Section 1 - Connecting the dataset and making basic exploration to understand the dataset.

In [426]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [427]:
```
!gdown 14BraZ_TLQ0umzhaknzLVtE_mD0ehvAyh
```

```
Downloading...
From: https://drive.google.com/uc?id=14BraZ_TLQ0umzhaknzLVtE_mD0ehvAyh (https://driv
e.google.com/uc?id=14BraZ_TLQ0umzhaknzLVtE_mD0ehvAyh)
To: /content/netflix_titles.csv
100% 3.40M/3.40M [00:00<00:00, 29.5MB/s]
```

In [428]:
```
df = pd.read_csv("netflix_titles.csv")
```

In [429]:
```
df.shape
# has 8807 rows and 12 columns.
```

Out[429]:
```
(8807, 12)
```

In [430]:
```
df.info()
# There is 1 int datatype and rest 11 is of of object datatype
# column names are in lower case which has to be changed----
# There are 6 columns out of the 12 with null values.
# These null values needed to be replaced or the respective row/ columns should be dr
```

```
<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 [431]:
```
df.sample(10)
# The 3 columns "cast", "director" and "listed_in" have multiple values.
# After careful considerations and depending on the need some columns above needed to
```

Out[431]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
6479	s6480	Movie	Christmas in the Smokies	Gary Wheeler	Sarah Lancaster, Barry Corbin, Alan Powell, Ji...	United States	March 1, 2019	2015	TV-G
3710	s3711	TV Show	Answer for Heaven	NaN	Sunny Suwanmethanont, Kan Kantathavorn, Pattar...	NaN	June 27, 2019	2019	TV-MA
1904	s1905	Movie	Carlos Almaraz: Playing with Fire	Elsa Flores Almaraz, Richard J Montoya	Edward James Olmos, Zach De La Rocha	United States	October 1, 2020	2019	TV-14
2935	s2936	Movie	Thottappan	Shanavas K. Bavakutty	Priyamvada Krishnan, Vinayakan, Roshan Mathew,...	India	February 8, 2020	2019	TV-14
581	s582	Movie	Mortal Kombat	Paul W.S. Anderson	Christophe Lambert, Robin Shou, Linden Ashby, ...	United States	July 1, 2021	1995	PG-13
7670	s7671	Movie	Operation Chromite	John H. Lee	Jung-jae Lee, Beom-su Lee, Liam Neeson, Se-yeo...	South Korea	January 15, 2018	2016	NR
8801	s8802	Movie	Zinzana	Majid Al Ansari	Ali Suliman, Saleh Bakri, Yasa, Ali Al-Jabri, ...	United Arab Emirates, Jordan	March 9, 2016	2015	TV-MA
8756	s8757	Movie	Woodstock	Barak Goodman	NaN	United States	August 13, 2019	2019	TV-MA
6478	s6479	Movie	Christmas Crush	Marita Grabiak	Rachel Boston, Jonathan Bennett, Jon Prescott,...	Canada, United States	November 4, 2019	2012	TV-PG
2390	s2391	TV Show	How to Get Away with Murder	NaN	Viola Davis, Billy Brown, Alfred Enoch, Jack F...	United States	June 13, 2020	2020	TV-14

In [432]:
```
df['director'].str.contains(", ").any()
# Proves that we have multiple values sep by ","
```

Out[432]:
```
True
```

In [433]:
```
df['cast'].str.contains(",").any()
# Proves that we have multiple values sep by ","
```

Out[433]:
```
True
```

In [434]:
```
df['listed_in'].str.contains(",").any()
# Proves that we have multiple values sep by ","
# Unnesting these column woulbe be unnecessary.
# As we can manage the popularity aspect of genres without breaking each generate int
```

Out[434]:
```
True
```

In [435]:
```
df.isnull().sum()
```

Out[435]:
```
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
```

Inference on Section 1 -

- We can see that the above dataset has content ranging from 1925 to 2021.
- We can notice that there are a lot of null values
- Dimension of the dataset is 8807 x 12
- Multiple values are present in country, cast and directors coluumn, so it it is imperative to unnest them to seperate rows.

Type *Markdown* and LaTeX: α^2

Section 2 - Data Cleaning

- Based on the previous section we have to do the following things to make the dataset EDA ready.
  - Unnesting the multiple values
  - Handling Null values

- ***Unnesting the multiple values***

a. Un-nesting the "director" column

```
In [436]: df['director']
          # The resulting multiple values has to be split and un-nested
```

```
Out[436]: 0       Kirsten Johnson
          1                   NaN
          2       Julien Leclercq
          3                   NaN
          4                   NaN
                       ...
          8802      David Fincher
          8803                NaN
          8804    Ruben Fleischer
          8805       Peter Hewitt
          8806        Mozez Singh
          Name: director, Length: 8807, dtype: object
```

```
In [437]: director_unnest = pd.DataFrame(df['director'].apply(lambda x: str(x).split(', ')).to_
          director_unnest = director_unnest.stack().reset_index().drop('level_1', axis = 1).ren
          director_unnest
          #
```

...

b. Un-nesting the "cast" column

```
In [438]: df['cast']
          # The resulting multiple values has to be split and un-nested
```

```
Out[438]: 0                                                     NaN
          1       Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...
          2       Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...
          3                                                     NaN
          4       Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...
                                        ...
          8802    Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...
          8803                                                  NaN
          8804    Jesse Eisenberg, Woody Harrelson, Emma Stone, ...
          8805    Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...
          8806    Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...
          Name: cast, Length: 8807, dtype: object
```

```
In [439]: cast_unnest = pd.DataFrame(df['cast'].apply(lambda x: str(x).split(', ')).to_list(), 
          cast_unnest = cast_unnest.stack().reset_index().drop('level_1', axis = 1).rename(colu
          cast_unnest
```

Out[439]:

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

64951 rows × 2 columns

c. Merging the unnested director and cast column together

```
In [440]: unnested_df = director_unnest.merge(cast_unnest, on = 'title', how = 'inner')
          unnested_df
```

Out[440]:

	title	director	cast
0	Dick Johnson Is Dead	Kirsten Johnson	nan
1	Blood & Water	nan	Ama Qamata
2	Blood & Water	nan	Khosi Ngema
3	Blood & Water	nan	Gail Mabalane
4	Blood & Water	nan	Thabang Molaba
...	...	...	...
70807	Zubaan	Mozez Singh	Manish Chaudhary
70808	Zubaan	Mozez Singh	Meghna Malik
70809	Zubaan	Mozez Singh	Malkeet Rauni
70810	Zubaan	Mozez Singh	Anita Shabdish
70811	Zubaan	Mozez Singh	Chittaranjan Tripathy

70812 rows × 3 columns

d. Changing the format of date_added column.

In [441]:
```
df['date_added'] = pd.to_datetime(df['date_added'], format = '%B %d, %Y', errors = 'c
df[df['date_added'].isnull()]
```

Out[441]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
6066	s6067	TV Show	A Young Doctor's Notebook and Other Stories	NaN	Daniel Radcliffe, Jon Hamm, Adam Godley, Chris...	United Kingdom	NaT	2013	TV-MA	2 Seasons
6079	s6080	TV Show	Abnormal Summit	Jung-ah Im, Seung-uk Jo	Hyun-moo Jun, Si-kyung Sung, Se-yoon Yoo	South Korea	NaT	2017	TV-PG	2 Seasons
6174	s6175	TV Show	Anthony Bourdain: Parts Unknown	NaN	Anthony Bourdain	United States	NaT	2018	TV-PG	5 Seasons
6177	s6178	TV Show	忍者ハットリくん	NaN	NaN	Japan	NaT	2012	TV-Y7	2 Seasons
6213	s6214	TV Show	Bad Education	NaN	Jack Whitehall, Mathew Horne, Sarah Solemani, ...	United Kingdom	NaT	2014	TV-MA	3 Seasons
...	...	...	...	...	...	...	...	...	...	...
8539	s8540	TV Show	The Tudors	NaN	Jonathan Rhys Meyers, Henry Cavill, James Frai...	Ireland, Canada, United States, United Kingdom	NaT	2010	TV-MA	4 Seasons
8557	s8558	TV Show	The West Wing	NaN	Martin Sheen, Rob Lowe, Allison Janney, John S...	United States	NaT	2005	TV-14	7 Seasons
8684	s8685	TV Show	Vroomiz	NaN	Joon-seok Song, Jeong-hwa Yang, Sang-hyun Um, ...	South Korea	NaT	2016	TV-Y	3 Seasons
8712	s8713	TV Show	Weird Wonders of the World	NaN	Chris Packham	United Kingdom	NaT	2016	TV-PG	2 Seasons
8755	s8756	TV Show	Women Behind Bars	NaN	NaN	United States	NaT	2010	TV-14	3 Seasons

98 rows × 12 columns

e. Merging the 'unnested_df' with rest of the columns to complete the dataset with all relevant information to facilitate EDA process.

In [442]:
```
netflix_df = unnested_df.merge(df[['show_id', 'type','title', 'country',
                                   'date_added','release_year', 'rating',
                                   'listed_in', 'duration']], on = 'title', how = 'in
netflix_df
```

Out[442]:

	title	director	cast	show_id	type	country	date_added	release_year	rating	lis
0	Dick Johnson Is Dead	Kirsten Johnson	nan	s1	Movie	United States	2021-09-25	2020	PG-13	Docume
1	Blood & Water	nan	Ama Qamata	s2	TV Show	South Africa	2021-09-24	2021	TV-MA	Intern TV Sho Dram My
2	Blood & Water	nan	Khosi Ngema	s2	TV Show	South Africa	2021-09-24	2021	TV-MA	Intern TV Sho Dram My
3	Blood & Water	nan	Gail Mabalane	s2	TV Show	South Africa	2021-09-24	2021	TV-MA	Intern TV Sho Dram My
4	Blood & Water	nan	Thabang Molaba	s2	TV Show	South Africa	2021-09-24	2021	TV-MA	Intern TV Sho Dram My
...	...	...	...	...	...	...	...	...	...	
70807	Zubaan	Mozez Singh	Manish Chaudhary	s8807	Movie	India	2019-03-02	2015	TV-14	D Intern Movies, & M
70808	Zubaan	Mozez Singh	Meghna Malik	s8807	Movie	India	2019-03-02	2015	TV-14	D Intern Movies, & M
70809	Zubaan	Mozez Singh	Malkeet Rauni	s8807	Movie	India	2019-03-02	2015	TV-14	D Intern Movies, & M
70810	Zubaan	Mozez Singh	Anita Shabdish	s8807	Movie	India	2019-03-02	2015	TV-14	D Intern Movies, & M
70811	Zubaan	Mozez Singh	Chittaranjan Tripathy	s8807	Movie	India	2019-03-02	2015	TV-14	D Intern Movies, & M

70812 rows × 11 columns

f. Unnesting Country Column and merge with rest of the final data set

In [443]:
```
df['country'].str.contains(', ').any()
# this proves that we have nested values in the country column
```

Out[443]:
```
True
```

```
In [444]: unnest_country = pd.DataFrame(df['country'].apply(lambda x: str(x).split(', ')).to_li
          unnest_country = unnest_country.stack().reset_index().drop('level_1', axis = 1).renam
          netflix_df = unnest_country.merge(netflix_df, left_on = ['title', 'country'], right_o
          netflix_df
```

Out[444]:

	title	country	director	cast	show_id	type	date_added	release_year	rating	lis
0	Dick Johnson Is Dead	United States	Kirsten Johnson	nan	s1	Movie	2021-09-25	2020	PG-13	Docume
1	Blood & Water	South Africa	nan	Ama Qamata	s2	TV Show	2021-09-24	2021	TV-MA	Intern TV Sho Dram My
2	Blood & Water	South Africa	nan	Khosi Ngema	s2	TV Show	2021-09-24	2021	TV-MA	Intern TV Sho Dram My
3	Blood & Water	South Africa	nan	Gail Mabalane	s2	TV Show	2021-09-24	2021	TV-MA	Intern TV Sho Dram My
4	Blood & Water	South Africa	nan	Thabang Molaba	s2	TV Show	2021-09-24	2021	TV-MA	Intern TV Sho Dram My
...	...	...	...	...	...	...	...	...	...	
53283	Zubaan	India	Mozez Singh	Manish Chaudhary	s8807	Movie	2019-03-02	2015	TV-14	D Intern Movies, & M
53284	Zubaan	India	Mozez Singh	Meghna Malik	s8807	Movie	2019-03-02	2015	TV-14	D Intern Movies, & M
53285	Zubaan	India	Mozez Singh	Malkeet Rauni	s8807	Movie	2019-03-02	2015	TV-14	D Intern Movies, & M
53286	Zubaan	India	Mozez Singh	Anita Shabdish	s8807	Movie	2019-03-02	2015	TV-14	D Intern Movies, & M
53287	Zubaan	India	Mozez Singh	Chittaranjan Tripathy	s8807	Movie	2019-03-02	2015	TV-14	D Intern Movies, & M

53288 rows × 11 columns

- ***Handling Null values***
  - As per instruction we are converting the value of the null values with "Unknown Column_name" if it is a Categorical variable.
  - If it is a Continuous valriable we are instructed replace the null values with '0'
  - I have added efficient ways at the end to efficiently replace the null values with more accurate which will help to increase the overall accuracy of the analysis.
- Continuous Variables
  - 'date_added', 'release_year'
- Categorical Varibles
  - 'show_id', 'type', 'title', 'country', 'rating', 'listed_in', 'duration'

In [445]:
```
df.isnull().sum()
```

Out[445]:
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        98
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

In [446]:
```
netflix_df['cast'].replace(['nan'], ['Unknown Actor'], inplace = True)
netflix_df['director'].replace(['nan'], ['Unknown Director'], inplace = True)
netflix_df['country'].replace(['nan'], ['Unknown Country'], inplace = True)
```

In [447]:
```
netflix_df.sample(10)
```

Out[447]:

	title	country	director	cast	show_id	type	date_added	release_year	rating	l
41902	Hokkabaz	Turkey	Ali Taner Baltacı	Mazhar Alanson	s6983	Movie	2017-03-10	2006	TV-MA	Co Inte
43077	Ken Burns: The Roosevelts: An Intimate History	United States	Ken Burns	Unknown Actor	s7177	TV Show	2017-02-22	2014	TV-PG	Do
49212	SuperNature: Wild Flyers	United Kingdom	Unknown Director	Unknown Actor	s8130	TV Show	2017-03-01	2016	TV-PG	E Doc S N
50143	The Flintstones in Viva Rock Vegas	United States	Brian Levant	Alan Cumming	s8306	Movie	2019-10-01	2000	PG	Cl Co F
19027	Como caído del cielo	Mexico	Pepe Bojórquez	Angélica María	s3105	Movie	2019-12-24	2019	TV-14	Co Inte
42303	I Am Wrath	United States	Chuck Russell	Jordan Whalen	s7044	Movie	2019-09-16	2016	R	Ac
24060	About Time	United Kingdom	Richard Curtis	Lindsay Duncan	s3909	Movie	2019-04-16	2013	R	Co Inte
5769	The Stand-In	United States	Jamie Babbit	Michael Zegen	s1084	Movie	2021-04-10	2020	R	C
1921	The Haunting in Connecticut 2: Ghosts of Georgia	United States	Tom Elkins	Brad James	s354	Movie	2021-08-01	2013	R	
13724	Adú	Spain	Salvador Calvo	Nora Navas	s2309	Movie	2020-07-01	2020	TV-MA	Inte

In [448]:
```
netflix_df.isnull().sum()
```

Out[448]:
```
title             0
country           0
director          0
cast              0
show_id           0
type              0
date_added      572
release_year      0
rating           36
listed_in         0
duration          3
dtype: int64
```

Handling the rest of the Categorical Null values

We can see we have handled the null value for 3 columns and for 'rating' and 'duration' columns we will take the data from the "IMDB" website and replace them.

Rating Column

In [449]:
```
# 1. Checking the null ratings
netflix_df[netflix_df['rating'].isnull()]
```

Out[449]:

	title	country	director	cast	show_id	type	date_added	release_year	rating	li
40979	Gargantia on the Verdurous Planet	Japan	Unknown Director	Kaito Ishikawa	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40980	Gargantia on the Verdurous Planet	Japan	Unknown Director	Hisako Kanemoto	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40981	Gargantia on the Verdurous Planet	Japan	Unknown Director	Ai Kayano	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40982	Gargantia on the Verdurous Planet	Japan	Unknown Director	Kana Asumi	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40983	Gargantia on the Verdurous Planet	Japan	Unknown Director	Shizuka Ito	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40984	Gargantia on the Verdurous Planet	Japan	Unknown Director	Sayaka Ohara	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40985	Gargantia on the Verdurous Planet	Japan	Unknown Director	Katsuyuki Konishi	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40986	Gargantia on the Verdurous Planet	Japan	Unknown Director	Yuka Terasaki	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40987	Gargantia on the Verdurous Planet	Japan	Unknown Director	Yuki Ono	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40988	Gargantia on the Verdurous Planet	Japan	Unknown Director	Tomokazu Sugita	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40989	Gargantia on the Verdurous Planet	Japan	Unknown Director	Ayumi Fujimura	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40990	Gargantia on the Verdurous Planet	Japan	Unknown Director	Alan Lee	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40991	Gargantia on the Verdurous Planet	Japan	Unknown Director	Cassandra Morris	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40992	Gargantia on the Verdurous Planet	Japan	Unknown Director	Natalie Hoover	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40993	Gargantia on the Verdurous Planet	Japan	Unknown Director	Janice Kawaye	s6828	TV Show	2016-12-01	2013	NaN	Inter T\
40994	Gargantia on the Verdurous Planet	Japan	Unknown Director	Laura Post	s6828	TV Show	2016-12-01	2013	NaN	Inter T\

	title	country	director	cast	show_id	type	date_added	release_year	rating	li
40995	Gargantia on the Verdurous Planet	Japan	Unknown Director	Julie Ann Taylor	s6828	TV Show	2016-12-01	2013	NaN	Inter T
40996	Gargantia on the Verdurous Planet	Japan	Unknown Director	Patrick Seitz	s6828	TV Show	2016-12-01	2013	NaN	Inter T
40997	Gargantia on the Verdurous Planet	Japan	Unknown Director	Michelle Ruff	s6828	TV Show	2016-12-01	2013	NaN	Inter T
40998	Gargantia on the Verdurous Planet	Japan	Unknown Director	Marc Diraison	s6828	TV Show	2016-12-01	2013	NaN	Inter T
40999	Gargantia on the Verdurous Planet	Japan	Unknown Director	Matthew Mercer	s6828	TV Show	2016-12-01	2013	NaN	Inter T
41000	Gargantia on the Verdurous Planet	Japan	Unknown Director	Karen Strassman	s6828	TV Show	2016-12-01	2013	NaN	Inter T
43773	Little Lunch	Australia	Unknown Director	Flynn Curry	s7313	TV Show	2018-02-01	2015	NaN	Kids C
43774	Little Lunch	Australia	Unknown Director	Olivia Deeble	s7313	TV Show	2018-02-01	2015	NaN	Kids C
43775	Little Lunch	Australia	Unknown Director	Madison Lu	s7313	TV Show	2018-02-01	2015	NaN	Kids C
43776	Little Lunch	Australia	Unknown Director	Oisín O'Leary	s7313	TV Show	2018-02-01	2015	NaN	Kids C
43777	Little Lunch	Australia	Unknown Director	Faith Seci	s7313	TV Show	2018-02-01	2015	NaN	Kids C
43778	Little Lunch	Australia	Unknown Director	Joshua Sitch	s7313	TV Show	2018-02-01	2015	NaN	Kids C
43779	Little Lunch	Australia	Unknown Director	Heidi Arena	s7313	TV Show	2018-02-01	2015	NaN	Kids C
45663	My Honor Was Loyalty	Italy	Alessandro Pepe	Leone Frisa	s7538	Movie	2017-03-01	2015	NaN	
45664	My Honor Was Loyalty	Italy	Alessandro Pepe	Paolo Vaccarino	s7538	Movie	2017-03-01	2015	NaN	
45665	My Honor Was Loyalty	Italy	Alessandro Pepe	Francesco Migliore	s7538	Movie	2017-03-01	2015	NaN	
45666	My Honor Was Loyalty	Italy	Alessandro Pepe	Albrecht Weimer	s7538	Movie	2017-03-01	2015	NaN	
45667	My Honor Was Loyalty	Italy	Alessandro Pepe	Giulia Dichiaro	s7538	Movie	2017-03-01	2015	NaN	
45668	My Honor Was Loyalty	Italy	Alessandro Pepe	Alessandra Oriti Niosi	s7538	Movie	2017-03-01	2015	NaN	
45669	My Honor Was Loyalty	Italy	Alessandro Pepe	Andreas Segeritz	s7538	Movie	2017-03-01	2015	NaN	

In [450]:
```
# 2. Filling the null cells with IMDB accurate ratings
netflix_df.loc[netflix_df['show_id'] == 's5990', :] = netflix_df[netflix_df['show_id'
netflix_df.loc[netflix_df['show_id'] == 's6828', :] = netflix_df[netflix_df['show_id'
netflix_df.loc[netflix_df['show_id'] == 's7313', :] = netflix_df[netflix_df['show_id'
netflix_df.loc[netflix_df['show_id'] == 's7538', :] = netflix_df[netflix_df['show_id'
```

Duration Column

In [451]:
```
# 1. Checking the null duration
netflix_df[netflix_df['duration'].isnull()]
```

Out[451]:

	title	country	director	cast	show_id	type	date_added	release_year	rating	listed_in	dur
33302	Louis C.K. 2017	United States	Louis C.K.	Louis C.K.	s5542	Movie	2017-04-04	2017	74 min	Movies	
34839	Louis C.K.: Hilarious	United States	Louis C.K.	Louis C.K.	s5795	Movie	2016-09-16	2010	84 min	Movies	
34906	Louis C.K.: Live at the Comedy Store	United States	Louis C.K.	Louis C.K.	s5814	Movie	2016-08-15	2015	66 min	Movies	

In [452]:
```
# 2. Filling the null duration cells
netflix_df.loc[netflix_df['show_id'] == 's5542', 'duration'] = netflix_df.loc[netflix_
netflix_df.loc[netflix_df['show_id'] == 's5542', 'rating'] = 'TV-MA'
# Test purpose
# netflix_df.loc[netflix_df['show_id'] == 's5542', :]

netflix_df.loc[netflix_df['show_id'] == 's5795', 'duration'] = netflix_df.loc[netflix_
netflix_df.loc[netflix_df['show_id'] == 's5795', 'rating'] = 'TV-MA'
# Test purpose
# netflix_df.loc[netflix_df['show_id'] == 's5795', :]

netflix_df.loc[netflix_df['show_id'] == 's5814', 'duration'] = netflix_df.loc[netflix_
netflix_df.loc[netflix_df['show_id'] == 's5814', 'rating'] = 'TV-MA'
# Test purpose
# netflix_df.loc[netflix_df['show_id'] == 's5814', :]
```

In [453]:
```
netflix_df.isnull().sum()
```

Out[453]:
```
title             0
country           0
director          0
cast              0
show_id           0
type              0
date_added      572
release_year      0
rating            0
listed_in         0
duration          0
dtype: int64
```

date_added column

- *As per the said guidance for this analysis the null values of continuous variable should be replaced by '0'*
- *'date_added' --> continuous variable*

- ***End of this session I will be adding a set code that can increase the efficiency of analysis***

```
In [454]: netflix_df['date_added'].fillna(0, inplace = True)
```

```
In [455]: netflix_df.isnull().sum()
```

```
Out[455]: title           0
          country         0
          director        0
          cast            0
          show_id         0
          type            0
          date_added      0
          release_year    0
          rating          0
          listed_in       0
          duration        0
          dtype: int64
```

All null values are handled and the data is ready for analysis

How to increase the Overall Analysis.

- In our case we are instructed to replace the continuous variable by '0' which makes sense because we had only 10 null values in the Continuous variable column out of 8807 rows.
- Therefore dropping them or replacing them with 0 makes sense, in this case.

Best Approach

- It is a good practice to replace null values with educated guess.
- Our data set have movies ranging from 1925 to 2021, whereas the Netflix was launched in the year 2007 and adds the movie to their platform within an year the movie getting released.
- But we can't say a movie that was relesased in 1925 was added tro netflix at 1925, that would be absurd.
- **We should find the mode for each year and replace values based on them instead of blindly upping years by 1.**

Finding mode value with respect to each year

```
In [456]: netflix_df['date_added'] = pd.to_datetime(netflix_df['date_added'], format = '%B %d, 
          df['date_added'].isnull().sum()
```

```
Out[456]: 98
```

```
In [457]: null_date_rows = netflix_df[netflix_df['date_added'].isnull()]
          null_year = null_date_rows['release_year'].unique()
          null_year.sort()
          null_year
```

```
Out[457]: array([1967, 1977, 1988, 1990, 1992, 1998, 2003, 2005, 2008, 2010, 2012,
                 2013, 2014, 2015, 2016, 2017, 2018])
```

So we have all the rows with null values w.r.t to the release_year column.

In [458]:
```
# calculating the mode with repective to each year and takking the first value
mode_year = df['date_added'].groupby(by = df['release_year']).agg(pd.Series.mode).to_
missing_dates_table = mode_year.loc[null_year,:]
missing_dates_table
```

Out[458]:

	date_added
release_year	
1967	2021-01-01 00:00:00
1977	2019-12-31 00:00:00
1988	2011-10-01 00:00:00
1990	[2011-10-01 00:00:00, 2019-12-31 00:00:00, 202...
1992	2020-01-01 00:00:00
1998	2019-11-01 00:00:00
2003	2021-09-01 00:00:00
2005	2020-01-01 00:00:00
2008	[2018-11-01 00:00:00, 2019-07-01 00:00:00, 201...
2010	2018-03-15 00:00:00
2012	2016-12-15 00:00:00
2013	2018-11-01 00:00:00
2014	[2017-07-01 00:00:00, 2018-10-01 00:00:00]
2015	[2016-10-01 00:00:00, 2018-07-26 00:00:00]
2016	2017-08-01 00:00:00
2017	2018-07-01 00:00:00
2018	[2018-11-30 00:00:00, 2019-03-01 00:00:00]

We have a list of years for which there are null values in their "date_added" column with their mode values. We can **replace the null with these mode values which will be more precise for analysis purpose** but as instructed we are reverting these values back to the '0' value to maintain the uniformity.

In [459]:
```
netflix_df['date_added'].fillna(0, inplace = True)
```

In [461]:
```
netflix_df.isnull().sum()
```

Out[461]:
```
title           0
country         0
director        0
cast            0
show_id         0
type            0
date_added      0
release_year    0
rating          0
listed_in       0
duration        0
dtype: int64
```

Inference from Section 2-

- It involves two steps one being unnesting the column with multiple rows.
- Second being Null values to make the dataset ready for analysis.
- Null values from Categorical variables are replaced by "Unknown Column_name" as per the guidance.

- Also few categorical null values are replaced by values from websites like "IMDB" and "Wikipedia".
- Employed a method showing how using functions like mode are used to take a eductive guess to enhance the accuracy of the analysis.

Section 3 - Analysis and Visualisation

Analysing the categorical Values

----Title Column----

Total Number of Available content

```
In [487]: len(df['title'].index)
```

```
Out[487]: 8807
```

Proportion of Movies and TV Shows

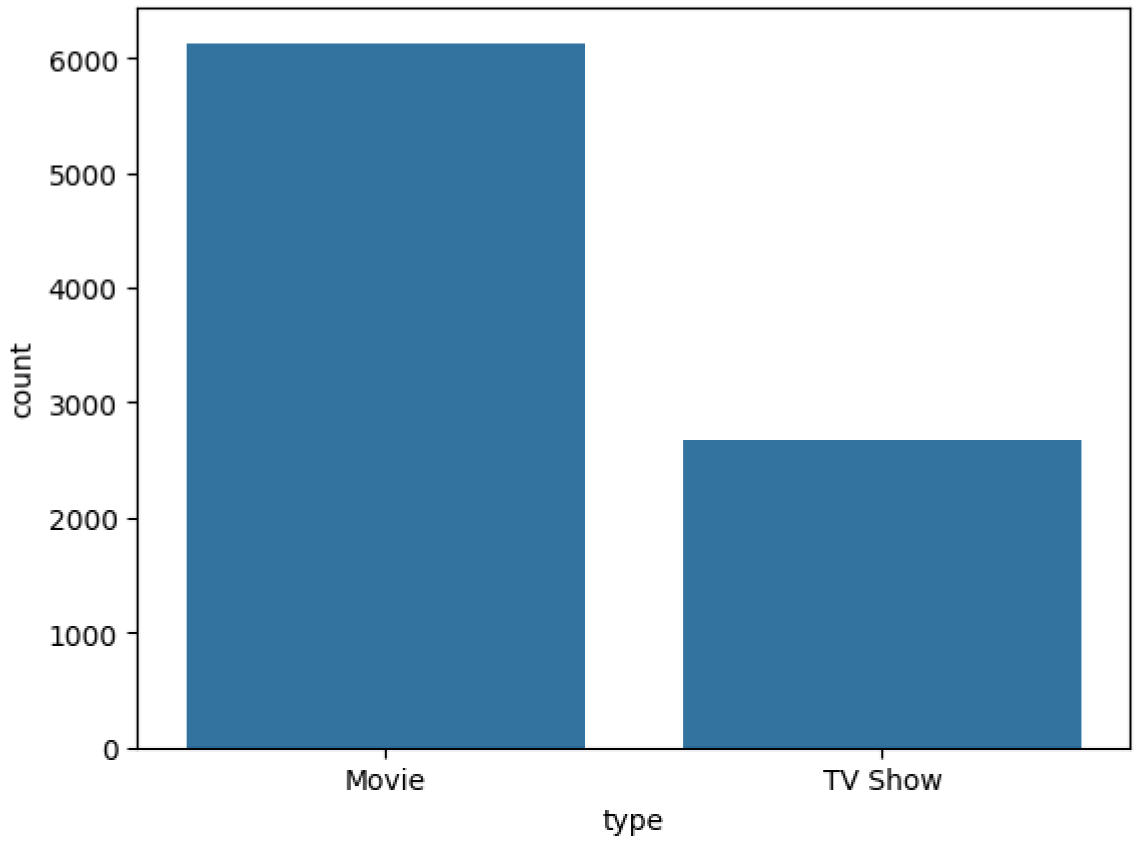
```
In [486]: df.groupby(by ='type')['title'].count()
```

```
Out[486]: type
          Movie      6131
          TV Show    2676
          Name: title, dtype: int64
```

Visual Representation of TV Shows and Movies proportion

```
In [491]: sns.countplot(x = 'type', data = df)
```

```
Out[491]: <Axes: xlabel='type', ylabel='count'>
```



Trends of Movies over the years

In [509]:
```
content_counts = df['release_year'].value_counts().sort_index().reset_index()
content_counts.columns = ['Year', 'Num_of_movies']
content_counts
```

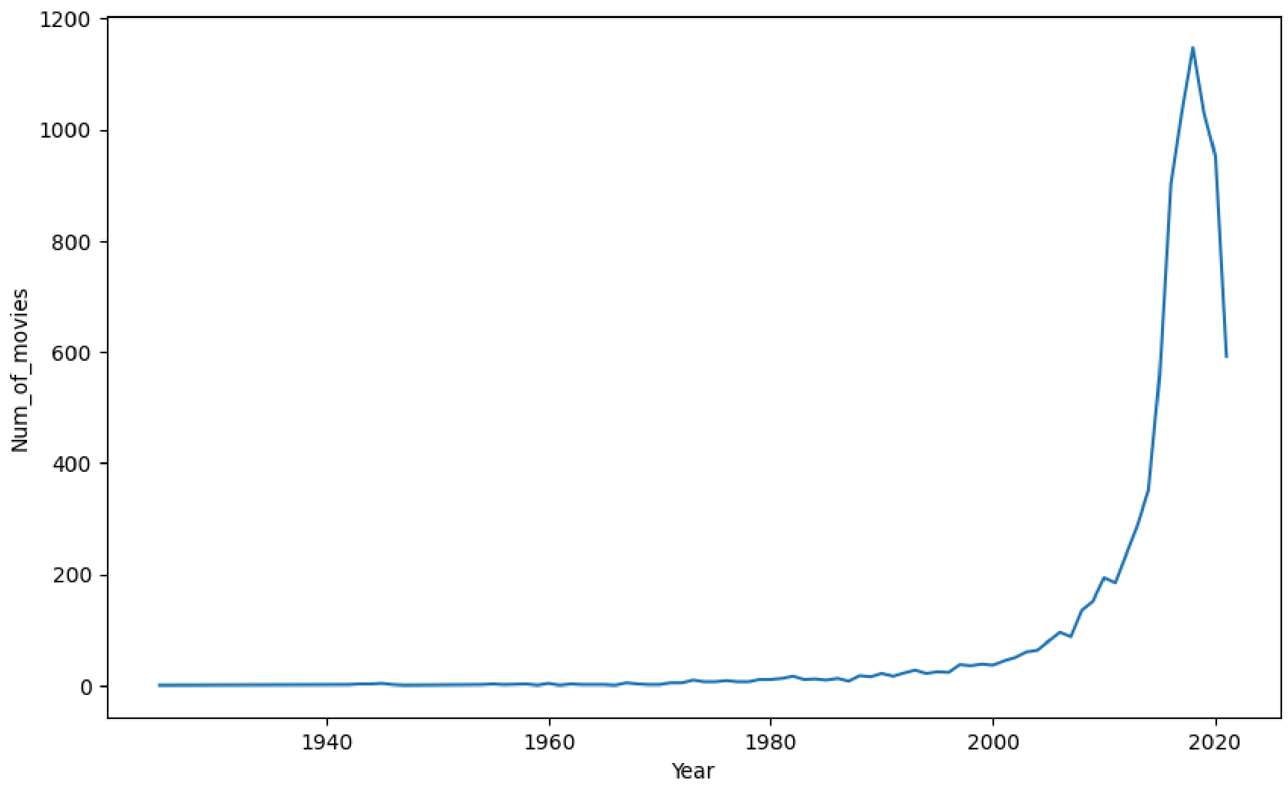
Out[509]:

	Year	Num_of_movies
0	1925	1
1	1942	2
2	1943	3
3	1944	3
4	1945	4
...	...	...
69	2017	1032
70	2018	1147
71	2019	1030
72	2020	953
73	2021	592

74 rows × 2 columns

In [510]:
```
plt.figure(figsize = (10, 6))
sns.lineplot(x = 'Year', y = 'Num_of_movies', data = content_counts)
```

Out[510]:
```
<Axes: xlabel='Year', ylabel='Num_of_movies'>
```



Inference-

- We can see the most of the movies in netflix from after 2000.
- There are some very good opportunities to add some old classic movies which are hard to find elsewhere.

----director column----

Total number of directors listed on Netflix

```
In [536]: netflix_df['director'].nunique()
```

```
Out[536]: 3921
```

Top 10 directors who have produced large number of movies/ TV shows

In [526]:
```python
top_directors = netflix_df.drop_duplicates(subset = ['title', 'director'])
top_directors

# We have droped the repeating titles w.r.t director name therefore no duplication an
```

Out[526]:

	title	country	director	cast	show_id	type	date_added	release_year	rating	
0	Dick Johnson Is Dead	United States	Kirsten Johnson	Unknown Actor	s1	Movie	2021-09-25 00:00:00	2020	PG-13	Doc
1	Blood & Water	South Africa	Unknown Director	Ama Qamata	s2	TV Show	2021-09-24 00:00:00	2021	TV-MA	I TV I
20	Kota Factory	India	Unknown Director	Mayur More	s5	TV Show	2021-09-24 00:00:00	2021	TV-MA	I R Sl
28	The Great British Baking Show	United Kingdom	Andy Devonshire	Mel Giedroyc	s9	TV Show	2021-09-24 00:00:00	2021	TV-14	Sho
32	The Starling	United States	Theodore Melfi	Melissa McCarthy	s10	Movie	2021-09-24 00:00:00	2021	PG-13	
...	...	...	...	...	...	...	...	...	...	
53245	Zindagi Gulzar Hai	Pakistan	Unknown Director	Sanam Saeed	s8801	TV Show	2016-12-15 00:00:00	2012	TV-PG	I R Sl
53254	Zodiac	United States	David Fincher	Mark Ruffalo	s8803	Movie	2019-11-20 00:00:00	2007	R	C
53264	Zombieland	United States	Ruben Fleischer	Jesse Eisenberg	s8805	Movie	2019-11-01 00:00:00	2009	R	Ho
53271	Zoom	United States	Peter Hewitt	Tim Allen	s8806	Movie	2020-01-11 00:00:00	2006	PG	Fan
53280	Zubaan	India	Mozez Singh	Vicky Kaushal	s8807	Movie	2019-03-02 00:00:00	2015	TV-14	I Mo

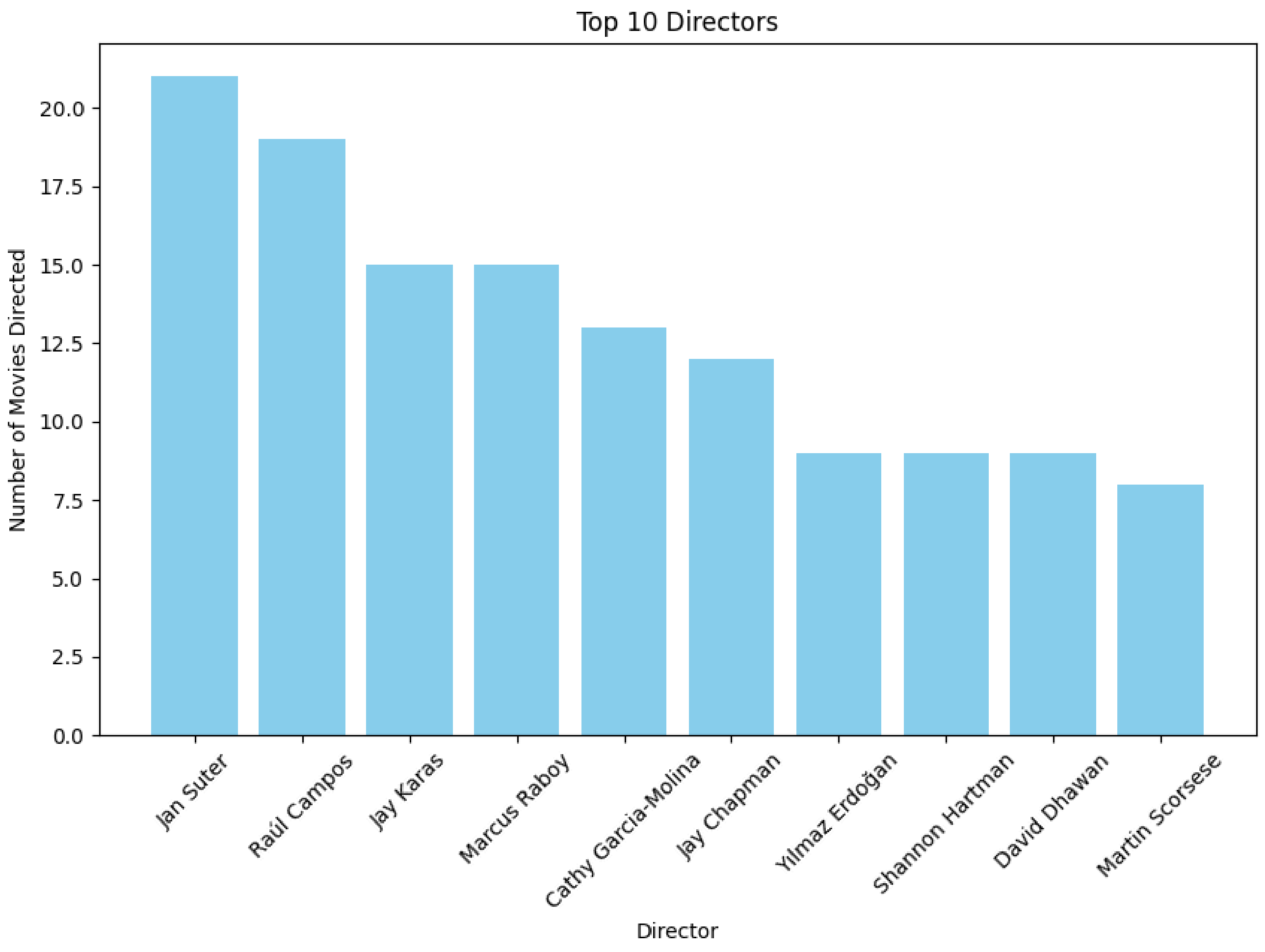
7261 rows × 11 columns

In [564]:
```
director_counts = top_directors['director'].value_counts().reset_index()
director_counts.sort_values(by = 'count', ascending = False)
director_counts = director_counts[director_counts['director'] != 'Unknown Director']
director_counts_top10 = director_counts.head(10)
director_counts_top10.columns = ['Director', 'Number of Movies']
director_counts_top10
```

Out[564]:

	Director	Number of Movies
1	Jan Suter	21
2	Raúl Campos	19
3	Jay Karas	15
4	Marcus Raboy	15
5	Cathy Garcia-Molina	13
6	Jay Chapman	12
7	Yılmaz Erdoğan	9
8	Shannon Hartman	9
9	David Dhawan	9
10	Martin Scorsese	8

In [566]:
```
plt.figure(figsize=(10, 6))
plt.bar(director_counts_top10['Director'], director_counts_top10['Number of Movies'],
plt.xlabel('Director')
plt.ylabel('Number of Movies Directed')
plt.title('Top 10 Directors')
plt.xticks(rotation=45)
plt.show()
```



Inference -

- These are most productive directors in the platform, having some colaborative efforts with these directors will go a long way.
- For example Netflix original shows with these directors.
- As a promotion, netflix came up with an idea of ' Learning spanish with Narcos', which was a big success.

---- cast column----

Top 10 directors who have produced large number of movies/ TV shows

In [568]:
```
top_actors = netflix_df.drop_duplicates(subset = ['title', 'cast'])
top_actors
```

Out[568]:

	title	country	director	cast	show_id	type	date_added	release_year	rating	li
0	Dick Johnson Is Dead	United States	Kirsten Johnson	Unknown Actor	s1	Movie	2021-09-25 00:00:00	2020	PG-13	Docume
1	Blood & Water	South Africa	Unknown Director	Ama Qamata	s2	TV Show	2021-09-24 00:00:00	2021	TV-MA	Interı TV Sho Drar M
2	Blood & Water	South Africa	Unknown Director	Khosi Ngema	s2	TV Show	2021-09-24 00:00:00	2021	TV-MA	Interı TV Sho Drar M
3	Blood & Water	South Africa	Unknown Director	Gail Mabalane	s2	TV Show	2021-09-24 00:00:00	2021	TV-MA	Interı TV Sho Drar M
4	Blood & Water	South Africa	Unknown Director	Thabang Molaba	s2	TV Show	2021-09-24 00:00:00	2021	TV-MA	Interı TV Sho Drar M
...	...	...	...	...	...	...	...	...	...	
53283	Zubaan	India	Mozez Singh	Manish Chaudhary	s8807	Movie	2019-03-02 00:00:00	2015	TV-14	D Interı Movies & M
53284	Zubaan	India	Mozez Singh	Meghna Malik	s8807	Movie	2019-03-02 00:00:00	2015	TV-14	D Interı Movies & M
53285	Zubaan	India	Mozez Singh	Malkeet Rauni	s8807	Movie	2019-03-02 00:00:00	2015	TV-14	D Interı Movies & M
53286	Zubaan	India	Mozez Singh	Anita Shabdish	s8807	Movie	2019-03-02 00:00:00	2015	TV-14	D Interı Movies & M
53287	Zubaan	India	Mozez Singh	Chittaranjan Tripathy	s8807	Movie	2019-03-02 00:00:00	2015	TV-14	D Interı Movies & M

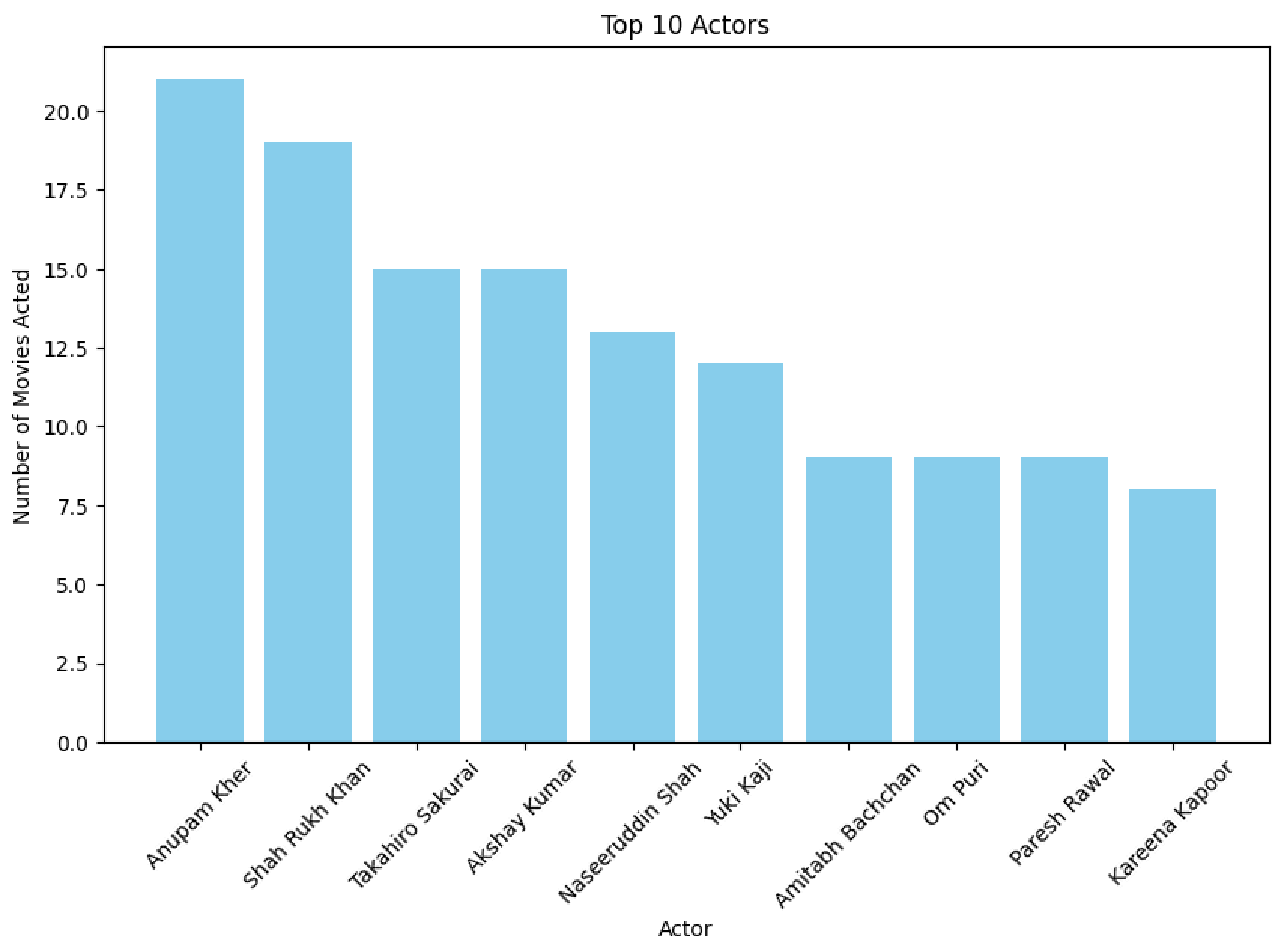
48894 rows × 11 columns

In [570]:
```
actor_counts = top_actors['cast'].value_counts().reset_index()
actor_counts.sort_values(by = 'count', ascending = False)
actor_counts = actor_counts[actor_counts['cast'] != 'Unknown Actor']
actor_counts_top10 = actor_counts.head(10)
actor_counts_top10.columns = ['Actor', 'Number of Movies']
actor_counts_top10
```

Out[570]:

	Actor	Number of Movies
1	Anupam Kher	41
2	Shah Rukh Khan	32
3	Takahiro Sakurai	29
4	Akshay Kumar	29
5	Naseeruddin Shah	29
6	Yuki Kaji	28
7	Amitabh Bachchan	28
8	Om Puri	27
9	Paresh Rawal	27
10	Kareena Kapoor	24

In [571]:
```
plt.figure(figsize=(10, 6))
plt.bar(actor_counts_top10['Actor'], director_counts_top10['Number of Movies'], color
plt.xlabel('Actor')
plt.ylabel('Number of Movies Acted')
plt.title('Top 10 Actors')
plt.xticks(rotation=45)
plt.show()
```



Inference -

- These are very famous actors, new shows can be endorsed by these actors since they are quite famous and familiar among the people.

Comparison of TV Shows and Movies

In [605]:
```
df1 = netflix_df[['country','title', 'type', 'release_year']]
df1 = df1.drop_duplicates(subset = ['title', 'country'])
df2 = df1.groupby(['country','release_year', 'type']).size().reset_index(name='Total 
top_10_trend_countries = pd.DataFrame(df2)
top_10_trend_countries.groupby(['country', 'type'])['Total Count'].sum()
```

Out[605]:
```
country       type   
Argentina     Movie      38
              TV Show    18
Australia     Movie      39
              TV Show    48
Austria       Movie       5
                         ..
Uruguay       Movie       3
Venezuela     Movie       1
Vietnam       Movie       7
West Germany  Movie       1
Zimbabwe      Movie       1
Name: Total Count, Length: 121, dtype: int64
```

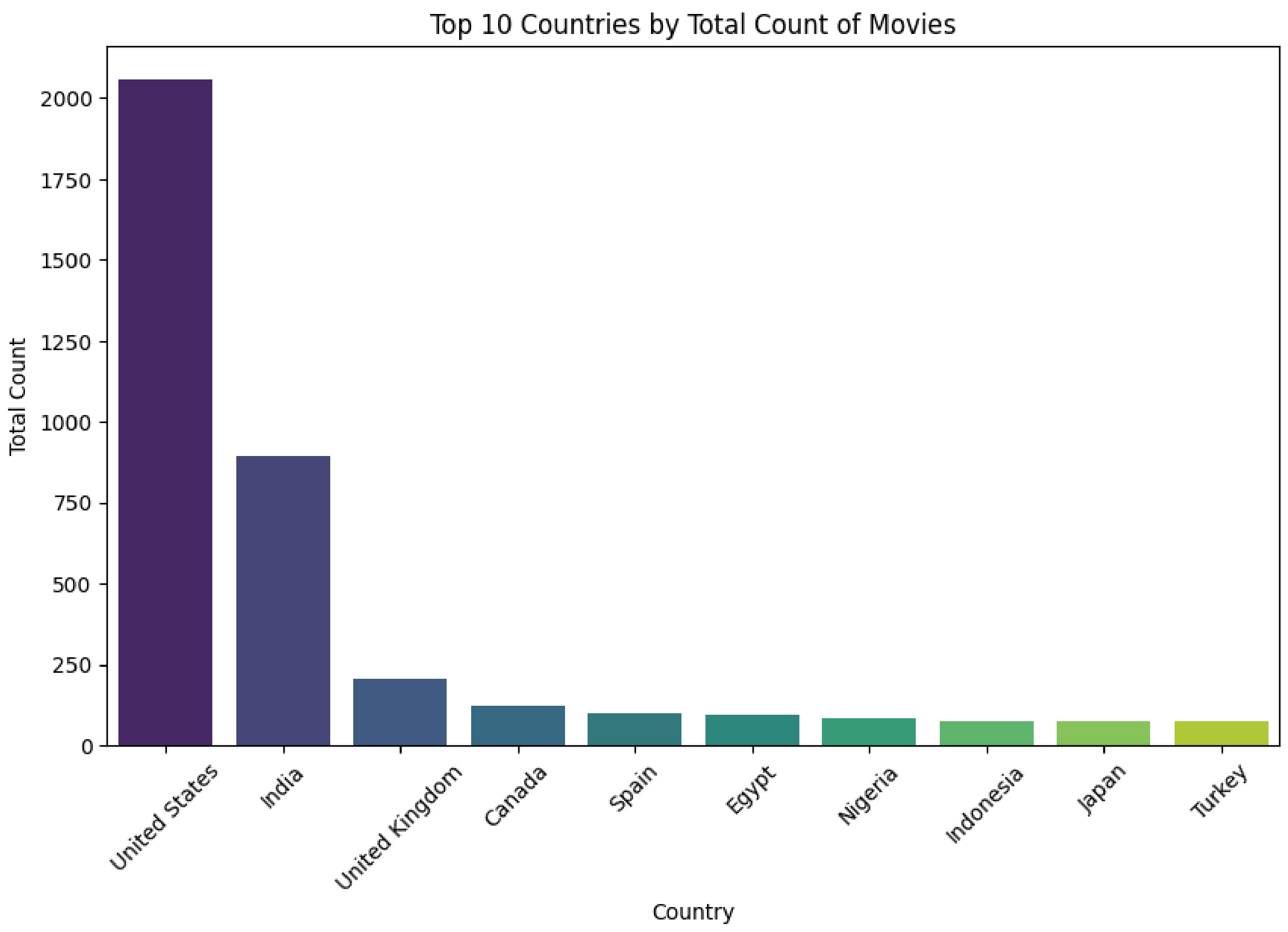
In [616]:
```
top_10_trend_countries_movies = top_10_trend_countries.loc[top_10_trend_countries['ty
top_10_trend_countries_movies = top_10_trend_countries_movies.groupby(['country'])['T
top_10_trend_countries_movies = pd.DataFrame(top_10_trend_countries_movies)
top_10_trend_countries_movies.reset_index(inplace = True)
top_10_trend_countries_movies
```

Out[616]:

	country	Total Count
0	United States	2058
1	India	893
2	United Kingdom	206
3	Canada	122
4	Spain	97
5	Egypt	92
6	Nigeria	86
7	Indonesia	77
8	Japan	76
9	Turkey	76

```
In [624]: plt.figure(figsize=(10, 6))
          sns.barplot(x='country', y='Total Count', data=top_10_trend_countries_movies, palette=

          # Adding titles and labels
          plt.title('Top 10 Countries by Total Count of Movies')
          plt.xlabel('Country')
          plt.ylabel('Total Count')

          # Rotating x-axis labels for better readability
          plt.xticks(rotation=45)

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

```
<ipython-input-624-b757bc22bf21>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='country', y='Total Count', data=top_10_trend_countries_movies, pale
tte='viridis')
```

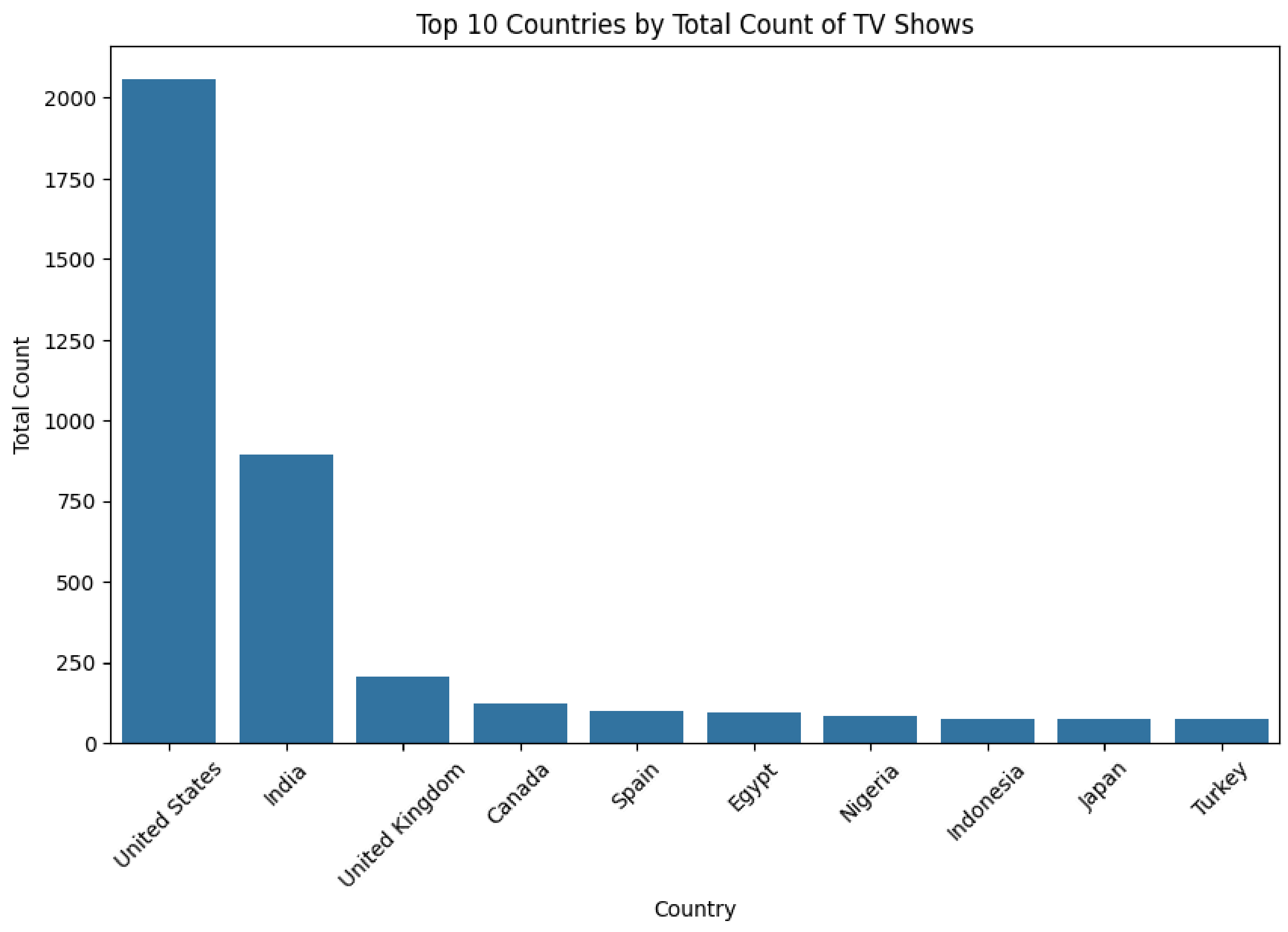


```
In [623]: top_10_trend_countries_tvshows = top_10_trend_countries.loc[top_10_trend_countries['t
          top_10_trend_countries_tvshows = top_10_trend_countries_tvshows.groupby(['country'])[
          top_10_trend_countries_tvshows = pd.DataFrame(top_10_trend_countries_tvshows)
          top_10_trend_countries_tvshows.reset_index(inplace = True)
          top_10_trend_countries_tvshows
```

Out[623]:

	country	Total Count
0	United States	760
1	United Kingdom	213
2	Japan	169
3	South Korea	158
4	India	79
5	Taiwan	68
6	Canada	59
7	France	49
8	Spain	48
9	Australia	48

In [622]:
```python
plt.figure(figsize=(10, 6))
sns.barplot(x='country', y='Total Count', data=top_10_trend_countries_movies)

# Adding titles and labels
plt.title('Top 10 Countries by Total Count of TV Shows')
plt.xlabel('Country')
plt.ylabel('Total Count')

# Rotating x-axis labels for better readability
plt.xticks(rotation=45)

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



Popular Genres

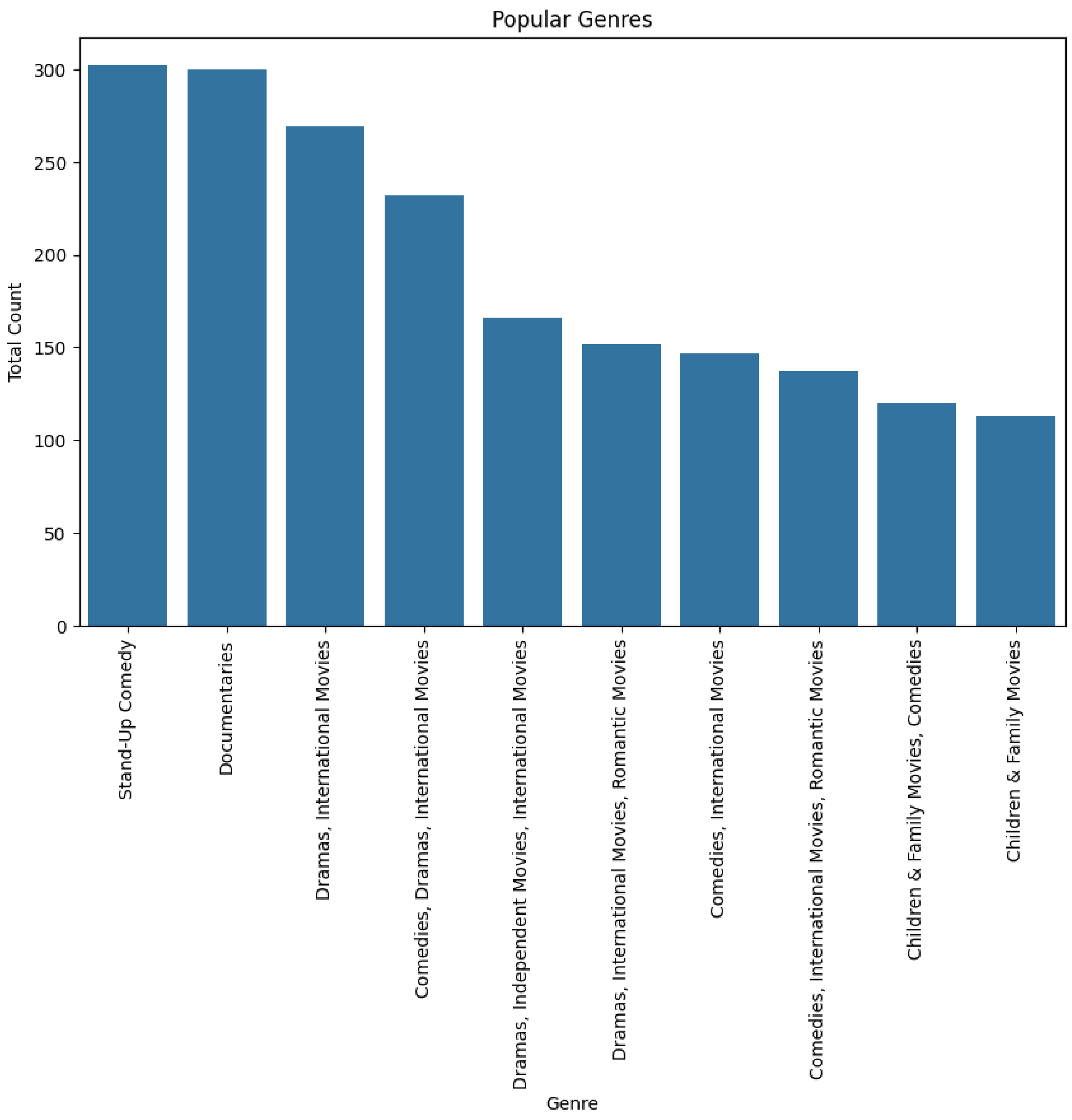
listed_in column

```
In [647]: popular_genres = netflix_df.drop_duplicates(subset = ['title'])
          popular_genres = popular_genres[popular_genres['type']=='Movie']
          popular_genres = pd.DataFrame(popular_genres['listed_in'].value_counts().sort_values(a
          popular_genres.reset_index(inplace = True)
          popular_genres.columns = ['Genre', 'Count']
          popular_genres = popular_genres.head(10)
          popular_genres
```

Out[647]:

	Genre	Count
0	Stand-Up Comedy	302
1	Documentaries	300
2	Dramas, International Movies	269
3	Comedies, Dramas, International Movies	232
4	Dramas, Independent Movies, International Movies	166
5	Dramas, International Movies, Romantic Movies	152
6	Comedies, International Movies	147
7	Comedies, International Movies, Romantic Movies	137
8	Children & Family Movies, Comedies	120
9	Children & Family Movies	113

In [650]:
```python
plt.figure(figsize=(10, 6))
sns.barplot(x='Genre', y='Count', data=popular_genres)

# Adding titles and labels
plt.title('Popular Genres')
plt.xlabel('Genre')
plt.ylabel('Total Count')

# Rotating x-axis labels for better readability
plt.xticks(rotation=90)

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



Inference -

- People tend to have a liking towards reality programs and documentaries which is quite surprising.
- Infotainment is still untapped area for netflix and there is lot opportunity for that.

```
In [ ]:
```