

```
In [524... import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [525... netflix = pd.read_csv('original_netflix.csv')
```

```
In [526... netflix
```

Out[526]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalan... Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...
...	...	...	...	...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	2 Seasons	Kids' TV, Korean TV Shows, TV Comedies	While living alone in a spooky town, a young g...
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	Comedies, Horror Movies	Looking to survive in a world taken over by zo...

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

8807 rows × 12 columns

## BASIC INFORMATION

In [527...

```
#Top 5 rows
netflix.head()
```

Out[527]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalané, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...

In [528...

```
#Last 10 rows
netflix.tail(5)
#Data is correctly inserted
```

Out[528]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
<b>8802</b>	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...
<b>8803</b>	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	2 Seasons	Kids' TV, Korean TV Shows, TV Comedies	While living alone in a spooky town, a young g...
<b>8804</b>	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	Comedies, Horror Movies	Looking to survive in a world taken over by zo...
<b>8805</b>	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...
<b>8806</b>	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

In [529...

```
netflix.dtypes
#Except Release year all other are in object dtypes
```

```
Out[529]: show_id      object
type          object
title         object
director      object
cast          object
country       object
date_added    object
release_year  int64
rating        object
duration      object
listed_in     object
description    object
dtype: object
```

```
In [530... netflix.info()
#There are null data and all are object dtype
```

```
<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 [531... netflix.isna().sum().sort_values(ascending = False)
#Director has the most null values followed by country and cast
#Out of 8807 values director has the highest null values 2634 followed by country and cast with null values of 831 and 825.
```

```
Out[531]: director      2634
country      831
cast         825
date_added   10
rating       4
duration     3
show_id      0
type         0
title        0
release_year 0
listed_in    0
description  0
dtype: int64
```

```
In [532... netflix.shape
#8807 rows and 12 columns
```

```
Out[532]: (8807, 12)
```

```
In [533... netflix.isna().sum().sum()
#Total null values in entire netflix dataframe
```

```
Out[533]: 4307
```

```
In [534... netflix.describe(include = 'object').T #Transpose
```

Out[534]:

	count	unique	top	freq
<b>show_id</b>	8807	8807	s1	1
<b>type</b>	8807	2	Movie	6131
<b>title</b>	8807	8807	Dick Johnson Is Dead	1
<b>director</b>	6173	4528	Rajiv Chilaka	19
<b>cast</b>	7982	7692	David Attenborough	19
<b>country</b>	7976	748	United States	2818
<b>date_added</b>	8797	1767	January 1, 2020	109
<b>rating</b>	8803	17	TV-MA	3207
<b>duration</b>	8804	220	1 Season	1793
<b>listed_in</b>	8807	514	Dramas, International Movies	362
<b>description</b>	8807	8775	Paranormal activity at a lush, abandoned prope...	4

In [535...

```
netflix.head()
```



Out[535]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...

## HANDLING MISSING VALUES

In [536...

```
netflix.director.fillna('No Director', inplace = True)
netflix.country.fillna('No Country', inplace = True)
netflix.date_added.fillna('No date available', inplace = True)
netflix.cast.fillna('No Cast', inplace = True)
```

In [537...

```
netflix.isna().sum()
```

```
Out[537]: show_id      0
          type         0
          title        0
          director     0
          cast         0
          country      0
          date_added   0
          release_year  0
          rating       4
          duration     3
          listed_in    0
          description   0
          dtype: int64
```

In [538...

```
netflix.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        8807 non-null  object 
4   cast            8807 non-null  object 
5   country         8807 non-null  object 
6   date_added      8807 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
```

## Conversion of date\_added (object dtype) to datetime dtype

In [539...

```
netflix['date_add'] = pd.to_datetime(netflix['date_added'], errors = 'coerce')
netflix['month'] = netflix['date_add'].dt.month
netflix['year'] = netflix['date_add'].dt.year
```

# Un-Nesting of Columns

```
In [540... netflix['cast'] = netflix['cast'].apply(lambda x: str(x).split(", ").tolist())
netflix['director'] = netflix['director'].apply(lambda x: str(x).split(", ").tolist())
netflix['listed_in'] = netflix['listed_in'].apply(lambda x: str(x).split(", ").tolist())
```

```
In [541... explode_cast = netflix.explode('cast')
explode_cast = explode_cast[explode_cast['cast'] != 'No Cast']
explode_director = netflix.explode('director')
explode_director = explode_director[explode_director['director'] != 'No Director']
explode_listedin = netflix.explode('listed_in')
```

```
In [542... #Merging all the un-nested columns
cast_data = explode_cast.loc[:, ['title', 'cast']]
director_data = explode_director.loc[:, ['title', 'director']]
listedin_data = explode_listedin.loc[:, ['title', 'listed_in']]

df_1 = cast_data.merge(director_data, on = 'title', how = 'inner')

df_data = df_1.merge(listedin_data, on = 'title', how = 'inner')
```

```
In [543... df_data['listed_in'].value_counts()
```

```
Out[543]: Dramas                21958
           International Movies  21921
           Comedies             16995
           Action & Adventure    8773
           Independent Movies    7112
           Children & Family Movies 6774
           Romantic Movies       5406
           Thrillers             5074
           Horror Movies         3476
           Sci-Fi & Fantasy       2564
           Music & Musicals       2557
           International TV Shows 1481
           Documentaries         1335
           Sports Movies         1200
           Classic Movies        1078
           TV Dramas            1034
           Anime Features        916
           Cult Movies           780
           LGBTQ Movies          696
           Crime TV Shows        556
           Faith & Spirituality   526
           Stand-Up Comedy       475
           TV Comedies           388
           TV Shows              326
           Romantic TV Shows     298
           Movies               272
           TV Action & Adventure  232
           TV Mysteries          201
           British TV Shows      179
           Spanish-Language TV Shows 146
           Kids' TV              135
           TV Horror             119
           Anime Series          113
           Korean TV Shows       101
           Docuseries            87
           TV Thrillers          78
           TV Sci-Fi & Fantasy    62
           Teen TV Shows         48
           Stand-Up Comedy & Talk Shows 31
           Classic & Cult TV      29
           Reality TV            7
           Science & Nature TV    7
           Name: listed_in, dtype: int64
```

# 1. MOVIES vs TV Shows

In [544...

```
netflix.head()
```

Out[544]:

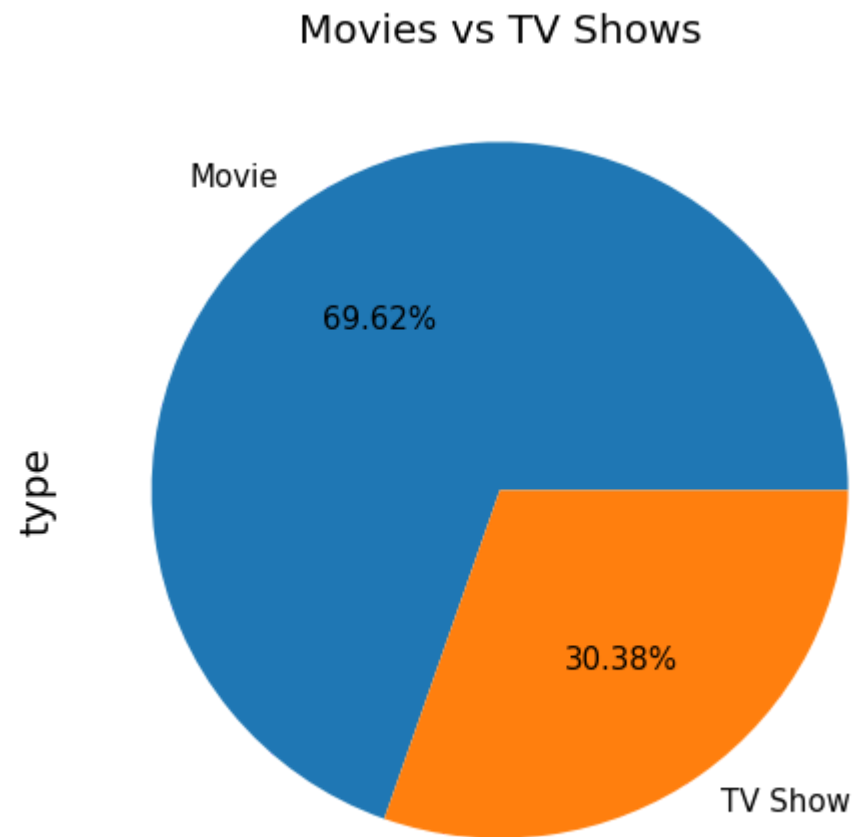
	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	date_add	month
0	s1	Movie	Dick Johnson Is Dead	[Kirsten Johnson]	[No Cast]	United States	September 25, 2021	2020	PG-13	90 min	[Documentaries]	As her father nears the end of his life, filmm...	2021-09-25	9.0
1	s2	TV Show	Blood & Water	[No Director]	[Ama Qamata, Khosi Ngema, Gail Mabalane, Thaba...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	[International TV Shows, TV Dramas, TV Mysteries]	After crossing paths at a party, a Cape Town t...	2021-09-24	9.0
2	s3	TV Show	Ganglands	[Julien Leclercq]	[Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nab...	No Country	September 24, 2021	2021	TV-MA	1 Season	[Crime TV Shows, International TV Shows, TV Ac...	To protect his family from a powerful drug lor...	2021-09-24	9.0
3	s4	TV Show	Jailbirds New Orleans	[No Director]	[No Cast]	No Country	September 24, 2021	2021	TV-MA	1 Season	[Docuseries, Reality TV]	Feuds, flirtations and toilet talk go down amo...	2021-09-24	9.0
4	s5	TV Show	Kota Factory	[No Director]	[Mayur More, Jitendra Kumar, Ranjan Raj, Alam ...	India	September 24, 2021	2021	TV-MA	2 Seasons	[International TV Shows, Romantic TV Shows, TV...	In a city of coaching centers known to train l...	2021-09-24	9.0

In [583...

```
# PIE CHART_Univariate analysis

Title_percent = netflix['type'].value_counts()/len(netflix)*100
```

```
plt.figure(figsize=(8, 8))
Title_percent.plot(kind='pie', y = Title_percent, labels=['Movie', 'TV Show'], autopct='%.2f%', fontsize = 15)
plt.ylabel('type', fontsize = 20)
plt.title('Movies vs TV Shows', fontsize = 20)
plt.show()
'''
Movie has a higher percentage of 69.62% compared to TV Shows 30.38%
'''
```



Out[583]: '\nMovie has a higher percentage of 69.62% compared to TV Shows 30.38%\n'

## 2. Top 10 countries producing movies or TV Shows?

```
In [546... netflix['country'].value_counts().head(10)
```

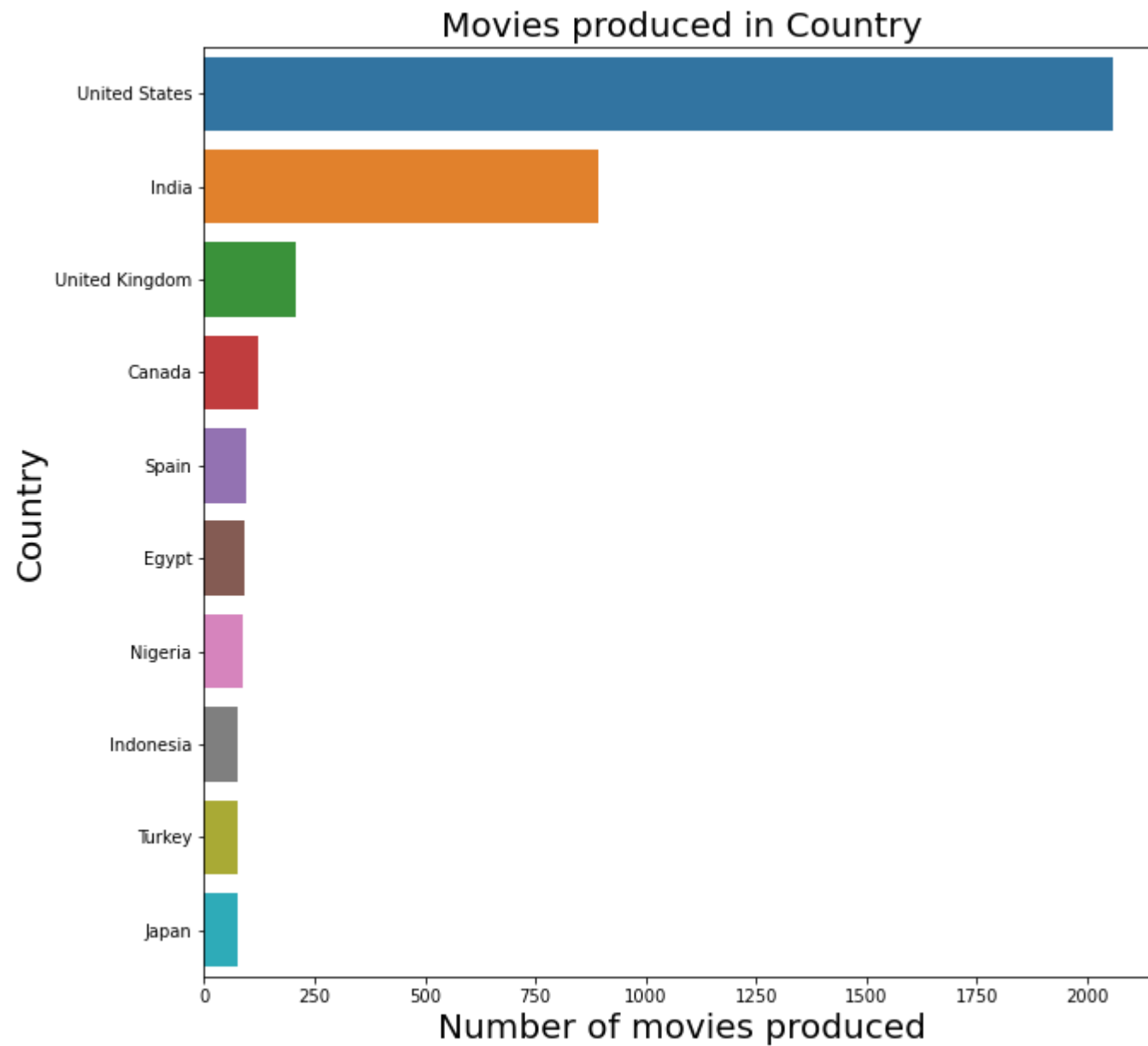
```
Out[546]: United States    2818  
India                972  
No Country           831  
United Kingdom       419  
Japan                245  
South Korea          199  
Canada               181  
Spain               145  
France              124  
Mexico              110  
Name: country, dtype: int64
```

```
In [584... # BAR PLOT: Countries where movies are produced?
```

```
Movies = netflix[netflix['type'] == 'Movie']  
Movies_country = Movies[Movies['country']!= 'No Country']  
  
plt.figure(figsize = (10, 10))  
  
sns.barplot(x = Movies_country['country'].value_counts().head(10),  
            y = Movies_country['country'].value_counts().head(10).index, data = Movies)  
plt.ylabel('Country', fontsize = 20)  
plt.xlabel('Number of movies produced', fontsize = 20)  
plt.title('Movies produced in Country', fontsize = 20)  
plt.show()
```

```
#As per data top 5 country produced the most Movies are United States, India, United Kingdom, Canada and Spain
```





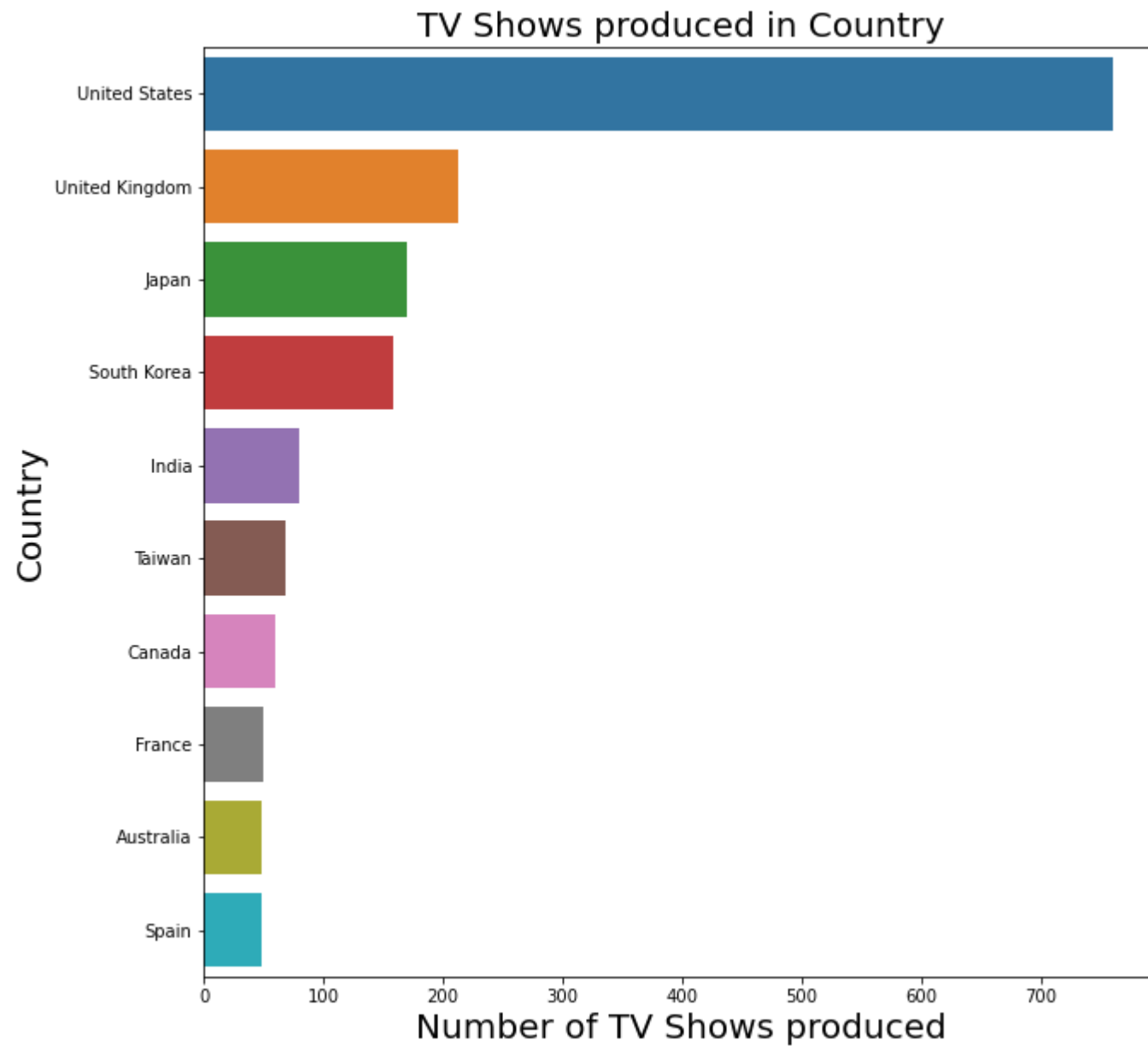
In [585...

```
# BAR PLOT: Countries where TV Shows are produced?  
  
TV_shows = netflix[netflix['type'] == 'TV Show']  
TV_shows_country = TV_shows[TV_shows['country'] != 'No Country']
```

```
plt.figure(figsize = (10, 10))

sns.barplot(x = TV_shows_country['country'].value_counts().head(10),
            y = TV_shows_country['country'].value_counts().head(10).index, data = TV_shows)
plt.ylabel('Country', fontsize = 20)
plt.xlabel('Number of TV Shows produced', fontsize = 20)
plt.title('TV Shows produced in Country', fontsize = 20)
plt.show()
```

*#As per data top 5 country produced the most TV Shows are United States, United Kingdom, Japan, South Korea and India*



3. Most productive movie director?

```
In [549... explode_director['director'].value_counts()
```

```
Out[549]: Rajiv Chilaka      22  
          Jan Suter      21  
          Raúl Campos    19  
          Suhas Kadav    16  
          Marcus Raboy   16  
          ..  
          Raymie Muzquiz  1  
          Stu Livingston  1  
          Joe Menendez   1  
          Eric Bross     1  
          Mozez Singh    1  
          Name: director, Length: 4993, dtype: int64
```

```
In [550... movie_data = explode_director[explode_director['type'] == 'Movie']  
Prod_dir = movie_data.groupby('director')[['title', 'release_year']].aggregate({'release_year': ['min', 'max'],  
                                                                                'title': 'count'})
```

```
In [551... Prod_dir
```

Out[551]:

	release_year		title
	min	max	count
director			
A. L. Vijay	2016	2019	2
A. Raajdheep	2020	2020	1
A. Salaam	1975	1975	1
A.R. Murugadoss	2017	2018	2
Aadish Keluskar	2018	2018	1
...	...	...	...
Éric Warin	2016	2016	1
Ísold Uggadóttir	2018	2018	1
Óskar Thór Axelsson	2017	2017	1
Ömer Faruk Sorak	2001	2011	3
Şenol Sönmez	2015	2019	2

4777 rows × 3 columns

In [552...]: Prod\_dir.columns

Out[552]: MultiIndex([('release\_year', 'min'),  
             ('release\_year', 'max'),  
             ('title', 'count')],  
             )

In [553...]: Prod\_dir['release\_year']

Out[553]:

	min	max
director		
A. L. Vijay	2016	2019
A. Raajdheep	2020	2020
A. Salaam	1975	1975
A.R. Murugadoss	2017	2018
Aadish Keluskar	2018	2018
...	...	...
Éric Warin	2016	2016
Ísold Uggadóttir	2018	2018
Óskar Thór Axelsson	2017	2017
Ömer Faruk Sorak	2001	2011
Şenol Sönmez	2015	2019

4777 rows × 2 columns

```
In [554... [i for i in Prod_dir.columns]
```

```
Out[554]: [('release_year', 'min'), ('release_year', 'max'), ('title', 'count')]
```

```
In [555... ['-'.join(i) for i in Prod_dir.columns]
```

```
Out[555]: ['release_year-min', 'release_year-max', 'title-count']
```

```
In [556... Prod_dir.columns = ['-'.join(i) for i in Prod_dir.columns]
```

```
In [557... Prod_dir.reset_index()
```

Out[557]:

	director	release_year-min	release_year-max	title-count
0	A. L. Vijay	2016	2019	2
1	A. Raajidheep	2020	2020	1
2	A. Salaam	1975	1975	1
3	A.R. Murugadoss	2017	2018	2
4	Aadish Keluskar	2018	2018	1
...	...	...	...	...
4772	Éric Warin	2016	2016	1
4773	Ísold Uggadóttir	2018	2018	1
4774	Óskar Thór Axelsson	2017	2017	1
4775	Ömer Faruk Sorak	2001	2011	3
4776	Şenol Sönmez	2015	2019	2

4777 rows × 4 columns

```
In [558... Prod_dir['years_active'] = Prod_dir['release_year-max'] - Prod_dir['release_year-min']
```

```
In [559... Prod_dir['per_year'] = Prod_dir['title-count']/Prod_dir['years_active']  
Prod_dir.sort_values('title-count', ascending = False)
```

Out[559]:

	release_year-min	release_year-max	title-count	years_active	per_year
director					
Rajiv Chilaka	2009	2019	22	10	2.200000
Jan Suter	2016	2018	21	2	10.500000
Raúl Campos	2016	2018	19	2	9.500000
Suhas Kadav	2013	2019	16	6	2.666667
Marcus Raboy	2012	2020	15	8	1.875000
...	...	...	...	...	...
José Ortuño	2018	2018	1	0	inf
Bob Persichetti	2018	2018	1	0	inf
Jovanka Vuckovic	2017	2017	1	0	inf
Bob Odenkirk	2003	2003	1	0	inf
Awı Suryadi	2017	2017	1	0	inf

4777 rows × 5 columns

In [560...

```
#Director's data of 10 years

'''
Director Rajiv Chilaka has produced a total of 22 movies in the last 10 years with 2.2 movies/year
therefore the most productive director

'''

Data = Prod_dir[Prod_dir['years_active'] == 10]
Data.sort_values('per_year',ascending = False).head(50)
```



Out[560]:

	release_year-min	release_year-max	title-count	years_active	per_year
director					
Rajiv Chilaka	2009	2019	22	10	2.2
Shannon Hartman	2009	2019	9	10	0.9
Ozan Açıktan	2010	2020	7	10	0.7
Priyadarshan	2006	2016	6	10	0.6
Matt Askem	2008	2018	6	10	0.6
Ava DuVernay	2008	2018	5	10	0.5
Prakash Jha	2003	2013	5	10	0.5
Michael Simon	2008	2018	5	10	0.5
Clay Glen	2011	2021	4	10	0.4
Farah Khan	2004	2014	4	10	0.4
Farhan Akhtar	2001	2011	4	10	0.4
Vikramaditya Motwane	2010	2020	3	10	0.3
Ribhu Dasgupta	2011	2021	3	10	0.3
Pablo Larraín	2006	2016	3	10	0.3
Noriyuki Abe	2008	2018	3	10	0.3
Neeraj Pandey	2008	2018	3	10	0.3
Neal Brennan	2009	2019	3	10	0.3
Dante Lam	2008	2018	3	10	0.3
Ömer Faruk Sorak	2001	2011	3	10	0.3
Michael J. Bassett	2009	2019	2	10	0.2
Lee Hirsch	2002	2012	2	10	0.2
Jonathan Hensleigh	2011	2021	2	10	0.2
Erik Canuel	2006	2016	2	10	0.2

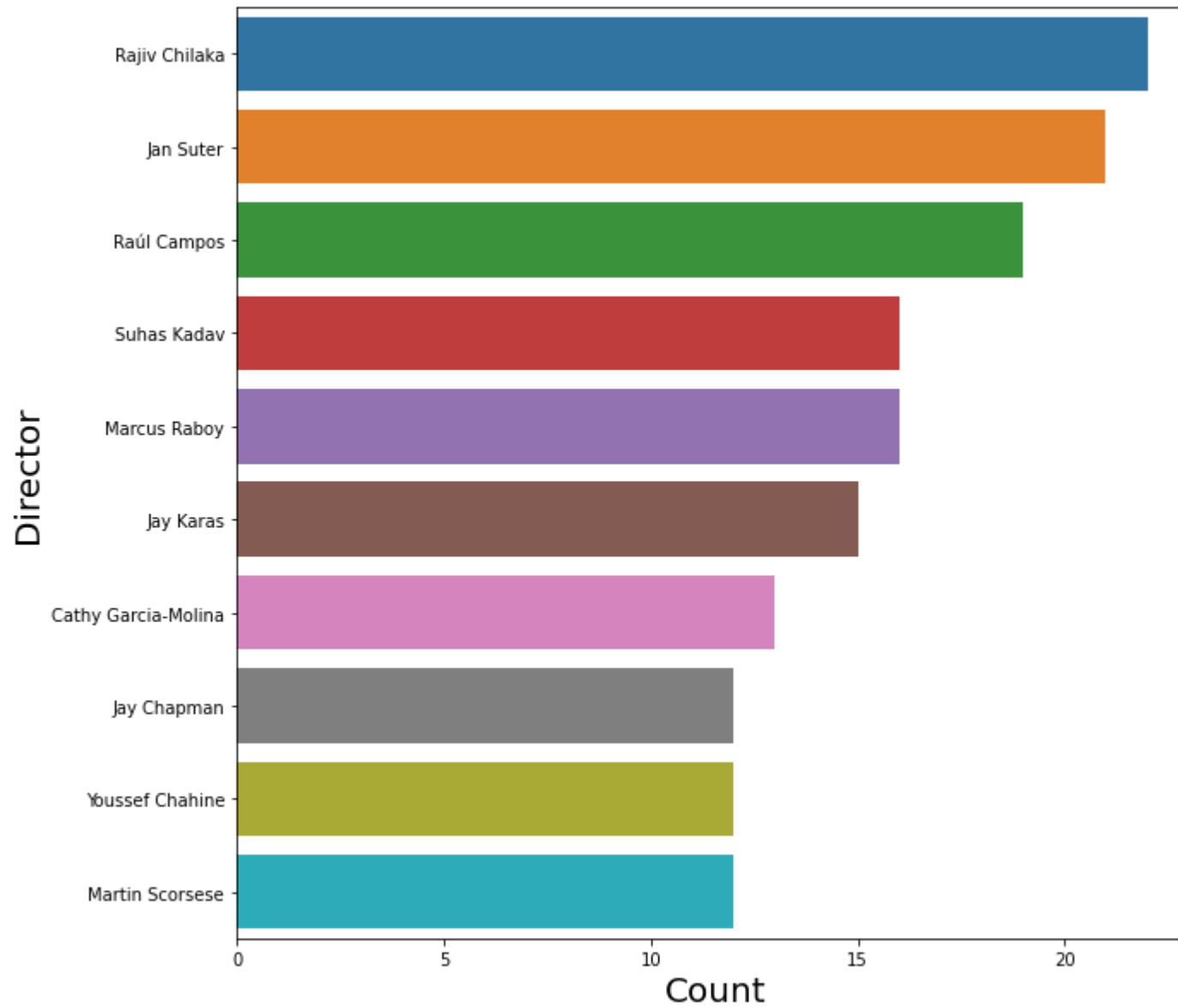
	release_year-min	release_year-max	title-count	years_active	per_year
director					
Ryan Murphy	2010	2020	2	10	0.2
Dominic Sena	2001	2011	2	10	0.2
Stephen Chow	1994	2004	2	10	0.2
Suparn Verma	2005	2015	2	10	0.2
Tyler Perry	2010	2020	2	10	0.2

```
In [561... #There are total 4777 directors producing movies-
movie_data['director'].nunique()
```

```
Out[561]: 4777
```

```
In [562... #Best Directors:
plt.figure(figsize = (10, 10))
sns.barplot(x = explode_director['director'].value_counts().head(10),
            y = explode_director['director'].value_counts().head(10).index, data = explode_director)
plt.title('Movie Directors content', fontsize = 30)
plt.xlabel('Count', fontsize = 20)
plt.ylabel('Director', fontsize = 20)
plt.show()
```

## Movie Directors content

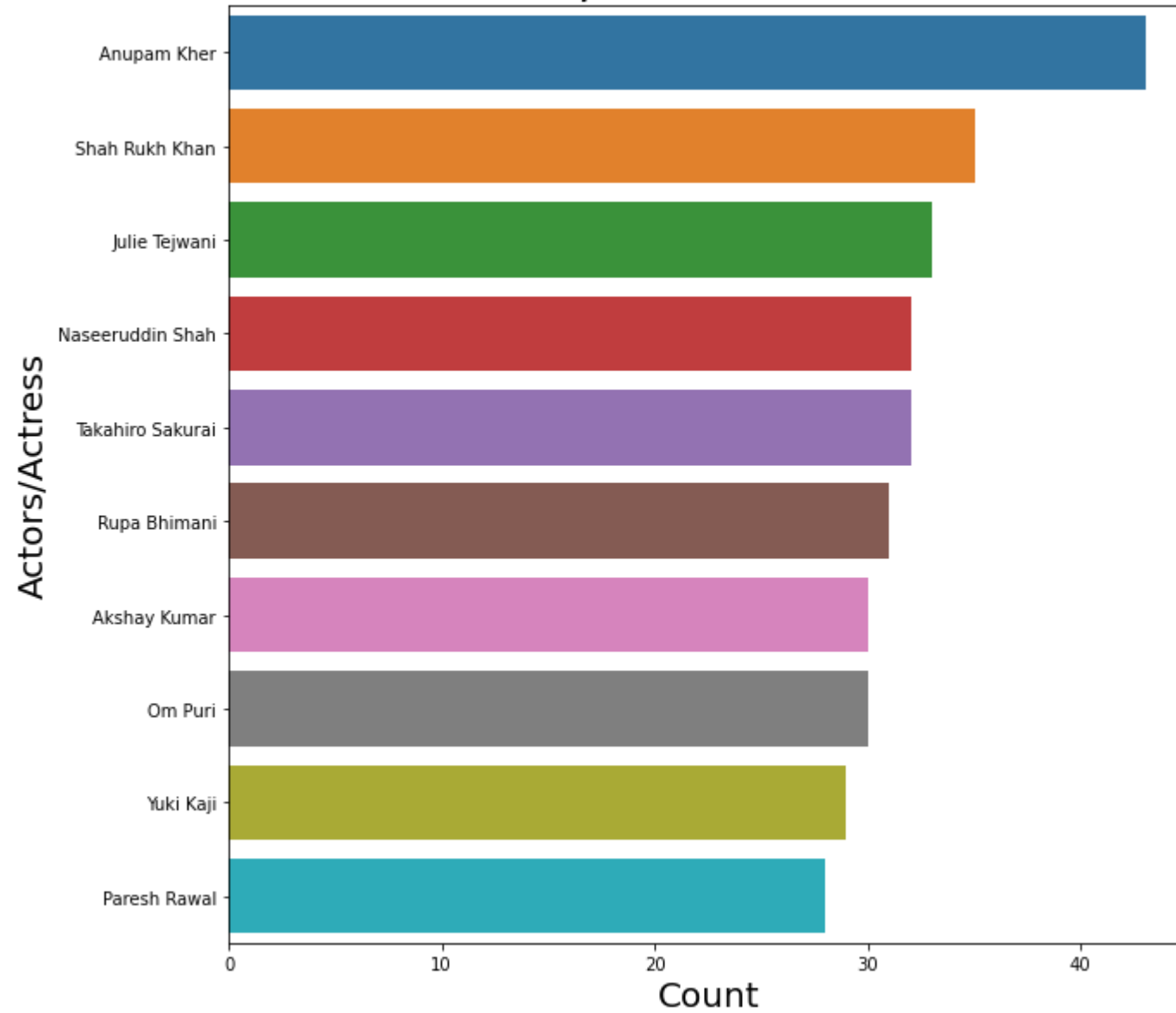


In [563... `explode_cast['cast'].value_counts()`

```
Out[563]: Anupam Kher          43
          Shah Rukh Khan     35
          Julie Teiwani      33
          Naseeruddin Shah   32
          Takahiro Sakurai   32
          ..
          Maryam Zaree       1
          Melanie Straub     1
          Gabriela Maria Schmeide 1
          Helena Zengel      1
          Chittaranjan Tripathy 1
          Name: cast, Length: 36439, dtype: int64
```

```
In [564... #Best Actors/Actress:
plt.figure(figsize = (10, 10))
sns.barplot(x = explode_cast['cast'].value_counts().head(10),
            y = explode_cast['cast'].value_counts().head(10).index, data = explode_cast)
plt.title('Actors/Actress content', fontsize = 30)
plt.xlabel('Count', fontsize = 20)
plt.ylabel('Actors/Actress', fontsize = 20)
plt.show()
```

## Actors/Actress content

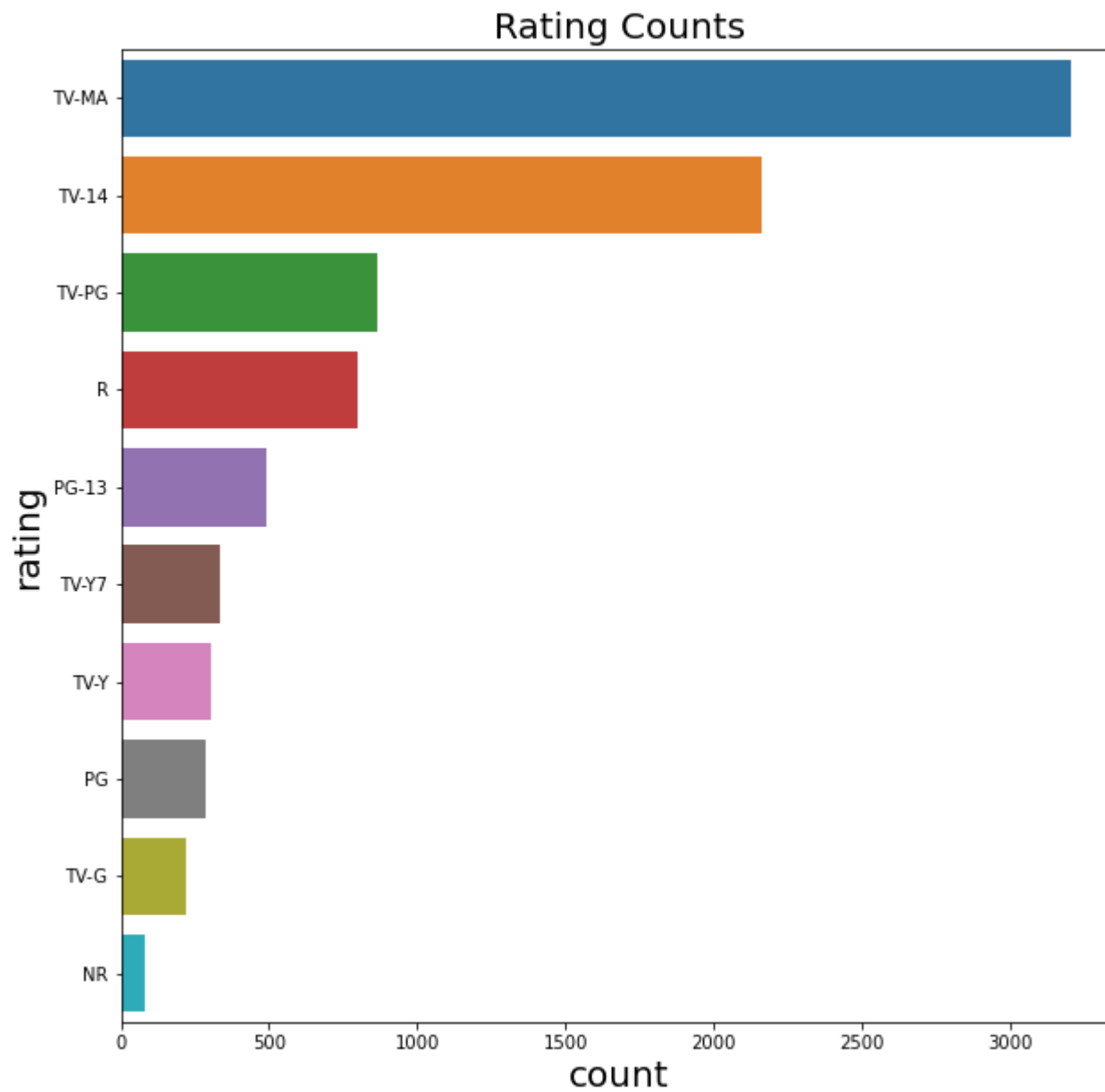


Top ratings

```
In [565... #Which title has got the highest rating?  
netflix['rating'].value_counts()
```

```
Out[565]: TV-MA      3207  
TV-14      2160  
TV-PG       863  
R           799  
PG-13       490  
TV-Y7       334  
TV-Y        307  
PG          287  
TV-G        220  
NR          80  
G           41  
TV-Y7-FV     6  
NC-17        3  
UR           3  
74 min       1  
84 min       1  
66 min       1  
Name: rating, dtype: int64
```

```
In [586... plt.figure(figsize = (10, 10))  
  
sns.barplot(x = netflix['rating'].value_counts().head(10),  
            y = netflix['rating'].value_counts().head(10).index, data = netflix)  
plt.ylabel('rating', fontsize = 20)  
plt.xlabel('count', fontsize = 20)  
plt.title('Rating Counts', fontsize = 20)  
plt.show()  
#TV-MA has the highest count, TV-MA means Mature Audience only for above 17 years of age
```



```
In [587... #Difference in Movie release and Movie added to netflix with ratings
netflix['year difference'] = netflix['year'] - netflix['release_year']

plt.figure(figsize = (20, 15))
```

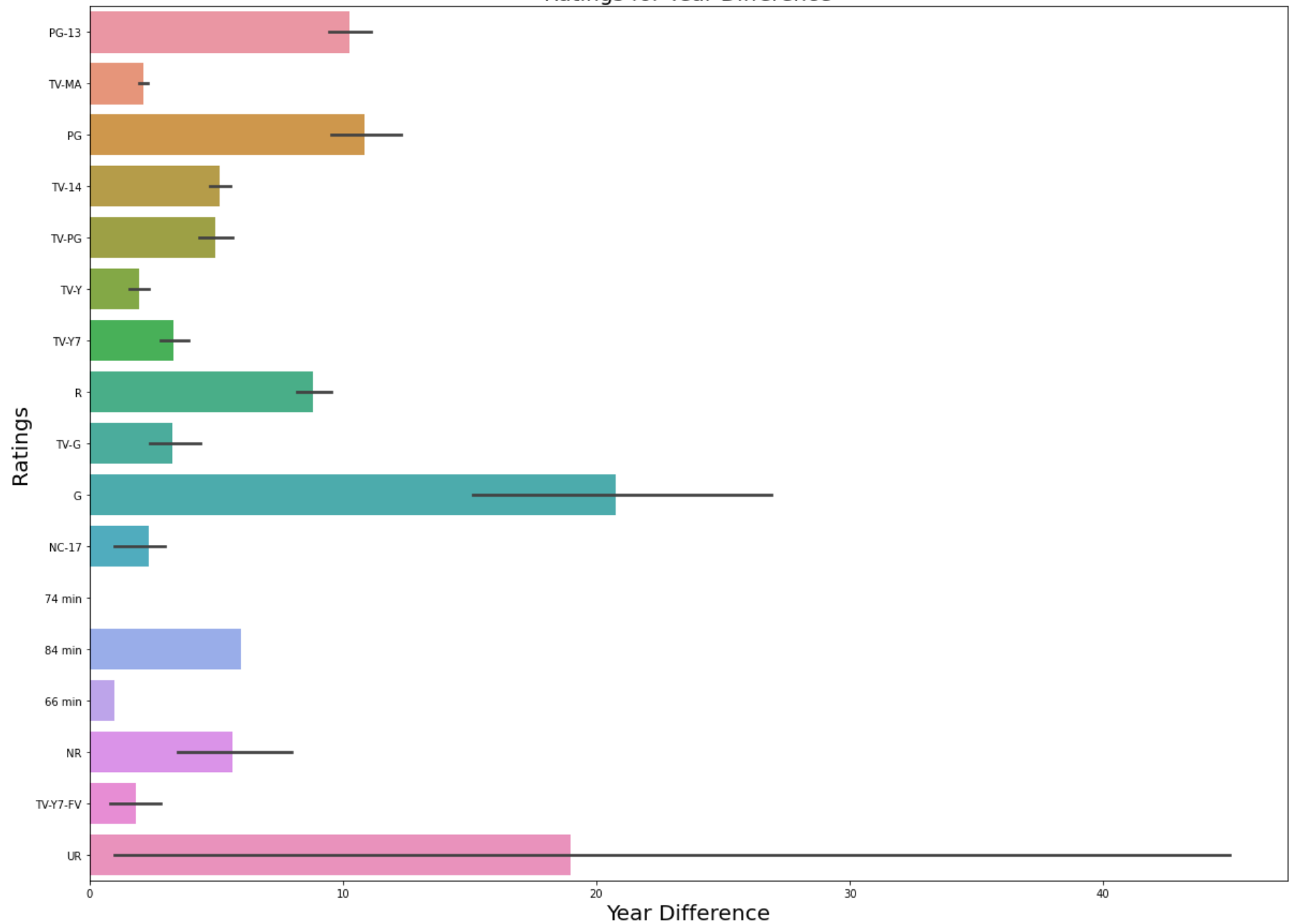
```
sns.barplot(x = 'year difference', y = 'rating', data = netflix)
plt.ylabel('Ratings', fontsize = 20)
plt.xlabel('Year Difference', fontsize = 20)
plt.title('Ratings for Year Difference', fontsize = 20)
plt.show()
'''
```

From the data we observe that TV\_MA which is highly rated has lesser time difference

```
'''
```



Ratings for Year Difference



```
Out[587]: '\nFrom the data we observe that TV_MA which is highly rated has lesser time difference\n\n\n'
```

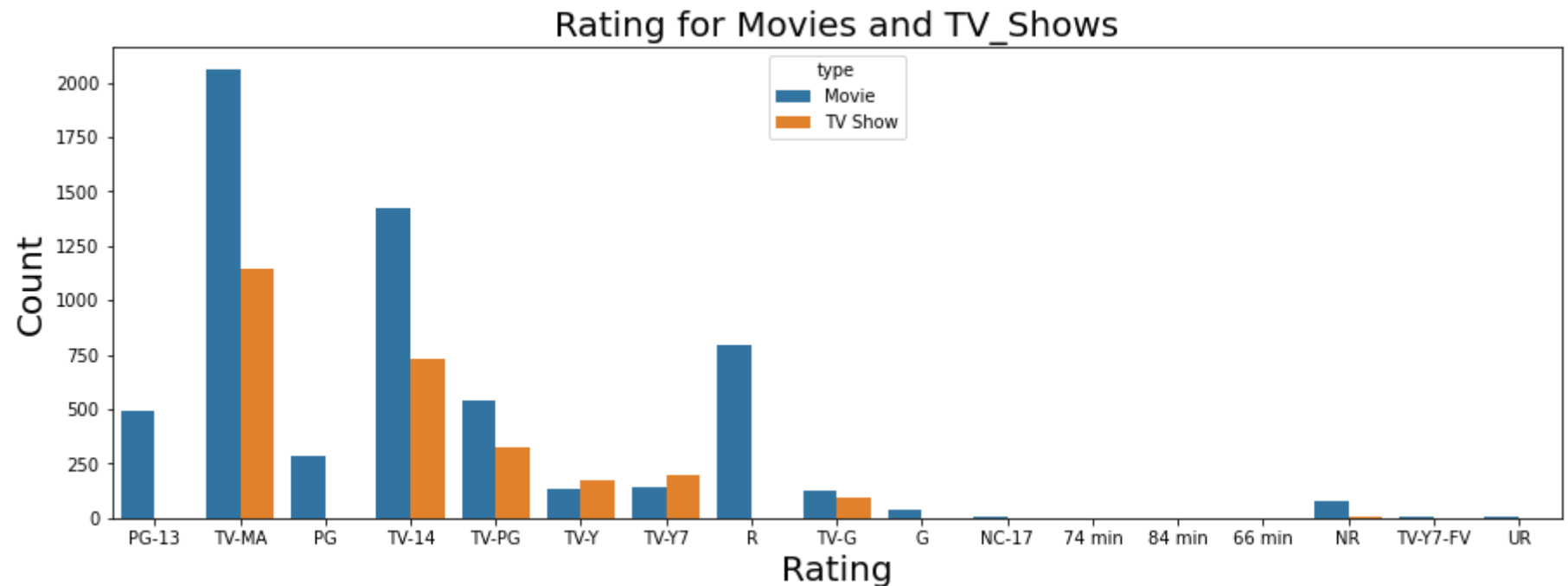
```
In [568... ##Categorical and Categorical:
```

```
plt.figure(figsize = (15, 5))
sns.countplot(x = 'rating', hue = 'type', data = netflix)
plt.title('Rating for Movies and TV_Shows', fontsize = 20)
plt.ylabel('Count', fontsize = 20)
plt.xlabel('Rating', fontsize = 20)
```

```
plt.show()
'''
```

Overall rating is more for movies than TV\_Shows, where TV\_MA is highly rated for movies as well as TV\_shows

```
'''
```



```
Out[568]: '\nOverall rating is more for movies than TV_Shows, where TV_MA is highly rated for movies as well as TV_shows\n\n'
```

# Release Year

```
In [569... netflix['release_year'].value_counts()
```

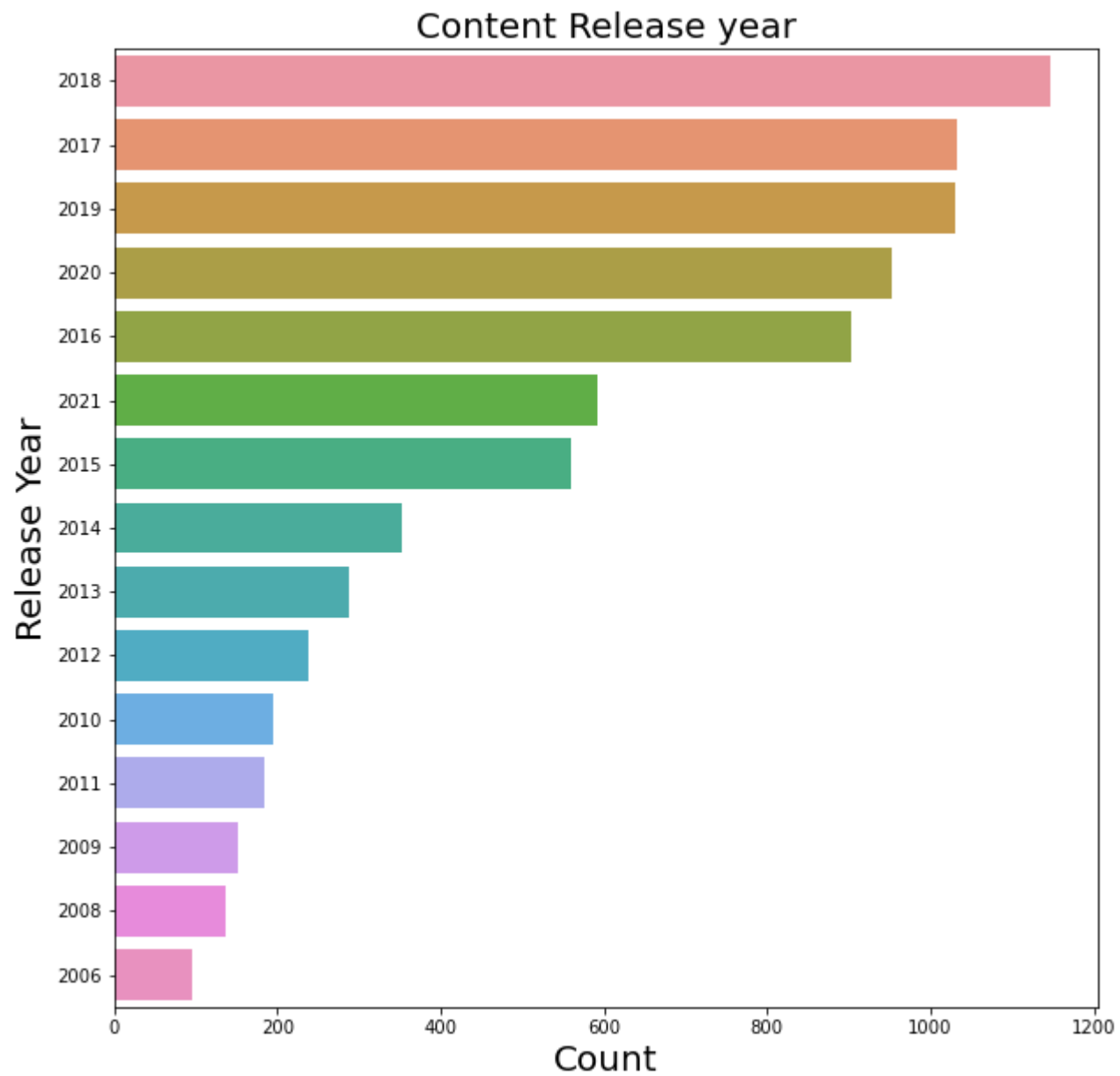
```
Out[569]: 2018    1147
          2017    1032
          2019    1030
          2020     953
          2016     902
          ...
          1959      1
          1925      1
          1961      1
          1947      1
          1966      1
          Name: release_year, Length: 74, dtype: int64
```

```
In [588... # COUNTPLOT
plt.figure(figsize = (10, 10))
ax = sns.countplot(y = 'release_year', data = netflix, order = netflix.release_year.value_counts().index[0:15])
plt.title('Content Release year', fontsize = 20)
plt.ylabel('Release Year', fontsize = 20)
plt.xlabel('Count', fontsize = 20)
plt.show()

...

Movies/TVShows were mostly release in the year 2018

...
```



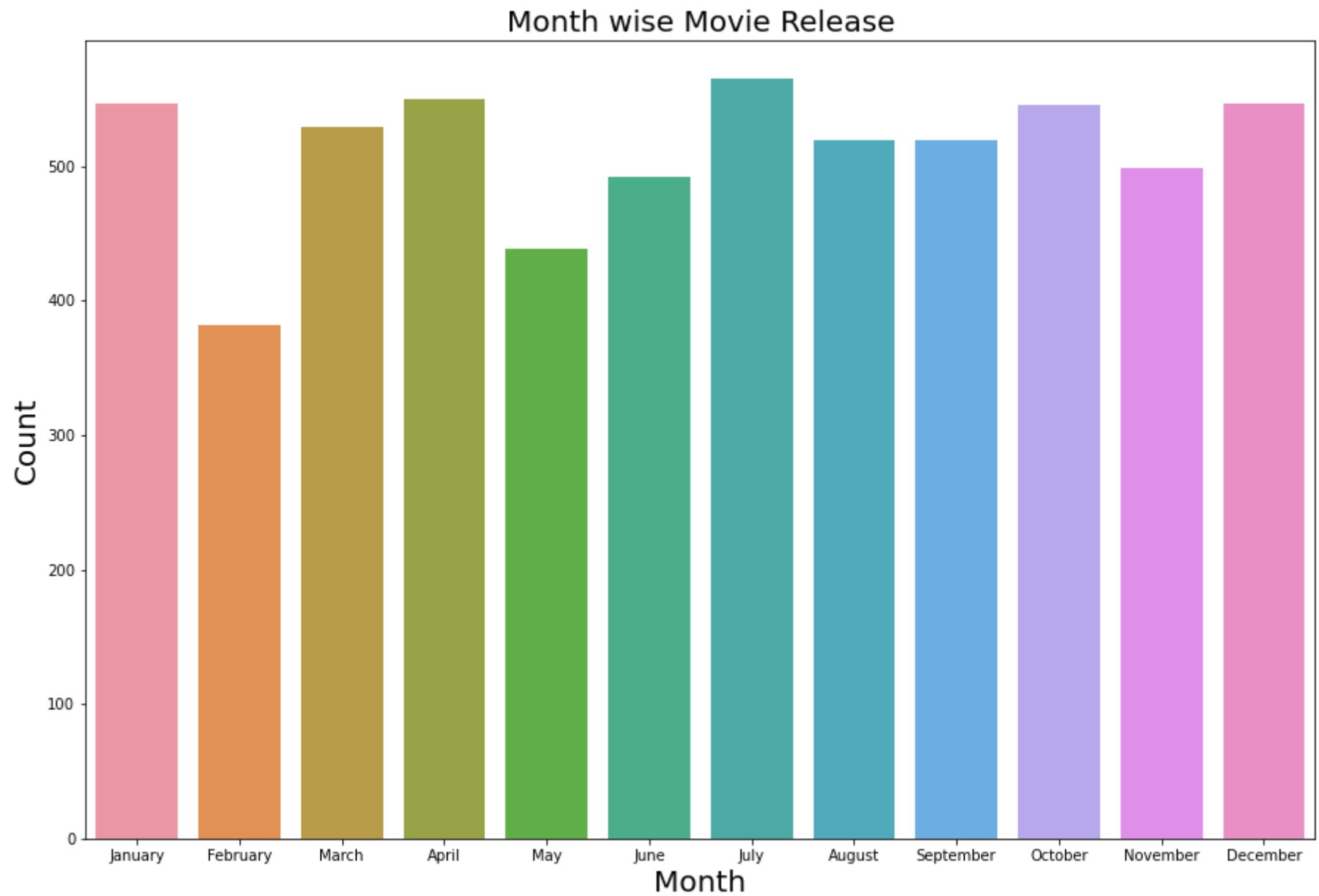
Out[588]: '\nMovies/TVShows were mostly release in the year 2018\n\n'

In [571... *#Month wise release information of Movies*

```
plt.figure(figsize = (15, 10))
```

```
Movie_release = netflix[netflix['type'] == 'Movie']
plot = sns.countplot(x = 'month', data = Movie_release)
plot.set_xticklabels(['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September',
                     'October', 'November', 'December'])
plt.title('Month wise Movie Release', fontsize = 20)
plt.ylabel('Count', fontsize = 20)
plt.xlabel('Month', fontsize = 20)
plt.show()

'''
Netflix releases the movies in all the months except July month is at higher side
'''
```

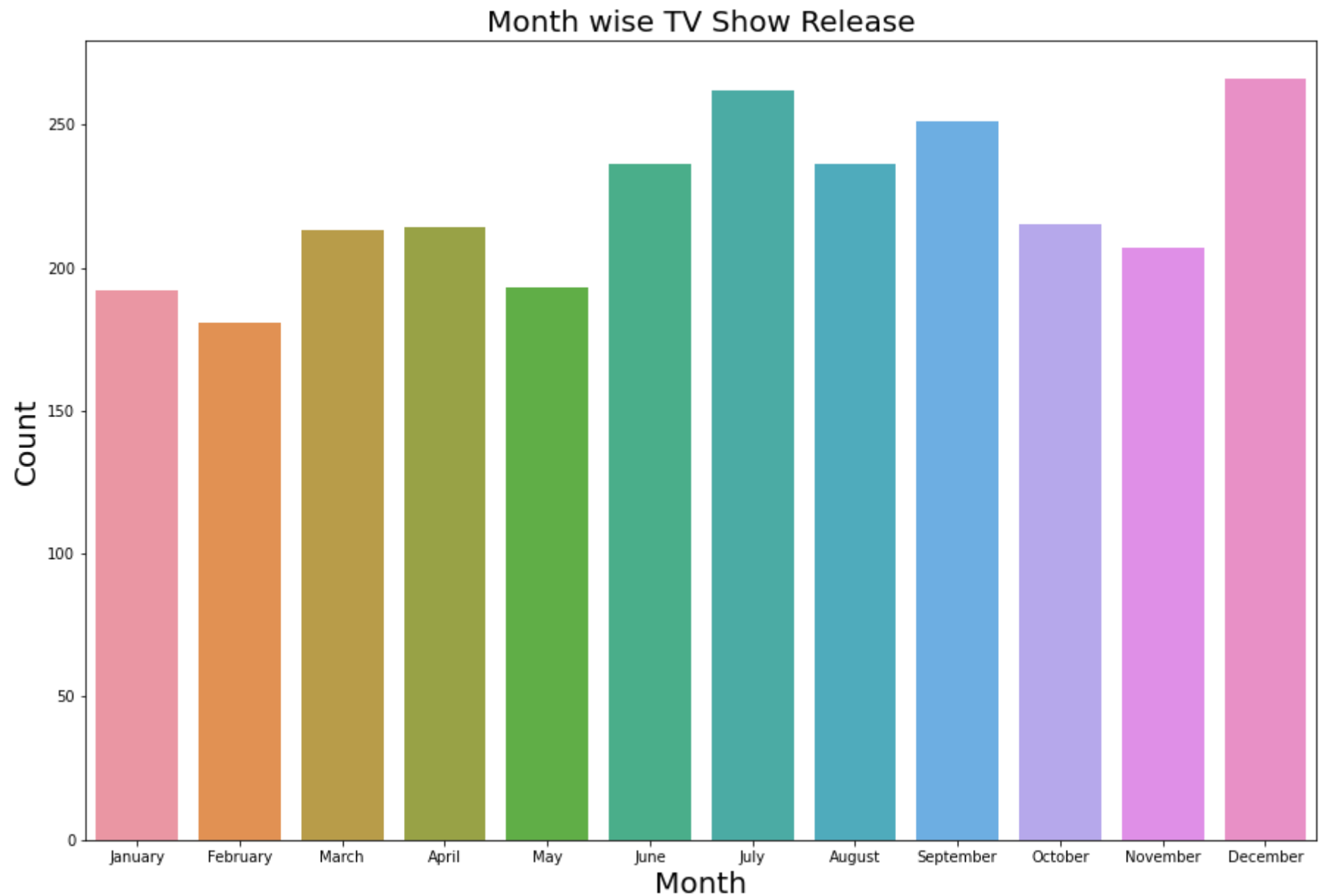


```
Out[571]: '\nNetflix releases the movies in all the months except July month is at higher side\n'
```

```
In [572... #Month wise release information of TV Shows
```

```
plt.figure(figsize = (15, 10))
TVShow_release = netflix[netflix['type'] == 'TV Show']
plot = sns.countplot(x = 'month', data = TVShow_release)
plot.set_xticklabels(['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September',
                     'October', 'November', 'December'])
plt.title('Month wise TV Show Release', fontsize = 20)
plt.ylabel('Count', fontsize = 20)
plt.xlabel('Month', fontsize = 20)
plt.show()

'''
Netflix release the Tv Show majorly in month of December and July
'''
```



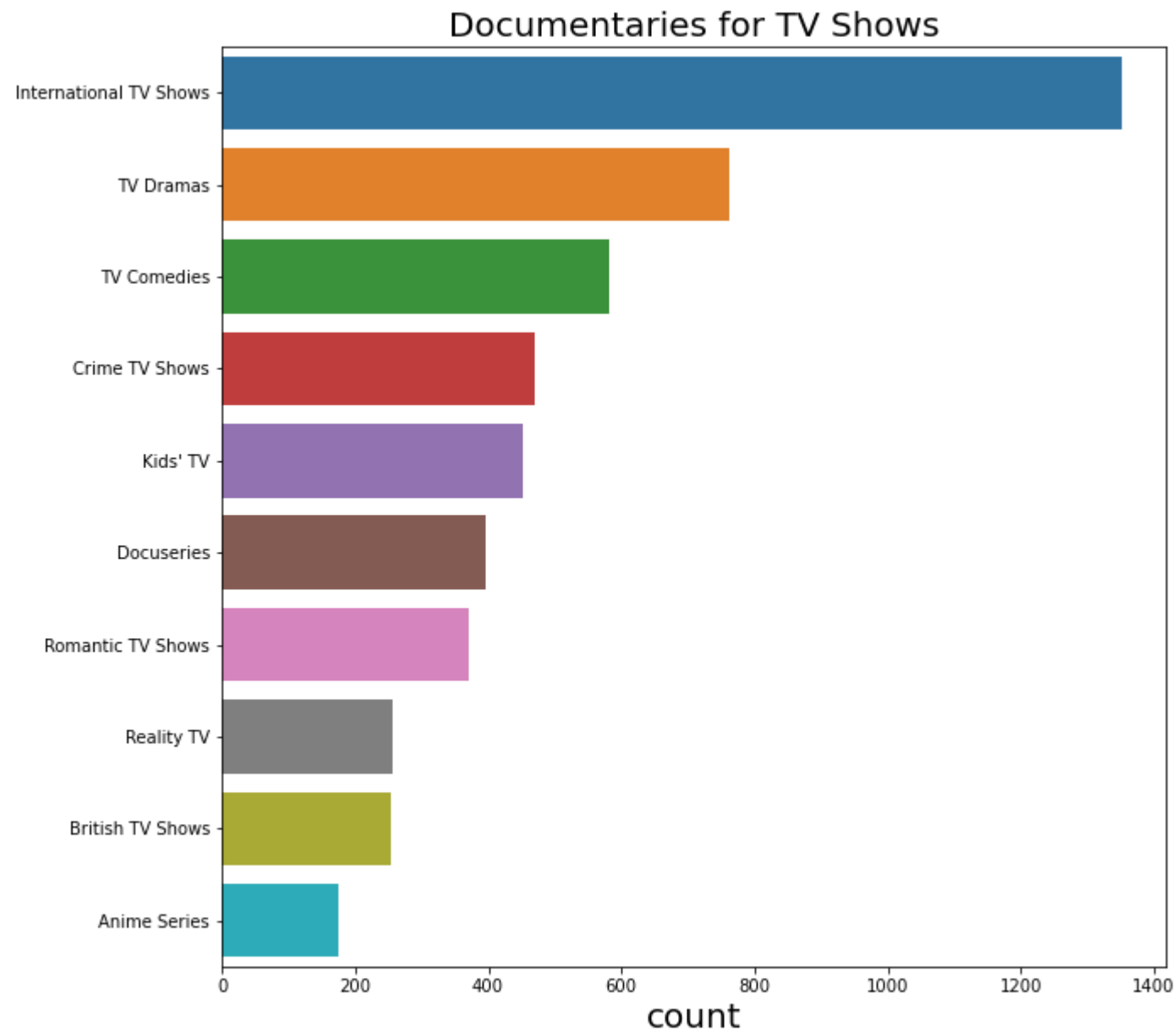
Out[572]: '\nNetflix release the Tv Show majorly in month of December and July\n\n'

```
In [589... plt.figure(figsize = (10, 10))
TVShow_release = explode_listedin[explode_listedin['type'] == 'TV Show']
sns.barplot(x = TVShow_release['listed_in'].value_counts().head(10),
```



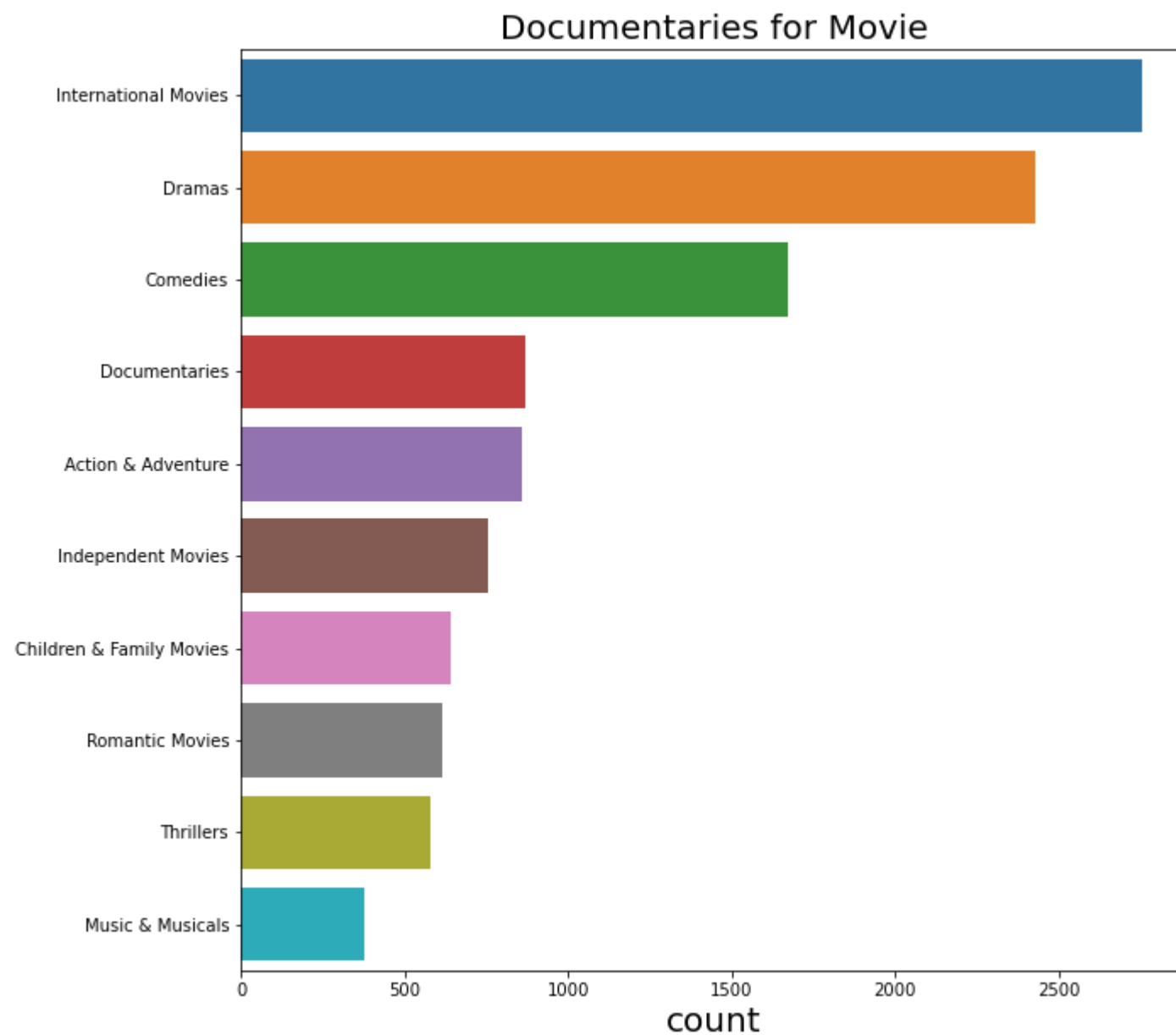
```
y = TVShow_release['listed_in'].value_counts().head(10).index, data = TVShow_release )

plt.xlabel('count', fontsize = 20)
plt.title('Documentaries for TV Shows', fontsize = 20)
plt.show()
```



```
In [590... plt.figure(figsize = (10, 10))
Movie_list = explode_listedin[explode_listedin['type'] == 'Movie']
sns.barplot(x = Movie_list['listed_in'].value_counts().head(10),
            y = Movie_list['listed_in'].value_counts().head(10).index, data = Movie_list )
```

```
plt.xlabel('count', fontsize = 20)
plt.title('Documentaries for Movie', fontsize = 20)
plt.show()
```



# Duration of Movies and TV Shows on Netflix??

```
In [575... Movies['duration'] = pd.to_numeric(Movies['duration'].str.split(" ").str[0], errors = 'coerce').convert_dtypes()
```

C:\Users\Chanchal Gupta\AppData\Local\Temp\ipykernel\_5500\293250842.py: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](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
Movies['duration'] = pd.to_numeric(Movies['duration'].str.split(" ").str[0], errors = 'coerce').convert_dtypes()
```

```
In [576... Movies['duration'].value_counts().sort_values(ascending = False)
```

*#Mostly Duration of Movie falls within 90 to 120min (1.5 to 2 hrs)*

Out[576]:

```
90      152
94      146
93      146
97      146
91      144
```

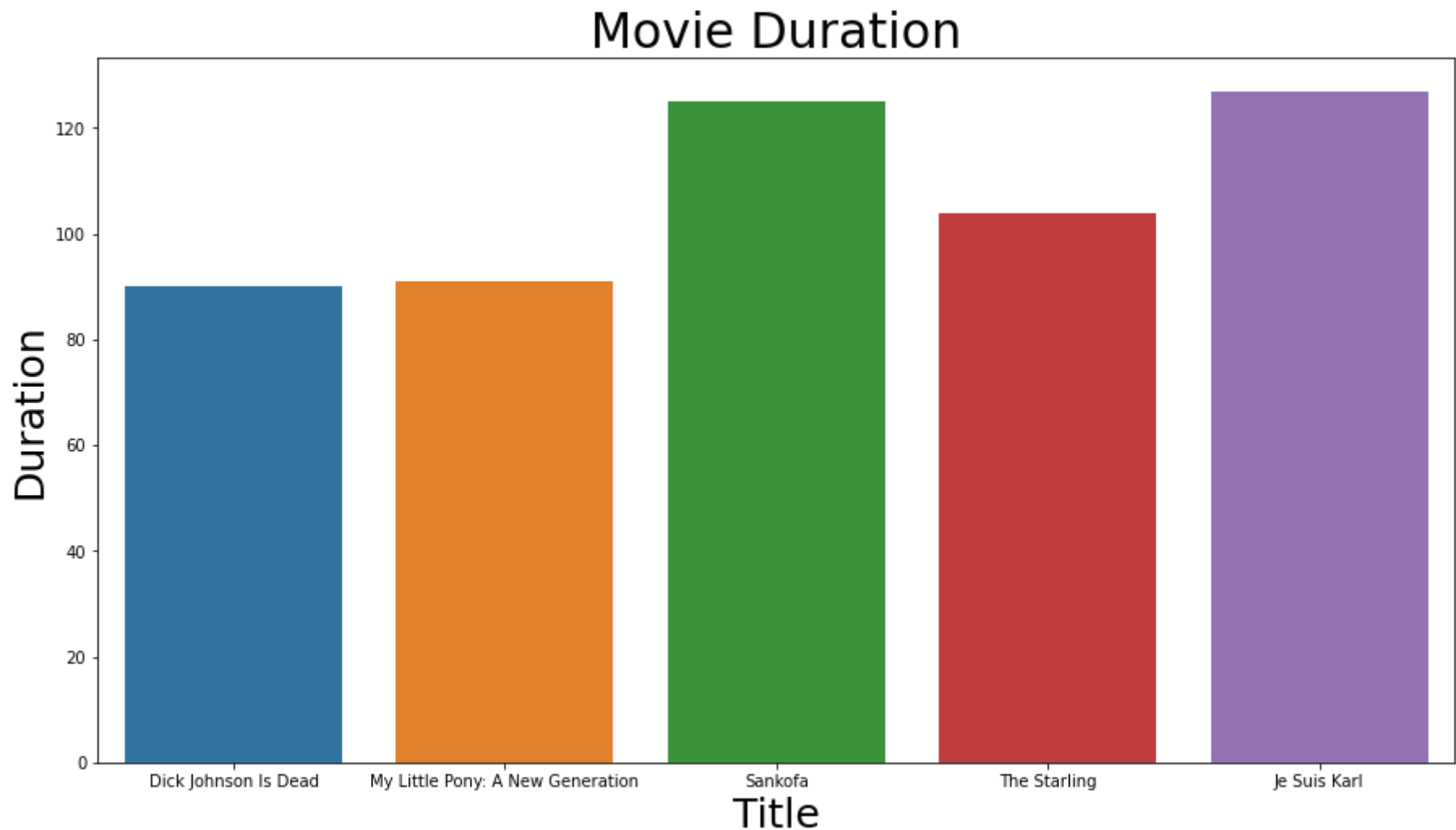
...

```
43       1
273      1
9        1
214      1
191      1
```

Name: duration, Length: 205, dtype: Int64

```
In [577... plt.figure(figsize = (15, 8))
sns.barplot(x = 'title', y = 'duration', data = Movies.head(5))
```

```
plt.xlabel('Title', fontsize = 25)
plt.ylabel('Duration', fontsize = 25)
plt.title('Movie Duration', fontsize = 30)
plt.show()
```



In [578...

```
TV_shows['duration'] = pd.to_numeric(TV_shows['duration'].str.split(" ").str[0], errors = 'coerce').convert_dtypes()
```

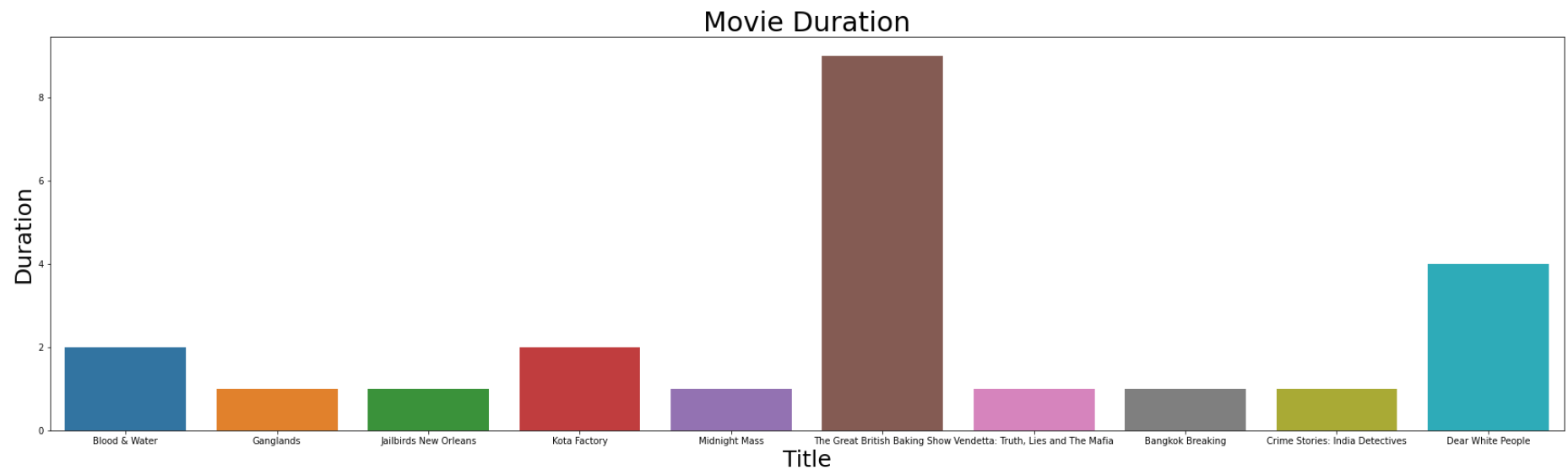
C:\Users\Chanchal Gupta\AppData\Local\Temp\ipykernel\_5500\3669160606.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
TV_shows['duration'] = pd.to_numeric(TV_shows['duration'].str.split(" ").str[0], errors = 'coerce').convert_dtypes()
```

```
In [580... plt.figure(figsize = (30, 8))
sns.barplot(x = 'title',y = 'duration', data = TV_shows.head(10))

plt.xlabel('Title', fontsize = 25)
plt.ylabel('Duration', fontsize = 25)
plt.title('Movie Duration', fontsize = 30)
plt.show()
```



```
In [581... TV_shows['duration'].value_counts().sort_values(ascending = False)

#TV_shows on netflix has a duration of 1 season on average
```

```
Out[581]: 1      1793
          2       425
          3       199
          4        95
          5        65
          6        33
          7        23
          8        17
          9         9
         10         7
         13         3
         15         2
         12         2
         11         2
         17         1
Name: duration, dtype: Int64
```

```
In [582...
```

```
'''
SUMMARY:

-> Netflix is a streaming service that offers a wide variety of award winning TV shows, movies,
anime, comedies and many more.
-> As per data, The percentage of total content contributes more for Movies 69.62% than TV Shows 30.38%.
-> Country that produces larger content on Netflix is United States with 2000++ for Movies and 700++ for TV Shows.
-> Director Rajiv Chilaka has directed a total of 22 movies in the last 10 years with 2.2 movies/year
therefore the most productive director
-> The actor with the larger number of contents on netflix is Anupam Kher with 40++ Movies
-> TV-MA(Mature Audience) has the highest rating for Movies as well as TV_shows
-> Less time difference between release date and date_added results in high ratings.
-> The number of content title Movies/TVShows were mostly released in year 2018 and
overall July and December month seems to be the
best time to release the content.
-> The larger number of content titles is International Movies with 2500++ and International TV Shows with 1200++
-> On an average the duration for movies is 110 mins and 1 season for TV shows.

'''
```

```
Out[582]: '\nSUMMARY:\n\n-> Netflix is a streamin service that offers a wide variety of award winning TV shows, movies, anime, comedies and many more.\n-> As per data, The percentage of total content contributes more for Movies 69.62% than TV Shows 30.38%.\n-> Country that produces larger content on Netflix is United States with 2000++ for Movies and 700++ for TV Shows.\n-> Director Rajiv Chilaka has directed a total of 22 movies in the last 10 years with 2.2 movies/year \ntherefore the most productive director\n-> The actor with the larger number of contents on netflix is Anupam Kher with 40++ Movies\n-> TV-MA(Mature Audience) has the highest rating for Movies as well as TV_shows\n-> Less time difference between release date and date_added results in high ratings.\n-> The number of content title Movies/TVShows were mostly released in year 2018 and overall July and December month seems to be the \nbest time to release the content.\n-> The larger number of content titles is International Movies with 2500++ and International TV Shows with 1200++\n-> On an average the duration for movies is 110 mins and 1 season for TV shows.\n\n'
```