Netflix Exploratory Data Analysis - Pavithran S

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
# using pd.read_csv we can load a csv file.
df = pd.read_csv("netflix.csv")
df.head(5)
```

Out[2]:

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

Features of the dataset:

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

FINDING SHAPE OF THE DATAFRAME

In [3]:
```
# using df.shape we come to know the given dataset have 8807 rows and 12 columns
df.shape
```

Out[3]: (8807, 12)

In [4]:
```
# using df.ndim we come to know the given dataset is a 2dimnesion
df.ndim
```

Out[4]: 2

FINDING DATA TYPES OF EACH COLUMN

In [5]:
```
# using df.info we come to know the data type of each series
# most of the series is in object datatype and release year is in int datatype format
# we found that the columns = (director, cast, country, date_added, rating, duration) have null (or) missing values
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
```

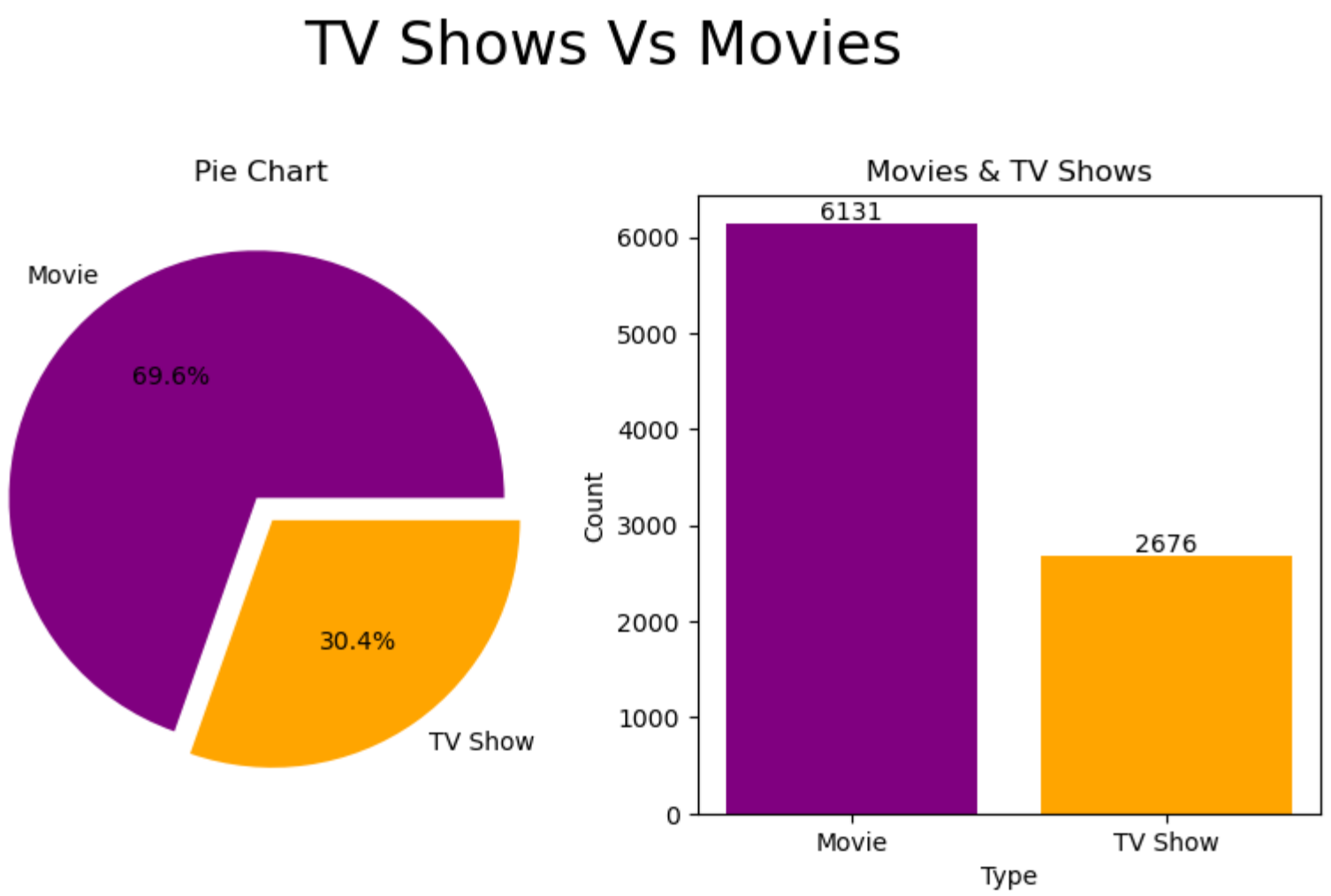
In [6]:
```
df["type"].value_counts()
```

Out[6]:
```
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

We got 6131 Movies and 2676 Tv Shows in the given dataset

In [7]:
```
# TV Shows Vs Movies in a Graphical Representation
df_type = df["type"].value_counts()
x = df_type.index
y = df_type
fig = plt.figure (figsize= (10,10))
plt.subplot (2,2,1)
plt.pie(y,explode = (0.03, 0.08), labels = x, colors= ["purple", "orange"], autopct = "%.1f%%")
plt.title("Pie Chart")

plt.subplot (2,2,2)
height = plt.bar(x,y, color = ["purple" ,"orange"])
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("Movies & TV Shows")
plt.xlabel("Type")
plt.ylabel("Count")
plt.suptitle("TV Shows Vs Movies " , fontsize = 24)
plt.show()
```



UNNESTING THE VALUES

In [8]:
```
# Before Exploratry Data Analysis we have to clean the given data set because it contains null values and nested values.
# first we will unest the nested values
# we have nested values in following columns (director, cast, country, listed_in)
# we are going to unnest director column
df_director = df[["title", "director"]]
df_director["director_unnested"] = df_director["director"].apply(lambda a: str(a).split(", "))
df_director = df_director.explode("director_unnested")
df_director.head(5)
```

Out[8]:

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

In [9]:
```
# we have successfully unnested director's name. but if we check the shape of the dataframe it will increase because after unn
df_director.shape
```

Out[9]: (9612, 3)

In [10]:
```
# Now we will do unnesting for cast column
df_cast = df[["title", "cast"]]
df_cast["cast_unnested"] = df_cast["cast"].apply(lambda a: str(a).split(", "))
df_cast = df_cast.explode("cast_unnested")
df_cast.head(5)
```

Out[10]:

	title	cast	cast_unnested
0	Dick Johnson Is Dead	NaN	nan
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Ama Qamata
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Khosi Ngema
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Gail Mabalane
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Thabang Molaba

In [11]:
```
# Now we will do unnesting for country column
df_country = df[["title", "country"]]
df_country["country_unnested"] = df_country["country"].apply(lambda a: str(a).split(", "))
df_country = df_country.explode("country_unnested")
df_country.head(5)
```

Out[11]:

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

In [12]:
```
# Now we will do unnesting for listed_in column
df_listed_in = df[["title", "listed_in"]]
df_listed_in["listed_in_unnested"] = df_listed_in["listed_in"].apply(lambda a: str(a).split(", "))
df_listed_in = df_listed_in.explode("listed_in_unnested")
df_listed_in.head(5)
```

Out[12]:

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

In [13]:
```
# we have unnested all 4 columns. now we have to merge the unnested values with the original dataframe
# first we are going to join the original dataframe with df_director
# then we drop the duplicate columns director_x, director_y
# then we rename the column director_unnested to director
df_final = df.merge(df_director, on= "title", how = "left")
df_final.drop(["director_x","director_y"], axis = 1, inplace = True)
df_final.rename(columns = {"director_unnested": "director"}, inplace = True)
df_final.head(5)
```

Out[13]:

	show_id	type	title	cast	country	date_added	release_year	rating	duration	listed_in	description	director
0	s1	Movie	Dick Johnson Is Dead	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	Kirsten Johnson
1	s2	TV Show	Blood & Water	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...	nan
2	s3	TV Show	Ganglands	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...	Julien Leclercq
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...	nan
4	s5	TV Show	Kota Factory	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...	nan

In [14]:
```
df_final.shape
```

Out[14]:
```
(9612, 12)
```

In [15]:
```
# Now we are going to join the original dataframe with df_cast
# Then we drop the duplicate columns cast_x, cast_y
# Then we rename the column cast_unnested to cast
df_final = df_final.merge(df_cast, on= "title", how = "left")
df_final.drop(["cast_x","cast_y"], axis = 1, inplace = True)
df_final.rename(columns = {"cast_unnested": "cast"}, inplace = True)
```

```
df_final.head(5)
```

Out[15]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description	director	cast
0	s1	Movie	Dick Johnson Is Dead	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	Kirsten Johnson	nan
1	s2	TV Show	Blood & Water	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...	nan	Ama Qamata
2	s2	TV Show	Blood & Water	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...	nan	Khosi Ngema
3	s2	TV Show	Blood & Water	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...	nan	Gail Mabalane
4	s2	TV Show	Blood & Water	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...	nan	Thabang Molaba

In [16]:
```
df_final.shape
```

Out[16]: (70812, 12)

In [17]:
```
# Now we are going to join the original dataframe with df_country
# Then we drop the duplicate columns country_x, country_y
# Then we rename the column country_unnested to country
df_final = df_final.merge(df_country, on= "title", how = "left")
df_final.drop(["country_x","country_y"], axis = 1, inplace = True)
df_final.rename(columns = {"country_unnested": "country"}, inplace = True)
df_final.head(5)
```

Out[17]:

	show_id	type	title	date_added	release_year	rating	duration	listed_in	description	director	cast	country
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	Kirsten Johnson	nan	United States
1	s2	TV Show	Blood & Water	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...	nan	Ama Qamata	South Africa
2	s2	TV Show	Blood & Water	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...	nan	Khosi Ngema	South Africa
3	s2	TV Show	Blood & Water	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...	nan	Gail Mabalane	South Africa
4	s2	TV Show	Blood & Water	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...	nan	Thabang Molaba	South Africa

In [18]:
```
df_final.shape
```

Out[18]: (89382, 12)

In [19]:
```
# Now we are going to join the original dataframe with df_listed_in
# Then we drop the duplicate columns listed_in_x, listed_in_y
# Then we rename the column listed_in_unnested to listed_in
df_final = df_final.merge(df_listed_in, on= "title", how = "left")
df_final.drop(["listed_in_x","listed_in_y"], axis = 1, inplace = True)
df_final.rename(columns = {"listed_in_unnested": "listed_in"}, inplace = True)
df_final.head(5)
```

Out[19]:

	show_id	type	title	date_added	release_year	rating	duration	description	director	cast	country	listed_in
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	As her father nears the end of his life, filmm...	Kirsten Johnson	nan	United States	Documentaries
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	nan	Ama Qamata	South Africa	International TV Shows
2	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	nan	Ama Qamata	South Africa	TV Dramas
3	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	nan	Ama Qamata	South Africa	TV Mysteries
4	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	nan	Khosi Ngema	South Africa	International TV Shows

In [20]:
```
# Finally after unnesting and merging the 4 columns with the original dataframe now we see we have 327001 rows.
df_final.shape
```

Out[20]: (201991, 12)

UNIQUE ATTRIBUTES

In [21]:
```
#unique attributes in Director Column:
print("Unique Directors: ", df_final["director"].unique())
print("Total No of Directors: ", df_final["director"].nunique())
```

```
Unique Directors:  ['Kirsten Johnson' 'nan' 'Julien Leclercq' ... 'Majid Al Ansari'
 'Peter Hewitt' 'Mozez Singh']
Total No of Directors:  4994
```

In [22]:
```
#unique attributes in Cast Column:
print("Unique Actors: ", df_final["cast"].unique())
print("Total No of Actors: ", df_final["cast"].nunique())
```

```
Unique Actors:  ['nan' 'Ama Qamata' 'Khosi Ngema' ... 'Malkeet Rauni' 'Anita Shabdish'
 'Chittaranjan Tripathy']
Total No of Actors:  36440
```

In [23]:
```
#unique attributes in Listed_in Column:
print("Unique Genre: ", df_final["listed_in"].unique())
print("Total No of Genre: ", df_final["listed_in"].nunique())
```

```
Unique Genre:  ['Documentaries' 'International TV Shows' 'TV Dramas' 'TV Mysteries'
 'Crime TV Shows' 'TV Action & Adventure' 'Docuseries' 'Reality TV'
 'Romantic TV Shows' 'TV Comedies' 'TV Horror' 'Children & Family Movies'
 'Dramas' 'Independent Movies' 'International Movies' 'British TV Shows'
 'Comedies' 'Spanish-Language TV Shows' 'Thrillers' 'Romantic Movies'
 'Music & Musicals' 'Horror Movies' 'Sci-Fi & Fantasy' 'TV Thrillers'
 "Kids' TV" 'Action & Adventure' 'TV Sci-Fi & Fantasy' 'Classic Movies'
 'Anime Features' 'Sports Movies' 'Anime Series' 'Korean TV Shows'
 'Science & Nature TV' 'Teen TV Shows' 'Cult Movies' 'TV Shows'
 'Faith & Spirituality' 'LGBTQ Movies' 'Stand-Up Comedy' 'Movies'
 'Stand-Up Comedy & Talk Shows' 'Classic & Cult TV']
Total No of Genre:  42
```

In [24]:
```
#unique attributes in Rating Column:
print("Unique Ratings: ", df_final["rating"].unique())
print("Total No of Ratings: ", df_final["rating"].nunique())
```

```
Unique Ratings:  ['PG-13' 'TV-MA' 'PG' 'TV-14' 'TV-PG' 'TV-Y' 'TV-Y7' 'R' 'TV-G' 'G'
 'NC-17' '74 min' '84 min' '66 min' 'NR' nan 'TV-Y7-FV' 'UR']
Total No of Ratings:  17
```

In [25]:
```
#unique attributes in Country Column:
print("Unique Country: ", df_final["country"].unique())
print("Total No of Country: ", df_final["country"].nunique())
```

```
Unique Country:  ['United States' 'South Africa' 'nan' 'India' 'Ghana' 'Burkina Faso'
 'United Kingdom' 'Germany' 'Ethiopia' 'Czech Republic' 'Mexico' 'Turkey'
 'Australia' 'France' 'Finland' 'China' 'Canada' 'Japan' 'Nigeria' 'Spain'
 'Belgium' 'South Korea' 'Singapore' 'Italy' 'Romania' 'Argentina'
 'Venezuela' 'Hong Kong' 'Russia' '' 'Ireland' 'Nepal' 'New Zealand'
 'Brazil' 'Greece' 'Jordan' 'Colombia' 'Switzerland' 'Israel' 'Taiwan'
 'Bulgaria' 'Algeria' 'Poland' 'Saudi Arabia' 'Thailand' 'Indonesia'
 'Egypt' 'Denmark' 'Kuwait' 'Netherlands' 'Malaysia' 'Vietnam' 'Hungary'
 'Sweden' 'Lebanon' 'Syria' 'Philippines' 'Iceland' 'United Arab Emirates'
 'Norway' 'Qatar' 'Mauritius' 'Austria' 'Cameroon' 'Palestine' 'Uruguay'
 'United Kingdom,' 'Kenya' 'Chile' 'Luxembourg' 'Cambodia' 'Bangladesh'
 'Portugal' 'Cayman Islands' 'Senegal' 'Serbia' 'Malta' 'Namibia' 'Angola'
 'Peru' 'Mozambique' 'Cambodia,' 'Belarus' 'Zimbabwe' 'Puerto Rico'
 'Pakistan' 'Cyprus' 'Guatemala' 'Iraq' 'Malawi' 'Paraguay' 'Croatia'
 'Iran' 'West Germany' 'United States,' 'Albania' 'Georgia' 'Soviet Union'
 'Morocco' 'Slovakia' 'Ukraine' 'Bermuda' 'Ecuador' 'Armenia' 'Mongolia'
 'Bahamas' 'Sri Lanka' 'Latvia' 'Liechtenstein' 'Cuba' 'Nicaragua'
 'Poland,' 'Slovenia' 'Dominican Republic' 'Samoa' 'Azerbaijan' 'Botswana'
 'Vatican City' 'Jamaica' 'Kazakhstan' 'Lithuania' 'Afghanistan' 'Somalia'
 'Sudan' 'Panama' 'Uganda' 'East Germany' 'Montenegro']
Total No of Country:  128
```

In [26]:
```
# There are 36440 Actors in the data Frame
df_final["cast"].nunique()
```

Out[26]: 36440

In [27]:
```
# To Avoid the missing Values we replace Nan with Unknown
df_final['cast'].replace(["nan"], ["Unknown Actor"], inplace = True)
df_final['director'].replace(["nan"], ["Unknown Directors"], inplace = True)
df_final['country'].replace(["nan"], [np.nan], inplace =True)
```

In [28]:
```
df_final.head(5)
```

Out[28]:

	show_id	type	title	date_added	release_year	rating	duration	description	director	cast	country	listed_in
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	As her father nears the end of his life, filmm...	Kirsten Johnson	Unknown Actor	United States	Documentaries
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	International TV Shows
2	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Dramas
3	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Mysteries
4	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Unknown Directors	Khosi Ngema	South Africa	International TV Shows

REMOVING/FILLING NULL OR MISSING VALUES

In [29]:
```
df_final.isnull().sum()
```

Out[29]:
```
show_id            0
type               0
title              0
date_added       158
release_year       0
rating            67
duration           3
description        0
director           0
cast               0
country        11897
listed_in          0
dtype: int64
```

In [30]:
```
#we found that there are three rows where the duration is typed in rating Column
df_final[df_final['duration'].isnull()]
```

Out[30]:

	show_id	type	title	date_added	release_year	rating	duration	description	director	cast	country	listed_in
126537	s5542	Movie	Louis C.K. 2017	April 4, 2017	2017	74 min	NaN	Louis C.K. muses on religion, eternal love, gi...	Louis C.K.	Louis C.K.	United States	Movies
131603	s5795	Movie	Louis C.K.: Hilarious	September 16, 2016	2010	84 min	NaN	Emmy-winning comedy writer Louis C.K. brings h...	Louis C.K.	Louis C.K.	United States	Movies
131737	s5814	Movie	Louis C.K.: Live at the Comedy Store	August 15, 2016	2015	66 min	NaN	The comic puts his trademark hilarious/ thought...	Louis C.K.	Louis C.K.	United States	Movies

In [31]:
```
# we now changing the mistaken rating column to the duration
df_final.loc[df_final['duration'].isnull(), 'duration'] = df_final.loc[df_final['duration'].isnull(), 'duration'].fillna(df_fi
```

In [32]:
```
df_final[df_final['duration'].isnull()]
```

Out[32]:

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

In [33]:
```
# now we are filling the missing values of rating Column
df_final.loc[df_final['rating'].str.contains('min', na =False),'rating'] = 'NR'
df_final['rating'].fillna('NR', inplace = True)
df_final.head()
```

Out[33]:

	show_id	type	title	date_added	release_year	rating	duration	description	director	cast	country	listed_in
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	As her father nears the end of his life, filmm...	Kirsten Johnson	Unknown Actor	United States	Documentaries
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	International TV Shows
2	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Dramas
3	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Mysteries
4	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Unknown Directors	Khosi Ngema	South Africa	International TV Shows

In [34]:
```
df_final.isna().sum()
```

Out[34]:
```
show_id            0
type               0
title              0
date_added       158
release_year       0
rating             0
duration           0
description        0
director           0
cast               0
country        11897
listed_in          0
dtype: int64
```

In [35]:
```
df_final['date_added']
```

Out[35]:
```
0         September 25, 2021
1         September 24, 2021
2         September 24, 2021
3         September 24, 2021
4         September 24, 2021
                 ...
201986         March 2, 2019
201987         March 2, 2019
201988         March 2, 2019
201989         March 2, 2019
201990         March 2, 2019
Name: date_added, Length: 201991, dtype: object
```

DATA TYPE CONVERSION (OBJECT TYPE TO DATETIME TYPE)

In [36]:
```
# Date added column is in object data type. we need that in datetime date type
df_final['date_added']=pd.to_datetime(df_final['date_added'],format='mixed')
```

In [37]:
```
df_final['date_added']
```

Out[37]:
```
0        2021-09-25
1        2021-09-24
2        2021-09-24
3        2021-09-24
4        2021-09-24
            ...
201986   2019-03-02
201987   2019-03-02
201988   2019-03-02
201989   2019-03-02
201990   2019-03-02
Name: date_added, Length: 201991, dtype: datetime64[ns]
```

In [38]:
```
df_final.isna().sum()
```

Out[38]:
```
show_id            0
type               0
title              0
date_added       158
release_year       0
rating             0
duration           0
description        0
director           0
cast               0
country        11897
listed_in          0
dtype: int64
```

In [39]:
```
# we are filling the missing country rows. when the same director have country name in some other entry we are imputing them t
for i in df_final[df_final['country'].isnull()]['director'].unique():
  if i in df_final[~df_final['country'].isnull()]['director'].unique():
    imp=df_final[df_final['director']==i]['country'].mode().values[0]
    df_final.loc[df_final['director']==i,'country']=df_final.loc[df_final['director']==i,'country'].fillna(imp)
```

In [40]:
```
# we are filling the missing country rows . when the same cast have country name in some other entry we are imputing them to t
for i in df_final[df_final['country'].isnull()]['cast'].unique():
  if i in df_final[~df_final['country'].isnull()]['cast'].unique():
    imp=df_final[df_final['cast']==i]['country'].mode().values[0]
    df_final.loc[df_final['cast']==i,'country']=df_final.loc[df_final['cast']==i,'country'].fillna(imp)
```

In [41]:
```
#after imputation also there are some missing values so we fill unknown country for the remaining values
df_final["country"].fillna("Unknown Country",inplace=True)
```

In [42]:
```
df_final.isna().sum()
```

Out[42]:
```
show_id          0
type             0
title            0
date_added     158
release_year     0
rating           0
duration         0
description      0
director         0
cast             0
country          0
listed_in        0
dtype: int64
```

In [43]:
```
# we are filling the missing date_added rows.
for i in df_final[df_final['date_added'].isnull()]['release_year'].unique():
```

```
    imp=df_final[df_final['release_year']==i]['date_added'].mode().values[0]
    df_final.loc[df_final['release_year']==i,'date_added']=df_final.loc[df_final['release_year']==i,'date_added'].fillna(imp)
```

In [44]:
```
#now data cleaning is done successfully. we can see there are no missing values
df_final.isna().sum()
```

Out[44]:
```
show_id         0
type            0
title           0
date_added      0
release_year    0
rating          0
duration        0
description     0
director        0
cast            0
country         0
listed_in       0
dtype: int64
```

In [45]:
```
df_final["duration"].value_counts()
```

Out[45]:
```
duration
1 Season      35035
2 Seasons      9559
3 Seasons      5084
94 min         4343
106 min        4040
              ...
3 min             4
5 min             3
11 min            2
8 min             2
9 min             2
Name: count, Length: 220, dtype: int64
```

In [46]:
```
# we cannot use duration column for EDA when it has Season/ Minutes with the data. so we replace the Season/ Minutes keyword
df_final["duration"] = df_final["duration"].str.replace("min", "")
df_final["duration"] = df_final["duration"].str.replace("Seasons", "")
df_final["duration"] = df_final["duration"].str.replace("Season", "")
df_final["duration"] = df_final["duration"].astype("int")
df_final.head()
```

Out[46]:

	show_id	type	title	date_added	release_year	rating	duration	description	director	cast	country	listed_in
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90	As her father nears the end of his life, filmm...	Kirsten Johnson	Unknown Actor	United States	Documentaries
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	International TV Shows
2	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Dramas
3	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Mysteries
4	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Khosi Ngema	South Africa	International TV Shows

In [47]:
```
df_final["duration"].unique()
```

Out[47]:
```
array([ 90,   2,   1,  91, 125,   9, 104, 127,   4,  67,  94,   5, 161,
        61, 166, 147, 103,  97, 106, 111,   3, 110, 105,  96, 124, 116,
        98,  23, 115, 122,  99,  88, 100,   6, 102,  93,  95,  85,  83,
       113,  13, 182,  48, 145,  87,  92,  80, 117, 128, 119, 143, 114,
       118, 108,  63, 121, 142, 154, 120,  82, 109, 101,  86, 229,  76,
        89, 156, 112, 107, 129, 135, 136, 165, 150, 133,  70,  84, 140,
        78,   7,  64,  59, 139,  69, 148, 189, 141, 130, 138,  81, 132,
        10, 123,  65,  68,  66,  62,  74, 131,  39,  46,  38,   8,  17,
       126, 155, 159, 137,  12, 273,  36,  34,  77,  60,  49,  58,  72,
       204, 212,  25,  73,  29,  47,  32,  35,  71, 149,  33,  15,  54,
       224, 162,  37,  75,  79,  55, 158, 164, 173, 181, 185,  21,  24,
        51, 151,  42,  22, 134, 177,  52,  14,  53,  57,  28,  50,  26,
        45, 171,  27,  44, 146,  20, 157, 203,  41,  30, 194, 233, 237,
       230, 195, 253, 152, 190, 160, 208, 180, 144, 174, 170, 192, 209,
       187, 172,  16, 186,  11, 193, 176,  56, 169,  40, 168, 312, 153,
       214,  31, 163,  19, 179,  43, 200, 196, 167, 178, 228,  18, 205,
       201, 191])
```

Value_counts Of Each Columns

In [54]:
```
#Value Counts For Movies in year Wise
df_value_release_year_m = df_movies.groupby("release_year").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=Fal
df_value_release_year_m.head()
```

Out[54]:

	release_year	title
0	2018	767
1	2017	767
2	2016	658
3	2019	633
4	2020	517

In [55]:
```
#Value Counts For TV Shows in year Wise
df_value_release_year_t = df_tv.groupby("release_year").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).
df_value_release_year_t.head()
```

Out[55]:

	release_year	title
0	2020	436
1	2019	397
2	2018	380
3	2021	315
4	2017	265

In [56]:
```
#Value Counts For Movies in Rating Wise
df_value_rating_m = df_movies.groupby("rating").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_in
df_value_rating_m.head(5)
```

Out[56]:

	rating	title
0	TV-MA	2062
1	TV-14	1427
2	R	797
3	TV-PG	540
4	PG-13	490

In [57]:
```
#Value Counts For TV Shows in Rating Wise
df_value_rating_t = df_tv.groupby("rating").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index(
df_value_rating_t.head(5)
```

Out[57]:

	rating	title
0	TV-MA	1145
1	TV-14	733
2	TV-PG	323
3	TV-Y7	195
4	TV-Y	176

In [58]:
```
#Value Counts For Movies in Duration Wise
df_value_duration_m = df_movies.groupby("duration").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).rese
df_value_duration_m.head(5)
```

Out[58]:

	duration	title
0	90	152
1	97	146
2	93	146
3	94	146
4	91	144

In [59]:
```
#Value Counts For TV Shows in Duration Wise
df_value_duration_t = df_tv.groupby("duration").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_in
df_value_duration_t.head(5)
```

Out[59]:

	duration	title
0	1	1793
1	2	425
2	3	199
3	4	95
4	5	65

In [60]:
```
#Value Counts For Movies in Country Wise
df_value_country_m = df_movies.groupby("country").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_
df_value_country_m = df_value_country_m[df_value_country_m["country"]!= "Unknown Country"]
df_value_country_m.head(5)
```

Out[60]:

	country	title
0	United States	2940
1	India	1052
2	United Kingdom	556
3	Canada	334
4	France	318

In [61]:
```
#Value Counts For TV Shows in Country Wise
df_value_country_t = df_tv.groupby("country").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_inde
df_value_country_t = df_value_country_t[df_value_country_t["country"]!= "Unknown Country"]
df_value_country_t.head(5)
```

Out[61]:

	country	title
0	United States	1308
1	United Kingdom	273
2	Japan	200
3	South Korea	171
4	Canada	126

In [62]:
```
#Value Counts For Movies in Listed_in Wise
df_value_list_m = df_movies.groupby("listed_in").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_i
df_value_list_m.head(5)
```

Out[62]:

	listed_in	title
0	International Movies	2752
1	Dramas	2427
2	Comedies	1674
3	Documentaries	869
4	Action & Adventure	859

In [63]:
```
#Value Counts For TV Shows in Listed_in Wise
df_value_list_t = df_tv.groupby("listed_in").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index
df_value_list_t.head(5)
```

Out[63]:

	listed_in	title
0	International TV Shows	1351
1	TV Dramas	763
2	TV Comedies	581
3	Crime TV Shows	470
4	Kids' TV	451

Statistical Summary

In [64]:
```
# Statistical Summary of Duration Column in Movies
df_value_duration_m["duration"].describe()
```

Out[64]:
```
count    205.000000
mean     110.395122
std       63.942797
min        3.000000
25%       57.000000
50%      108.000000
75%      159.000000
max      312.000000
Name: duration, dtype: float64
```

we can see the minimum duration of a movie is 3 minutes and maximum Duration is 312 minutes

In [65]:
```
# Statistical Summary of Duration Column in TV Shows
df_value_duration_t["duration"].describe()
```

Out[65]:
```
count    15.000000
mean      8.200000
std       4.813671
min       1.000000
25%       4.500000
50%       8.000000
75%      11.500000
max      17.000000
Name: duration, dtype: float64
```

we can see the minimum Season for a TV Show is 1 Season and maximum is 17 Season

In [66]:
```
#we are splitting the date_added column into mutiple columns for EDA purpose
df_final['year'] = df_final['date_added'].dt.year
df_final['month'] = df_final['date_added'].dt.month
df_final['day'] = df_final['date_added'].dt.day
df_final['day_name'] = df_final['date_added'].dt.day_name()
```

In [67]:
```
df_final.head()
```

Out[67]:

	show_id	type	title	date_added	release_year	rating	duration	description	director	cast	country	listed_in	year	r
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90	As her father nears the end of his life, filmm...	Kirsten Johnson	Unknown Actor	United States	Documentaries	2021	
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	International TV Shows	2021	
2	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Dramas	2021	
3	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Mysteries	2021	
4	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Khosi Ngema	South Africa	International TV Shows	2021	

In [68]:
```
# Seperated the movies from the final dataframe
df_movies = df_final[df_final["type"]== "Movie"]
df_movies.head(5)
```

Out[68]:

	show_id	type	title	date_added	release_year	rating	duration	description	director	cast	country	listed_in	yea
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90	As her father nears the end of his life, filmm...	Kirsten Johnson	Unknown Actor	United States	Documentaries	202
159	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	Equestria's divided. But a bright-eyed hero be...	Robert Cullen	Vanessa Hudgens	United States	Children & Family Movies	202
160	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	Equestria's divided. But a bright-eyed hero be...	Robert Cullen	Kimiko Glenn	United States	Children & Family Movies	202
161	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	Equestria's divided. But a bright-eyed hero be...	Robert Cullen	James Marsden	United States	Children & Family Movies	202
162	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	Equestria's divided. But a bright-eyed hero be...	Robert Cullen	Sofia Carson	United States	Children & Family Movies	202

In [69]:
```
# Starting Year and Ending Year of Movies:
print("Minimum Year : ",df_movies["release_year"].min())
print("Maximum Year : ",df_movies["release_year"].max())
```

```
Minimum Year :  1942
Maximum Year :  2021
```

In [70]:
```
# Starting Year and Ending Year of TV Shows:
print("Minimum Year : ",df_tv["release_year"].min())
print("Maximum Year : ",df_tv["release_year"].max())
```

```
Minimum Year :  1925
Maximum Year :  2021
```

In [71]:
```
# Seperated the tv shows from the final dataframe
df_tv = df_final[df_final["type"]== "TV Show"]
df_tv.head(5)
```

Out[71]:

	show_id	type	title	date_added	release_year	rating	duration	description	director	cast	country	listed_in	year	month
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	International TV Shows	2021	9
2	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Dramas	2021	9
3	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Ama Qamata	South Africa	TV Mysteries	2021	9
4	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Khosi Ngema	South Africa	International TV Shows	2021	9
5	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	After crossing paths at a party, a Cape Town t...	Unknown Directors	Khosi Ngema	South Africa	TV Dramas	2021	9

Now we are going to do Exploratory Data Analysis for the Netflix Dataset:

I) Uni Variate Analysis for the follwing variables:

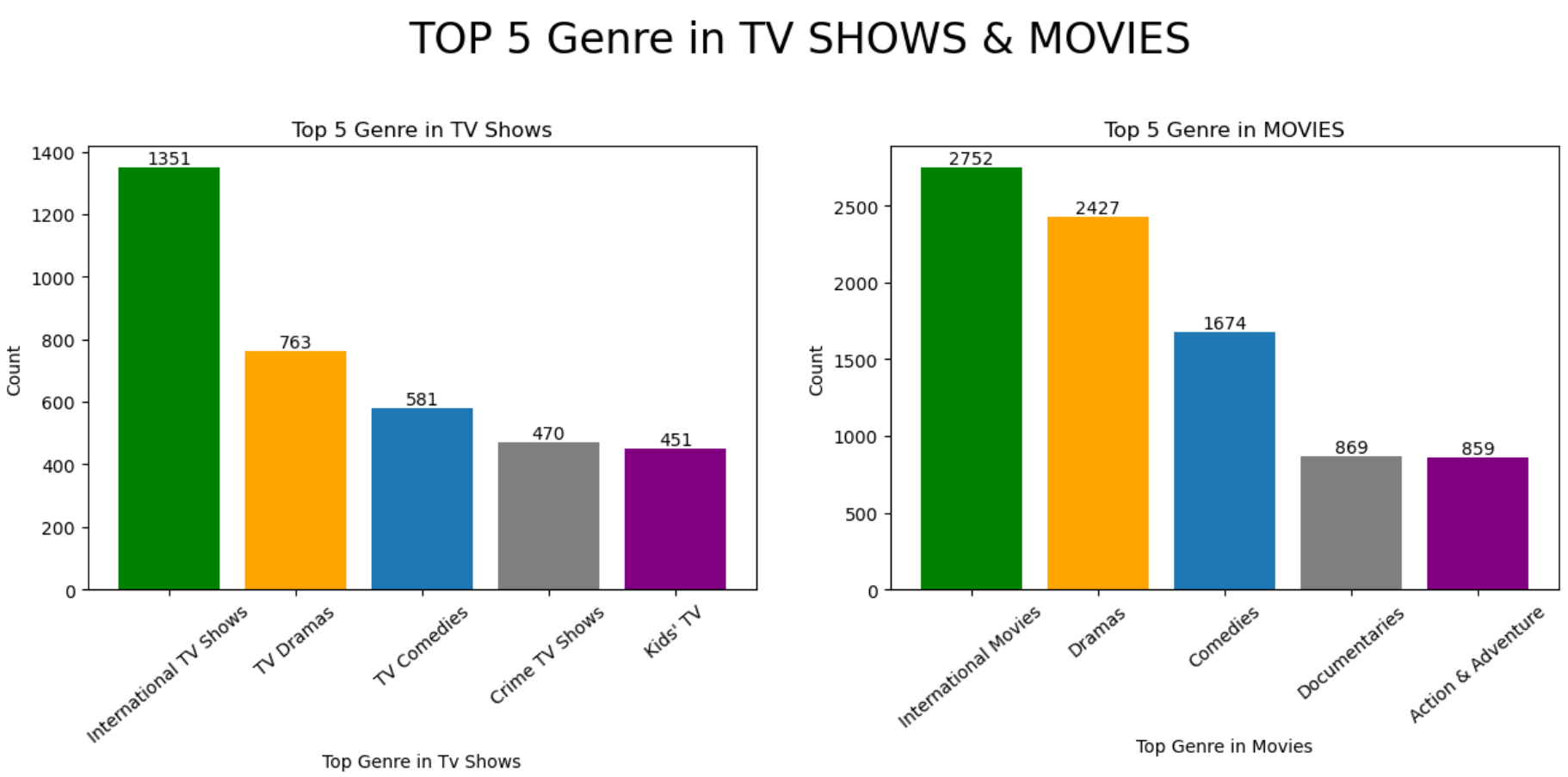
```
1) Top 5 Genre in TV Shows & Movies
2) Top 5 Directors in TV Shows & Movies
3) Top 5 Cast in TV Shows & Movies
4) Highest No.of TV Shows & Movies Added in Netflix in Years
5) No.of TV Shows & Movies Released in Years
6) Top 5 Ratings in TV Shows & Movies
7) Top 5 Actors in Indian TV Shows & Movies
8) Top 5 Directors in Indian TV Shows & Movies
9) TOP 10 Countries Releasing TV SHOWS & MOVIES
```

TOP 5 Genre in TV SHOWS & MOVIES

In [73]:
```
#TOP 5 Genre in TV SHOWS & MOVIES
#Graphical RepresentatioTn using bar plot
fig = plt.figure (figsize= (15,10))
plt.subplot (2,2,1)
df_genre = df_tv.groupby("listed_in").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index().head
x = df_genre["listed_in"]
y = df_genre["title"]
height = plt.bar(x,y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.xticks(rotation = 40)
plt.title("Top 5 Genre in TV Shows")
plt.xlabel("Top Genre in Tv Shows")
plt.ylabel("Count")

plt.subplot (2,2,2)
df_genre_m = df_movies.groupby("listed_in").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index(
x = df_genre_m["listed_in"]
y = df_genre_m["title"]
height = plt.bar(x,y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
for bar in height:
    yval = bar.get_height()
```

```
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.xticks(rotation = 40)
plt.title("Top 5 Genre in MOVIES")
plt.xlabel("Top Genre in Movies")
plt.ylabel("Count")
plt.suptitle("TOP 5 Genre in TV SHOWS & MOVIES", fontsize = 24)
plt.show()
```



TOP 5 Directors in TV SHOWS & MOVIES

In [74]:
```
# to find which director had more movies
df_more_movies = df_movies.groupby("director").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_ind
df_more_movies = df_more_movies[df_more_movies["director"]!= "Unknown Directors"]
df_more_movies.head(5)
```

Out[74]:

	director	title
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Suhas Kadav	16
5	Marcus Raboy	15

In [75]:
```
# to find which director had more TV Shows
df_more_tv = df_tv.groupby("director").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index()
df_more_tv = df_more_tv[df_more_tv["director"]!= "Unknown Directors"]
df_more_tv.head(5)
```

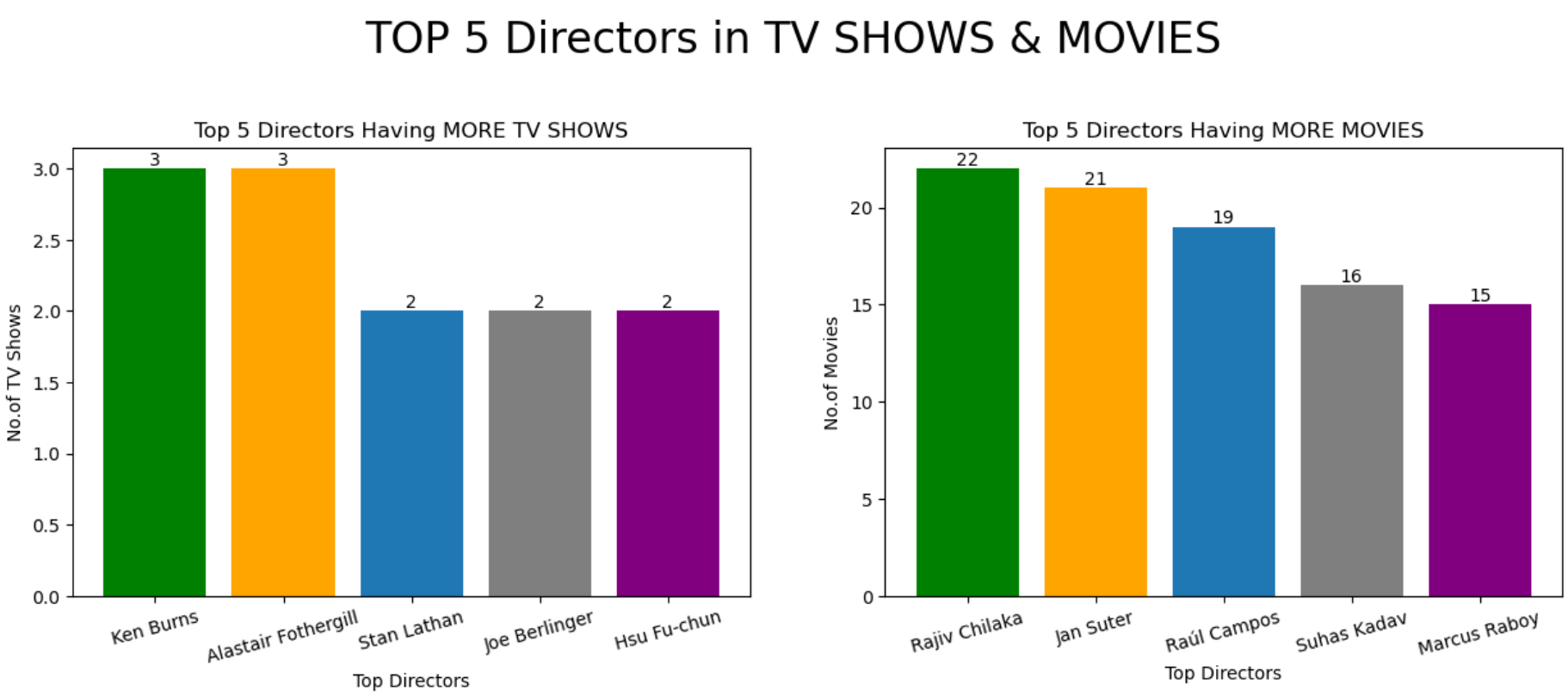
Out[75]:

	director	title
1	Ken Burns	3
2	Alastair Fothergill	3
3	Stan Lathan	2
4	Joe Berlinger	2
5	Hsu Fu-chun	2

In [76]:
```
#TOP 5 Directors in TV SHOWS & MOVIES
#Graphical RepresentatioTn using bar plot
fig = plt.figure (figsize= (15,10))
plt.subplot (2,2,1)
df_more_tv_top5 = df_more_tv.head(5)
x= df_more_tv_top5["director"]
y = df_more_tv_top5["title"]
height = plt.bar(x, y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("Top 5 Directors Having MORE TV SHOWS")
plt.xlabel("Top Directors")
plt.ylabel("No.of TV Shows")
```

```
plt.xticks(rotation = 15)

plt.subplot (2,2,2)
df_more_movies_top5 = df_more_movies.head(5)
x= df_more_movies_top5["director"]
y = df_more_movies_top5["title"]
height = plt.bar(x, y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("Top 5 Directors Having MORE MOVIES")
plt.xlabel("Top Directors")
plt.ylabel("No.of Movies")
plt.xticks(rotation = 15)
plt.suptitle("TOP 5 Directors in TV SHOWS & MOVIES", fontsize = 24)
plt.show()
```



TOP 5 Cast in TV SHOWS & MOVIES

In [77]:
```
# to find Top 5 Cast in Movies
df_c_movies = df_movies.groupby("cast").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index()
df_c_movies = df_c_movies[df_c_movies["cast"]!= "Unknown Actor"]
df_c_movies.head(5)
```

Out[77]:

	cast	title
1	Anupam Kher	42
2	Shah Rukh Khan	35
3	Naseeruddin Shah	32
4	Om Puri	30
5	Akshay Kumar	30

In [78]:
```
# to find Top 5 Cast in TV Shows
df_c_tv = df_tv.groupby("cast").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index()
df_c_tv = df_c_tv[df_c_tv["cast"]!= "Unknown Actor"]
df_c_tv.head(5)
```

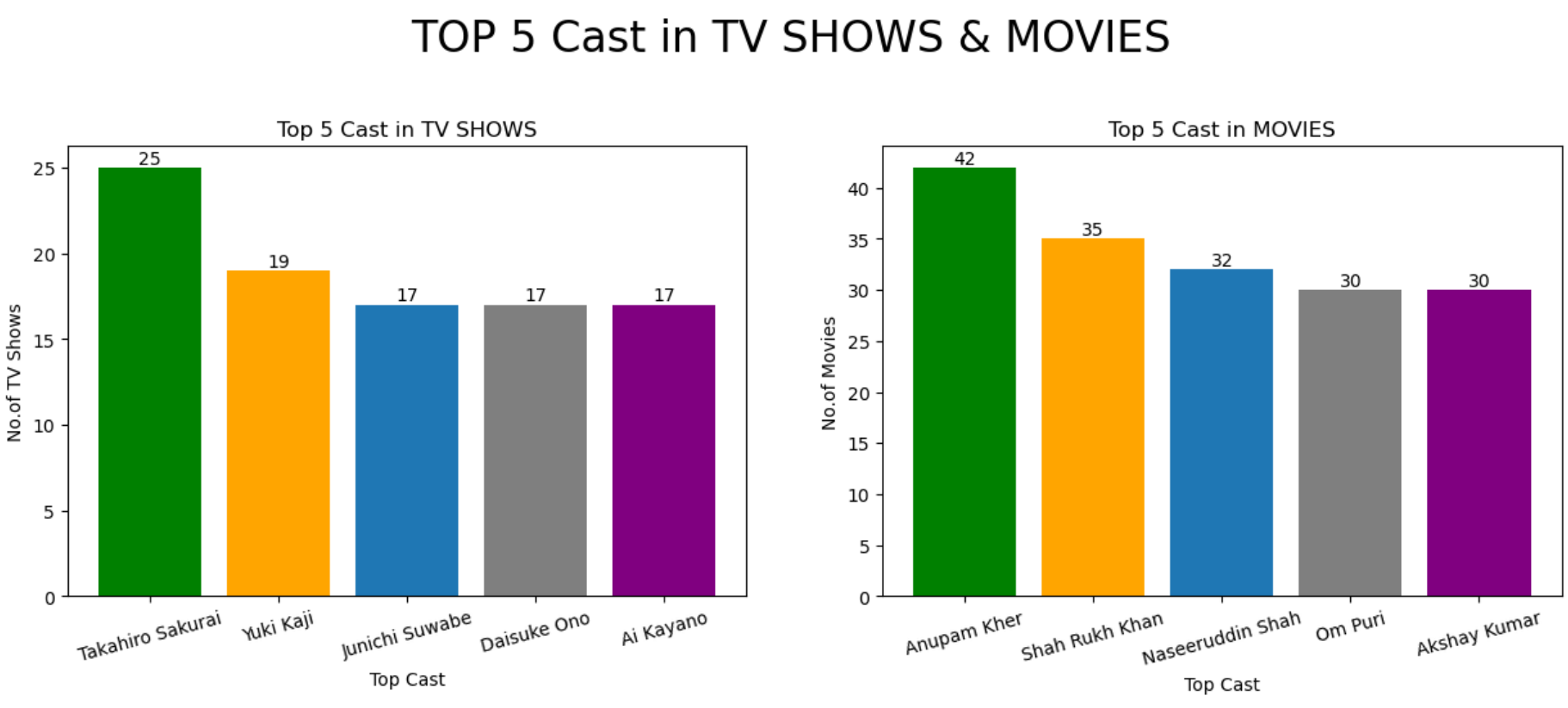
Out[78]:

	cast	title
1	Takahiro Sakurai	25
2	Yuki Kaji	19
3	Junichi Suwabe	17
4	Daisuke Ono	17
5	Ai Kayano	17

In [79]:
```
#TOP 5 Cast in TV SHOWS & MOVIES
#Graphical RepresentatioTn using bar plot
fig = plt.figure (figsize= (15,10))
plt.subplot (2,2,1)
df_c_tv_top5 = df_c_tv.head(5)
x= df_c_tv_top5["cast"]
y = df_c_tv_top5["title"]
height = plt.bar(x, y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
```

```
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("Top 5 Cast in TV SHOWS")
plt.xlabel("Top Cast")
plt.ylabel("No.of TV Shows")
plt.xticks(rotation = 15)

plt.subplot (2,2,2)
df_c_movies_top5 = df_c_movies.head(5)
x= df_c_movies_top5["cast"]
y = df_c_movies_top5["title"]
height = plt.bar(x, y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("Top 5 Cast in MOVIES")
plt.xlabel("Top Cast")
plt.ylabel("No.of Movies")
plt.xticks(rotation = 15)
plt.suptitle("TOP 5 Cast in TV SHOWS & MOVIES", fontsize = 24)
plt.show()
```



Highest No.Of Shows added in Netflix in the Year 2020

In [80]:
```
# In which year highest no.of TV Schows were Added?
df_tv_year = df_tv.groupby("year").agg({"title":"nunique"}).reset_index()
df_tv_year.head()
```

Out[80]:

	year	title
0	2008	1
1	2013	5
2	2014	5
3	2015	26
4	2016	179

Highest No.Of Movies was added in Netflix in the Year 2019

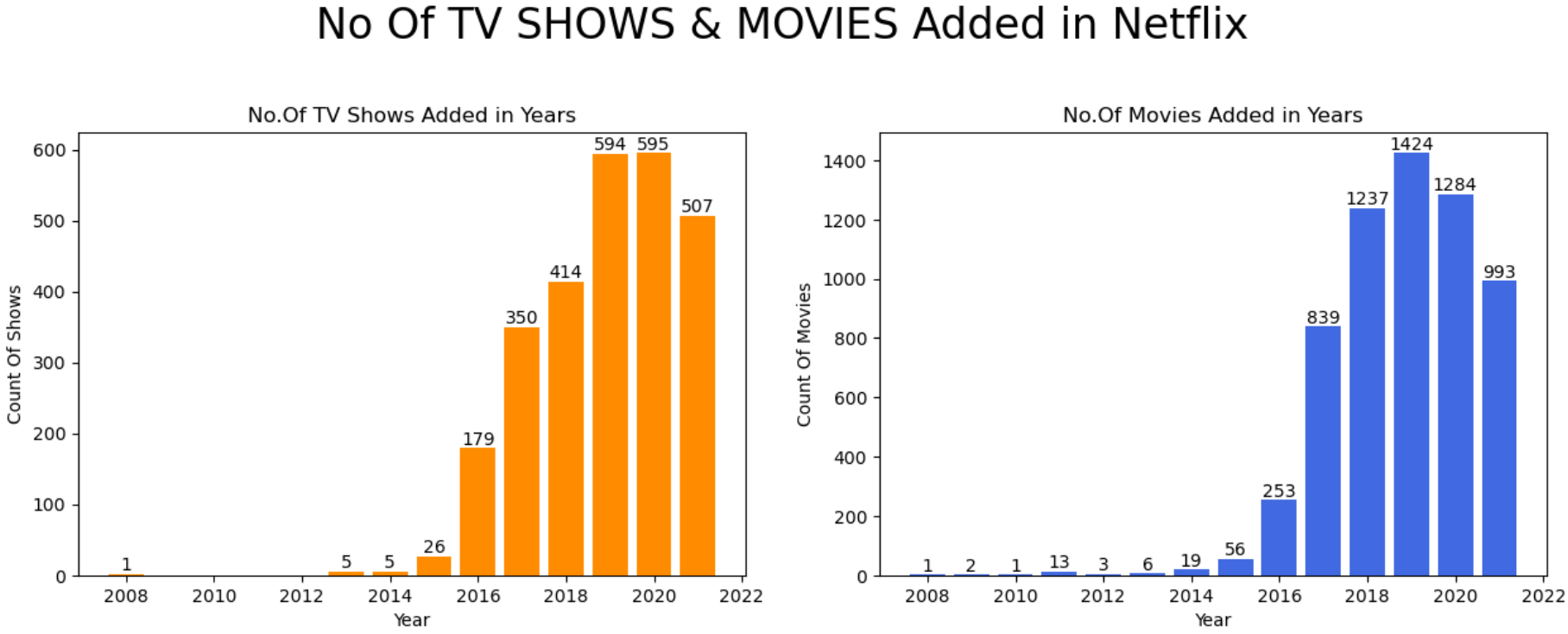
In [81]:
```
# In which year highest no.of Movies were Added?
df_mov_year = df_movies.groupby("year").agg({"title":"nunique"}).reset_index()
df_mov_year.head()
```

Out[81]:

	year	title
0	2008	1
1	2009	2
2	2010	1
3	2011	13
4	2012	3

In [82]:
```
#Graphical Representation using bar plot
fig = plt.figure (figsize= (15,10))
plt.subplot (2,2,1)
x = df_tv_year["year"]
y = df_tv_year["title"]
height = plt.bar(x,y, color ="darkorange")
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("No.Of TV Shows Added in Years")
plt.xlabel("Year")
plt.ylabel("Count Of Shows")

plt.subplot (2,2,2)
x = df_mov_year["year"]
y = df_mov_year["title"]
height = plt.bar(x,y, color ="royalblue")
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("No.Of Movies Added in Years")
plt.xlabel("Year")
plt.ylabel("Count Of Movies")
plt.suptitle("No Of TV SHOWS & MOVIES Added in Netflix ", fontsize = 24)
plt.show()
```



Highest No.Of Movies released in the Year 2017 & 2018

In [83]:
```
# In which year highest no.of Movies were released?
df_mov_y = df_movies.groupby("release_year").agg({"title":"nunique"}).reset_index()
df_mov_y.head()
```

Out[83]:

	release_year	title
0	1942	2
1	1943	3
2	1944	3
3	1945	3
4	1946	1

Highest No.Of TV Shows released in the Year 2020

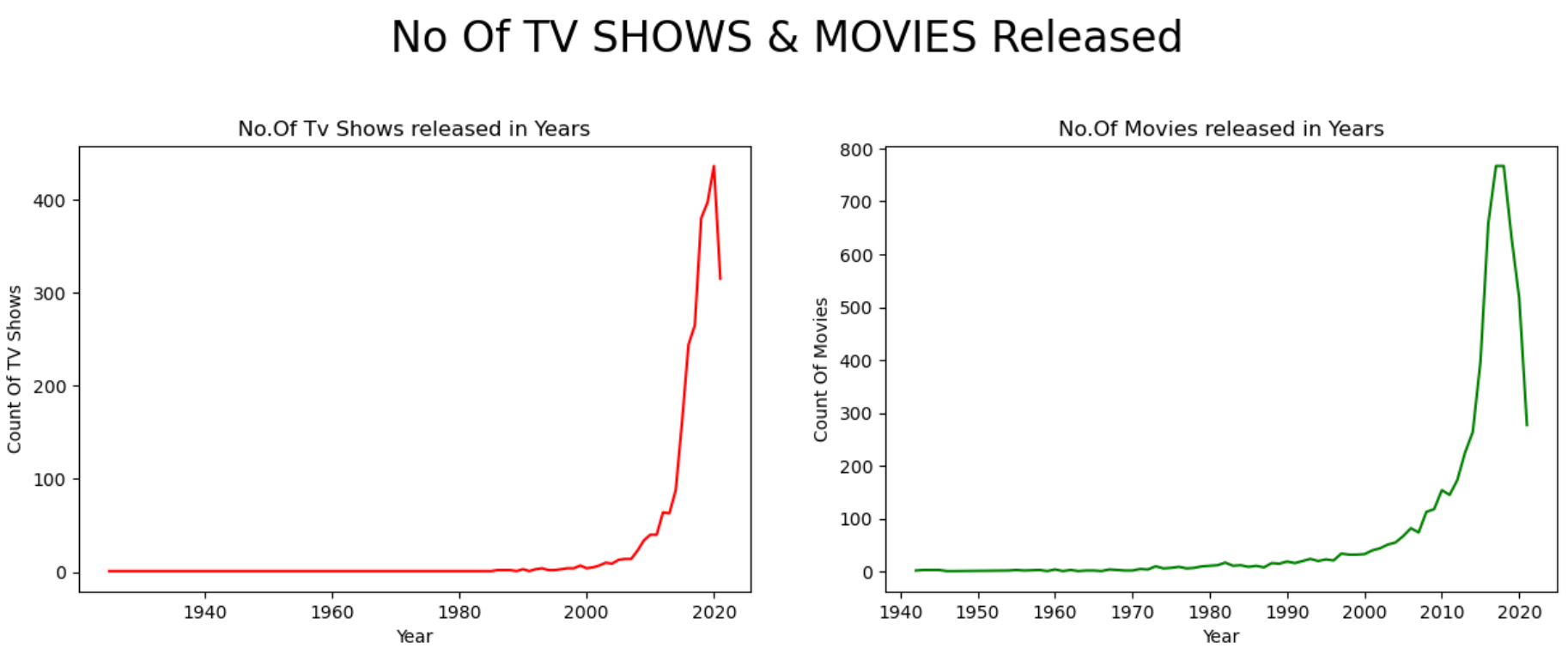
In [84]:
```
# In which year highest no.of TV Schows were Released?
df_tv_y = df_tv.groupby("release_year").agg({"title":"nunique"}).reset_index()
df_tv_y.tail()
```

Out[84]:

	release_year	title
41	2017	265
42	2018	380
43	2019	397
44	2020	436
45	2021	315

In [85]:
```
#Graphical Representation using Line plot
fig = plt.figure (figsize= (15,10))
plt.subplot (2,2,1)
x = df_tv_y["release_year"]
y = df_tv_y["title"]
plt.plot(x,y, color ="r")
plt.title("No.Of Tv Shows released in Years")
plt.xlabel("Year")
plt.ylabel("Count Of TV Shows")

plt.subplot (2,2,2)
x = df_mov_y["release_year"]
y = df_mov_y["title"]
plt.plot(x,y, color ="g")
plt.title("No.Of Movies released in Years")
plt.xlabel("Year")
plt.ylabel("Count Of Movies")
plt.suptitle("No Of TV SHOWS & MOVIES Released ", fontsize = 24)
plt.show()
```



Top 5 Ratings in TV SHOWS & MOVIES

In [86]:
```
# the ratings of the most movies and tvshows
movies = df_movies.groupby(["rating"]).agg({"title":"nunique"}).reset_index().sort_values(by=["title"],ascending=False).head(5
tvshows = df_tv.groupby(["rating"]).agg({"title":"nunique"}).reset_index().sort_values(by=["title"],ascending=False).head(5)
print(movies)
print()
print(tvshows)
```

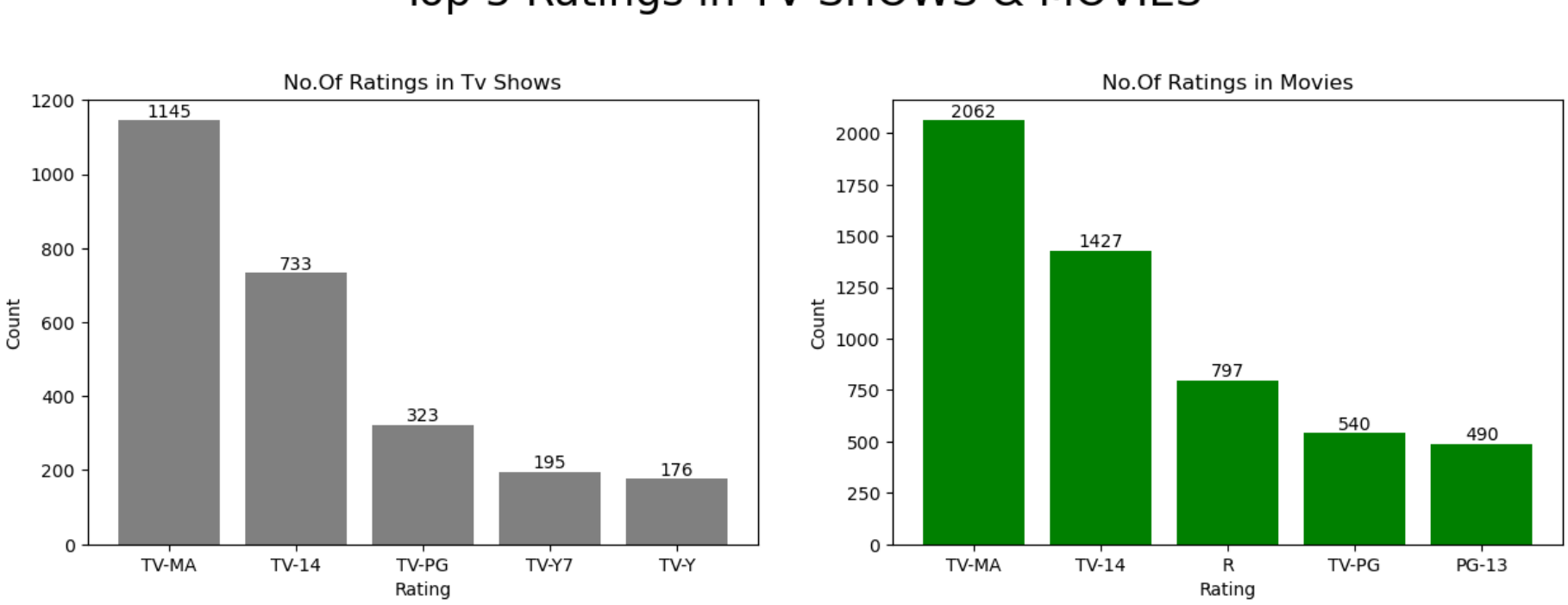
```
  rating  title
8  TV-MA   2062
6  TV-14   1427
5      R    797
9  TV-PG    540
4  PG-13    490

  rating  title
4  TV-MA   1145
2  TV-14    733
5  TV-PG    323
7  TV-Y7    195
6   TV-Y    176
```

In [87]:
```
#Graphical Representation using Bar plot
fig = plt.figure (figsize= (15,10))
plt.subplot (2,2,1)
```

```
x = tvshows["rating"]
y = tvshows["title"]
height = plt.bar(x,y, color ="gray")
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("No.Of Ratings in Tv Shows")
plt.xlabel("Rating")
plt.ylabel("Count")

plt.subplot (2,2,2)
x = movies["rating"]
y = movies["title"]
height = plt.bar(x,y, color ="g")
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("No.Of Ratings in Movies")
plt.xlabel("Rating")
plt.ylabel("Count")
plt.suptitle("Top 5 Ratings in TV SHOWS & MOVIES", fontsize = 24)
plt.show()
```



TOP 5 Actors in Indian TV SHOWS & MOVIES

In [88]:
```
indian_movies = df_movies[df_movies["country"]== "India"]
indian_movies_g = indian_movies.groupby("cast").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_in
indian_movies_g= indian_movies_g[indian_movies_g["cast"]!= "Unknown Actor"]
indian_movies_g.head()
```

Out[88]:

	cast	title
0	Anupam Kher	40
1	Shah Rukh Khan	35
2	Naseeruddin Shah	32
3	Akshay Kumar	29
4	Om Puri	29

In [89]:
```
indian_tv = df_tv[df_tv["country"]== "India"]
indian_tv_g = indian_tv.groupby("cast").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index()
indian_tv_g= indian_tv_g[indian_tv_g["cast"]!= "Unknown Actor"]
indian_tv_g.head(5)
```

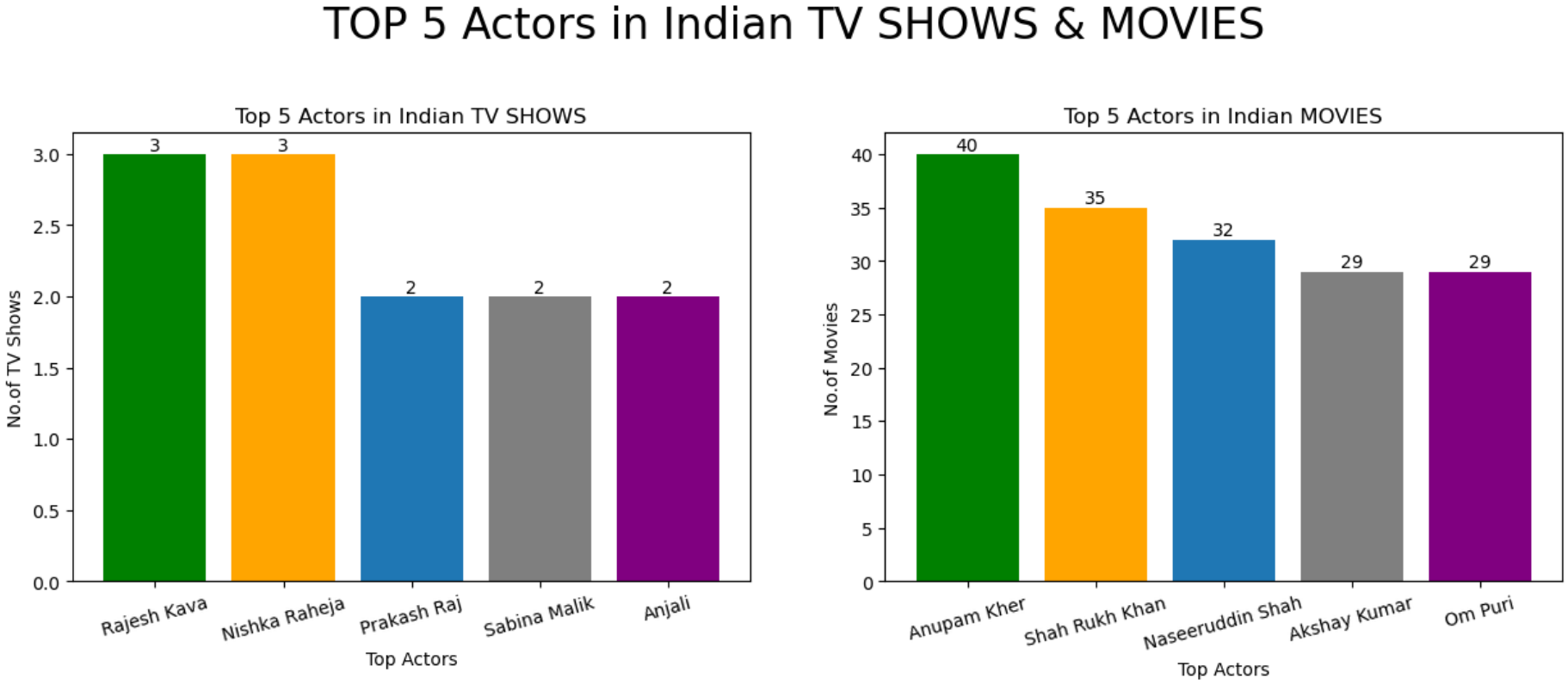
Out[89]:

	cast	title
1	Rajesh Kava	3
2	Nishka Raheja	3
3	Prakash Raj	2
4	Sabina Malik	2
5	Anjali	2

In [90]:
```
#TOP 5 Cast in Indian TV SHOWS & MOVIES
#Graphical RepresentatioTn using bar plot
fig = plt.figure (figsize= (15,10))
```

```
plt.subplot (2,2,1)
indian_tv_t_c = indian_tv_g.head(5)
x= indian_tv_t_c["cast"]
y = indian_tv_t_c["title"]
height = plt.bar(x, y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("Top 5 Actors in Indian TV SHOWS")
plt.xlabel("Top Actors")
plt.ylabel("No.of TV Shows")
plt.xticks(rotation = 15)

plt.subplot (2,2,2)
indian_mov_t_c = indian_movies_g.head(5)
x= indian_mov_t_c["cast"]
y = indian_mov_t_c["title"]
height = plt.bar(x, y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("Top 5 Actors in Indian MOVIES")
plt.xlabel("Top Actors")
plt.ylabel("No.of Movies")
plt.xticks(rotation = 15)
plt.suptitle("TOP 5 Actors in Indian TV SHOWS & MOVIES", fontsize = 24)
plt.show()
```



TOP 5 Directors in Indian TV SHOWS & MOVIES

In [91]:
```
# to find which indian director had more Movies
indian_movies = df_movies[df_movies["country"]== "India"]
indian_movies_g_d = indian_movies.groupby("director").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).re
indian_movies_g_d= indian_movies_g_d[indian_movies_g_d["director"]!= "Unknown Directors"]
indian_movies_g_d.head()
```

Out[91]:

	director	title
0	Rajiv Chilaka	22
1	Suhas Kadav	16
3	David Dhawan	9
4	Umesh Mehra	8
5	Anurag Kashyap	8

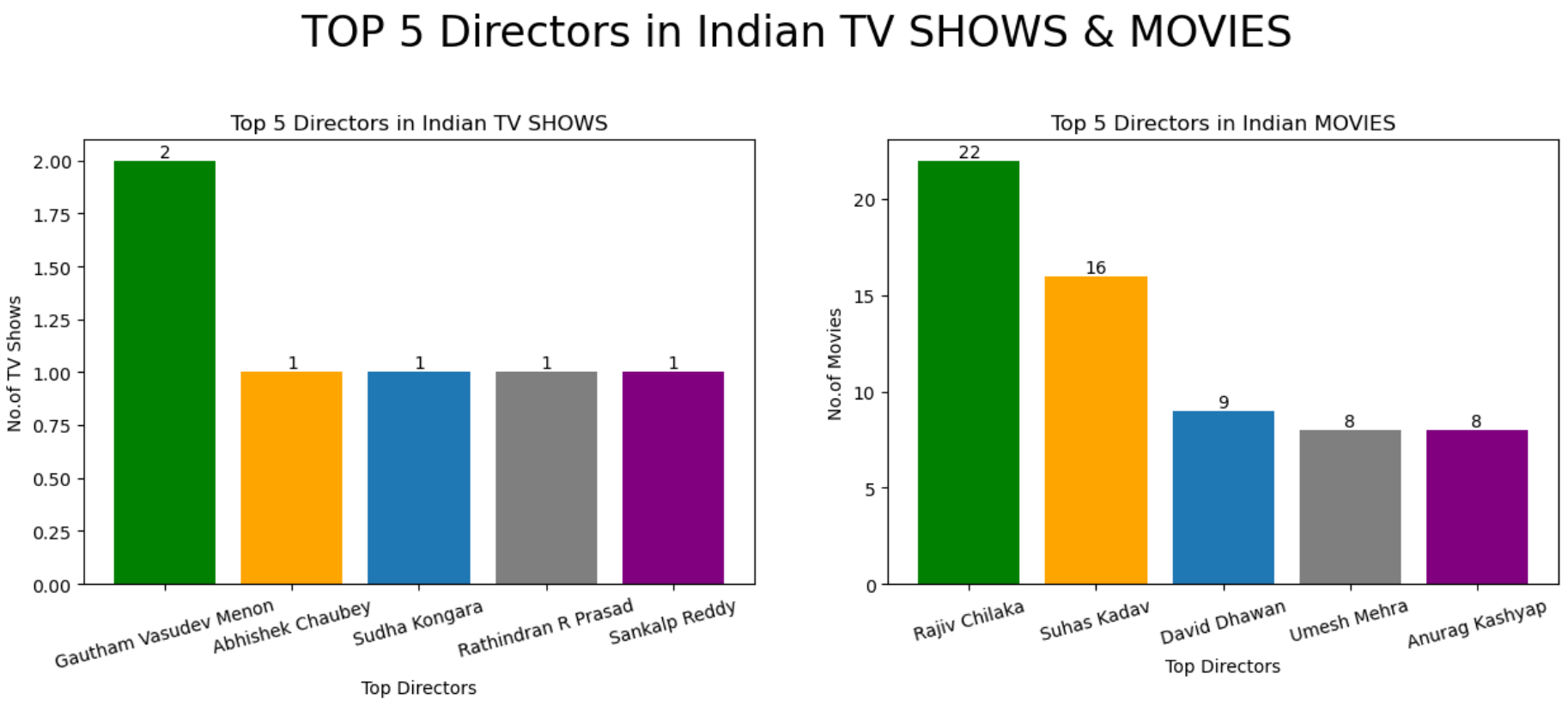
In [92]:
```
# to find which indian director had more Tv Shows
indian_tv = df_tv[df_tv["country"]== "India"]
indian_tv_g_d = indian_tv.groupby("director").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_inde
indian_tv_g_d= indian_tv_g_d[indian_tv_g_d["director"]!= "Unknown Directors"]
indian_tv_g_d.head(5)
```

Out[92]:

	director	title
1	Gautham Vasudev Menon	2
2	Abhishek Chaubey	1
3	Sudha Kongara	1
4	Rathindran R Prasad	1
5	Sankalp Reddy	1

In [93]:
```
#TOP 5 Directors in Indian TV SHOWS & MOVIES
#Graphical RepresentatioTn using bar plot
fig = plt.figure (figsize= (15,10))
plt.subplot (2,2,1)
indian_tv_t_d = indian_tv_g_d.head(5)
x= indian_tv_t_d["director"]
y = indian_tv_t_d["title"]
height = plt.bar(x, y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("Top 5 Directors in Indian TV SHOWS")
plt.xlabel("Top Directors")
plt.ylabel("No.of TV Shows")
plt.xticks(rotation = 15)

plt.subplot (2,2,2)
indian_mov_t_d = indian_movies_g_d.head(5)
x= indian_mov_t_d["director"]
y = indian_mov_t_d["title"]
height = plt.bar(x, y, color = ("g","orange","tab:blue", "tab:gray", "purple"))
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("Top 5 Directors in Indian MOVIES")
plt.xlabel("Top Directors")
plt.ylabel("No.of Movies")
plt.xticks(rotation = 15)
plt.suptitle("TOP 5 Directors in Indian TV SHOWS & MOVIES", fontsize = 24)
plt.show()
```



TOP 10 Countries Releasing TV SHOWS & MOVIES

We can understand United States, United Kingdom, Japan, South Korea & Canada are producing More TV Shows

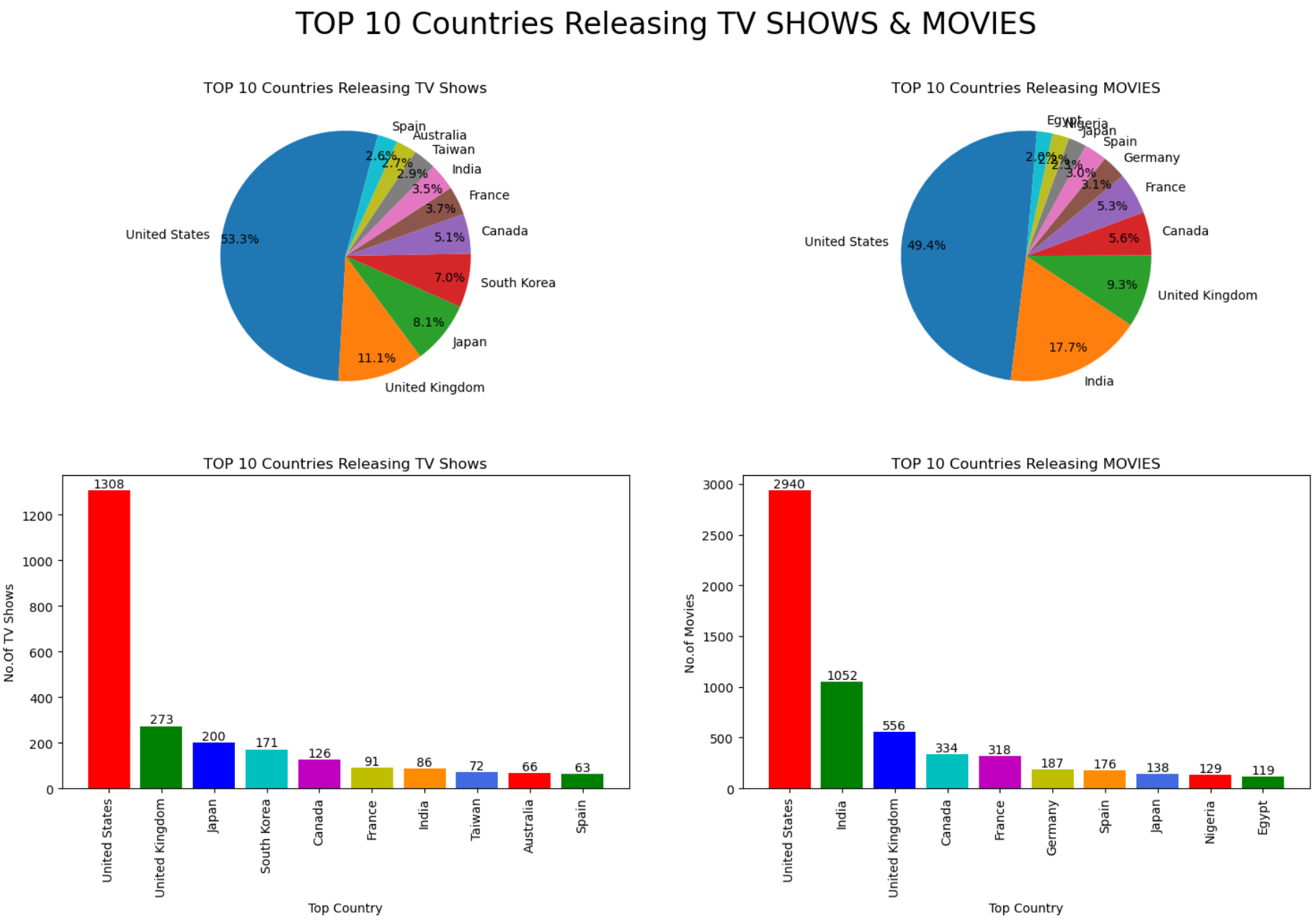
If we look into Movies We can see United States, India, United Kingdom, Canada, France are Producing More Movies.

In [94]:
```
# Percentage of TV SHOWS & MOVIES Releasing by Countries
#Graphical RepresentatioTn using Pie Chart
fig = plt.figure (figsize= (18,10))
plt.subplot (2,2,1)
df_country_t = df_tv.groupby("country").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index().he
x = df_country_t["country"]
y = df_country_t["title"]
plt.pie(y, labels = x, autopct = "%1.1f%%",startangle=75, pctdistance = 0.85)
plt.title("TOP 10 Countries Releasing TV Shows")
```

```
plt.subplot (2,2,2)
df_country_m = df_movies.groupby("country").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index(
df_country_m= df_country_m[df_country_m["country"]!= "Unknown Country"]
df_country_m = df_country_m.head(10)
x = df_country_m["country"]
y = df_country_m["title"]
plt.pie(y,labels = x, autopct = "%1.1f%%",startangle=85, pctdistance = 0.8)
plt.title("TOP 10 Countries Releasing MOVIES")
plt.suptitle("TOP 10 Countries Releasing TV SHOWS & MOVIES", fontsize = 24)

# Top 10 Countries Releasing TV SHOWS & MOVIES
#Graphical RepresentatioTn using Bar plot
colors = ["r","g","b","c","m","y","darkorange","royalblue"]
plt.subplot (2,2,3)
df_country_t = df_tv.groupby("country").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index().he
x = df_country_t["country"]
y = df_country_t["title"]
height = plt.bar(x,y, color = colors)
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.xlabel("Top Country")
plt.ylabel("No.Of TV Shows")
plt.xticks(rotation = 90)
plt.title("TOP 10 Countries Releasing TV Shows")

plt.subplot (2,2,4)
df_country_m = df_movies.groupby("country").agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index(
df_country_m= df_country_m[df_country_m["country"]!= "Unknown Country"]
df_country_m = df_country_m.head(10)
x = df_country_m["country"]
y = df_country_m["title"]
height = plt.bar(x,y, color = colors)
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("TOP 10 Countries Releasing MOVIES")
plt.xlabel("Top Country")
plt.ylabel("No.of Movies")
plt.xticks(rotation = 90)
plt.suptitle("TOP 10 Countries Releasing TV SHOWS & MOVIES", fontsize = 24)
plt.show()
```



How many Movies/ TV Shows Released in Which Month of the Year

In [95]:
```
df_mov1 = df_movies.groupby(["month","type"]).agg({"title":"nunique"}).reset_index()
df_mov1.head()
```

Out[95]:

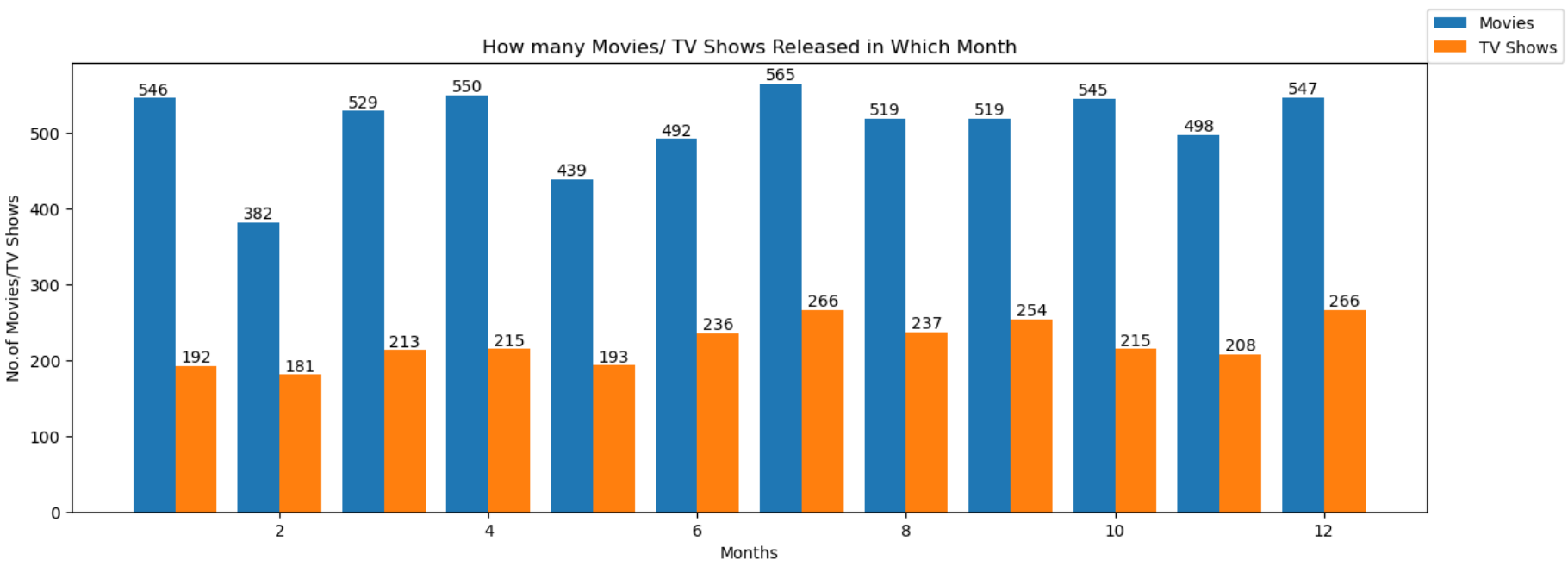
	month	type	title
0	1	Movie	546
1	2	Movie	382
2	3	Movie	529
3	4	Movie	550
4	5	Movie	439

In [96]:
```
df_tv1 = df_tv.groupby(["month","type"]).agg({"title":"nunique"}).reset_index()
df_tv1.head()
```

Out[96]:

	month	type	title
0	1	TV Show	192
1	2	TV Show	181
2	3	TV Show	213
3	4	TV Show	215
4	5	TV Show	193

In [97]:
```
fig = plt.figure(figsize = (15, 5))
height = plt.bar(df_mov1["month"]  -0.2 , df_mov1["title"] , - 0.4, label = "Movies" )
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
height = plt.bar(df_tv1["month"]  + 0.2 ,df_tv1["title"] , +0.4, label = "TV Shows")
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.legend(loc = (1,1))
plt.title("How many Movies/ TV Shows Released in Which Month")
plt.xlabel("Months")
plt.ylabel("No.of Movies/TV Shows")
plt.show()
```



Heat Map & Pair plots

1) we will create more numerical fields to get a Heat map and pairplot here
2) although there are no meaningful numerical features in this dataset, but we can attempt to see if we can find something interesting in the data that we have.

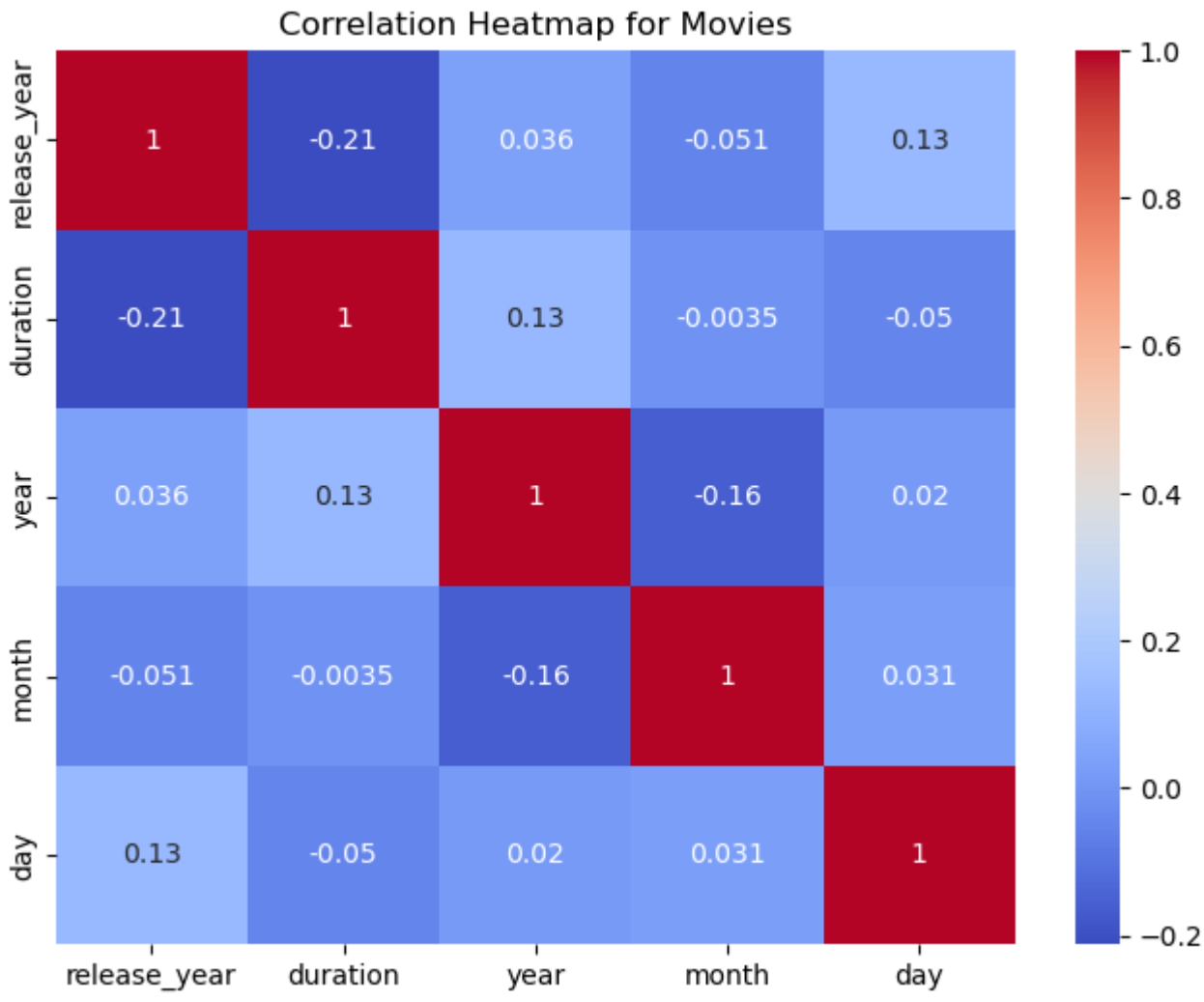
In [98]:
```
df_dur = df_movies.groupby(["duration","release_year","year","month","day"]).agg({"title":"nunique"}).sort_values(by = ["title
df_dur.head(5)
```

Out[98]:

	duration	release_year	year	month	day	title
0	53	2017	2019	7	1	6
1	54	2017	2019	7	1	6
2	64	2013	2021	7	22	5
3	122	2021	2021	8	23	4
4	97	2010	2020	1	1	3

In [99]:
```
# Correlation Heatmap for Movies
```

```
correlation_matrix = df_dur[["release_year", "duration","year","month","day"]].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title('Correlation Heatmap for Movies')
plt.show()
```



In [100…

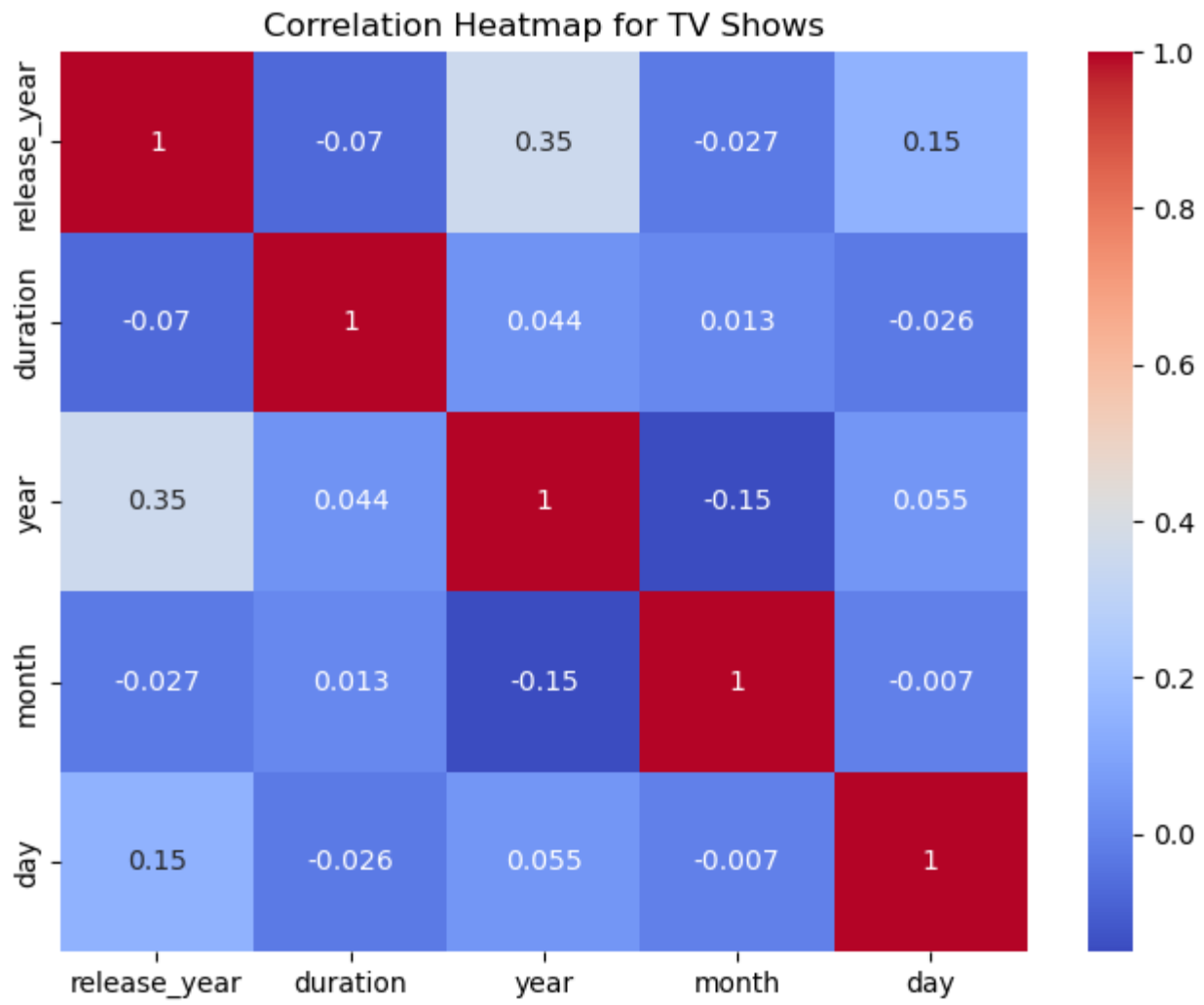
```
df_dur_tv = df_tv.groupby(["duration","release_year","year","month","day"]).agg({"title":"nunique"}).sort_values(by = ["title"
df_dur_tv.head(5)
```

Out[100…

	duration	release_year	year	month	day	title
0	1	2016	2017	8	1	8
1	1	2012	2016	12	15	7
2	1	2018	2018	10	1	6
3	1	2014	2017	7	1	6
4	1	2019	2019	8	16	6

In [101…

```
# Correlation Heatmap for TV Shows
correlation_matrix = df_dur_tv[["release_year", "duration","year","month","day"]].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title('Correlation Heatmap for TV Shows')
plt.show()
```



Pair Plot

In [102…
```
df_dur_pair = df_final.groupby(["duration","release_year","year","month","day","type"]).agg({"title":"nunique"}).sort_values(b
df_dur_pair.head(5)
```

Out[102…

	duration	release_year	year	month	day	type	title
0	1	2016	2017	8	1	TV Show	8
1	1	2012	2016	12	15	TV Show	7
2	54	2017	2019	7	1	Movie	6
3	53	2017	2019	7	1	Movie	6
4	1	2019	2019	8	16	TV Show	6

In [105…
```
# Pair plot for the Numerical columns available in the data set
sns.pairplot(df_dur_pair[["release_year", "duration","year","month","day","type"]], hue="type")
plt.show()
```

How has the number of movies/Tv shows released per year changed over the last 20–30 years?

• Growth in Number of Movies and TV shows skyrocketed since 2014

• The pace of Movie releases was somewhat slow until 1996, but since 1996 ther's been notable increase in the realese of movies and Tv shows started out with good pace since 2005

• It is interesting to note that, TV Shows ended up in higher side on comparison with movies

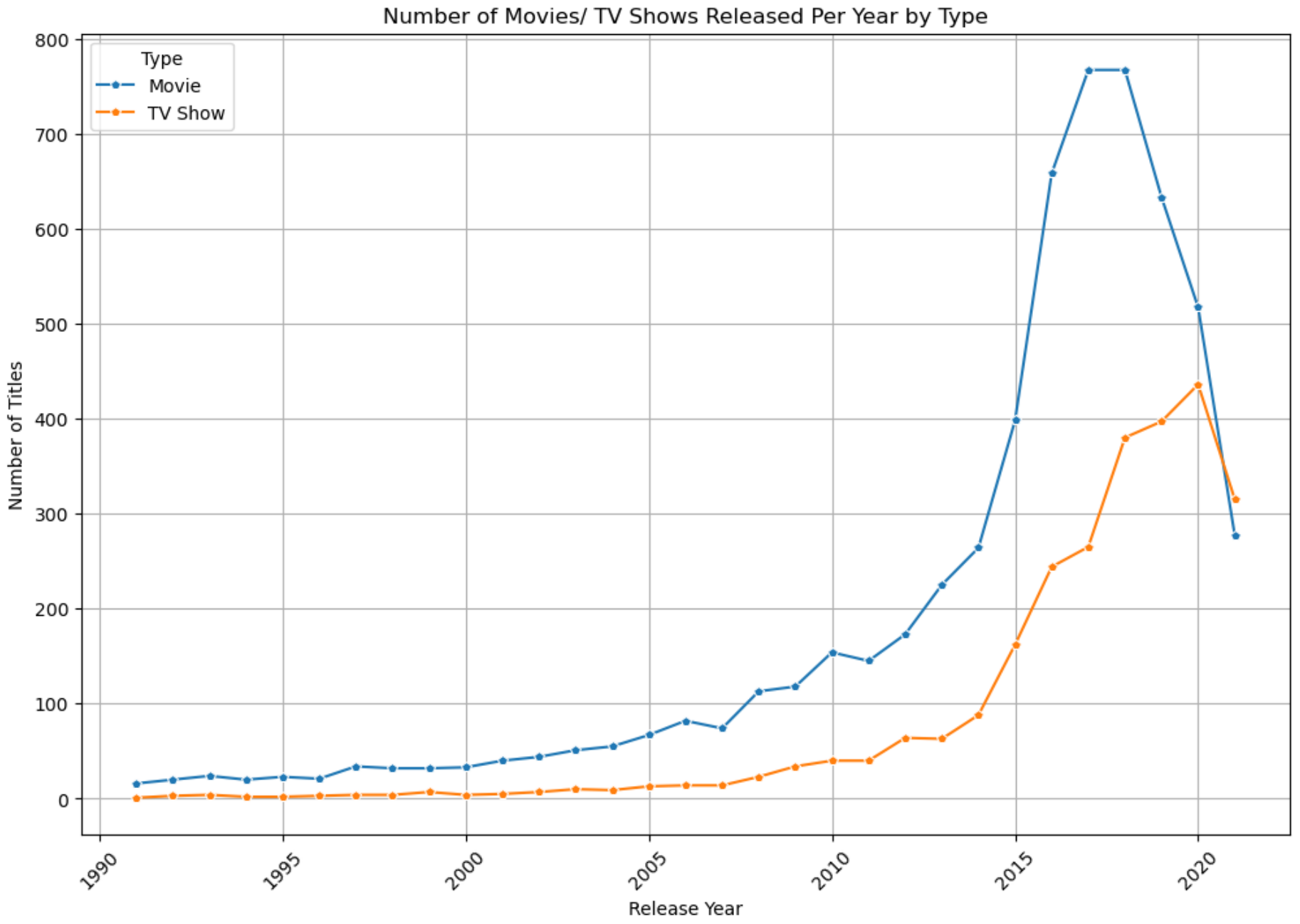
In [104…
```
titles_per_year_type = df_final.groupby(['release_year', 'type']).agg({"title": "nunique"}).reset_index()[57:]
titles_per_year_type.head()
```

Out[104…

	release_year	type	title
57	1991	Movie	16
58	1991	TV Show	1
59	1992	Movie	20
60	1992	TV Show	3
61	1993	Movie	24

In [106…
```
# Plot the number of titles released per year with hue as type
plt.figure(figsize=(12, 8))
sns.lineplot(x='release_year', y='title', hue='type', data=titles_per_year_type, marker='p')
plt.xlabel('Release Year')
plt.ylabel('Number of Titles')
plt.title('Number of Movies/ TV Shows Released Per Year by Type')
plt.xticks(rotation=45)
plt.grid(True)
```

```
plt.legend(title='Type')
plt.show()
```



Finding Outliers in No.of Seasons in TV Shows & the Duration of Movies

In [107…
```
# Finding Outliers in No.of Seasons in TV Shows
df_dur = df_tv.groupby(["duration","type"]).agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_index(
df_dur.head(5)
```

Out[107…

	duration	type	title
0	1	TV Show	1793
1	2	TV Show	425
2	3	TV Show	199
3	4	TV Show	95
4	5	TV Show	65

In [108…
```
# Finding Outliers in movie duration in Movies
df_dur_m = df_movies.groupby(["duration","type"]).agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).reset_
df_dur_m.head(5)
```

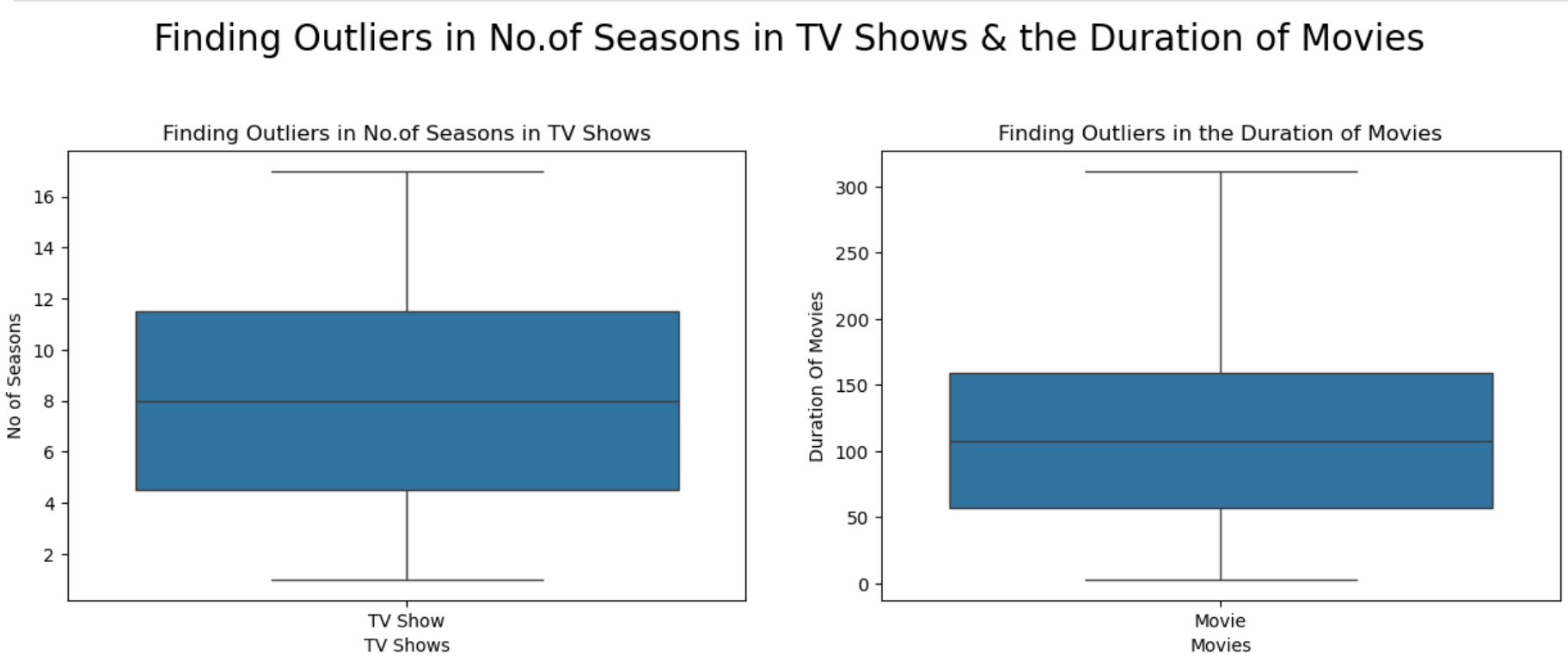
Out[108…

	duration	type	title
0	90	Movie	152
1	97	Movie	146
2	93	Movie	146
3	94	Movie	146
4	91	Movie	144

In [109…
```
#Finding Outliers in No.of Seasons in TV Shows & the Duration of Movies
#Graphical RepresentatioTn using box plot
fig = plt.figure (figsize= (15,10))
plt.subplot (2,2,1)
sns.boxplot (x = "type", y = "duration", data = df_dur)
plt.title("Finding Outliers in No.of Seasons in TV Shows")
plt.xlabel("TV Shows")
```

```
plt.ylabel("No of Seasons")

plt.subplot (2,2,2)
sns.boxplot (x = "type", y = "duration", data = df_dur_m)
plt.title("Finding Outliers in the Duration of Movies")
plt.xlabel("Movies")
plt.ylabel("Duration Of Movies")
plt.suptitle("Finding Outliers in No.of Seasons in TV Shows & the Duration of Movies", fontsize = 20)
plt.show()
```



What is the best time to launch a TV show?

Best Month to release a TV Show is July and December

Best Month to release a Movie is July

In [110…
```
df_mov_month = df_movies.groupby("month").agg({"title":"nunique"}).reset_index()
df_mov_month.head()
```

Out[110…

	month	title
0	1	546
1	2	382
2	3	529
3	4	550
4	5	439

In [111…
```
df_tv_month = df_tv.groupby("month").agg({"title":"nunique"}).reset_index()
df_tv_month.head()
```

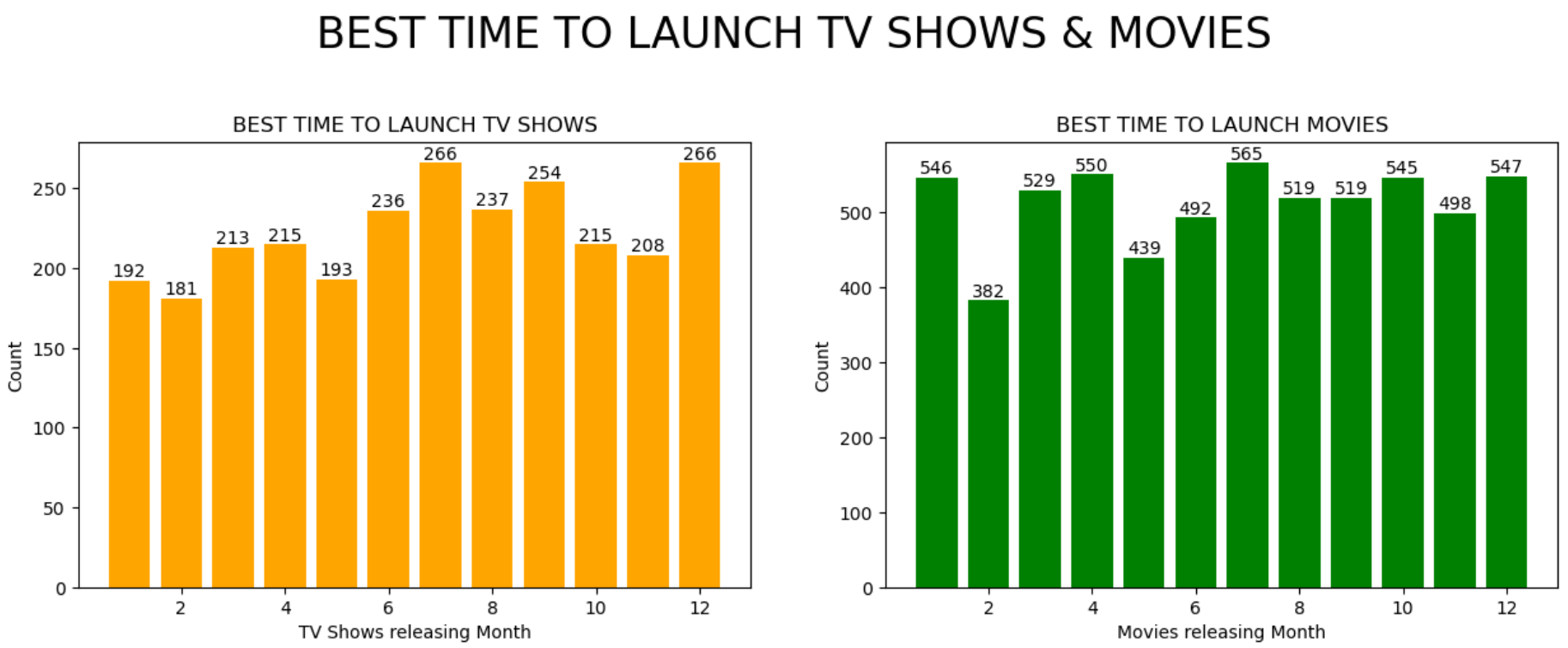
Out[111…

	month	title
0	1	192
1	2	181
2	3	213
3	4	215
4	5	193

In [112…
```
#Graphical Representation using Bar plot
fig = plt.figure (figsize= (15,10))
plt.subplot (2,2,1)
x = df_tv_month["month"]
y = df_tv_month["title"]
height = plt.bar(x,y, color ="orange")
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("BEST TIME TO LAUNCH TV SHOWS")
plt.xlabel("TV Shows releasing Month")
plt.ylabel("Count")

plt.subplot (2,2,2)
x = df_mov_month["month"]
```

```
y = df_mov_month["title"]
height = plt.bar(x,y, color ="g")
for bar in height:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
plt.title("BEST TIME TO LAUNCH MOVIES")
plt.xlabel("Movies releasing Month")
plt.ylabel("Count")
plt.suptitle("BEST TIME TO LAUNCH TV SHOWS & MOVIES", fontsize = 24)
plt.show()
```



Understanding what content is available in different countries

Top 5 Countries with Number of Titles per Top 5 Genre in Movies

In [113…

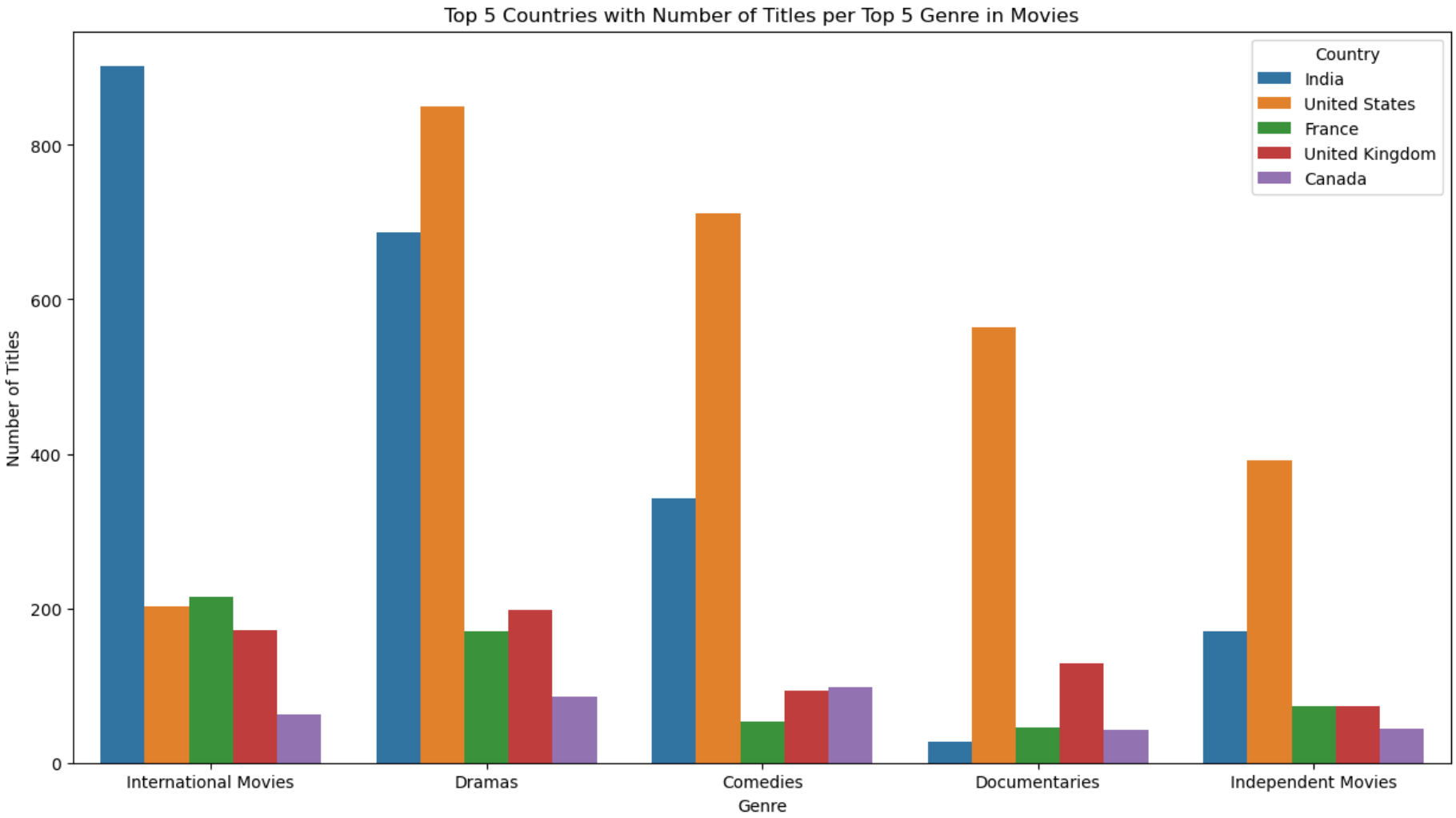
```
# Group by country and count the number of unique titles
country_titles_m = df_movies.groupby('country').agg({"title": "nunique"}).reset_index()
top_5_countries = country_titles_m.sort_values(by='title', ascending=False).head(5)

# Filter the dataset to include only the top 5 countries
top_5_country_filter = df_movies[df_movies['country'].isin(top_5_countries['country'])]
country_genre_titles = top_5_country_filter.groupby(['country', 'listed_in']).agg({"title": "nunique"}).reset_index()
genre_counts = top_5_country_filter.groupby('listed_in')['title'].nunique().reset_index()

top_genres = genre_counts.sort_values(by='title', ascending=False).head(5)['listed_in'].tolist()

country_genre_titles_top5 = country_genre_titles[country_genre_titles['listed_in'].isin(top_genres)].sort_values(by='title', a

plt.figure(figsize=(15, 8))
sns.barplot(y='title', x='listed_in', hue='country', data=country_genre_titles_top5)
plt.xlabel('Genre')
plt.ylabel('Number of Titles')
plt.title('Top 5 Countries with Number of Titles per Top 5 Genre in Movies')
plt.legend(title='Country')
plt.show()
```



Top 5 Countries with Number of Titles per Top 5 Genre in TV Shows

In [114…

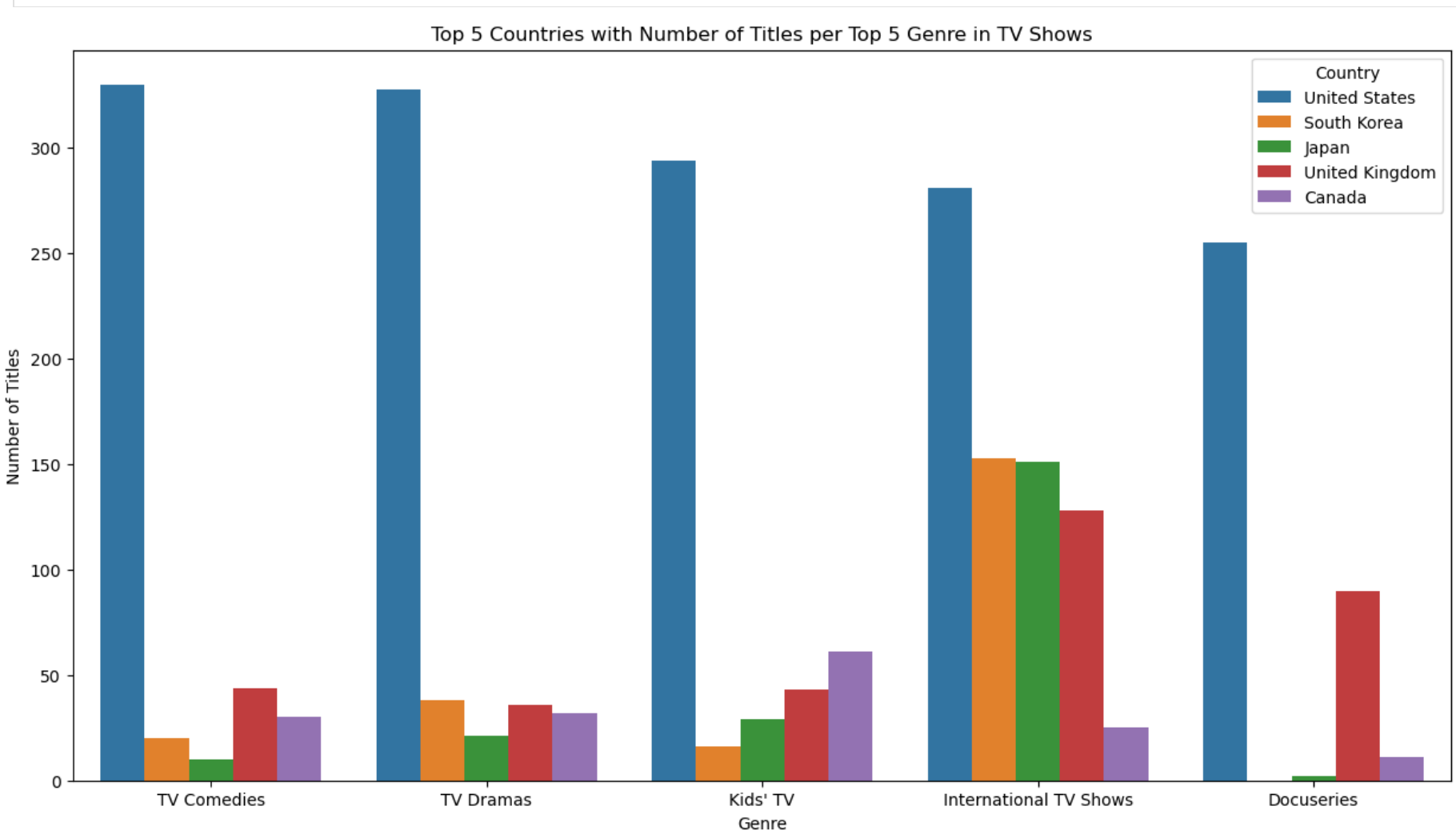
```
# Group by country and count the number of unique titles
country_titles_t = df_tv.groupby('country').agg({"title": "nunique"}).reset_index()
top_5_countries_t = country_titles_t.sort_values(by='title', ascending=False).head(5)

# Filter the dataset to include only the top 5 countries
top_5_country_filter_t = df_tv[df_tv['country'].isin(top_5_countries_t['country'])]
country_genre_titles_t = top_5_country_filter_t.groupby(['country', 'listed_in']).agg({"title": "nunique"}).reset_index()
genre_counts_t = top_5_country_filter_t.groupby('listed_in')['title'].nunique().reset_index()

top_genres_t = genre_counts_t.sort_values(by='title', ascending=False).head(5)['listed_in'].tolist()

country_genre_titles_top5_t = country_genre_titles_t[country_genre_titles_t['listed_in'].isin(top_genres_t)].sort_values(by='t

plt.figure(figsize=(15, 8))
sns.barplot(y='title', x='listed_in', hue='country', data=country_genre_titles_top5_t)
plt.xlabel('Genre')
plt.ylabel('Number of Titles')
plt.title('Top 5 Countries with Number of Titles per Top 5 Genre in TV Shows')
plt.legend(title='Country')
plt.show()
```



Distribution of TV SHOWS & MOVIES by Date Added in Netflix

In [115…
```
df_t_d = df_tv.groupby("date_added").agg({"title":"nunique"}).reset_index()
df_t_d.head(5)
```

Out[115…

	date_added	title
0	2008-02-04	1
1	2013-03-31	1
2	2013-08-02	1
3	2013-09-01	1
4	2013-10-08	1

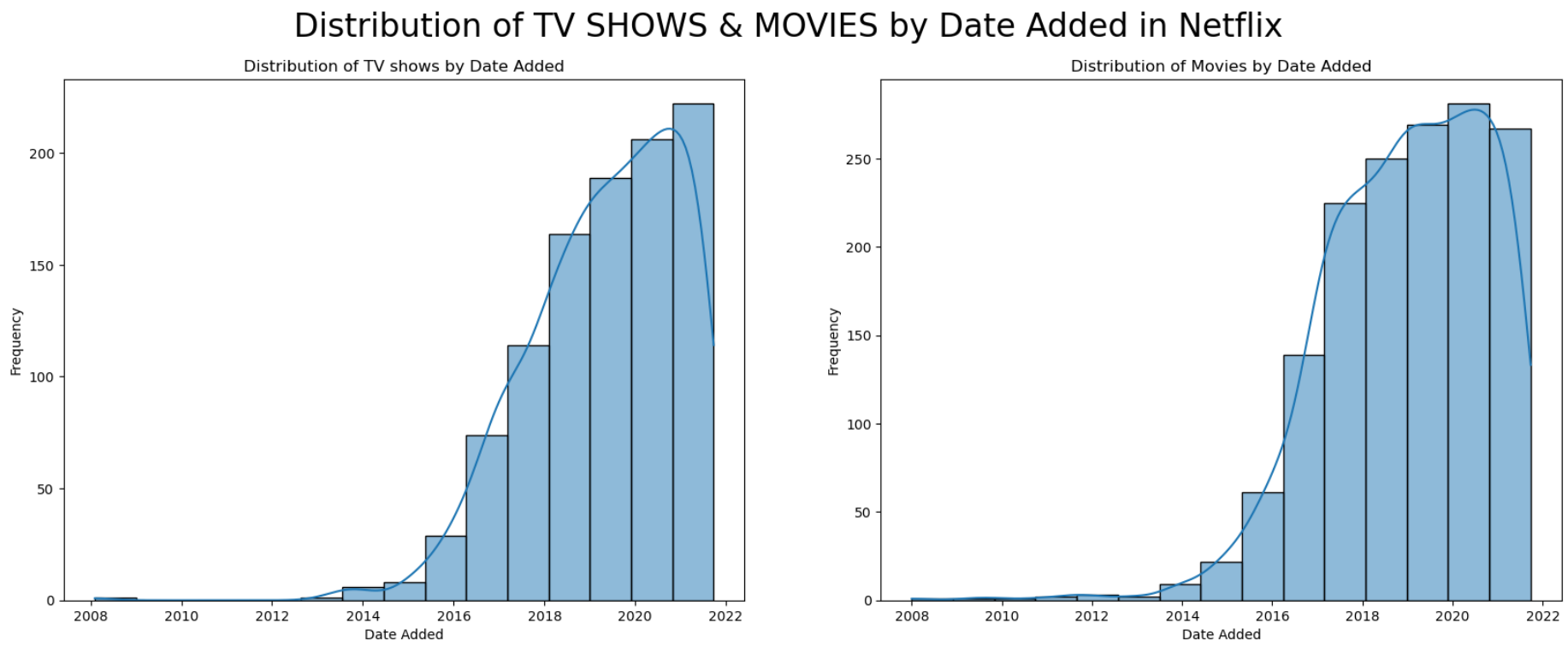
In [116…
```
df_m_d = df_movies.groupby("date_added").agg({"title":"nunique"}).reset_index()
df_m_d.head(5)
```

Out[116…

	date_added	title
0	2008-01-01	1
1	2009-05-05	1
2	2009-11-18	1
3	2010-11-01	1
4	2011-05-17	1

In [117…
```
plt.figure(figsize=(20, 7))
plt.subplot(1, 2, 1)
sns.histplot(df_t_d['date_added'], bins=15, kde=True)
plt.title('Distribution of TV shows by Date Added')
plt.xlabel('Date Added')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
sns.histplot(df_m_d['date_added'], bins=15, kde=True)
plt.title('Distribution of Movies by Date Added')
plt.xlabel('Date Added')
plt.ylabel('Frequency')
plt.suptitle("Distribution of TV SHOWS & MOVIES by Date Added in Netflix", fontsize = 24)
plt.show()
```



Distribution of TV SHOWS & MOVIES by Seasons & Durations

In [118…
```
df_t_dur = df_tv.groupby("duration").agg({"title":"nunique"}).reset_index()
df_t_dur.head(5)
```

Out[118…

	duration	title
0	1	1793
1	2	425
2	3	199
3	4	95
4	5	65

In [119…

```
df_mo_dur = df_movies.groupby("duration").agg({"title":"nunique"}).reset_index()
df_mo_dur.head(5)
```

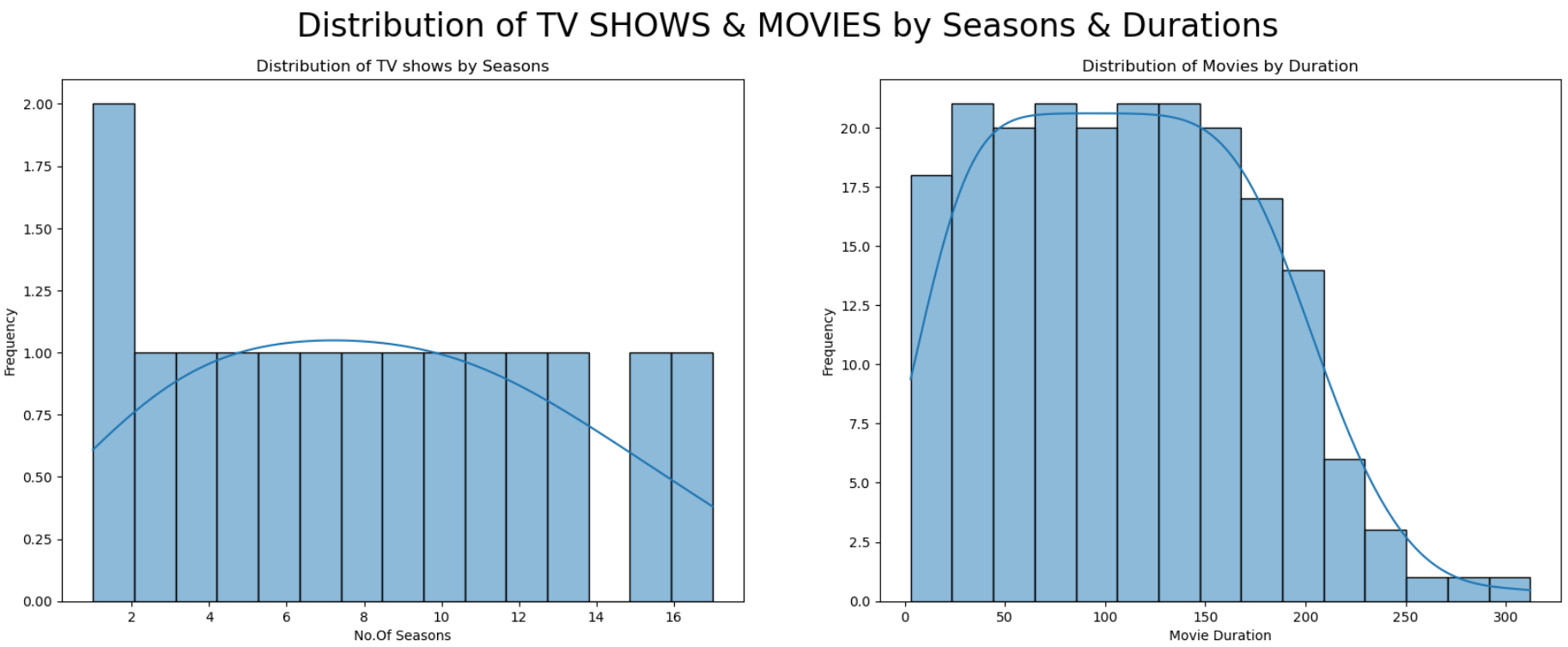
Out[119…

	duration	title
0	3	1
1	5	1
2	8	1
3	9	1
4	10	1

In [120…

```
plt.figure(figsize=(20, 7))
plt.subplot(1, 2, 1)
sns.histplot(df_t_dur['duration'], bins=15, kde=True)
plt.title('Distribution of TV shows by Seasons')
plt.xlabel('No.Of Seasons')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
sns.histplot(df_mo_dur['duration'], bins=15, kde=True)
plt.title('Distribution of Movies by Duration')
plt.xlabel('Movie Duration')
plt.ylabel('Frequency')
plt.suptitle("Distribution of TV SHOWS & MOVIES by Seasons & Durations", fontsize = 24)
plt.show()
```



Finding the Top 5 Actor & Director Pair for Movies & TV Shows

In [121…

```
cast_director_m = df_movies.groupby(["cast","director"]).agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False)
cast_director_m= cast_director_m[(cast_director_m["director"]!= "Unknown Directors") & (cast_director_m["cast"]!= "Unknown Act
cast_director_m.head(5)
```

Out[121…

	cast	director	title
1	Rajesh Kava	Rajiv Chilaka	19
2	Julie Tejwani	Rajiv Chilaka	19
3	Rupa Bhimani	Rajiv Chilaka	18
4	Jigna Bhardwaj	Rajiv Chilaka	18
5	Vatsal Dubey	Rajiv Chilaka	16

In [122…

```
cast_director_t = df_tv.groupby(["cast","director"]).agg({"title":"nunique"}).sort_values(by = ["title"], ascending=False).res
cast_director_t= cast_director_t[(cast_director_t["director"]!= "Unknown Directors") & (cast_director_t["cast"]!= "Unknown Act
```

```
cast_director_t.head(5)
```

Out[122...

	cast	director	title
796	David Attenborough	Alastair Fothergill	3
971	Dave Chappelle	Stan Lathan	2
1653	Sung Dong-il	Shin Won-ho	2
1797	Gautham Vasudev Menon	Gautham Vasudev Menon	2
1846	Anjali	Gautham Vasudev Menon	2

Insights based on Non-Graphical and Visual Analysis

• Distribution of Netflix collection in Movies is 69.6% and in TV Shows it is 30.4%

• Netflix collection of Tv shows span from Year 1925 to 2021 and Movies spans from Year 1942 to 2021.

• collection of Netflix consist 8807 Titles, 36440 Actors, 4994 Directors, 42 Genre, 124 Countries and 15 Content Ratings are there.

• TOP 5 GENRE IN TV SHOWS:

1. International Movies
2. Dramas
3. Tv Comedies
4. Crime TV Shows
5. Kids' TV

• TOP 5 GENRE IN MOVIES:

1. International Movies
2. Dramas
3. Comedies
4. Documentaries
5. Action & Adventure

• TOP 5 DIRECTOR IN TV SHOWS:

1. Ken Burns
2. Alastair Fothergill
3. Stan Lathan
4. Joe Berlinger
5. Hsu Fu-chun

• TOP 5 DIRECTOR IN MOVIES:

1. Rajiv Chilaka
2. Jan Suter
3. Raúl Campos
4. Suhas Kadav
5. Marcus Raboy

• TOP 5 ACTORS IN TV SHOWS:

1. Takahiro Sakurai
2. Yuki Kaji
3. Junichi Suwabe
4. Daisuke Ono
5. Ai Kayano

• TOP 5 ACTORS IN MOVIES:

1. Anupam Kher
2. Shah Rukh Khan
3. Naseeruddin Shah
4. Om Puri
5. Akshay Kumar

• TOP 5 Ratings IN TV SHOWS:

1. TV-MA
2. TV-14

3. TV-PG
4. TV-Y7
5. TV-Y

• TOP 5 Ratings IN MOVIES:

1. TV-MA
2. TV-14
3. R
4. TV-PG
5. PG-13

Most of Movies Movies fall under 60–145 Min duration Range

It is observed that Netflix is More focused in Mature rated Movies / TV shows

Business Insights

It is known that R rating in india has very few titles (6)

93.52% of the content was added in the netfix in 5 years(2017–2021) and the rest 6.48% was added in 9 years (2008–2017)

Average Duration of Movies is nearly 106 Min and Tv shows is 2 seasons

Top 10 Genre covers 65.77% and the rest 32 Genre covers 34.23%

Recommendation:

• In a Tech platform like Netflix, Business metrics like DAU (Daily active users), WAU (Weekly active users) & MAU (Monthly active users) are crucially important to look out for, and the comparison of Movie and TV series, Tv series would increase the watch time of users, in turn users would be more likely of renewing their subscription as well as more watch time ensures higher Ad revenues.

• As seen in the business insight, a movie has a possibility of 106 minutes of watch time against a Tv show of 2 seasons

• Say 2 seasons has around 10 episodes equates to 20 episodes with each assumed duration of 40 minutes totalling to 800 minutes

• As seen in the Business Insight, Most of the country's content potential is yet to unfold, and many regional content is witnessed to breakthrough the popularity in recent years, so it is recommended to search for such regional content that may have the right potential to attract more users with the help of newly introduced novelty.

• It is understood that out of selective revenue models of Netflix, Ad revenue is indeed an important one

• Countries like Japan, Greece, Italy, Germany etc are known to have higher % of old people, Netflix can study the demographics of all countries to target the most optimal audience, in general it is undoubtedly the adult users that is targeted but, tailoring the content based on Demographics of country might help expand the userbase.

• Since pandemic, a boom in online education is seen (also known as zoom era), Netflix can tap into this demand and enable content for kids where entertaining education can help expand userbase under "Netflix for kids!".

• We can interpret top Genres here as widely accepted genre and content produced under such genre would have higher probability of popularity and acceptance by the audience.

In [107…

```
df_final.to_csv("netflix_cleaned.csv", index = False)
```