As Netflix operates in a competitive streaming market, it is facing issues with keeping subscribers and standing out with its content. With many alternatives offering different options, Netflix needs to find ways to reduce the chances of people leaving. The main challenge is to come up with strategies that improve user engagement through personalized content suggestions.

Loading the Dataset

In [1]:
```
!gdown 1vr10Pug_YjASBxENTY7jKURi40TmzOZW
```

```
Downloading...
From: https://drive.google.com/uc?id=1vr10Pug_YjASBxENTY7jKURi40TmzOZW
To: /content/netflix.csv

  0% 0.00/3.40M [00:00<?, ?B/s]
 46% 1.57M/3.40M [00:00<00:00, 15.0MB/s]
100% 3.40M/3.40M [00:00<00:00, 26.9MB/s]
```

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

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

Out[3]:

	show_id	type	title	director	cast	country	date_added	r
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	
...	...	...	...	...	...	...	...	
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy	United States	January 11, 2020	

	show_id	type	title	director	cast	country	date_added	r
					Chase, Kate Ma...			
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	

8807 rows × 12 columns

In [4]:
```
df.shape
```

Out[4]:
```
(8807, 12)
```

In [5]:
```
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
```

In [6]:
```
df.index
```

Out[6]:
```
RangeIndex(start=0, stop=8807, step=1)
```

In [8]:
```
df.describe()
```

Out[8]:

	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

In [9]:
```
df.describe(include = "all")
```

Out[9]:

	show_id	type	title	director	cast	country	date_added
count	8807	8807	8807	6173	7982	7976	8797
unique	8807	2	8807	4528	7692	748	1767
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020
freq	1	6131	1	19	19	2818	109
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Checking for any Null Values present or Not

In [10]:
```
df.isnull()
```

Out[10]:

	show_id	type	title	director	cast	country	date_added	release_year
0	False	False	False	False	True	False	False	False
1	False	False	False	True	False	False	False	False
2	False	False	False	False	False	True	False	False
3	False	False	False	True	True	True	False	False
4	False	False	False	True	False	False	False	False
...	...	...	...	...	...	...	...	...
8802	False	False	False	False	False	False	False	False
8803	False	False	False	True	True	True	False	False
8804	False	False	False	False	False	False	False	False
8805	False	False	False	False	False	False	False	False
8806	False	False	False	False	False	False	False	False

8807 rows × 12 columns

In [11]:
```
df.isnull().sum()
```

Out[11]:

	0
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 columns director, cast, country, date_added, rating, and duration contain missing values.

Check for any duplicated rows are present or not

In [12]:
```
df.duplicated().value_counts()
```

Out[12]:

	count
False	8807

dtype: int64

Major Concerns in the data

- Missing Values
- Nested Values
- Date Formate
- Units present in the duration column

Fix the data

In [13]:
```
# we are not aware of any NLP-related resources to get information for the c

df.drop(columns = "description" , inplace = True)
```

In [14]:
```
df.head()
```

Out[14]:

	show_id	type	title	director	cast	country	date_added	releas
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	

Dividing the dataframe apart

- Df1 : Title and Cast

In [15]:
```
df1 = df[["title" , "cast"]]
```

In [16]:
```
df1.head()
```

Out[16]:

	title	cast
0	Dick Johnson Is Dead	NaN
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...
2	Ganglands	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...
3	Jailbirds New Orleans	NaN
4	Kota Factory	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...

In [17]:
```
# Checking for the null values in those 2 cols
df1.isnull().sum()
```

Out[17]:

	0
title	0
cast	825

dtype: int64

In [18]:
```
# cast column contains null values.

# Those null values are being filed as Not Mentioned.

# I don't want to remove those rows or introduce any bias into the cast colu
```

In [19]:
```
df1["cast"] = df1["cast"].fillna("Not Mentioned")
```

```
<ipython-input-19-04d3b7eabbe2>:1: 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-a-copy
  df1["cast"] = df1["cast"].fillna("Not Mentioned")
```

In [20]:
```
df1.isnull().sum()
```

Out[20]:

	0
title	0
cast	0

dtype: int64

In [21]:
```
# Let's explode the data
```

In [22]:
```
df1["cast"] = df1["cast"].str.split(",")
```

```
<ipython-input-22-70f587058fd6>:1: 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-a-copy
  df1["cast"] = df1["cast"].str.split(",")
```

In [23]:
```
df1 = df1.explode("cast")
```

In [24]:
```
df1.head(25)
```

Out[24]:

	title	cast
0	Dick Johnson Is Dead	Not Mentioned
1	Blood & Water	Ama Qamata
1	Blood & Water	Khosi Ngema
1	Blood & Water	Gail Mabalane
1	Blood & Water	Thabang Molaba
1	Blood & Water	Dillon Windvogel
1	Blood & Water	Natasha Thahane
1	Blood & Water	Arno Greeff
1	Blood & Water	Xolile Tshabalala
1	Blood & Water	Getmore Sithole
1	Blood & Water	Cindy Mahlangu
1	Blood & Water	Ryle De Morny
1	Blood & Water	Greteli Fincham
1	Blood & Water	Sello Maake Ka-Ncube
1	Blood & Water	Odwa Gwanya
1	Blood & Water	Mekaila Mathys
1	Blood & Water	Sandi Schultz
1	Blood & Water	Duane Williams
1	Blood & Water	Shamilla Miller
1	Blood & Water	Patrick Mofokeng
2	Ganglands	Sami Bouajila
2	Ganglands	Tracy Gotoas
2	Ganglands	Samuel Jouy
2	Ganglands	Nabiha Akkari
2	Ganglands	Sofia Lesaffre

- Df2 : Title and director

In [25]:
```
df2 = df[["title" , "director"]]
```

In [26]:
```
df2.head()
```

Out[26]:

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

In [27]:
```
df2.isnull().sum()
```

Out[27]:

	0
title	0
director	2634

dtype: int64

In [28]:
```
# We may also use mode to fill in the missing data, but in this case, I beli

# In categorical columns, we must only use mode; however, I don't want to in
```

In [29]:
```
df2["director"] = df2["director"].fillna("Not Mentioned")
```

```
<ipython-input-29-fd1ee436f140>:1: 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-a-copy
  df2["director"] = df2["director"].fillna("Not Mentioned")
```

In [30]:
```
# Exploding the columns
```

In [31]:
```
df2["director"] = df2["director"].str.split(",")
```

```
<ipython-input-31-49e21a1bb9c9>:1: 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-a-copy
  df2["director"] = df2["director"].str.split(",")
```

In [32]:
```
df2 = df2.explode("director")
```

In [33]:
```
df2.head(10)
```

Out[33]:

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	Not Mentioned
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	Not Mentioned
4	Kota Factory	Not Mentioned
5	Midnight Mass	Mike Flanagan
6	My Little Pony: A New Generation	Robert Cullen
6	My Little Pony: A New Generation	José Luis Ucha
7	Sankofa	Haile Gerima
8	The Great British Baking Show	Andy Devonshire

- Df3 : Title and Country

In [34]:
```
df3 = df[["title","country"]]
```

In [35]:
```
df3.head()
```

Out[35]:

	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	NaN
3	Jailbirds New Orleans	NaN
4	Kota Factory	India

In [36]:
```
df3.isnull().sum()
```

Out[36]:

	0
title	0
country	831

dtype: int64

In [37]:
```
df3["country"].mode()[0]
```

Out[37]:
```
'United States'
```

In [38]:
```
# Let's fill the missing values with mode
```

In [39]:
```
df3["country"] = df3["country"].fillna(df["country"].mode()[0])
```

```
<ipython-input-39-02561a1638b4>:1: 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-a-copy
  df3["country"] = df3["country"].fillna(df["country"].mode()[0])
```

In [40]:
```
# Let's explode the data
```

In [41]:
```
df3["country"] = df3["country"].str.split(",")
```

```
<ipython-input-41-bb011acba023>:1: 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-a-copy
  df3["country"] = df3["country"].str.split(",")
```

In [42]:
```
df3 = df3.explode("country")
```

In [43]:
```
df3.head(10)
```

Out[43]:

	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	United States
3	Jailbirds New Orleans	United States
4	Kota Factory	India
5	Midnight Mass	United States
6	My Little Pony: A New Generation	United States
7	Sankofa	United States
7	Sankofa	Ghana
7	Sankofa	Burkina Faso

- Df4 : Title and listed_in

In [44]:
```
df4 = df[["title","listed_in"]]
```

In [45]:
```
df4.head()
```

Out[45]:

	title	listed_in
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows, TV Dramas, TV Mysteries
2	Ganglands	Crime TV Shows, International TV Shows, TV Act...
3	Jailbirds New Orleans	Docuseries, Reality TV
4	Kota Factory	International TV Shows, Romantic TV Shows, TV ...

In [46]:
```
df4.isnull().sum()
```

Out[46]:

	0
title	0
listed_in	0

dtype: int64

In [47]:
```
# There are no null values so we can directly explode the data
```

In [48]:
```
df4["listed_in"] = df4["listed_in"].str.split(",")
```

```
<ipython-input-48-70a29f76871e>:1: 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-a-copy
  df4["listed_in"] = df4["listed_in"].str.split(",")
```

In [49]:
```
df4 = df4.explode("listed_in")
```

In [50]:
```
df4.head()
```

Out[50]:

	title	listed_in
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows
1	Blood & Water	TV Dramas
1	Blood & Water	TV Mysteries
2	Ganglands	Crime TV Shows

Merging all 4 dataframes together

In [51]:
```
df_final = df1.merge(df2, on = "title" , how = "inner")
```

In [52]:
```
df_final.head()
```

Out[52]:

	title	cast	director
0	Dick Johnson Is Dead	Not Mentioned	Kirsten Johnson
1	Blood & Water	Ama Qamata	Not Mentioned
2	Blood & Water	Khosi Ngema	Not Mentioned
3	Blood & Water	Gail Mabalane	Not Mentioned
4	Blood & Water	Thabang Molaba	Not Mentioned

In [53]:
```
# merging the above dataframe with df3
```

In [54]:
```
df_final = df_final.merge(df3 , on = "title" , how = "inner")
```

In [55]:
```
df_final.head()
```

Out[55]:

	title	cast	director	country
0	Dick Johnson Is Dead	Not Mentioned	Kirsten Johnson	United States
1	Blood & Water	Ama Qamata	Not Mentioned	South Africa
2	Blood & Water	Khosi Ngema	Not Mentioned	South Africa
3	Blood & Water	Gail Mabalane	Not Mentioned	South Africa
4	Blood & Water	Thabang Molaba	Not Mentioned	South Africa

In [56]:
```
df_final = df_final.merge(df4 , on = "title" , how = "inner")
```

In [57]:
```
df_final.head()
```

Out[57]:

	title	cast	director	country	listed_in
0	Dick Johnson Is Dead	Not Mentioned	Kirsten Johnson	United States	Documentaries
1	Blood & Water	Ama Qamata	Not Mentioned	South Africa	International TV Shows
2	Blood & Water	Ama Qamata	Not Mentioned	South Africa	TV Dramas
3	Blood & Water	Ama Qamata	Not Mentioned	South Africa	TV Mysteries
4	Blood & Water	Khosi Ngema	Not Mentioned	South Africa	International TV Shows

Merging df_final with remaining cols in the original dataframe

In [58]:
```
df_original = df[["show_id","type","title","date_added","release_year","rati
```

In [59]:
```
df_original
```

Out[59]:

	show_id	type	title	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons
2	s3	TV Show	Ganglands	September 24, 2021	2021	TV-MA	1 Season
3	s4	TV Show	Jailbirds New Orleans	September 24, 2021	2021	TV-MA	1 Season
4	s5	TV Show	Kota Factory	September 24, 2021	2021	TV-MA	2 Seasons
...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	November 20, 2019	2007	R	158 min
8803	s8804	TV Show	Zombie Dumb	July 1, 2019	2018	TV-Y7	2 Seasons
8804	s8805	Movie	Zombieland	November 1, 2019	2009	R	88 min
8805	s8806	Movie	Zoom	January 11, 2020	2006	PG	88 min
8806	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min

8807 rows × 7 columns

In [60]:
```
df_new = df_original.merge(df_final , on = "title" , how = "inner")
```

In [61]:
```
df_new
```

Out[61]:

	show_id	type	title	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons
2	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons
3	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons
4	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons
...	...	...	...	...	...	...	...
202060	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min
202061	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min
202062	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min
202063	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min
202064	s8807	Movie	Zubaan	March 2, 2019	2015	TV-14	111 min

202065 rows × 11 columns

In [62]:
```
df_new.isnull().sum()
```

Out[62]:

	0
show_id	0
type	0
title	0
date_added	158
release_year	0
rating	67
duration	3
cast	0
director	0
country	0
listed_in	0

dtype: int64

In [63]:
```
df_new["date_added"].mode()[0]
```

Out[63]:
```
'January 1, 2020'
```

In [64]:
```
df_new["date_added"] = df_new["date_added"].fillna(df_new["date_added"].mode
```

In [65]:
```
df_new["date_added"].isnull().sum()
```

Out[65]:
```
0
```

In [66]:
```
df_new["rating"] = df_new["rating"].fillna(df_new["rating"].mode()[0])
```

In [67]:
```
df_new["rating"].isnull().sum()
```

Out[67]:
```
0
```

In [68]:
```
# i want to drop those 3 rows where duration is null
```

In [69]:
```
df_new = df_new.dropna()
```

In [70]:
```
df_new.isnull().sum()
```

Out[70]:

	0
show_id	0
type	0
title	0
date_added	0
release_year	0
rating	0
duration	0
cast	0
director	0
country	0
listed_in	0

dtype: int64

In [71]:
```
df_new = df_new[~df_new["rating"].str.contains("min")]
```

In [72]:
```
df_new["rating"].unique()
```

Out[72]:
```
array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
       'TV-G', 'G', 'NC-17', 'NR', 'TV-Y7-FV', 'UR'], dtype=object)
```

In [73]:
```
df_new.head()
```

Out[73]:

	show_id	type	title	date_added	release_year	rating	duration	c
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	Mentio
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	A Qam
2	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	A Qam
3	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	A Qam
4	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	Kh Nge

In [74]:
```
df_new.shape
```

Out[74]:
```
(202062, 11)
```

In [75]:
```
# Let's work on duration column to ge the first part of the string
```

In [76]:
```
df_new["duration"] = df_new["duration"].str.split(" ")
```

In [77]:
```
df_new["duration"] = df_new['duration'].apply(lambda x: int(x[0]))
```

In [78]:
```
df_new.head()
```

Out[78]:

	show_id	type	title	date_added	release_year	rating	duration	c
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90	Mentio
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2	A Qam
2	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2	A Qam
3	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2	A Qam
4	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2	Kh Nge

In [79]:
```
# Now let's work on the date_added column
```

In [80]:
```
df_new["date_added"].head()
```

Out[80]:

	date_added
0	September 25, 2021
1	September 24, 2021
2	September 24, 2021
3	September 24, 2021
4	September 24, 2021

dtype: object

In [81]:
```
df_new["date_added"] = pd.to_datetime(df_new["date_added"].str.strip())
```

In [82]:
```
df_new.head()
```

Out[82]:

	show_id	type	title	date_added	release_year	rating	duration	c
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90	Mentio
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	A Qam
2	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	A Qam
3	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	A Qam
4	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	Kh Nge

In [83]:
```
df_new.dtypes
```

Out[83]:

	0
show_id	object
type	object
title	object
date_added	datetime64[ns]
release_year	int64
rating	object
duration	int64
cast	object
director	object
country	object
listed_in	object

dtype: object

In [84]:
```
df_backup = df_new.copy()
```

In [85]:
```
# The data is completely cleaned now now let's get insights from the data
```

Insights

In [86]:
```
# no .of movies and Tv shows in the data
```

- **Netflix has more number of movies than TV Shows**

In [200…
```
unique_counts = df_new.groupby(by = "type")["title"].nunique()
unique_counts
```

Out[200…

type	title
Movie	6128
TV Show	2676

dtype: int64

In [220…

In [89]:
```
# No. of movies every director has directed
# In this way we can get the lead director
```

- **Rajiv Chilaka is the director who directed most number of movies and TV shows**

In [90]:
```
lead_director = df_new.groupby(by = "director")["title"].nunique().sort_valu
```

In [91]:
```
lead_director = lead_director[lead_director["director"] != "Not Mentioned"]
```

In [92]:
```
lead_director
```

Out[92]:

	director	title
1	Rajiv Chilaka	22
2	Raúl Campos	18
3	Jan Suter	18
4	Suhas Kadav	16
5	Marcus Raboy	16
...	...	...
5115	J. Davis	1
5116	J. Lee Thompson	1
5117	J. Michael Long	1
5118	Smriti Keshari	1
5119	Joaquín Mazón	1

5119 rows × 2 columns

In [93]:
```
# Most popular cast
# In this way we can get the actor who acted in most number of movies
```

- **Anupam Kher is the actor who acted in most number of movies**

In [94]:
```
popular_cast = df_new.groupby(by = "cast")["title"].nunique().sort_values(as
```

In [95]:
```
popular_cast = popular_cast[popular_cast["cast"] != "Not Mentioned"]
```

In [96]:
```
popular_cast
```

Out[96]:

	cast	title
1	Anupam Kher	39
2	Rupa Bhimani	31
3	Takahiro Sakurai	30
4	Julie Tejwani	28
5	Om Puri	27
...	...	...
39291	João Côrtes	1
39292	João Assunção	1
39293	Joziah Lagonoy	1
39294	Jozef Gjura	1
39295	Ṣọpẹ́ Dìrísù	1

39295 rows × 2 columns

In [97]:
```
# Most popular Director and Cast pair
```

- **rajiv chilaka and julie tejwani is the most popular director and cast pair**

In [98]:
```
df_new['cast'] = df_new['cast'].str.strip().str.lower()
df_new['director'] = df_new['director'].str.strip().str.lower()
```

In [99]:
```
pop_dir_cast = df_new.groupby(by = ["director","cast"])["title"].nunique().s
```

In [100…
```
pop_dir_cast = pop_dir_cast[pop_dir_cast["director"] != "not mentioned"]
```

In [101…
```
pop_dir_cast
```

Out[101…

	director	cast	title
2	rajiv chilaka	julie tejwani	19
3	rajiv chilaka	rajesh kava	19
4	rajiv chilaka	jigna bhardwaj	18
5	rajiv chilaka	rupa bhimani	18
9	rajiv chilaka	vatsal dubey	16
...	...	...	...
62721	joseduardo giordano	khristian clausen	1
62722	joseduardo giordano	leonardo ortizgris	1
62723	joseduardo giordano	marco méndez	1
62724	joseduardo giordano	mónica dionne	1
62725	şenol sönmez	özgür emre yıldırım	1

48722 rows × 3 columns

In [102…
```
# Most popular Genre
```

In [103…
```
df_new["listed_in"].value_counts().reset_index()
```

Out[103…

	listed_in	count
0	International Movies	27141
1	Dramas	19657
2	Comedies	13894
3	Action & Adventure	12216
4	Dramas	10149
...	...	...
68	Stand-Up Comedy	24
69	Romantic Movies	20
70	TV Sci-Fi & Fantasy	7
71	LGBTQ Movies	5
72	Sports Movies	3

73 rows × 2 columns

In [104…
```
# Most titles got published in which genere
```

- **Most of the movie titles got published in International Movies**

- Next most of the titles got published in Dramas

In [105…
```
pop_listed_in = df_new.groupby(by = "listed_in")["title"].nunique().sort_val
```

In [106…
```
pop_listed_in
```

Out[106…

	listed_in	title
0	International Movies	2624
1	Dramas	1600
2	Comedies	1210
3	Action & Adventure	859
4	Documentaries	829
...	...	...
68	Romantic Movies	3
69	Spanish-Language TV Shows	2
70	TV Sci-Fi & Fantasy	1
71	LGBTQ Movies	1
72	Sports Movies	1

73 rows × 2 columns

In [107…
```
# Every year how many movies and Tv Shows  had released
```

- **In the year of 2018 most of the movies got released in Netflix**

In [108…
```
df_new.groupby(by = "release_year")["title"].nunique().sort_values(ascending
```

Out[108…

	release_year	title
0	2018	1147
1	2017	1031
2	2019	1030
3	2020	953
4	2016	902
...	...	...
69	1959	1
70	1961	1
71	1947	1
72	1966	1
73	1925	1

74 rows × 2 columns

In [109…

```
# Number of movies released in each rating category
```

- **Most of the movies have rating of TV - MA**

In [110…

```
df_new.groupby(by = "rating")["title"].nunique().sort_values(ascending = Fal
```

Out[110…

	rating	title
0	TV-MA	3211
1	TV-14	2160
2	TV-PG	863
3	R	799
4	PG-13	490
5	TV-Y7	334
6	TV-Y	307
7	PG	287
8	TV-G	220
9	NR	80
10	G	41
11	TV-Y7-FV	6
12	NC-17	3
13	UR	3

In [202…

```
df_new.groupby(by = ["type" , "rating"])["title"].nunique().sort_values(asce
```

Out[202…

	type	rating	title
0	Movie	TV-MA	2064
1	Movie	TV-14	1427
2	TV Show	TV-MA	1147
3	Movie	R	797
4	TV Show	TV-14	733
5	Movie	TV-PG	540
6	Movie	PG-13	490
7	TV Show	TV-PG	323
8	Movie	PG	287
9	TV Show	TV-Y7	195
10	TV Show	TV-Y	176
11	Movie	TV-Y7	139
12	Movie	TV-Y	131
13	Movie	TV-G	126
14	TV Show	TV-G	94
15	Movie	NR	75
16	Movie	G	41
17	TV Show	NR	5
18	Movie	TV-Y7-FV	5
19	Movie	UR	3
20	Movie	NC-17	3
21	TV Show	R	2
22	TV Show	TV-Y7-FV	1

In [111…

```
# No of shows avilable in each country
```

- **Most of the movies are released in United States**

In [112…

```
df_new.groupby(by = "country")["title"].nunique().sort_values(ascending = Fa
```

Out[112…

	country	title
0	United States	4039
1	India	1008
2	United Kingdom	628
3	United States	479
4	Canada	271
...	...	...
192	Cameroon	1
193	Lithuania	1
194	Paraguay	1
195	Liechtenstein	1
196	Zimbabwe	1

197 rows × 2 columns

In [113…

```
# Country wise number of people acted
```

- **Most of the cast members are in united States only**

In [114…

```
df_new.groupby(by = ["country"])["cast"].nunique().sort_values(ascending = F
```

Out[114…

	country	cast
0	United States	16325
1	India	3710
2	United States	3357
3	United Kingdom	2905
4	Canada	1609
...	...	...
192	Palestine	1
193	Panama	1
194	Samoa	1
195	Afghanistan	1
196	Guatemala	1

197 rows × 2 columns

In [115…
```
# country wise who is the most popular actor
```

- **In india Anupam Kher is the most popular actor**

In [116…
```
country_pop_cast = df_new.groupby(by = ["country","cast"])["title"].nunique(
```

In [117…
```
country_pop_cast = country_pop_cast[country_pop_cast["cast"] != "not mentior
```

In [118…
```
country_pop_cast
```

Out[118…

	country	cast	title
3	India	anupam kher	40
4	India	shah rukh khan	33
6	India	naseeruddin shah	31
7	India	akshay kumar	29
8	Japan	takahiro sakurai	29
...	...	...	...
56755	China	sonu sood	1
56756	China	stephen fry	1
56757	China	steven seagal	1
56758	China	su ke	1
56759	Zimbabwe	zihlo	1

56665 rows × 3 columns

In [119…
```
# country wise most popular genere
```

- **In India most of the titles got published in International movies only and in USA most of the titles got published in Dramas**

In [120…
```
df_new.groupby(by = ["country","listed_in"])["title"].nunique().sort_values(
```

Out[120…

	country	listed_in	title
0	India	International Movies	807
1	United States	Dramas	501
2	United States	Documentaries	499
3	United States	Comedies	436
4	United States	Children & Family Movies	425
...	...	...	...
2436	Thailand	Action & Adventure	1
2437	Thailand	International Movies	1
2438	Taiwan	International Movies	1
2439	Taiwan	Documentaries	1
2440	Zimbabwe	Comedies	1

2441 rows × 3 columns

In [121…
```
# country wise most popular director
```

- **Rajiv Chilaka is the most popular director in United States**

In [122…
```
country_pop_dir = df_new.groupby(by = ["country" , "director"])["title"].nur
```

In [123…
```
country_pop_dir = country_pop_dir[country_pop_dir["director"] != "not mentic
```

In [124…
```
country_pop_dir
```

Out[124…

	country	director	title
25	United States	rajiv chilaka	17
28	United States	marcus raboy	16
29	United States	jay karas	15
30	United States	suhas kadav	15
32	Philippines	cathy garcia-molina	13
...	...	...	...
7239	Denmark	brian de palma	1
7240	Denmark	bille august	1
7241	Denmark	andrew tan	1
7242	Denmark	amalie næsby fick	1
7243	Zimbabwe	tomas brickhill	1

7143 rows × 3 columns

In [125…
```
# When is the director first title and last title got released
```

In [126…
```
first_and_last_movie = df_new.groupby(by = "director")["release_year"].agg([
```

In [127…
```
first_and_last_movie.head()
```

Out[127…

	director	min	max
0	a. l. vijay	2016	2019
1	a. raajdheep	2020	2020
2	a. salaam	1975	1975
3	a.r. murugadoss	2017	2018
4	aadish keluskar	2018	2018

In [128…
```
first_and_last_movie["duration"] = first_and_last_movie["max"] - first_and_l
```

In [129…
```
first_and_last_movie.sort_values(by = "duration" , ascending = False)
```

Out[129…

	director	min	max	duration
3390	not mentioned	1925	2021	96
2939	martin scorsese	1967	2019	52
4937	youssef chahine	1954	1999	45
659	brian de palma	1976	2019	43
4421	steven spielberg	1975	2016	41
...	...	...	...	...
2170	john scheinfeld	2016	2016	0
461	athiyan athirai	2019	2019	0
2172	john smithson	1994	1994	0
2173	john stephenson	2013	2013	0
2494	khalid kashogi	2008	2008	0

4989 rows × 4 columns

In [130…
```
# Find the number of unique titles directed by each director in each genre
```

- **Most of the rajiv chilaka movies got listed in Children & Family Movies**

In [131…
```
dir_genre = df_new.groupby(by = ["director" ,"listed_in"])["title"].nunique(
```

In [132…
```
dir_genre = dir_genre[dir_genre["director"] != "not mentioned"]
```

In [133…
```
dir_genre
```

Out[133…

	director	listed_in	title
35	rajiv chilaka	Children & Family Movies	22
36	jan suter	Stand-Up Comedy	21
38	raúl campos	Stand-Up Comedy	19
41	suhas kadav	Children & Family Movies	16
42	marcus raboy	Stand-Up Comedy	15
...	...	...	...
12420	joey curtis	Independent Movies	1
12421	joey curtis	Sci-Fi & Fantasy	1
12422	joey curtis	Action & Adventure	1
12423	joey kern	Comedies	1
12424	a. l. vijay	Dramas	1

12365 rows × 3 columns

Movies DataFrame

In [134…

```
df_movies = df_new[df_new["type"] == "Movie"]
df_movies.head()
```

Out[134…

	show_id	type	title	date_added	release_year	rating	duration	
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90	n
159	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	
160	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	
161	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	
162	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	

In [135…

```
# Mean Duration of Movies
```

In [136…
```
mean_duration_movies = df_movies["duration"].mean()
mean_duration_movies
```

Out[136…
```
106.84038543251505
```

In [137…
```
# country wise average runtime of the movies
```

- The average duration of the movies in Liechtenstein country is 200.000000
- Compared to other countries it is very much higher

In [138…
```
df_movies.groupby(by = "country")["duration"].mean().sort_values(ascending =
```

Out[138…

	country	duration
0	Liechtenstein	200.000000
1	Soviet Union	162.142857
2	Montenegro	157.000000
3	Croatia	157.000000
4	West Germany	150.000000
...	...	...
182	Guatemala	67.000000
183	Kazakhstan	67.000000
184	Kenya	54.470588
185	Namibia	29.000000
186	Syria	13.000000

187 rows × 2 columns

In [139…
```
# Genre wise average runtime of movies
```

- Classic movies are the one which is having the highest mean duration

In [140…
```
df_movies.groupby(by = "listed_in")["duration"].mean().sort_values(ascending
```

Out[140…

	listed_in	duration
0	Classic Movies	129.494970
1	Classic Movies	125.363636
2	Dramas	114.808651
3	Action & Adventure	113.166339
4	Cult Movies	113.146476
5	Music & Musicals	112.821713
6	Dramas	112.459226
7	International Movies	112.040861
8	International Movies	110.368421
9	Romantic Movies	110.098404
10	Faith & Spirituality	109.006954
11	Sci-Fi & Fantasy	108.758968
12	Thrillers	108.713672
13	Sci-Fi & Fantasy	107.962963
14	Thrillers	105.373786
15	Children & Family Movies	104.945493
16	Comedies	104.858860
17	Independent Movies	102.886763
18	Sports Movies	101.317408
19	Horror Movies	100.449695
20	LGBTQ Movies	100.292917
21	Cult Movies	99.686391
22	Independent Movies	99.114815
23	LGBTQ Movies	99.000000
24	Anime Features	98.486275
25	Horror Movies	98.443081
26	Comedies	96.345205
27	Anime Features	88.910714
28	Documentaries	87.409628
29	Sports Movies	87.000000
30	Children & Family Movies	84.430278
31	Romantic Movies	82.800000
32	Music & Musicals	76.076923

	listed_in	duration
33	Documentaries	75.911290
34	Stand-Up Comedy	74.625000
35	Stand-Up Comedy	68.575581
36	Movies	48.838631

In [141…
```
# Popular month to add movies to the netflix
```

- Popular month to add movies to the netflix is 7th month tht is JULY

In [203…
```
popular_month = df_movies["date_added"].dt.month.value_counts().idxmax()

# count of movies added in the most popular month
popular_month_count = df_movies["date_added"].dt.month.value_counts().max()
```

In [204…
```
popular_month
```

Out[204… 7

In [205…
```
popular_month_count
```

Out[205… 15075

In [206…
```
# No. of Movies released on Netflix in each Genre
```

- **Most Movies are uploaded under International Movies genre**

In [146…
```
df_movies.groupby(by = "listed_in")["title"].nunique().sort_values(ascending
```

Out[146…

	listed_in	title
0	International Movies	2624
1	Dramas	1600
2	Comedies	1210
3	Action & Adventure	859
4	Documentaries	829
5	Dramas	827
6	Independent Movies	736
7	Romantic Movies	613
8	Children & Family Movies	605
9	Thrillers	512
10	Comedies	464
11	Music & Musicals	357
12	Stand-Up Comedy	334
13	Horror Movies	275
14	Sci-Fi & Fantasy	230
15	Sports Movies	218
16	International Movies	128
17	LGBTQ Movies	101
18	Horror Movies	82
19	Classic Movies	80
20	Thrillers	65
21	Faith & Spirituality	65
22	Cult Movies	59
23	Movies	54
24	Anime Features	50
25	Documentaries	40
26	Children & Family Movies	36
27	Classic Movies	36
28	Anime Features	21
29	Independent Movies	20
30	Music & Musicals	18
31	Sci-Fi & Fantasy	13
32	Cult Movies	12

	listed_in	title
33	Stand-Up Comedy	9
34	Romantic Movies	3
35	LGBTQ Movies	1
36	Sports Movies	1

In [147…
```
# Lead actor in movies
```

- Here we get to see that Anupam keher is the lead actor in movies and shah rukh khan is the second most popular lead actor in movies

In [148…
```
movies_lead_actor = df_movies.groupby(by = "cast")["title"].nunique().sort_v
```

In [149…
```
movies_lead_actor = movies_lead_actor[movies_lead_actor["cast"] != "not ment
```

In [150…
```
movies_lead_actor
```

Out[150…

	cast	title
1	anupam kher	42
2	shah rukh khan	35
3	naseeruddin shah	32
4	akshay kumar	30
5	om puri	30
...	...	...
25939	jacob batalon	1
25940	jacob artist	1
25941	jaco muller	1
25942	jaclyn victor	1
25943	ṣọpẹ́ dìrísù	1

25943 rows × 2 columns

In [151…
```
# Let's do same with TV Shows
```

In [152…
```
df_shows = df_new[df_new["type"] == "TV Show"]
df_shows.head()
```

Out[152…

	show_id	type	title	date_added	release_year	rating	duration	cast
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	ama qamata
2	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	ama qamata
3	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	ama qamata
4	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	khosi ngema
5	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	khosi ngema

In [153…
```
# best month to add the tv shows into netflix
```

- 12th month is the best month to add TV shows to netflix

In [154…
```
df_shows["date_added"].dt.month.value_counts().idxmax()
```

Out[154… 12

In [155…
```
# Lead actor in TV shows
```

- takahiro sakurai is the lead actor in TV Shows

In [222…
```
df_shows.groupby(by = "cast")["title"].nunique().sort_values(ascending = Fal
```

Out[222…

	cast	title
0	not mentioned	350
1	takahiro sakurai	25
2	yuki kaji	19
3	junichi suwabe	17
4	daisuke ono	17
...	...	...
14858	iván pellicer	1
14859	iván álvarez de araya	1
14860	iza moreira	1
14861	izan llunas	1
14862	şükrü özyıldız	1

14863 rows × 2 columns

In [157…

```
# No. of TV Shows released on netflix in each genre
```

- Here also we can see that most of the TV shows got published in INternational Movies
- Then after International Movies most of the TV Shows got published in TV Dramas

In [158…

```
df_shows.groupby(by = "listed_in")["title"].nunique().sort_values(ascending
```

Out[158…

	listed_in	title
0	International TV Shows	774
1	TV Dramas	696
2	International TV Shows	577
3	TV Comedies	461
4	Crime TV Shows	399
5	Kids' TV	388
6	Romantic TV Shows	338
7	British TV Shows	253
8	Docuseries	221
9	Anime Series	176
10	Docuseries	174
11	Spanish-Language TV Shows	172
12	Korean TV Shows	151
13	Reality TV	135
14	TV Action & Adventure	128
15	Reality TV	120
16	TV Comedies	120
17	TV Mysteries	98
18	Science & Nature TV	92
19	TV Sci-Fi & Fantasy	83
20	Crime TV Shows	71
21	Teen TV Shows	69
22	TV Dramas	67
23	TV Horror	64
24	Kids' TV	63
25	TV Thrillers	57
26	TV Action & Adventure	40
27	Stand-Up Comedy & Talk Shows	34
28	Romantic TV Shows	32
29	Classic & Cult TV	22
30	Stand-Up Comedy & Talk Shows	22
31	TV Shows	16
32	TV Horror	11

	listed_in	title
33	Classic & Cult TV	6
34	Spanish-Language TV Shows	2
35	TV Sci-Fi & Fantasy	1

In [159…
```
# Mean seasons of TV shows in each genre
```

In [160…
```
avg_tvshow_length = df_shows.groupby(by = "listed_in")["duration"].mean().sc
```

In [161…
```
avg_tvshow_length["duration"] = avg_tvshow_length["duration"].round()
```

In [162…
```
avg_tvshow_length
```

Out[162…

	listed_in	duration
0	Classic & Cult TV	6.0
1	Classic & Cult TV	6.0
2	Stand-Up Comedy & Talk Shows	4.0
3	TV Horror	4.0
4	Romantic TV Shows	3.0
5	TV Sci-Fi & Fantasy	3.0
6	TV Comedies	3.0
7	TV Dramas	3.0
8	TV Sci-Fi & Fantasy	3.0
9	TV Action & Adventure	3.0
10	TV Action & Adventure	3.0
11	Teen TV Shows	3.0
12	TV Mysteries	2.0
13	TV Thrillers	2.0
14	British TV Shows	2.0
15	Reality TV	2.0
16	Kids' TV	2.0
17	TV Comedies	2.0
18	TV Horror	2.0
19	Kids' TV	2.0
20	Crime TV Shows	2.0
21	Crime TV Shows	2.0
22	TV Dramas	2.0
23	Spanish-Language TV Shows	2.0
24	Reality TV	2.0
25	International TV Shows	2.0
26	Stand-Up Comedy & Talk Shows	2.0
27	Anime Series	2.0
28	Docuseries	2.0
29	Docuseries	1.0
30	International TV Shows	1.0
31	Science & Nature TV	1.0
32	Romantic TV Shows	1.0

	listed_in	duration
33	Korean TV Shows	1.0
34	Spanish-Language TV Shows	1.0
35	TV Shows	1.0

In [207…
```
# Director with most number of movies in TV shows
```

In [220…
```
df_shows.groupby(by = "director")["title"].nunique().sort_values(ascending =
```

Out[220…

	director	title
0	not mentioned	2634
1	rajiv chilaka	22
2	jan suter	21
3	raúl campos	19
4	suhas kadav	16
...	...	...
4984	juan antonio de la riva	1
4985	juan camilo pinzon	1
4986	juan carlos medina	1
4987	juan carlos rulfo	1
4988	khalid kashogi	1

4989 rows × 2 columns

Graphical Analysis

In [218…
```
# Total number of  movies and Tv Shows releasaed in last 20 years
```

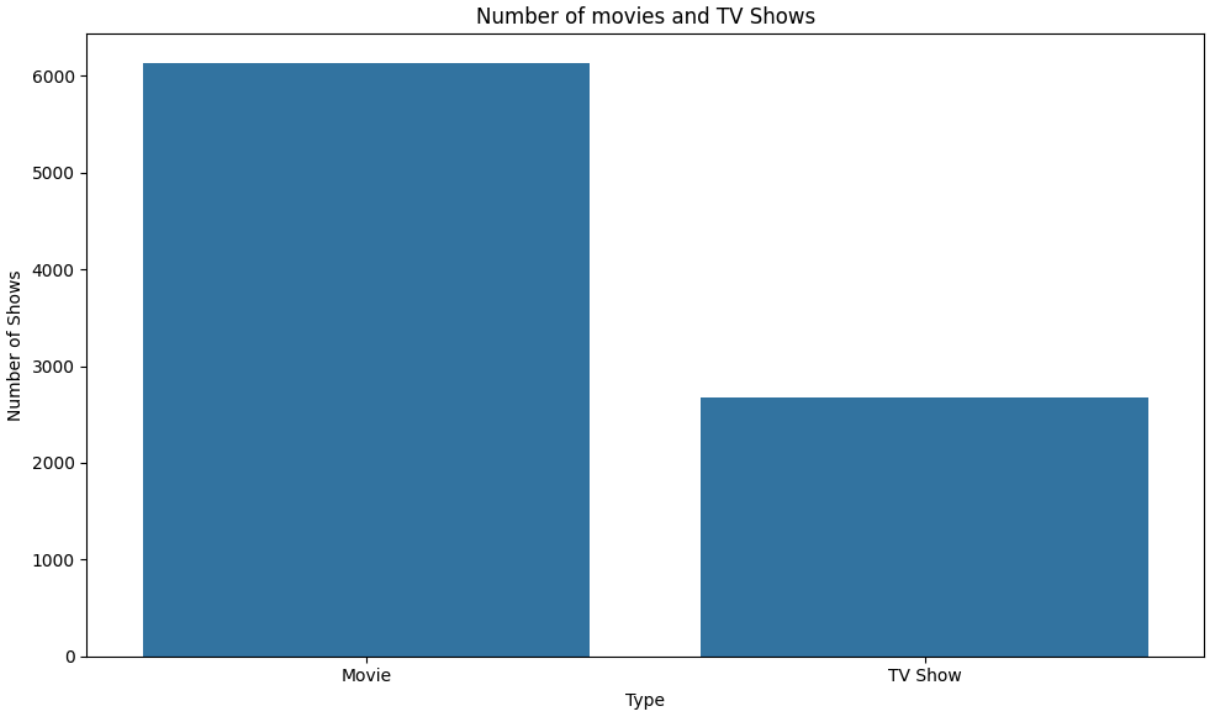
In [219…
```
df_new.groupby(by = "type")["title"].nunique().sort_values(ascending = False
```

Out[219…

	type	title
0	Movie	6128
1	TV Show	2676

In [221…
```
plt.figure(figsize = (10,6))
sns.barplot(data = df_new.groupby(by = "type")["title"].nunique().reset_inde
plt.title("Number of movies and TV Shows")
plt.xlabel("Type")
plt.ylabel("Number of Shows")
```

```
plt.tight_layout()
plt.show()
```



observation : The bar chart shows the number of movies and TV shows. Movies are significantly more numerous than TV shows.

No.of shows in each rating category

In [169…

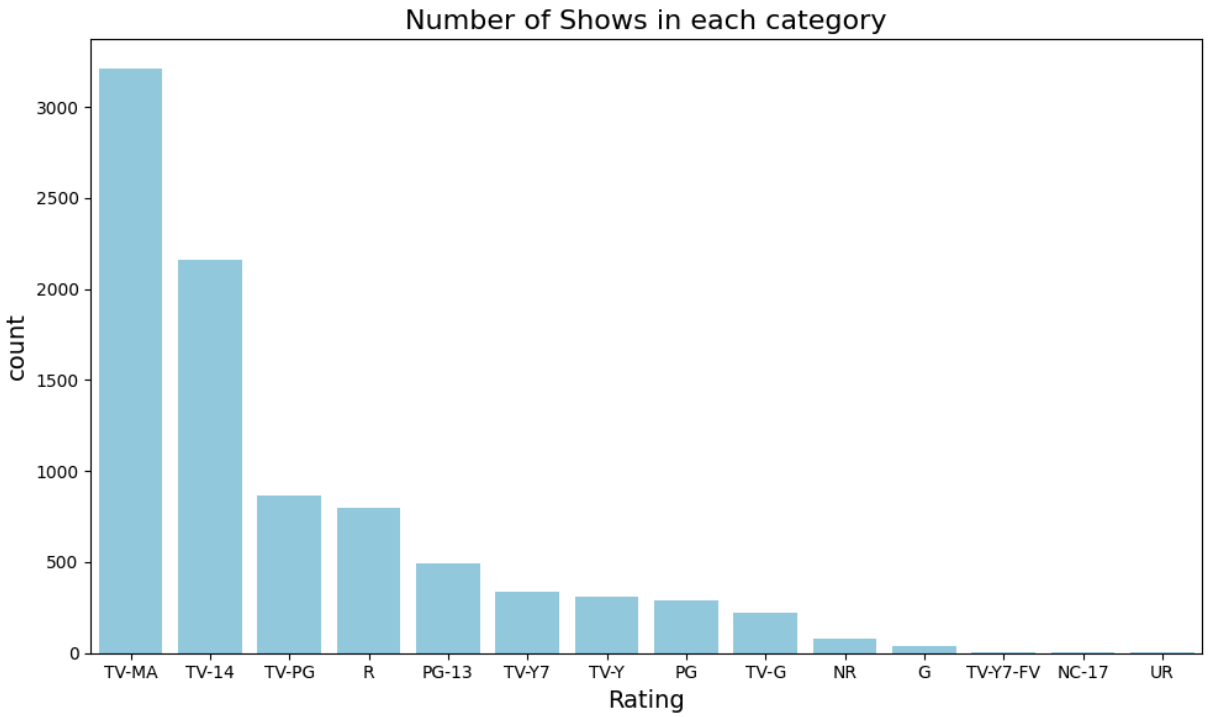
```
rating_category = df_new.groupby(by = "rating")["title"].nunique().sort_valu
rating_category
```

Out[169…

	rating	title
0	TV-MA	3211
1	TV-14	2160
2	TV-PG	863
3	R	799
4	PG-13	490
5	TV-Y7	334
6	TV-Y	307
7	PG	287
8	TV-G	220
9	NR	80
10	G	41
11	TV-Y7-FV	6
12	NC-17	3
13	UR	3

In [170…

```
plt.figure(figsize=(10, 6))
sns.barplot(data=rating_category, x='rating', y='title', color='skyblue')

plt.title('Number of Shows in each category', fontsize=16)
plt.xlabel('Rating', fontsize=14)
plt.ylabel('count', fontsize=14)
plt.tight_layout()

# Show the plot
plt.show()
```



Obeservations : The rating distribution chart reveals a skew towards mature content, with TV-MA, TV-14, and R being the most frequent ratings, indicating a preference for mature audiences. Less common ratings like PG, PG-13, and G show that family-friendly content is rarer. This suggests industry trends are favoring more mature themes

In [171…
```
# No. of shows added to netflix every year
```

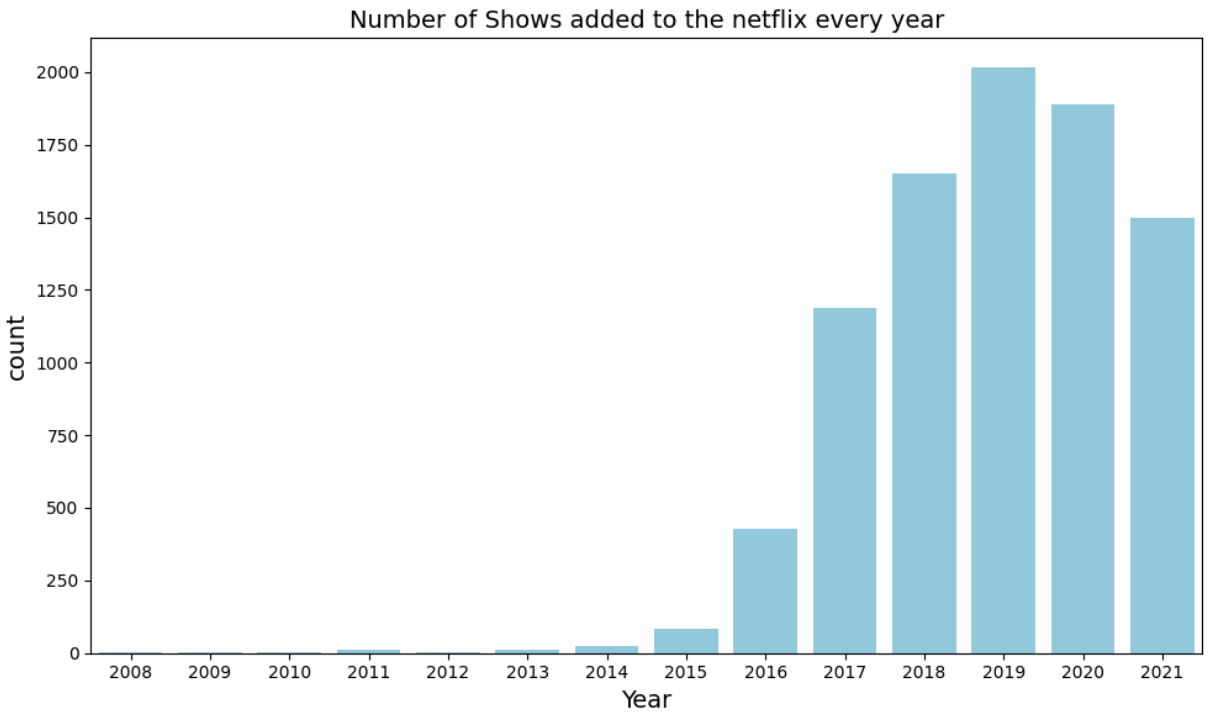
In [172…
```
shows_added_per_year = df_new.groupby(by = df_new["date_added"].dt.year)["ti
shows_added_per_year
```

Out[172…

	date_added	title
0	2019	2016
1	2020	1889
2	2018	1649
3	2021	1498
4	2017	1187
5	2016	427
6	2015	82
7	2014	24
8	2011	13
9	2013	11
10	2012	3
11	2008	2
12	2009	2
13	2010	1

In [173…

```
plt.figure(figsize=(10, 6))
sns.barplot(data=shows_added_per_year, x='date_added', y='title', color='sky

plt.title('Number of Shows added to the netflix every year', fontsize=14)
plt.xlabel('Year', fontsize=14)
plt.ylabel('count', fontsize=14)
plt.tight_layout()

# Show the plot
plt.show()
```

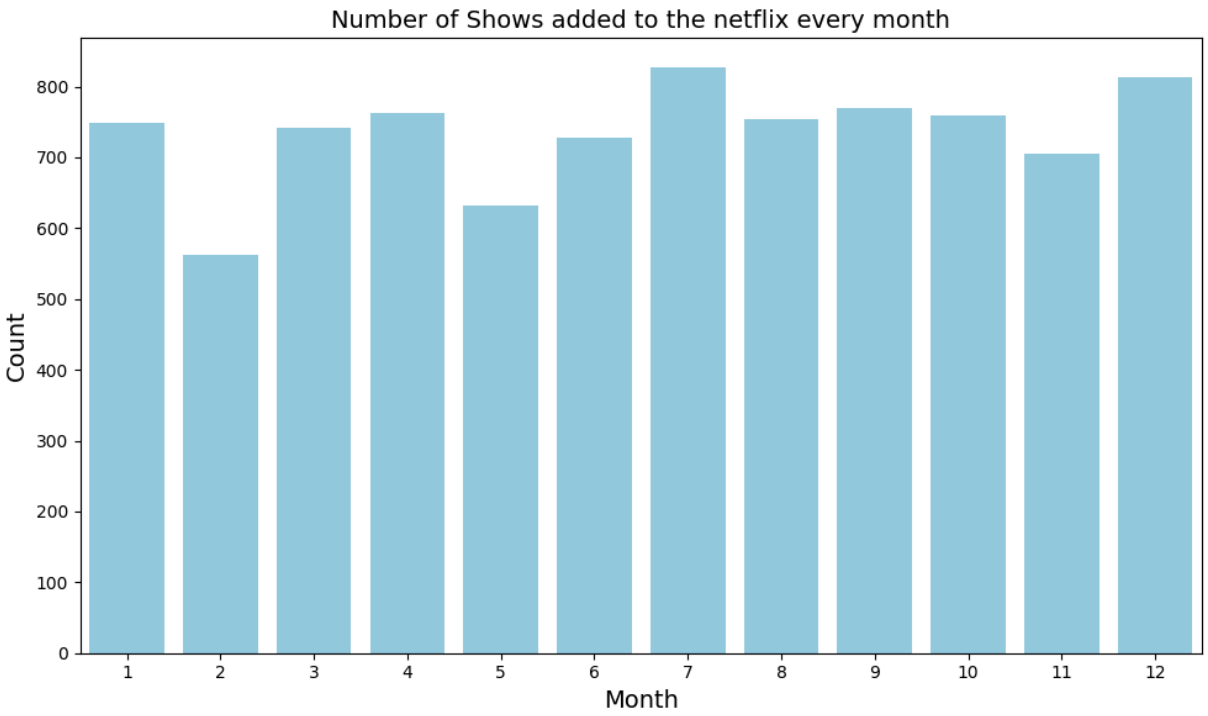


Observations : The Netflix Shows Added chart shows a clear upward trend in the number of shows added over the years, with significant growth between 2018 and 2019, peaking in 2019. After a slight drop in 2020, the numbers rebounded in 2021. This reflects Netflix's strategy to expand its content library, driven by rising audience demand and competition in the streaming industry.

In [174…
```
# No. of shows uploaded on Netflix each Month
```

In [175…
```
shows_added_per_month = df_new.groupby(by = df_new["date_added"].dt.month)["
shows_added_per_month= shows_added_per_month.sort_values(by = "title" , asce
```

In [176…
```
plt.figure(figsize = (10,6))

sns.barplot(data = shows_added_per_month , x = "date_added" , y = "title" ,
plt.title("Number of Shows added to the netflix every month" , fontsize = 14
plt.xlabel("Month" , fontsize = 14)
plt.ylabel("Count" , fontsize = 14)
plt.tight_layout()

plt.show()
```



Observation : The chart shows consistent monthly additions of new Netflix shows, with slight seasonal variations, potentially peaking in months like July and December. Some months have higher additions due to major releases, while others may see fewer. Influencing factors include production schedules, content strategy, viewer demand, and competition. These insights help Netflix optimize content planning, align with viewer expectations, and stay informed about industry trends.

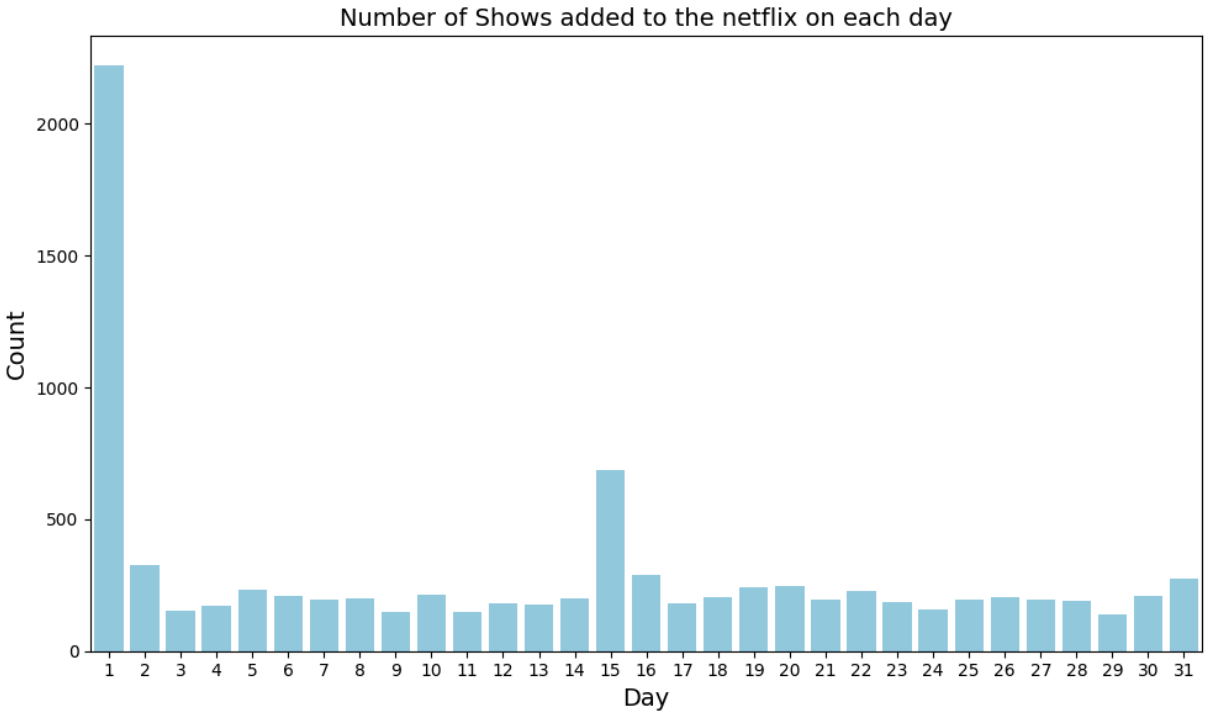
In [177…
```
# No. of shows uploaded on Netflix each Day
```

In [178…
```
shows_on_each_day = df_new.groupby(by = df_new["date_added"].dt.day)["title"
shows_on_each_day
```

Out[178…

	date_added	title
0	1	2222
1	15	686
2	2	325
3	16	288
4	31	274
5	20	249
6	19	243
7	5	231
8	22	230
9	10	214
10	30	210
11	6	210
12	18	207
13	26	206
14	8	201
15	14	198
16	25	197
17	27	195
18	7	194
19	21	193
20	28	190
21	23	184
22	12	181
23	17	180
24	13	175
25	4	174
26	24	159
27	3	151
28	11	149
29	9	147
30	29	141

In [179…
```
plt.figure(figsize = (10,6))

sns.barplot(data = shows_on_each_day , x = "date_added" , y = "title" , colo
plt.title("Number of Shows added to the netflix on each day" , fontsize = 14
plt.xlabel("Day" , fontsize = 14)
plt.ylabel("Count" , fontsize = 14)
plt.tight_layout()

plt.show()
```



Observations : The above chart shows that most of the movies are releasing on first of every month and followed by middile of the month few are releasing

In [ ]:
```
```

In [180…
```
# No. of shows available on Netflix in each Country
```

In [181…
```
country_wise_no_of_movies = df_new.groupby(by = "country")["title"].nunique(
country_wise_no_of_movies
```

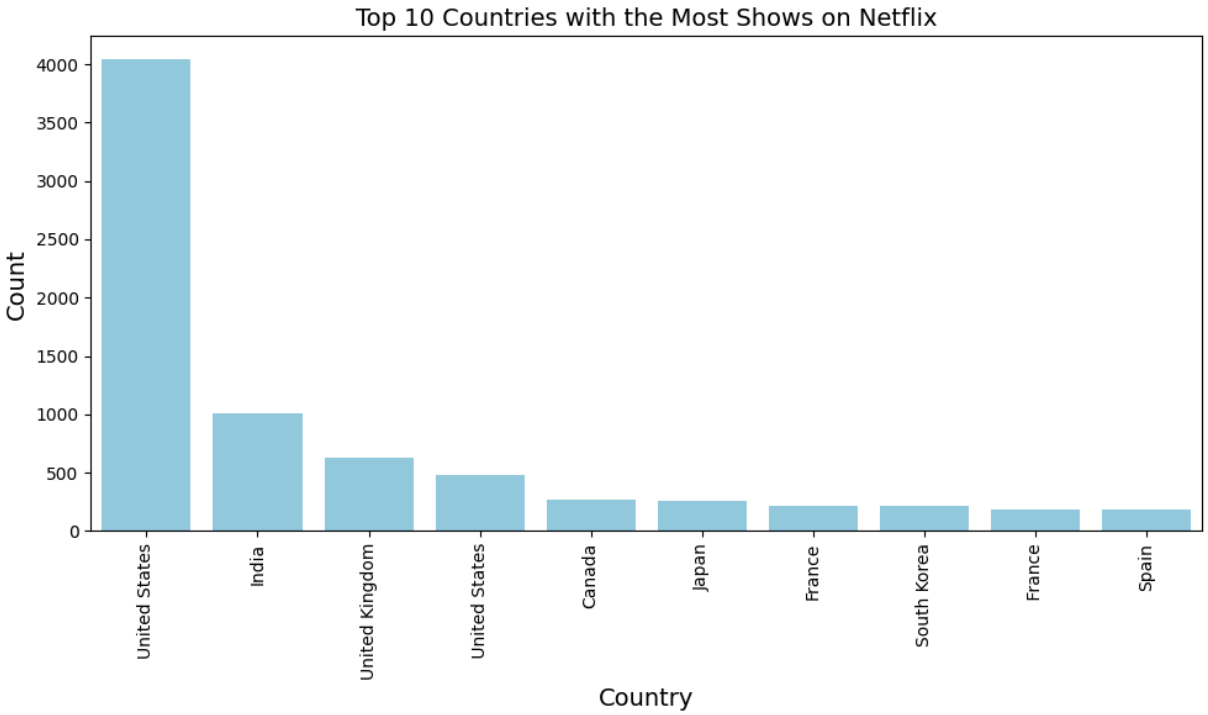
Out[181…

	country	title
0	United States	4039
1	India	1008
2	United Kingdom	628
3	United States	479
4	Canada	271
...	...	...
192	Cameroon	1
193	Lithuania	1
194	Paraguay	1
195	Liechtenstein	1
196	Zimbabwe	1

197 rows × 2 columns

In [182…

```
top_countries = country_wise_no_of_movies.head(10)

plt.figure(figsize=(10,6))
sns.barplot(data=top_countries, x="country", y="title", color="skyblue")
plt.title("Top 10 Countries with the Most Shows on Netflix", fontsize=14)
plt.xlabel("Country", fontsize=14)
plt.ylabel("Count", fontsize=14)
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

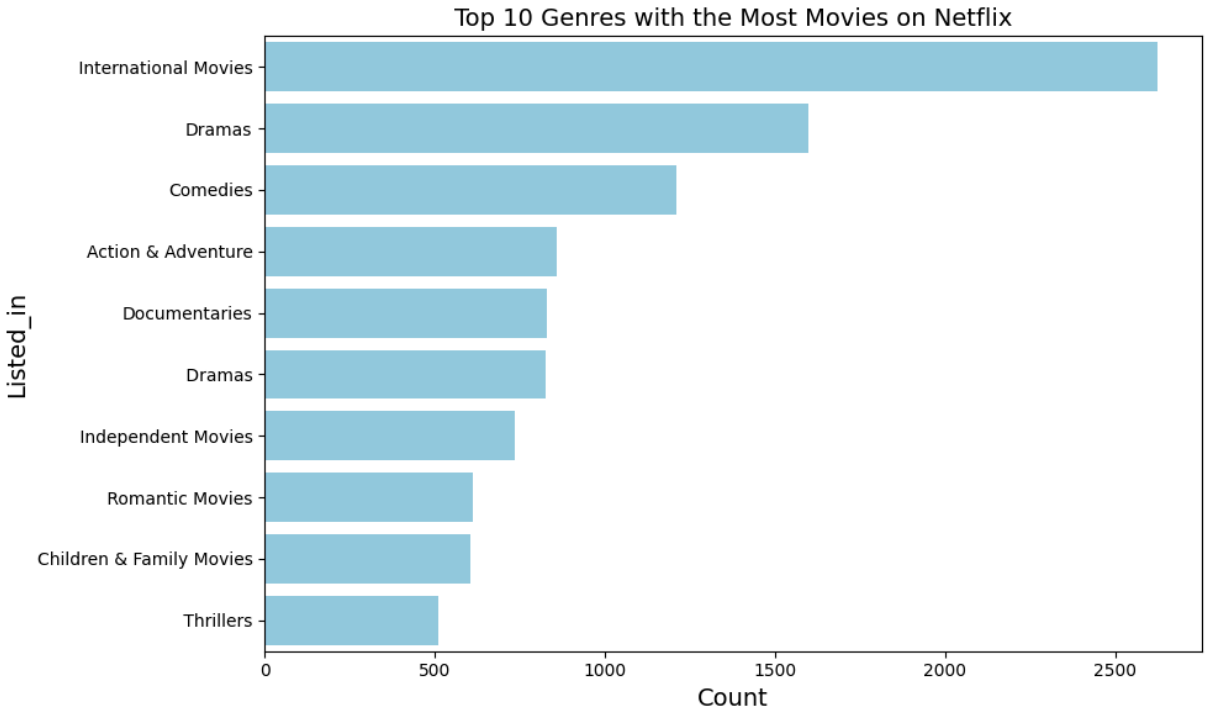


Observations : The above chart displays that United states is having the most number of titles followed by india this shows that most of the directors are releasing movies in united States

In [183…
```
# No. of Movies released on Netflix in each Genre
```

In [184…
```
movies_in_each_genre = df_movies.groupby(by = "listed_in")["title"].nunique(
movies_in_each_genre = movies_in_each_genre.head(10)
```

In [185…
```
plt.figure(figsize = (10,6))

sns.barplot(data = movies_in_each_genre , x = "title" , y = "listed_in" , co
plt.title("Top 10 Genres with the Most Movies on Netflix", fontsize=14)
plt.xlabel("Count", fontsize=14)
plt.ylabel("Listed_in", fontsize=14)
plt.tight_layout()

plt.show()
```

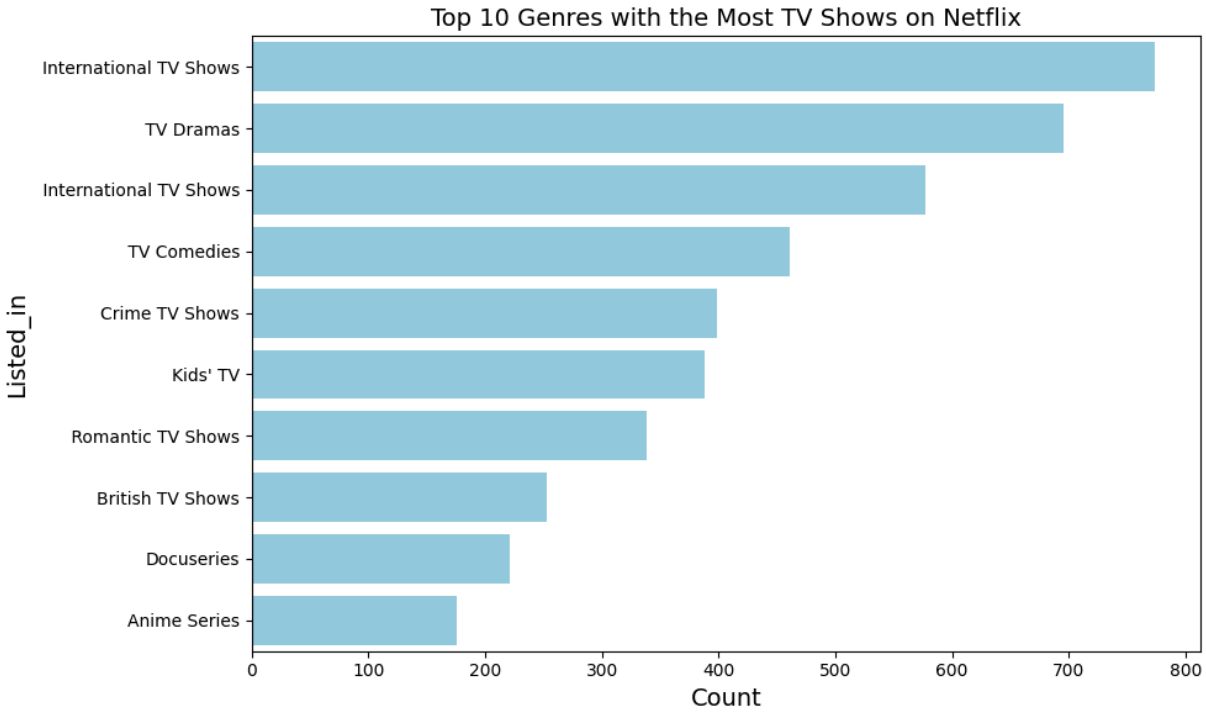


Observations: Most of the Movies in Netflix are releasing in International movies Genre and the second most popular genre is Dramas

In [186…
```
# No. of TV Shows released on Netflix in each Genre
```

In [187…
```
shows_in_each_genre = df_shows.groupby(by = "listed_in")["title"].nunique().
shows_in_each_genre = shows_in_each_genre.head(10)
```

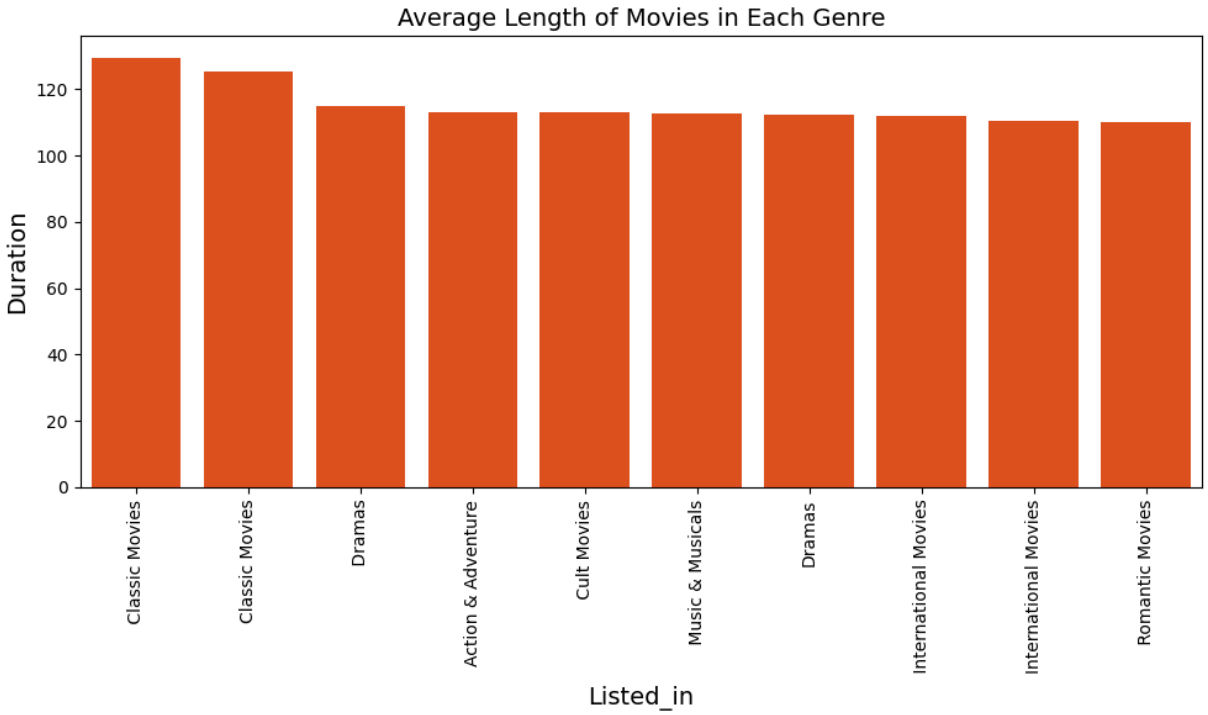
In [188…
```
plt.figure(figsize = (10,6))
sns.barplot(data = shows_in_each_genre , x = "title" , y = "listed_in" , col
plt.title("Top 10 Genres with the Most TV Shows on Netflix", fontsize=14)
plt.xlabel("Count", fontsize=14)
plt.ylabel("Listed_in", fontsize=14)
plt.tight_layout()


plt.show()
```



Observations : According to above plot most of the TV shows are releasing in International TV shows and followed by TV Dramas

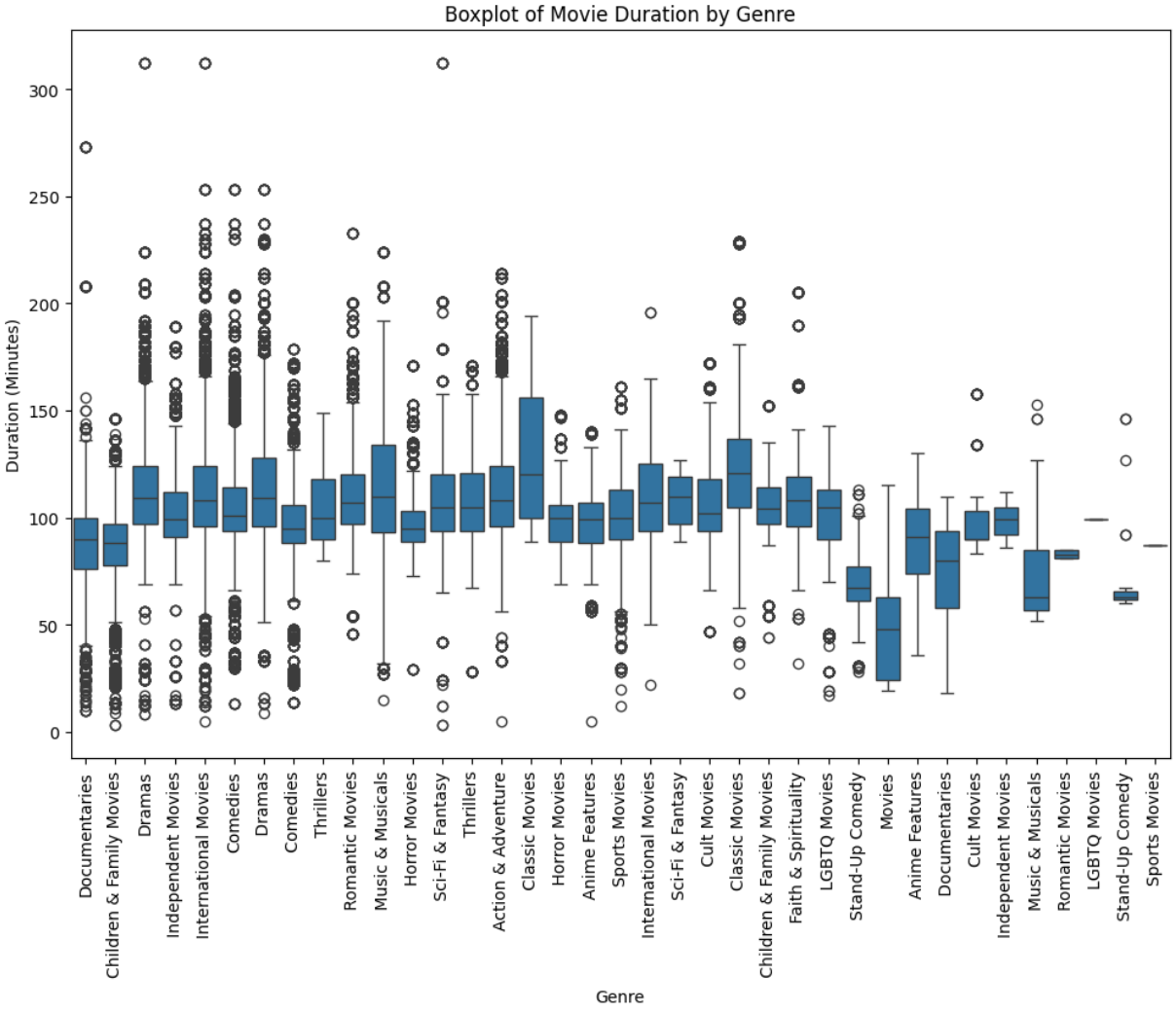
In [189…
```
avg_length_movies = df_movies.groupby(by = "listed_in")["duration"].mean().s
avg_length_movies = avg_length_movies.head(10)
```

In [190…
```
plt.figure(figsize = (10,6))
sns.barplot(data = avg_length_movies , x = "listed_in" , y = "duration" , co
plt.title("Average Length of Movies in Each Genre", fontsize=14)
plt.xlabel("Listed_in", fontsize=14)
plt.ylabel("Duration", fontsize=14)
plt.xticks(rotation=90)
plt.tight_layout()

plt.show()
```



In [191…
```
# Purpose: This boxplot shows the distribution of movie runtimes for each ge
```

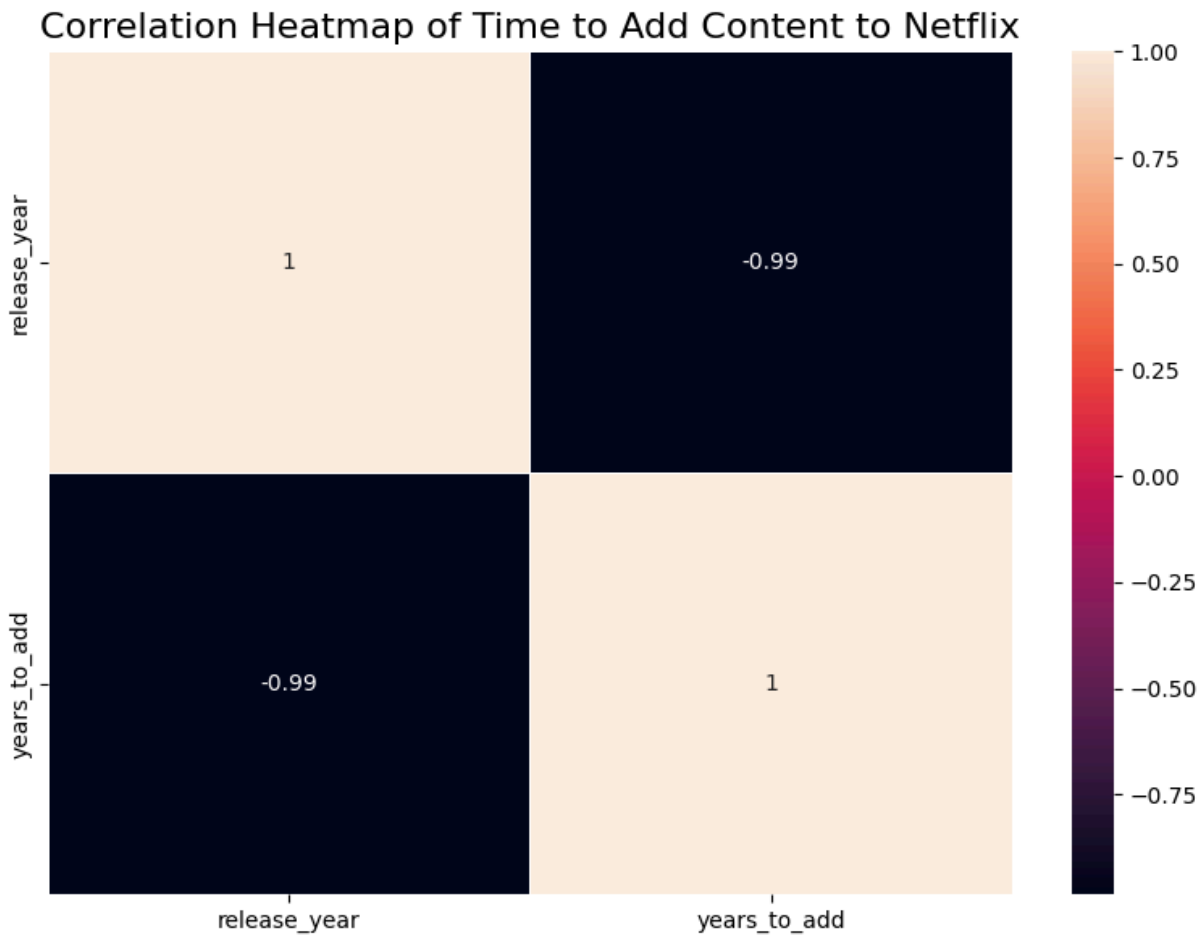
In [192…
```
# Boxplot of movie duration (runtime) by genre
plt.figure(figsize=(12, 8))
sns.boxplot(x='listed_in', y='duration', data=df_movies)
plt.title('Boxplot of Movie Duration by Genre')
plt.xlabel('Genre')
plt.ylabel('Duration (Minutes)')
plt.xticks(rotation=90)
plt.show()
```



In [193…
```
# Extract Year Difference (release_year vs. date_added)
# You can calculate how long after its release a show or movie was added to
```

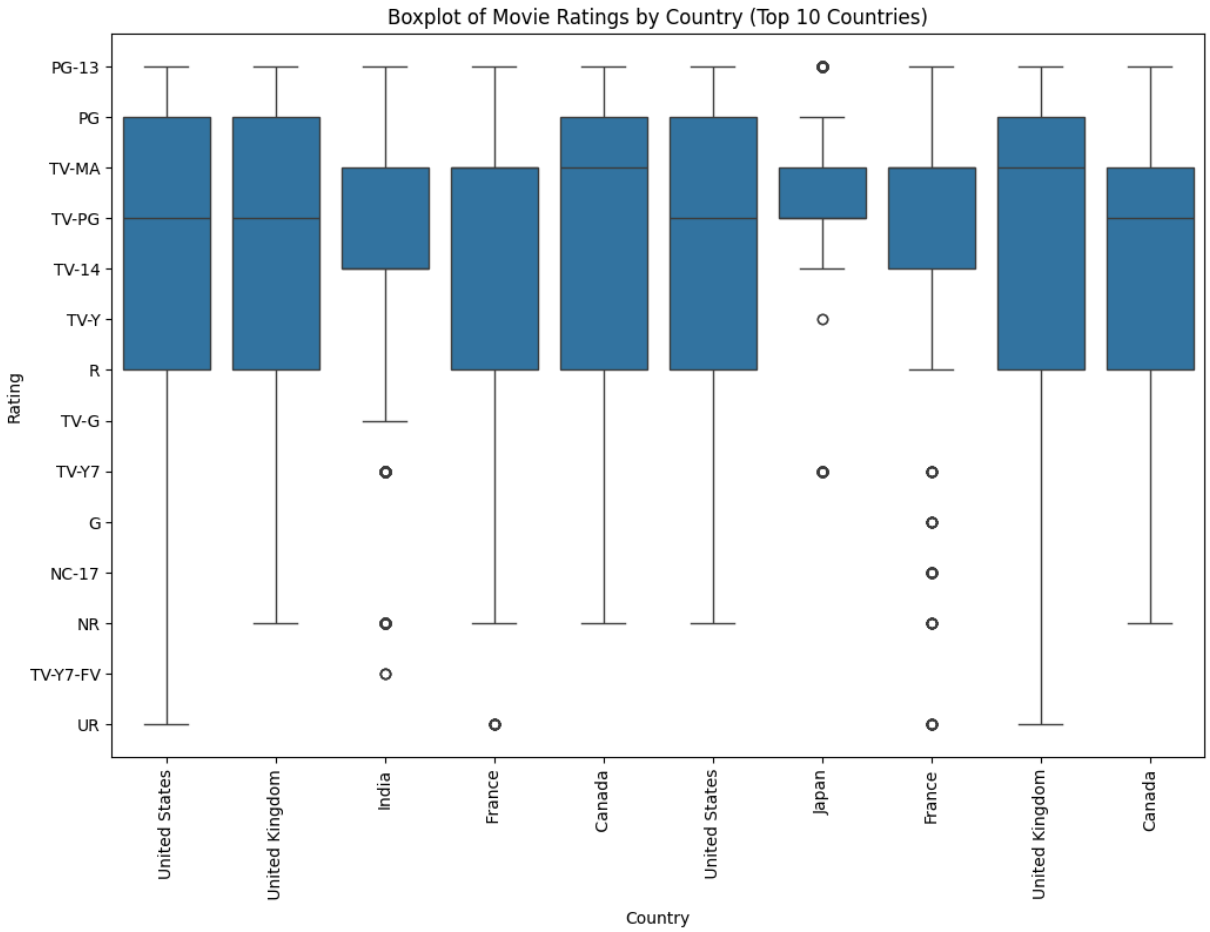
In [194…
```
df_new['years_to_add'] = df_new['date_added'].dt.year - df_new['release_year
```

In [195…
```
# Create a heatmap to see the correlation between 'years_to_add' and 'releas
numerical_cols = ['release_year', 'years_to_add']  # You can include 'durati

corr_matrix = df_new[numerical_cols].corr()

# Create the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, linewidths=0.5)
plt.title('Correlation Heatmap of Time to Add Content to Netflix', fontsize=
plt.tight_layout()
plt.show()
```



In [196…
```
# Plot boxplot of movie ratings by country (Top 10 countries with the most m
```

In [197…
```
top_countries = df_movies['country'].value_counts().nlargest(10).index

plt.figure(figsize=(12, 8))
sns.boxplot(x='country', y='rating', data=df_movies[df_movies['country'].isi
plt.title('Boxplot of Movie Ratings by Country (Top 10 Countries)')
plt.xlabel('Country')
plt.ylabel('Rating')
plt.xticks(rotation=90)
plt.show()
```



In [198…
```
# Duration of content by genre
```

In [199…
```
df_movies.groupby(by = "listed_in")["duration"].mean().sort_values(ascending
```

Out[199…

	listed_in	duration
0	Classic Movies	129.494970
1	Classic Movies	125.363636
2	Dramas	114.808651
3	Action & Adventure	113.166339
4	Cult Movies	113.146476
5	Music & Musicals	112.821713
6	Dramas	112.459226
7	International Movies	112.040861
8	International Movies	110.368421
9	Romantic Movies	110.098404
10	Faith & Spirituality	109.006954
11	Sci-Fi & Fantasy	108.758968
12	Thrillers	108.713672
13	Sci-Fi & Fantasy	107.962963
14	Thrillers	105.373786
15	Children & Family Movies	104.945493
16	Comedies	104.858860
17	Independent Movies	102.886763
18	Sports Movies	101.317408
19	Horror Movies	100.449695
20	LGBTQ Movies	100.292917
21	Cult Movies	99.686391
22	Independent Movies	99.114815
23	LGBTQ Movies	99.000000
24	Anime Features	98.486275
25	Horror Movies	98.443081
26	Comedies	96.345205
27	Anime Features	88.910714
28	Documentaries	87.409628
29	Sports Movies	87.000000
30	Children & Family Movies	84.430278
31	Romantic Movies	82.800000
32	Music & Musicals	76.076923

	listed_in	duration
33	Documentaries	75.911290
34	Stand-Up Comedy	74.625000
35	Stand-Up Comedy	68.575581
36	Movies	48.838631

Business Insights and Recommendations

1. Content Distribution between Movies and TV Shows

- **Insight :** The dataset shows a larger proportion of Movies compared to TV Shows.This suggexts that Netflix catalog is more of movies , Which might indicate viewer preference.
- **Recommendation :** Netflix could explore expanding its TV show offerings, especially in high-performing categories like international TV dramas and reality shows.

2. Release Year Trend

- **Insight:** Most content on Netflix was released between 2015 and 2021, with a peak in 2019. This indicates a strong focus on newer content.

3. Dominance of Certain Genres

- **Insight :** Genres like International Movies, Dramas, and Comedies dominate the Netflix catalog. International Movies alone account for a significant portion of all content.
- **Recommendation:** Given the popularity of International Movies, Netflix should continue investing in global content, potentially exploring underrepresented regions like Southeast Asia and Africa for content acquisition.

4. Popular Ratings

- **Insight :** The most common ratings are TV-MA and TV-14, indicating that Netflix’s catalog is geared more towards mature audiences, with fewer titles rated for younger viewers (e.g., TV-Y or TV-Y7).

5. Geographical Content Distribution

- **Insight :** The United States dominates the catalog in terms of both content production and cast members, followed by India and the United Kingdom.

Content from other regions is significantly less represented.

6. Director Insights

- **Insight :** Directors like Rajiv Chilaka and Marcus Raboy are among the most prolific, particularly in children's and stand-up comedy categories, respectively. This suggests that certain directors are key contributors to specific content types on the platform.
- **Recommendation:** Netflix should continue collaborating with high-performing directors in popular genres, while also identifying and promoting new talent in underrepresented areas like indie films or international dramas.

7. Popular Cast Members

- **Insight:** Actors like Anupam Kher and Shah Rukh Khan have appeared in the most titles, particularly in Indian movies. This highlights the significance of Bollywood stars in Netflix Indian content offerings.

8. Most Popular Content Genres Across Countries

- **Insight:** In the United States, genres like Dramas and Documentaries dominate the platform, while in India, International Movies and Children & Family Movies are the most popular. This shows that different markets have different preferences.
- **Recommendation:** Netflix should continue tailoring its content by region, offering more documentaries and dramas in the US while focusing on family-oriented and international films for the Indian market. A region-specific content strategy can boost local engagement.

9. Duration of Content by Genre

- **Insight:** On average, genres like Dramas, Action & Adventure, and Cult Movies have the longest movie runtimes, while Stand-Up Comedy and Children's content tend to be much shorter.
- **Recommendation:** For longer genres like Dramas and Action, Netflix could experiment with episodic or limited series formats to keep viewers engaged, while continuing to offer short-form content in genres like Stand-Up Comedy to cater to audiences looking for quick entertainment.

10. Release Trends by Year

- **Insight:** A significant number of movies were released in 2019, followed by a dip in subsequent years. This could reflect either a production trend or Netflix focus on recent content.

- **Recommendation:** Given the dip in releases after 2019, Netflix may need to reinvigorate its content acquisition strategy post-2020 to maintain a steady flow of new titles, particularly in trending genres like True Crime and International Documentaries.

11. Country-Specific Content Trends

- **Insight:** "The analysis shows that certain countries (e.g., India, the UK, and Japan) have distinctive genre preferences, with India focusing on family-friendly content and Japan on Anime."

In [199…