hosoya     13
 Yuichi Nakamura      13
 Takehito Koyasu      13
 Kana Hanazawa        12
Name: title, dtype: int64
```

[104]:
```
print("Top 10 Actors with the Highest Number of Unique Movie:")
filtered_df = df[df["cast"] != "No cast"]
filtered_df[filtered_df["type"]=="Movie"].groupby('cast')['title'].nunique().
 ↪sort_values(ascending=False).head(10)
```

```
Top 10 Actors with the Highest Number of Unique Movie:
```

[104]:
```
cast
 Anupam Kher       38
 Rupa Bhimani      27
 Om Puri           27
Shah Rukh Khan     26
 Boman Irani       25
 Paresh Rawal      25
 Julie Tejwani     24
Akshay Kumar       23
 Rajesh Kava       21
 Kareena Kapoor    20
Name: title, dtype: int64
```

[116]:
```
# 6. Which genre movies are more popular or produced more
print("Count of Unique Genres Present:")
df.listed_in.nunique()
```

```
Count of Unique Genres Present:
```

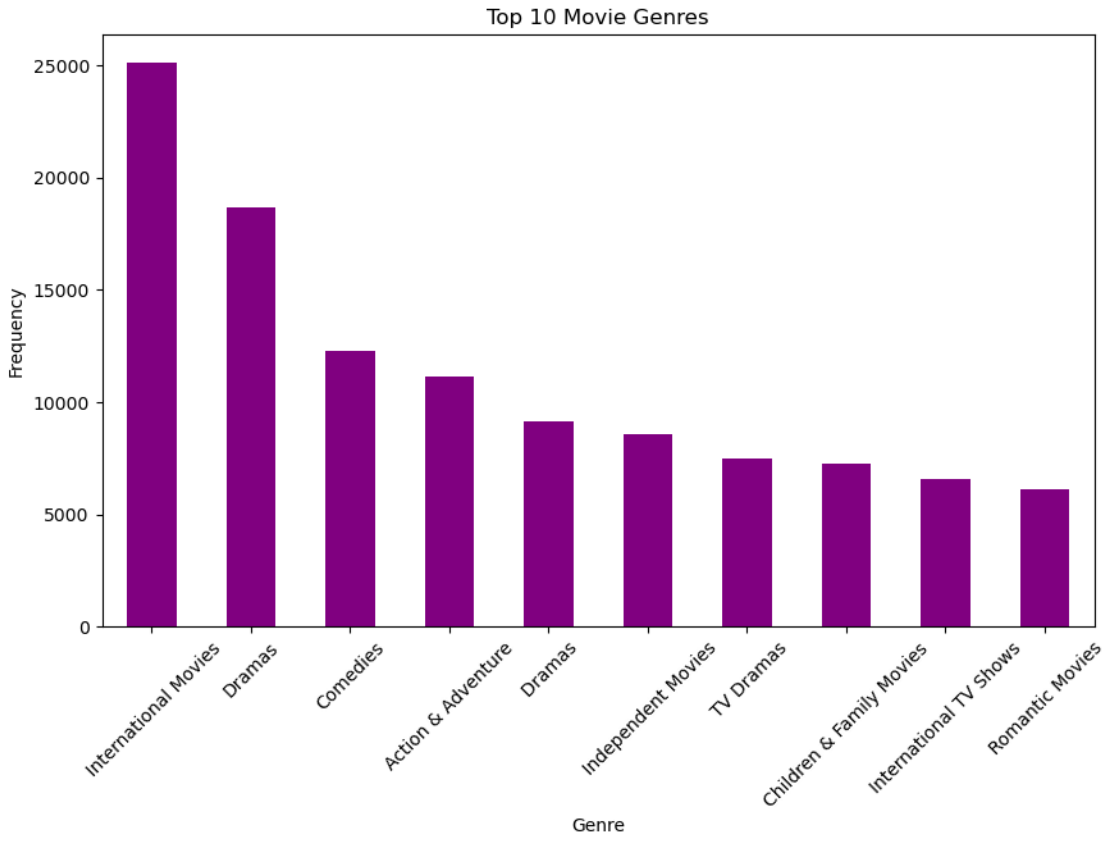
[116]:
```
73
```

[122]:
```
#Display the top genres
print("Top Genre Movies by Popularity: Arranged from High to Low:")
df[df["type"]=="Movie"].groupby('listed_in')['title'].nunique().
 ↪sort_values(ascending=False)
```

```
Top Genre Movies by Popularity: Arranged from High to Low:
```

[122]:
```
listed_in
 International Movies        2624
Dramas                       1599
Comedies                     1210
Action & Adventure            859
Documentaries                 829
 Dramas                       827
 Independent Movies           736
 Romantic Movies              613
Children & Family Movies      605
 Thrillers                    512
 Comedies                     464
 Music & Musicals             357
Stand-Up Comedy               334
Horror Movies                 275
 Sci-Fi & Fantasy             230
 Sports Movies                218
International Movies          128
 LGBTQ Movies                 101
 Horror Movies                 82
Classic Movies                 80
Thrillers                      65
 Faith & Spirituality          65
 Cult Movies                   59
Movies                         53
 Anime Features                50
 Documentaries                 40
 Children & Family Movies      36
 Classic Movies                36
Anime Features                 21
Independent Movies             20
Music & Musicals               18
Sci-Fi & Fantasy               13
Cult Movies                    12
 Stand-Up Comedy               9
Romantic Movies                3
LGBTQ Movies                   1
Sports Movies                  1
Name: title, dtype: int64
```

[121]:
```
plt.figure(figsize=(10, 6))
genre_counts.head(10).plot(kind='bar',color="purple")
plt.xlabel('Genre')
plt.ylabel('Frequency')
plt.title('Top 10 Movie Genres')
plt.xticks(rotation=45)  # Rotate genre labels for better readability
plt.show()
```



[137]:
```
# 7.Find After how many days the movie will be added to Netflix after the␣
 ↪release of
# the movie (you can consider the recent past data)
#  want you to get the difference between the columns having date added
# information and release year information and get the mode of difference. This
# will give an insight into what will be the better time to add in Netflix
print("Days Taken for Movies to be Added to Netflix After Release:")
# Step 1: Calculate the difference in days
df['release_year'] = pd.to_datetime(df['release_year'], format='%Y')  # Convert␣
 ↪release year to datetime
df['date_added'] = pd.to_datetime(df['date_added'])  # Convert date added to␣
 ↪datetime
```

```
df['days_to_add'] = (df['date_added'] - df['release_year']).dt.days  #␣
 ↪Calculate the difference in days

df['days_to_add']
```

Days Taken for Movies to be Added to Netflix After Release:

[137]:
```
0        633
1        266
1        266
1        266
1        266
        ...
8806    1521
8806    1521
8806    1521
8806    1521
8806    1521
Name: days_to_add, Length: 186171, dtype: int64
```

[140]:
```
# Step 2: Find the mode of the differences

mode_days_to_add = df['days_to_add'].mode()[0]
print("The Most common number of days taken for a movie to be added to Netflix␣
 ↪after its release:", mode_days_to_add)
```

The Most common number of days taken for a movie to be added to Netflix after its release: 243

[157]:
```
df.to_csv("Netflix_Updated.csv",index=False)
```