

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: data = pd.read_csv('netflix.csv')
```

```
In [3]: data.head()
```

	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...

```
In [4]: data.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 [5]: data.shape
```

```
Out[5]: (8807, 12)
```

```
In [7]: print(f"TOTAL ROWS : {data.shape[0]}")  
print(f"TOTAL COLUMNS : {data.shape[1]}")
```

TOTAL ROWS : 8807  
TOTAL COLUMNS : 12

```
In [9]: print(f"SIZE OF DataFrame : {data.size}")
```

SIZE OF DataFrame : 105684

```
In [12]: print(f"Index of the DataFrame : {data.index}")
```

Index of the DataFrame : RangeIndex(start=0, stop=8807, step=1)

```
In [14]: print(f"Columns : {data.columns}")
```

Columns : Index(['show\_id', 'type', 'title', 'director', 'cast', 'country', 'date\_added',  
'release\_year', 'rating', 'duration', 'listed\_in', 'description'],  
dtype='object')

```
In [15]: data.describe()
```

```
Out[15]: release_year
```

<b>count</b>	8807.000000
<b>mean</b>	2014.180198
<b>std</b>	8.819312
<b>min</b>	1925.000000
<b>25%</b>	2013.000000
<b>50%</b>	2017.000000
<b>75%</b>	2019.000000
<b>max</b>	2021.000000

```
In [16]: data.describe().T
```

```
Out[16]: count mean std min 25% 50% 75% max
```

<b>release_year</b>	8807.0	2014.180198	8.819312	1925.0	2013.0	2017.0	2019.0	2021.0
---------------------	--------	-------------	----------	--------	--------	--------	--------	--------

```
In [17]: data.isnull()
```

```
Out[17]:
```

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

8807 rows × 12 columns

```
In [18]: data.isnull().sum()
```

```
Out[18]:
```

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

```
In [19]: data.duplicated()
```

```
Out[19]:
```

0	False
1	False
2	False
3	False
4	False
...	
8802	False
8803	False
8804	False
8805	False
8806	False
Length: 8807, dtype: bool	

```
In [20]: data.duplicated().sum()
```

```
Out[20]: 0
```

```
In [21]: #There are null values for director, cast, country, date_added, rating, and duration
```

```
In [22]: #There are no duplicates int he DataFrame
```

```
In [36]: #Taking a copy of data into data_copy  
data_copy = data.copy()
```

```
In [300...]: # Convert categorical columns to 'category' data type  
data = data_copy.copy()  
categorical_columns = ['type', 'country', 'listed_in']  
for col in categorical_columns:  
    data[col] = data[col].astype('category')
```

```
In [301]: data.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   category  
 2   title       8807 non-null   object    
 3   director    6173 non-null   object    
 4   cast        7982 non-null   object    
 5   country     7976 non-null   category  
 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   category  
 11  description  8807 non-null   object    
dtypes: category(3), int64(1), object(8)  
memory usage: 704.7+ KB
```

```
In [302]: data.nunique()
```

```
Out[302]: show_id      8807  
type          2  
title        8807  
director     4528  
cast         7692  
country      748  
date_added   1767  
release_year  74  
rating       17  
duration     220  
listed_in    514  
description  8775  
dtype: int64
```

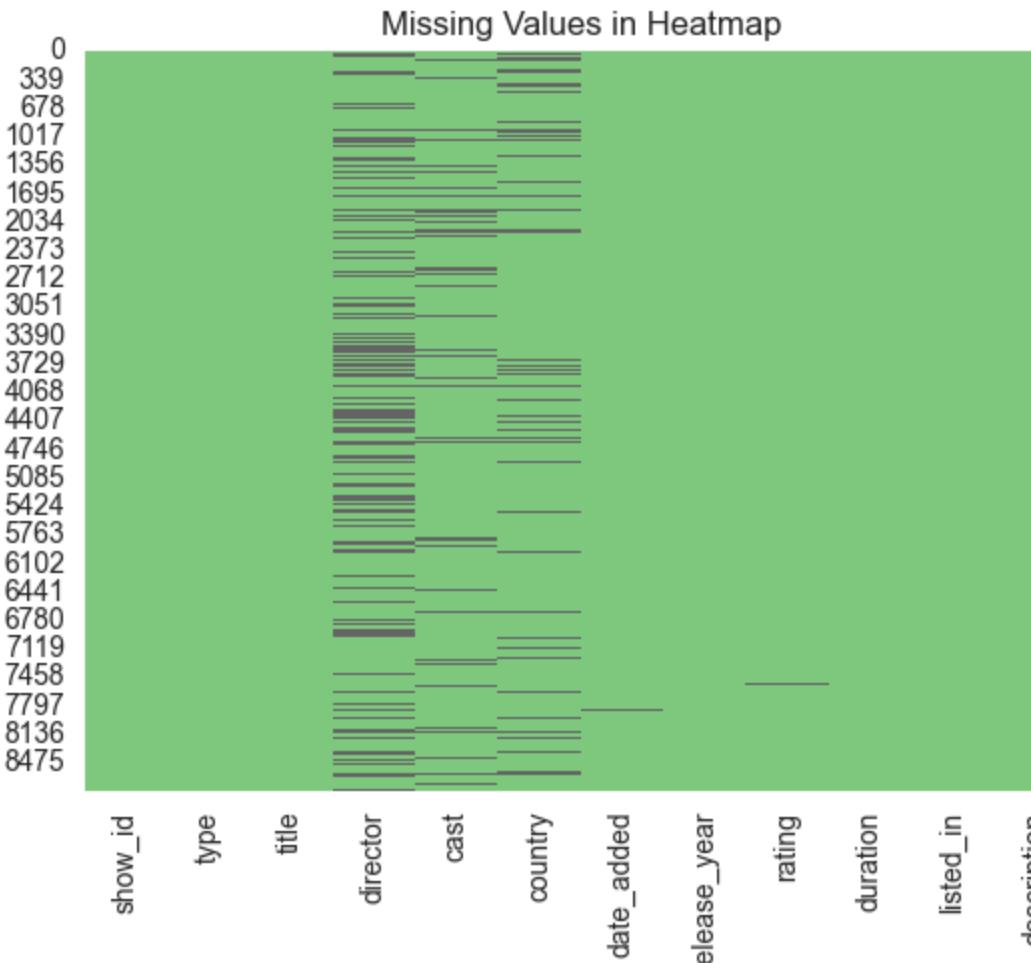
```
In [303...]: data.isnull().sum()
```

```
Out[303...]:
```

show_id	0
type	0
title	0
director	2634
cast	825
country	831
date_added	10
release_year	0
rating	4
duration	3
listed_in	0
description	0

dtype: int64

```
In [304...]: # Visualize missing values  
sns.heatmap(data.isnull(), cbar=False, cmap='Accent')  
plt.title('Missing Values in Heatmap')  
plt.show()
```



```
In [305...]: data['type'].unique()
```

```
Out[305...]:
```

['Movie', 'TV Show']
Categories (2, object): ['Movie', 'TV Show']

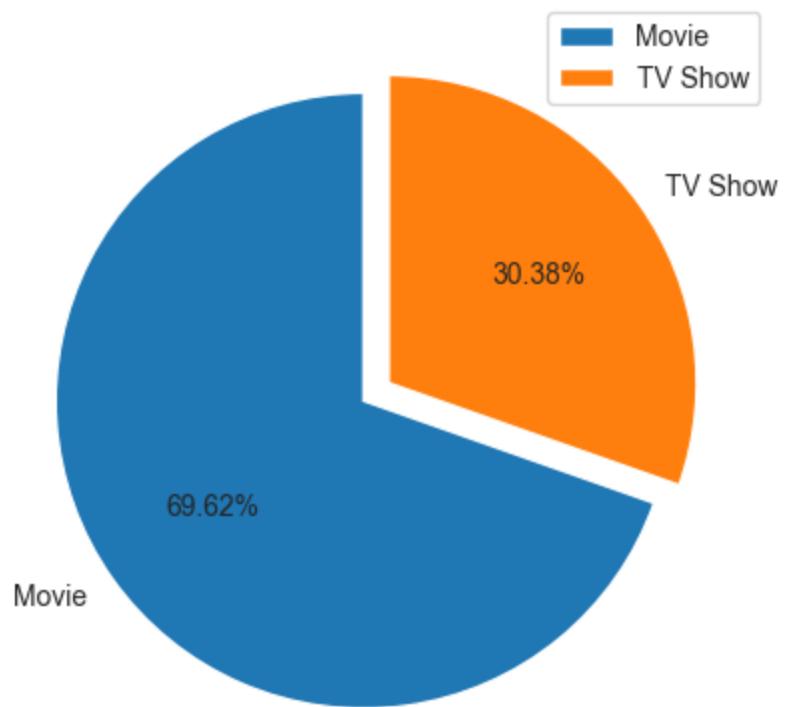
```
In [306... type_counts = data['type'].value_counts()  
type_counts
```

```
Out[306... type  
Movie      6131  
TV Show    2676  
Name: count, dtype: int64
```

```
In [307... #There are 6131 Movies and 2676 TV Shows
```

```
In [308... type_counts = data['type'].value_counts()  
plt.figure(figsize=(5,5))  
plt.pie(type_counts.values, labels = type_counts.index, autopct = '%.2f%%', startangle = 90, explode = [0.1, 0])  
plt.legend()  
plt.title('Contribution of Movies and TV Shows in Netflix', fontsize=16)  
plt.show()
```

Contribution of Movies and TV Shows in Netflix



```
In [309... #Movie count is more in Netflix compared to TV Shows. Movie - 30.38% and TV Shows - 69.62%
```

```
In [310... data['rating'].unique()
```

```
Out[310... array(['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'], dtype=object)
```

```
In [311... rating_values = data['rating'].value_counts()  
rating_values
```

```
Out[311... rating
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: count, dtype: int64
```

```
In [312... # Count of unique values in the rating column
data["rating"].value_counts().head()
```

```
Out[312... rating
TV-MA    3207
TV-14    2160
TV-PG    863
R        799
PG-13    490
Name: count, dtype: int64
```

```
In [313... data["rating"].value_counts().tail(5)
```

```
Out[313... rating
NC-17    3
UR      3
74 min   1
84 min   1
66 min   1
Name: count, dtype: int64
```

```
In [314... # Most frequent value in the rating
mode_rating = data["rating"].mode()[0]
mode_rating
```

```
Out[314... 'TV-MA'
```

```
In [315... # Filling the Null values with mode value
data["rating"] = data["rating"].fillna(mode_rating)
```

```
In [316... data["rating"].isna().sum()
```

```
Out[316... 0
```

```
In [317... #ALL the Null values in rating column are replaced with mode value
```

```
In [318... data["rating"].unique()
```

```
Out[318... array(['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',
   'TV-Y7-FV', 'UR'], dtype=object)
```

```
In [319... #There are "66 min, 74 min, 84 min" in the rating column which are wrong values
```

```
In [320... data[data["rating"].str.contains("min")]
```

```
Out[320...

| show_id | type  | title | director                             | cast       | country    | date_added | release_year  | rating             | duration | listed_in | description |        |                                                   |
|---------|-------|-------|--------------------------------------|------------|------------|------------|---------------|--------------------|----------|-----------|-------------|--------|---------------------------------------------------|
| 5541    | s5542 | Movie | Louis C.K.                           | 2017       | Louis C.K. | Louis C.K. | United States | April 4, 2017      | 2017     | 74 min    | NaN         | Movies | Louis C.K. muses on religion, eternal love, gi... |
| 5794    | s5795 | Movie | Louis C.K.: Hilarious                | Louis C.K. | Louis C.K. | Louis C.K. | United States | September 16, 2016 | 2010     | 84 min    | NaN         | Movies | Emmy-winning comedy writer Louis C.K. brings h... |
| 5813    | s5814 | Movie | Louis C.K.: Live at the Comedy Store | Louis C.K. | Louis C.K. | Louis C.K. | United States | August 15, 2016    | 2015     | 66 min    | NaN         | Movies | The comic puts his trademark hilarious/thought... |


```

```
In [321... data.loc[data["rating"].str.contains("min"), "rating"] = mode_rating
```

```
In [322... data["rating"].unique()
```

```
Out[322... array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
   'TV-G', 'G', 'NC-17', 'NR', 'TV-Y7-FV', 'UR'], dtype=object)
```

```
In [323... #There are wrong rating values now in the DataFrame
```

```
In [324... data["rating"].value_counts()
```

```
Out[324...

| rating   | count |
|----------|-------|
| TV-MA    | 3214  |
| 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     |


Name: count, dtype: int64
```

```
In [325... import warnings
```

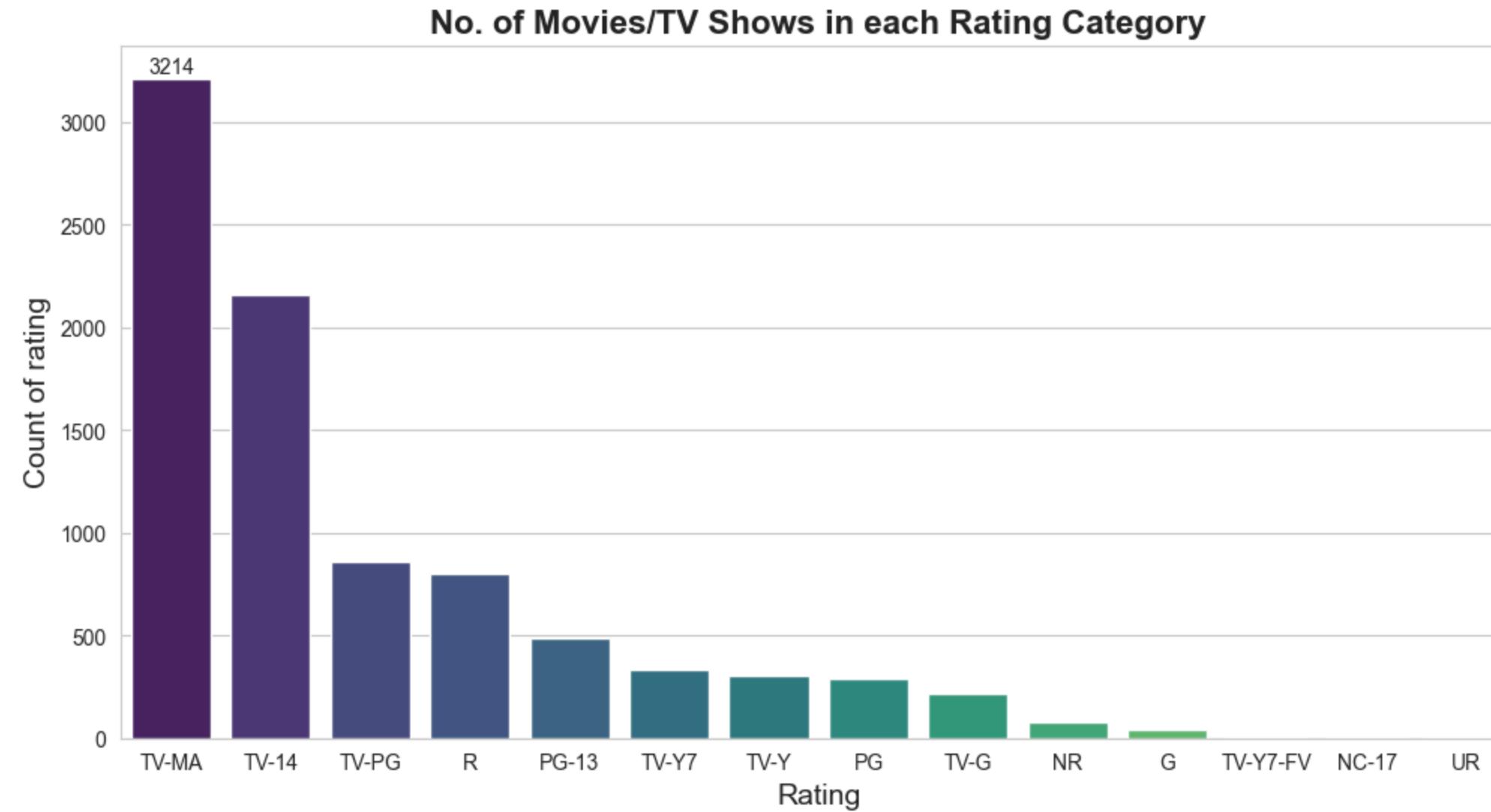
```
warnings.filterwarnings("ignore", category=FutureWarning)
```

```
In [326... sns.set_style("whitegrid")
fig, ax = plt.subplots(figsize=(12, 6))
```

```

ax = sns.countplot(data=data, x="rating", order=data["rating"].value_counts().index, saturation=0.75, palette="viridis", legend=False)
ax.set_title("No. of Movies/TV Shows in each Rating Category", fontsize=16, fontweight='bold')
ax.set_xlabel("Rating", fontsize=14)
ax.set_ylabel("Count of rating", fontsize=14)
ax.bar_label(ax.containers[0], fontsize=10)
plt.show()

```



In [327]: `data.shape`

Out[327]: `(8807, 12)`

In [328]: `#Rating "TV-MA" is in the top and "TV-14" in second top`

In [329]: `#Rating "NC-17", "UR" are having the bottom (less in number)`

In [330]: `#Type wise rating and total count in each`  
`data[["type","rating"]].value_counts()`

```
Out[330...]: type    rating
Movie      TV-MA     2067
          TV-14     1427
TV Show    TV-MA     1147
Movie      R         797
TV Show    TV-14     733
Movie      TV-PG     540
          PG-13     490
TV Show    TV-PG     323
Movie      PG        287
TV Show    TV-Y7     195
          TV-Y      176
Movie      TV-Y7     139
          TV-Y      131
          TV-G      126
TV Show    TV-G      94
Movie      NR        75
          G         41
TV Show    NR        5
Movie      TV-Y7-FV   5
          UR        3
          NC-17     3
TV Show    R         2
          TV-Y7-FV   1
Name: count, dtype: int64
```

```
In [331...]: #Movies ratings
data[data["type"] == "Movie"]["rating"].value_counts()
```

```
Out[331...]: rating
TV-MA      2067
TV-14      1427
R          797
TV-PG      540
PG-13      490
PG          287
TV-Y7      139
TV-Y       131
TV-G       126
NR          75
G           41
TV-Y7-FV    5
NC-17      3
UR          3
Name: count, dtype: int64
```

```
In [332...]: #TV Show ratings
data[data["type"] == "TV Show"]["rating"].value_counts()
```

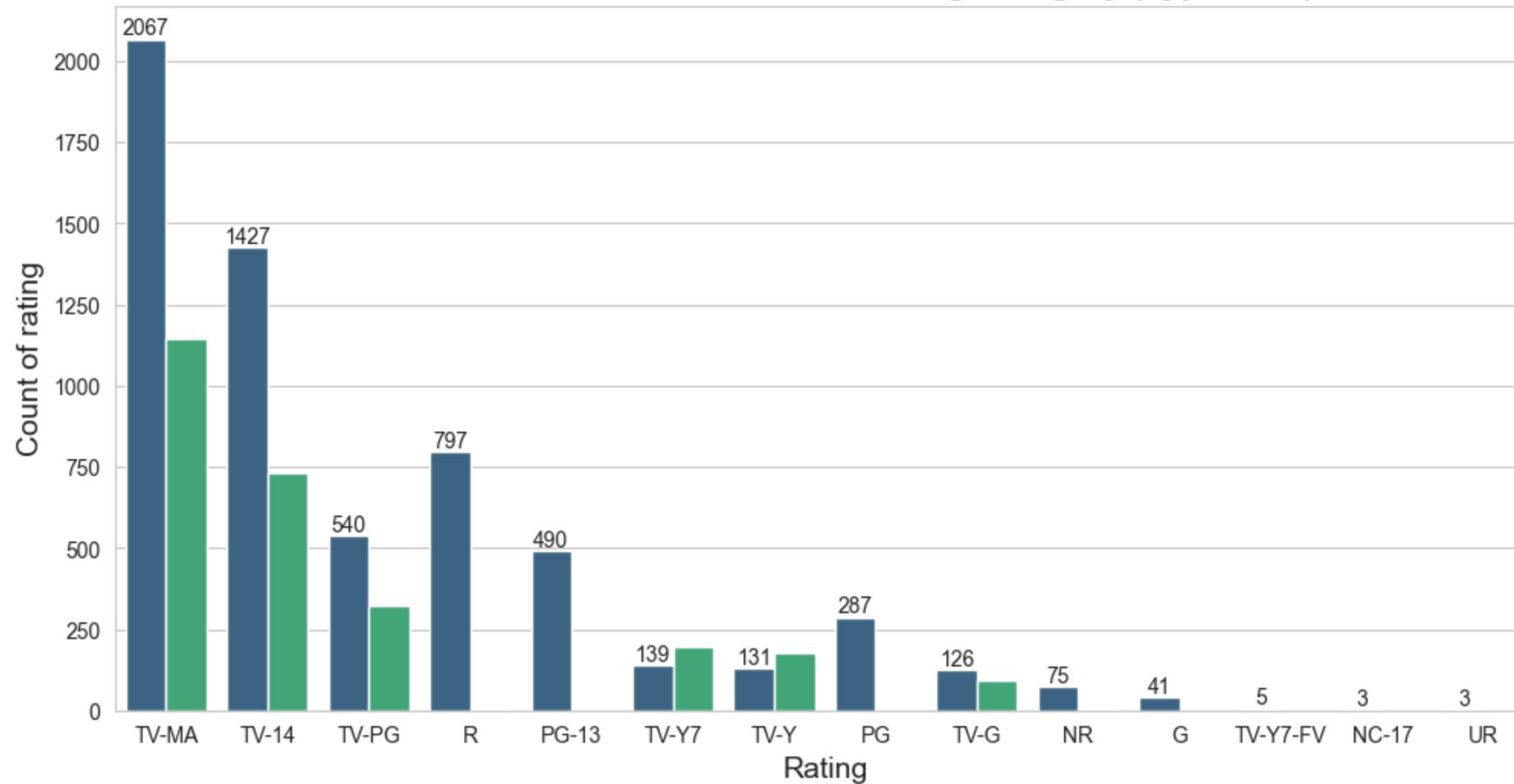
```
Out[332... rating
TV-MA      1147
TV-14       733
TV-PG       323
TV-Y7       195
TV-Y        176
TV-G        94
NR          5
R           2
TV-Y7-FV    1
Name: count, dtype: int64
```

```
In [333... print(f"Unique rating in Movies : ", data[data["type"] == "Movie"]["rating"].value_counts().shape[0])
print(f"Unique rating in TV Show : ", data[data["type"] == "TV Show"]["rating"].value_counts().shape[0])
```

```
Unique rating in Movies :  14
Unique rating in TV Show :  9
```

```
In [334... sns.set_style("whitegrid")
fig, ax = plt.subplots(figsize=(12, 6))
ax = sns.countplot(data=data, x="rating", hue = "type", order=data["rating"].value_counts().index, saturation=0.75, palette="viridis", legend=False)
ax.set_title("No. of Movies/TV Shows in each Rating Category (Type wise)", fontsize=16, fontweight='bold')
ax.set_xlabel("Rating", fontsize=14)
ax.set_ylabel("Count of rating", fontsize=14)
ax.bar_label(ax.containers[0], fontsize=10)
plt.show()
```

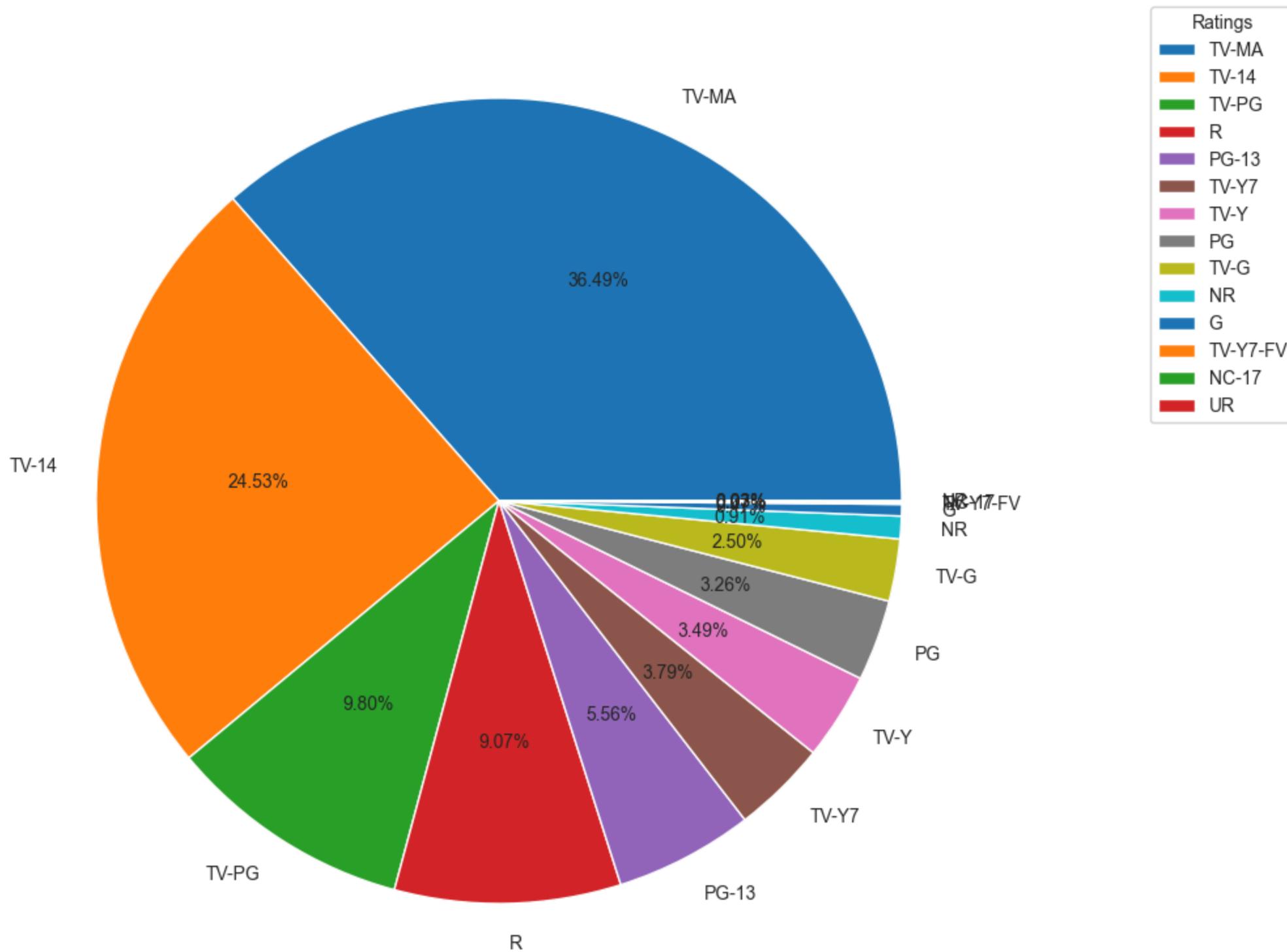
### No. of Movies/TV Shows in each Rating Category (Type wise)



```
In [335]: #Type wise the result is almost similar as well, but rating "TV-Y7, TV-Y" has more TV Shows than Movies
```

```
In [336]: sns.set_style("whitegrid")
fig, ax = plt.subplots(figsize=(10, 8))
ax.set_title("Distribution of Ratings", fontsize=16, fontweight='bold')
data["rating"].value_counts().plot.pie(autopct="%1.2f%%", figsize=(10,10))
ax.set_ylabel('')
ax.legend(
    labels=data["rating"].value_counts().index,
    loc="upper right",
    bbox_to_anchor=(1.3, 1),
    title="Ratings"
)
plt.show()
```

## Distribution of Ratings



```
In [337]: #36.49% of contribution are from the rating TV-MA, and 24.53 are from TV-14
```

```
In [338]: data.shape
```

```
Out[338]: (8807, 12)
```

```
In [339...]: data["duration"].value_counts()
```

```
Out[339]: duration
1 Season      1793
2 Seasons     425
3 Seasons     199
90 min        152
94 min        146
...
16 min         1
186 min        1
193 min        1
189 min        1
191 min        1
Name: count, Length: 220, dtype: int64
```

```
In [340...]: data["duration"].unique()
```

```
Out[340]: array(['90 min', '2 Seasons', '1 Season', '91 min', '125 min',
   '9 Seasons', '104 min', '127 min', '4 Seasons', '67 min', '94 min',
   '5 Seasons', '161 min', '61 min', '166 min', '147 min', '103 min',
   '97 min', '106 min', '111 min', '3 Seasons', '110 min', '105 min',
   '96 min', '124 min', '116 min', '98 min', '23 min', '115 min',
   '122 min', '99 min', '88 min', '100 min', '6 Seasons', '102 min',
   '93 min', '95 min', '85 min', '83 min', '113 min', '13 min',
   '182 min', '48 min', '145 min', '87 min', '92 min', '80 min',
   '117 min', '128 min', '119 min', '143 min', '114 min', '118 min',
   '108 min', '63 min', '121 min', '142 min', '154 min', '120 min',
   '82 min', '109 min', '101 min', '86 min', '229 min', '76 min',
   '89 min', '156 min', '112 min', '107 min', '129 min', '135 min',
   '136 min', '165 min', '150 min', '133 min', '70 min', '84 min',
   '140 min', '78 min', '7 Seasons', '64 min', '59 min', '139 min',
   '69 min', '148 min', '189 min', '141 min', '130 min', '138 min',
   '81 min', '132 min', '10 Seasons', '123 min', '65 min', '68 min',
   '66 min', '62 min', '74 min', '131 min', '39 min', '46 min',
   '38 min', '8 Seasons', '17 Seasons', '126 min', '155 min',
   '159 min', '137 min', '12 min', '273 min', '36 min', '34 min',
   '77 min', '60 min', '49 min', '58 min', '72 min', '204 min',
   '212 min', '25 min', '73 min', '29 min', '47 min', '32 min',
   '35 min', '71 min', '149 min', '33 min', '15 min', '54 min',
   '224 min', '162 min', '37 min', '75 min', '79 min', '55 min',
   '158 min', '164 min', '173 min', '181 min', '185 min', '21 min',
   '24 min', '51 min', '151 min', '42 min', '22 min', '134 min',
   '177 min', '13 Seasons', '52 min', '14 min', '53 min', '8 min',
   '57 min', '28 min', '50 min', '9 min', '26 min', '45 min',
   '171 min', '27 min', '44 min', '146 min', '20 min', '157 min',
   '17 min', '203 min', '41 min', '30 min', '194 min', '15 Seasons',
   '233 min', '237 min', '230 min', '195 min', '253 min', '152 min',
   '190 min', '160 min', '208 min', '180 min', '144 min', '5 min',
   '174 min', '170 min', '192 min', '209 min', '187 min', '172 min',
   '16 min', '186 min', '11 min', '193 min', '176 min', '56 min',
   '169 min', '40 min', '10 min', '3 min', '168 min', '312 min',
   '153 min', '214 min', '31 min', '163 min', '19 min', '12 Seasons',
   nan, '179 min', '11 Seasons', '43 min', '200 min', '196 min',
   '167 min', '178 min', '228 min', '18 min', '205 min', '201 min',
   '191 min'], dtype=object)
```

```
In [341]: data["duration"].isnull().sum()
```

```
Out[341]: 3
```

```
In [342]: #there are 3 Null values in duration
```

```
In [343]: data[data["duration"].isnull()]
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	
5541	s5542	Movie	Louis C.K.	2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	TV-MA	NaN	Movies	Louis C.K. muses on religion, eternal love, gi...
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	TV-MA	NaN	Movies	Emmy-winning comedy writer Louis C.K. brings h...	
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	TV-MA	NaN	Movies	The comic puts his trademark hilarious/thought...	

```
In [344...]: #there are 3 Null values in duration and those Movies, adding with the mode value
```

```
In [345...]: # Most frequent value in the duration  
mode_duration = data[data["type"] == "Movie"]["duration"].mode()[0]  
mode_duration
```

```
Out[345...]: '90 min'
```

```
In [346...]: # Filling the Null values with mode value  
data["duration"] = data["duration"].fillna(mode_duration)
```

```
In [347...]: data["duration"].isnull().sum()
```

```
Out[347...]: 0
```

```
In [348...]: #NULL values in duration are taken care here
```

```
In [349...]: # Count of unique values in the duration column  
data["duration"].value_counts().head()
```

```
Out[349...]: duration  
1 Season      1793  
2 Seasons     425  
3 Seasons     199  
90 min        155  
94 min        146  
Name: count, dtype: int64
```

```
In [352...]: data[["type", "duration"]].value_counts().head(6)
```

```
Out[352...]: type    duration  
TV Show 1 Season      1793  
          2 Seasons     425  
          3 Seasons     199  
Movie   90 min        155  
          93 min        146  
          94 min        146  
Name: count, dtype: int64
```

```
In [353...]: #in TV Shows duration with "1 Season" has more shows  
#in Movie duration with "90 mints" has more movies
```

```
In [378...]: tvshow_data = data[data["type"] == "TV Show"]  
tvshow_data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
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...
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	September 24, 2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...

In [379... tvshow\_data.shape

Out[379... (2676, 12)

In [390... movie\_data = data[data["type"] == "Movie"]  
movie\_data.head()

	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...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	2021	PG-13	104 min	Comedies, Dramas	A woman adjusting to life after a loss contend...
12	s13	Movie	Je Suis Karl	Christian Schwochow	Luna Wedler, Jannis Niewöhner, Milan Peschel, ...	Germany, Czech Republic	September 23, 2021	2021	TV-MA	127 min	Dramas, International Movies	After most of her family is murdered in a terr...

In [391... movie\_data.shape

Out[391... (6131, 12)

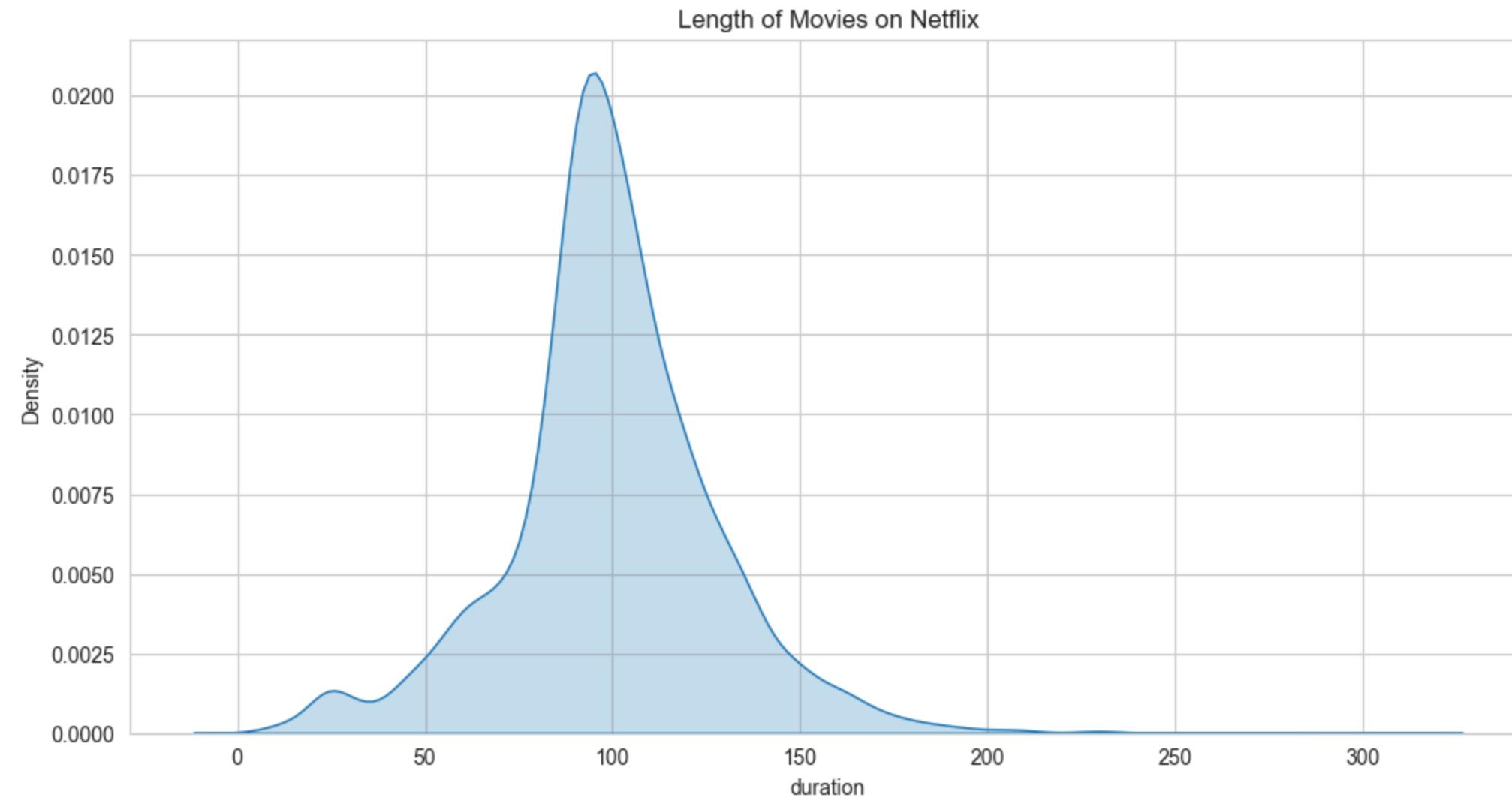
In [392... data.shape[0]

Out[392... 8807

In [393... tvshow\_data.shape[0] + movie\_data.shape[0]

Out[393... 8807

In [394...]	#Splitting duration column with spaces and taking first value in only in column											
In [395...]	data.head()											
Out[395...]												
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...
In [396...]	data.loc[:, 'duration'] = data["duration"].str.split(" ").str[0].astype(int)											
In [397...]	data.head()											
Out[397...]												
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	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	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	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	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	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...
In [399...]	#Length of Movies on Netflix											
In [402...]	fig, ax = plt.subplots(figsize=(12, 6)) ax = sns.kdeplot(data[data["type"] == "Movie"]["duration"], fill=True) ax.set_title("Length of Movies on Netflix") plt.show()											



```
In [411]: data[data["type"] == "Movie"]["duration"].value_counts()
```

```
Out[411]: duration
90      155
94      146
93      146
97      146
91      144
...
212      1
8       1
186      1
193      1
191      1
Name: count, Length: 205, dtype: int64
```

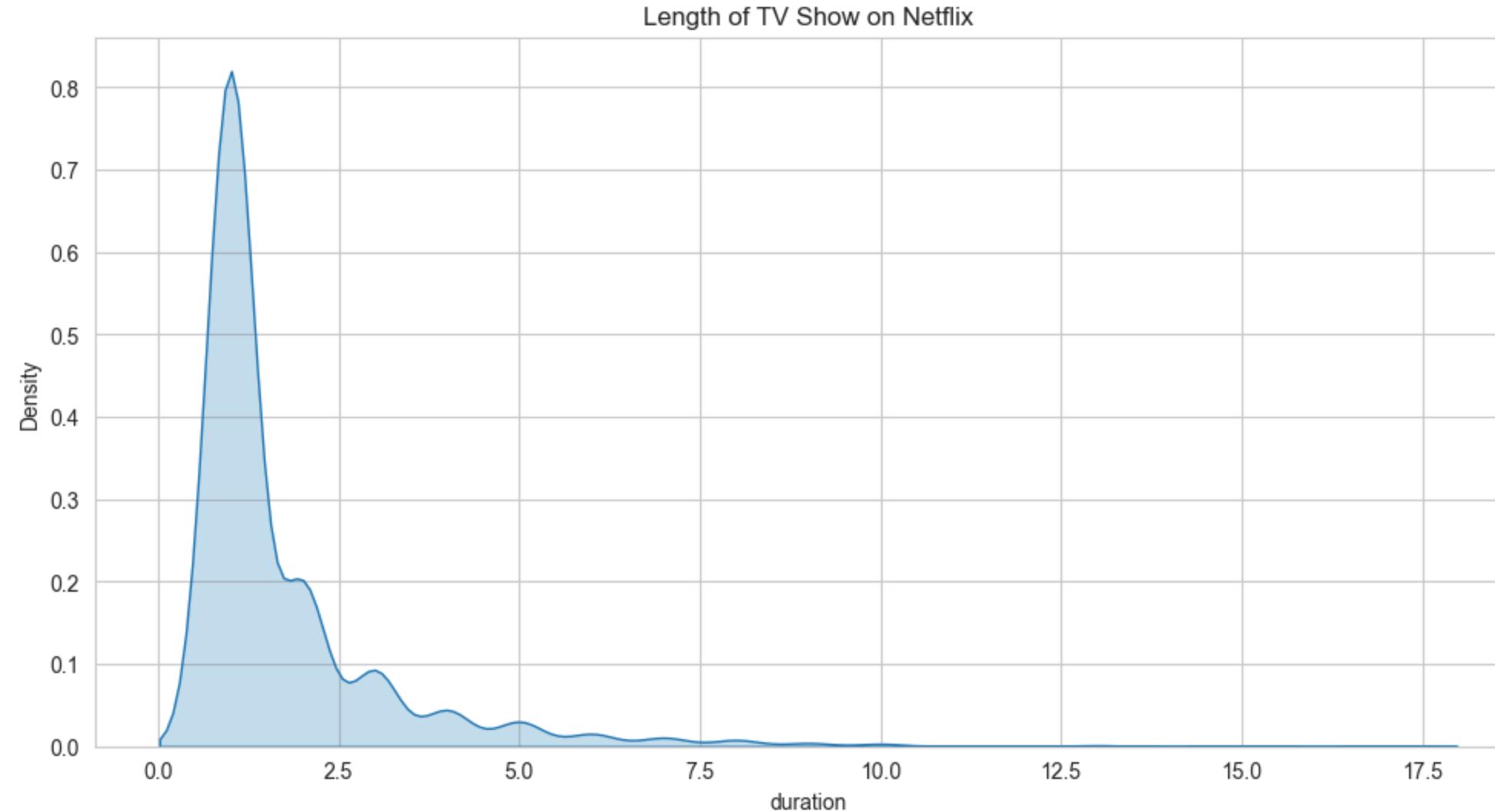
```
In [418]: #Most of the movies are in range of 90-110 minutes
```

```
In [419]: print("Minimum Length of Movie in minutes: " + str(data[data["type"] == "Movie"]["duration"].min()))
print("Maximum Length of Movie in minutes: " + str(data[data["type"] == "Movie"]["duration"].max()))
print("Mean Length of Movie in minutes: " + str(data[data["type"] == "Movie"]["duration"].mean()))
```

```
Minimum Length of Movie in minutes: 3
Maximum Length of Movie in minutes: 312
Mean Length of Movie in minutes: 99.57250040776383
```

```
In [420]: #Length of TV Show on Netflix
```

```
In [421]: fig, ax = plt.subplots(figsize=(12, 6))
ax = sns.kdeplot(data[data["type"] == "TV Show"]["duration"], fill=True)
ax.set_title("Length of TV Show on Netflix")
plt.show()
```



```
In [422]: data[data["type"] == "TV Show"]["duration"].value_counts()
```

```
Out[422...]: duration
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: count, dtype: int64
```

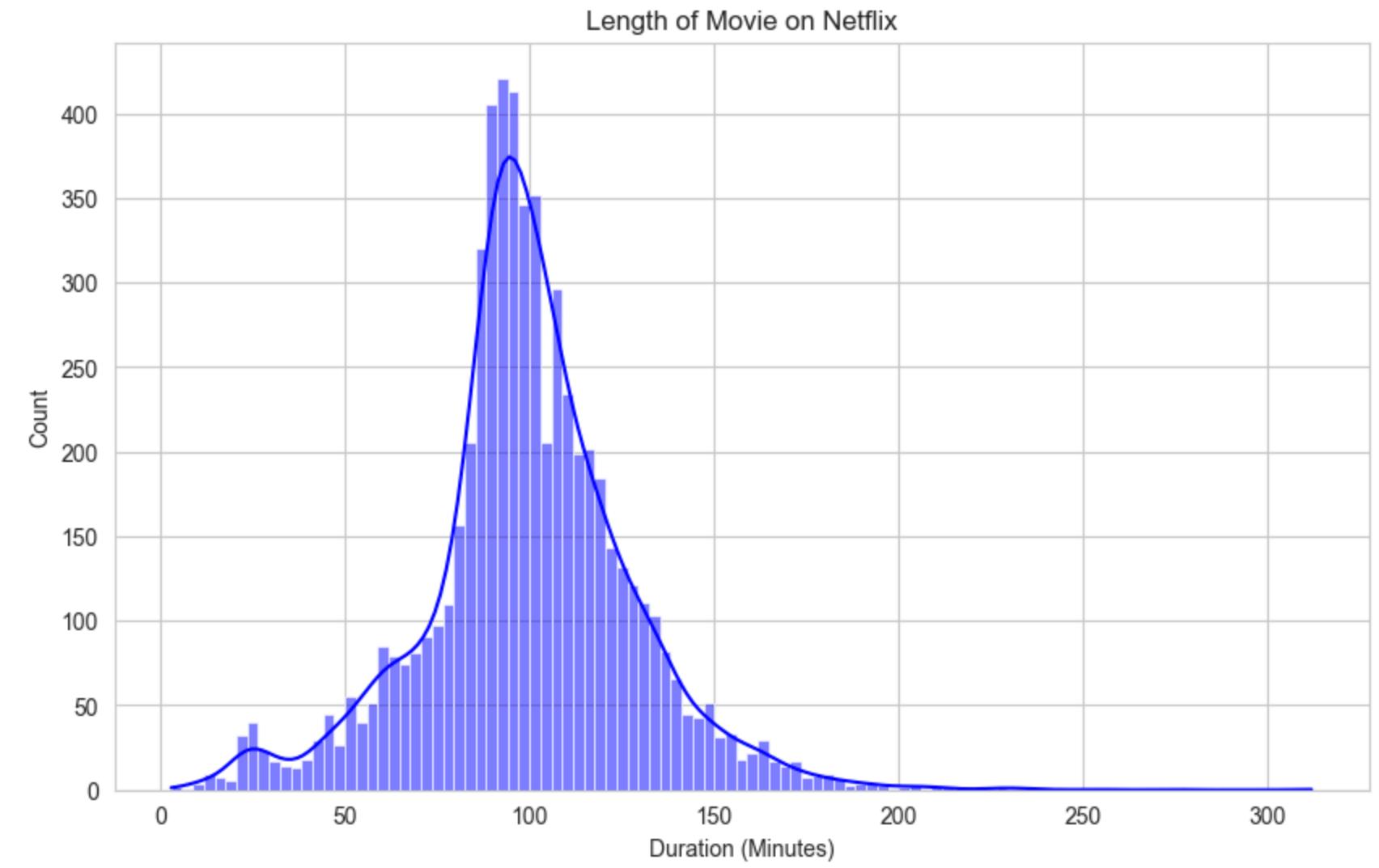
```
In [428...]: #Most of the TV Shows are in range of 1-2 Seasons
```

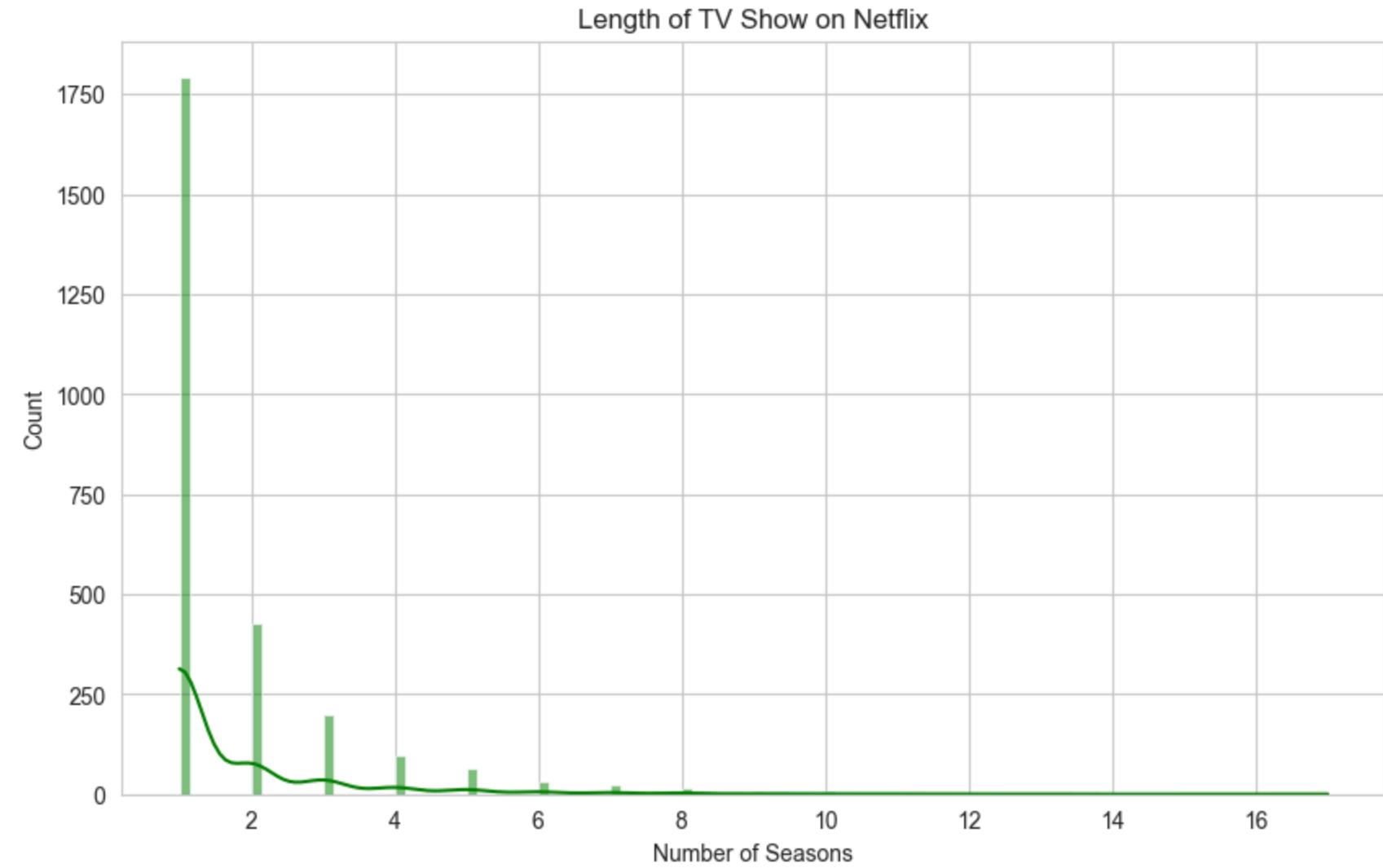
```
In [429...]: print("Minimum Length of TV Shows in Seasons: " + str(data[data["type"] == "TV Show"]["duration"].min()))
print("Maximum Length of TV Shows in Seasons: " + str(data[data["type"] == "TV Show"]["duration"].max()))
print("Mean Length of TV Shows in Seasons: " + str(data[data["type"] == "TV Show"]["duration"].mean()))
```

```
Minimum Length of TV Shows in Seasons: 1
Maximum Length of TV Shows in Seasons: 17
Mean Length of TV Shows in Seasons: 1.764947683109118
```

```
In [430...]: plt.figure(figsize=(10, 6))
sns.histplot(data[data["type"] == "Movie"]['duration'], kde=True, color='blue')
plt.title('Length of Movie on Netflix')
plt.xlabel('Duration (Minutes)')
plt.ylabel('Count')
plt.show()

plt.figure(figsize=(10, 6))
sns.histplot(data[data["type"] == "TV Show"]['duration'], kde=True, color='green')
plt.title('Length of TV Show on Netflix')
plt.xlabel('Number of Seasons')
plt.ylabel('Count')
plt.show()
```



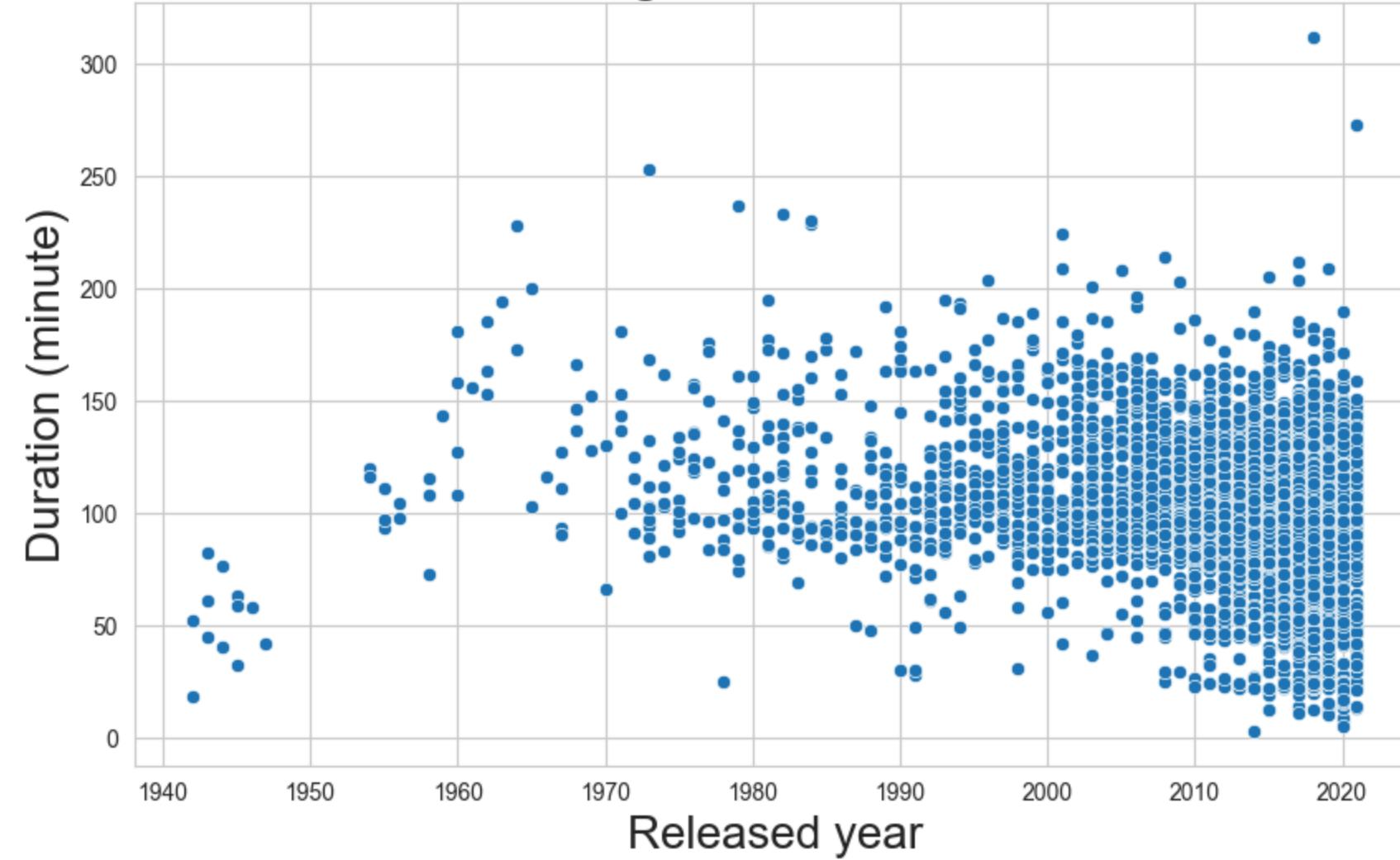


```
In [438]: tvshow_data = data[data["type"] == "TV Show"]
```

```
In [439]: movie_data = data[data["type"] == "Movie"]
```

```
In [440]: #Scatterplot of the "release_year" vs. "duration" columns using plotly.express
plt.figure(figsize = (10,6))
sns.scatterplot(movie_data, x = "release_year", y = "duration")
plt.title("Movie Length Vs. Release Year ", fontsize = 25)
plt.xlabel("Released year", fontsize = 20)
plt.ylabel("Duration (minute)", fontsize = 20)
plt.show()
```

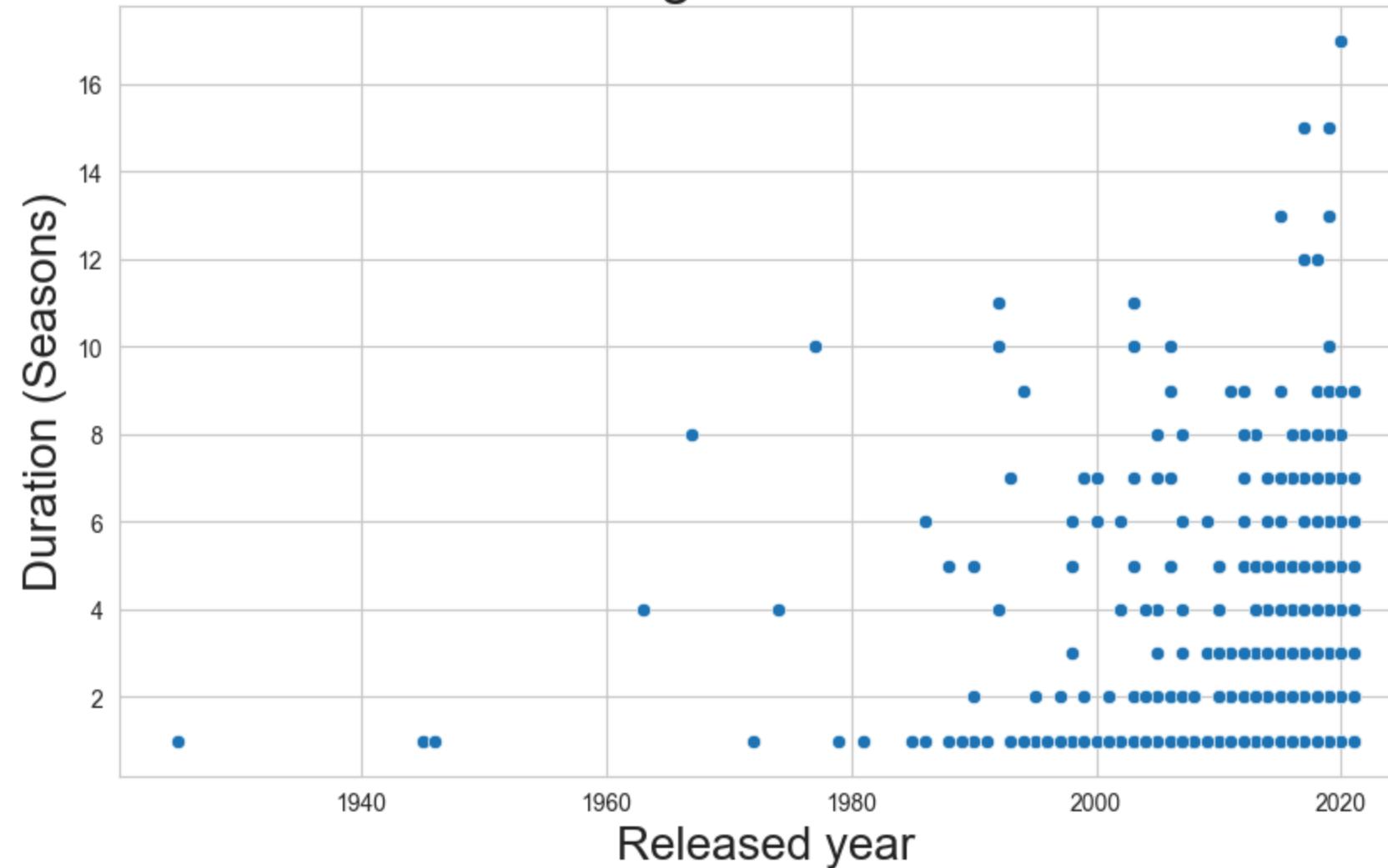
## Movie Length Vs. Release Year



In [441]:

```
#Scatterplot of the "release_year" vs. "duration" columns using plotly.express
plt.figure(figsize = (10,6))
sns.scatterplot(tvshow_data, x = "release_year", y = "duration")
plt.title("TV Show Length Vs. Release Year ", fontsize = 25)
plt.xlabel("Released year", fontsize = 20)
plt.ylabel("Duration (Seasons)", fontsize = 20)
plt.show()
```

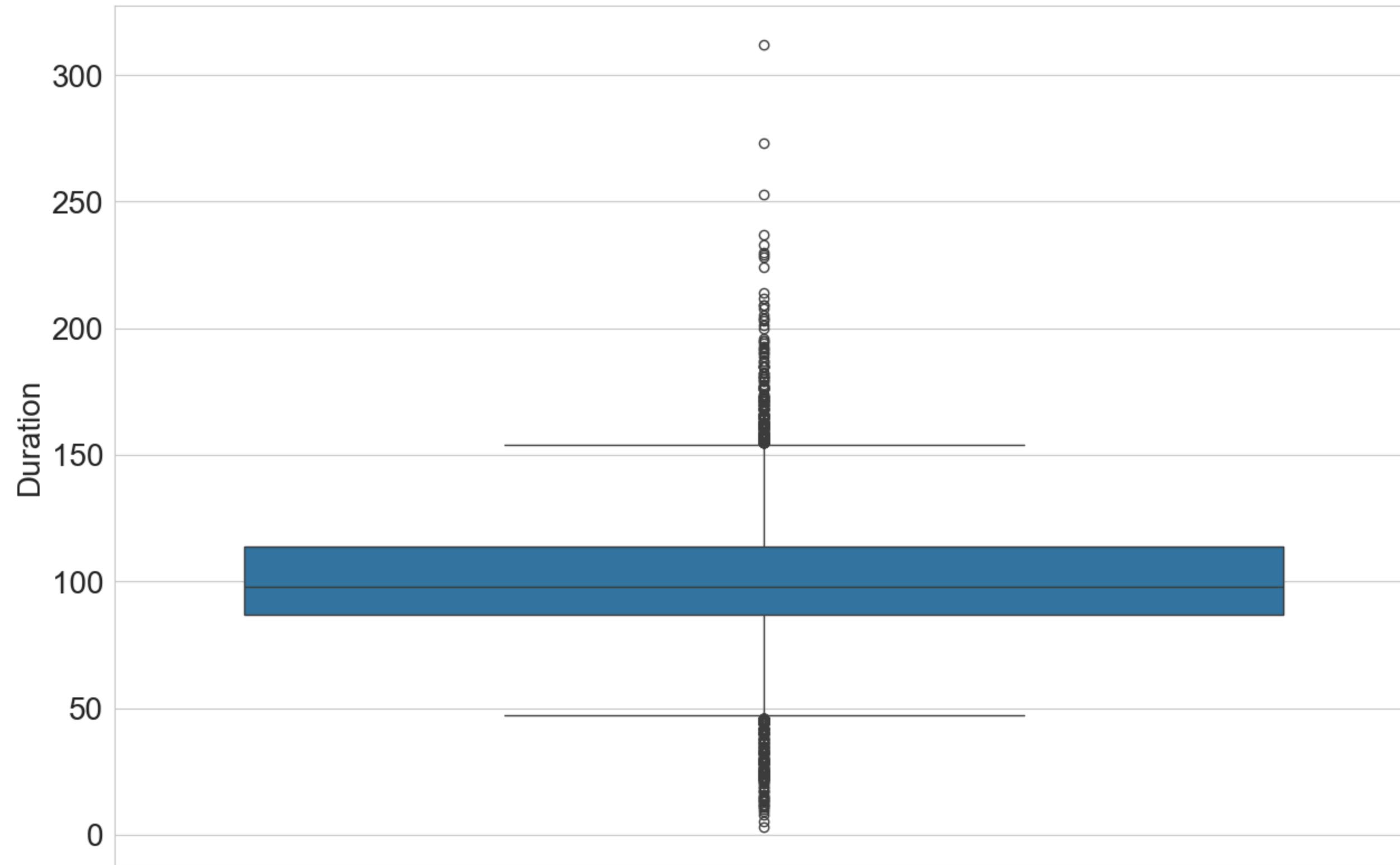
## TV Show Length Vs. Release Year



```
In [442]: plt.figure(figsize=(15,10))
sns.boxplot(y = movie_data["duration"])
plt.yticks(fontsize=20)
plt.ylabel('Duration', fontsize=20)
plt.title('Duration of Movies', fontsize=20)
```

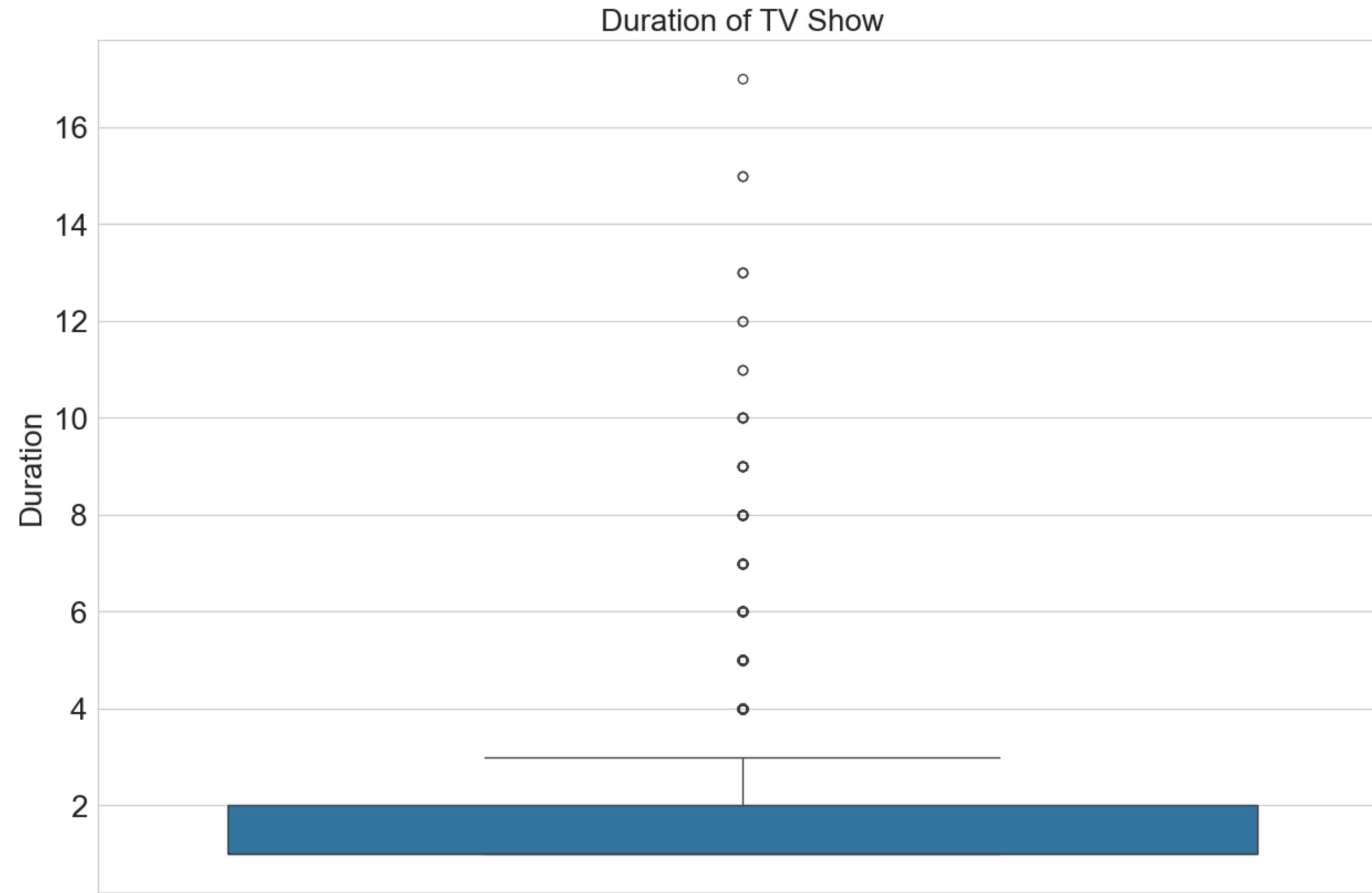
```
Out[442]: Text(0.5, 1.0, 'Duration of Movies')
```

## Duration of Movies



```
In [443]:  
plt.figure(figsize=(15,10))  
sns.boxplot(y = tvshow_data["duration"])  
plt.yticks(fontsize=20)  
plt.ylabel('Duration', fontsize=20)  
plt.title('Duration of TV Show', fontsize=20)
```

```
Out[443...]: Text(0.5, 1.0, 'Duration of TV Show')
```



```
In [444...]: data.head()
```

Out[444...]

	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	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	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	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	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	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...

In [445...]

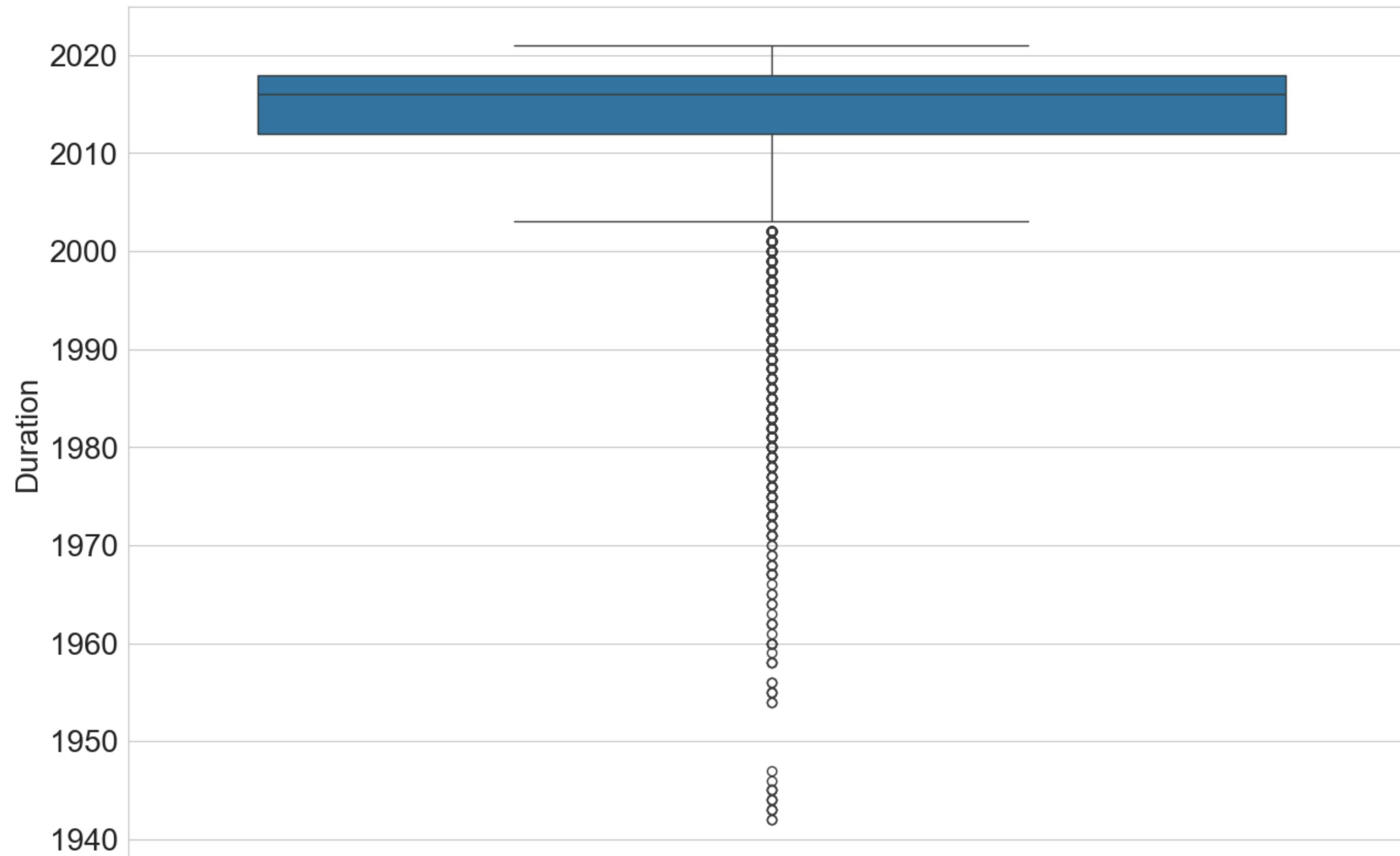
```
plt.figure(figsize=(15,10))
sns.boxplot(y = movie_data["release_year"])
plt.xticks(fontsize=20)
plt.ylabel('Duration', fontsize=20)
plt.title('Duration of Movies', fontsize=20)

plt.figure(figsize=(15,10))
sns.boxplot(y = tvshow_data["release_year"])
plt.xticks(fontsize=20)
plt.ylabel('Duration', fontsize=20)
plt.title('Duration of TV Show', fontsize=20)
```

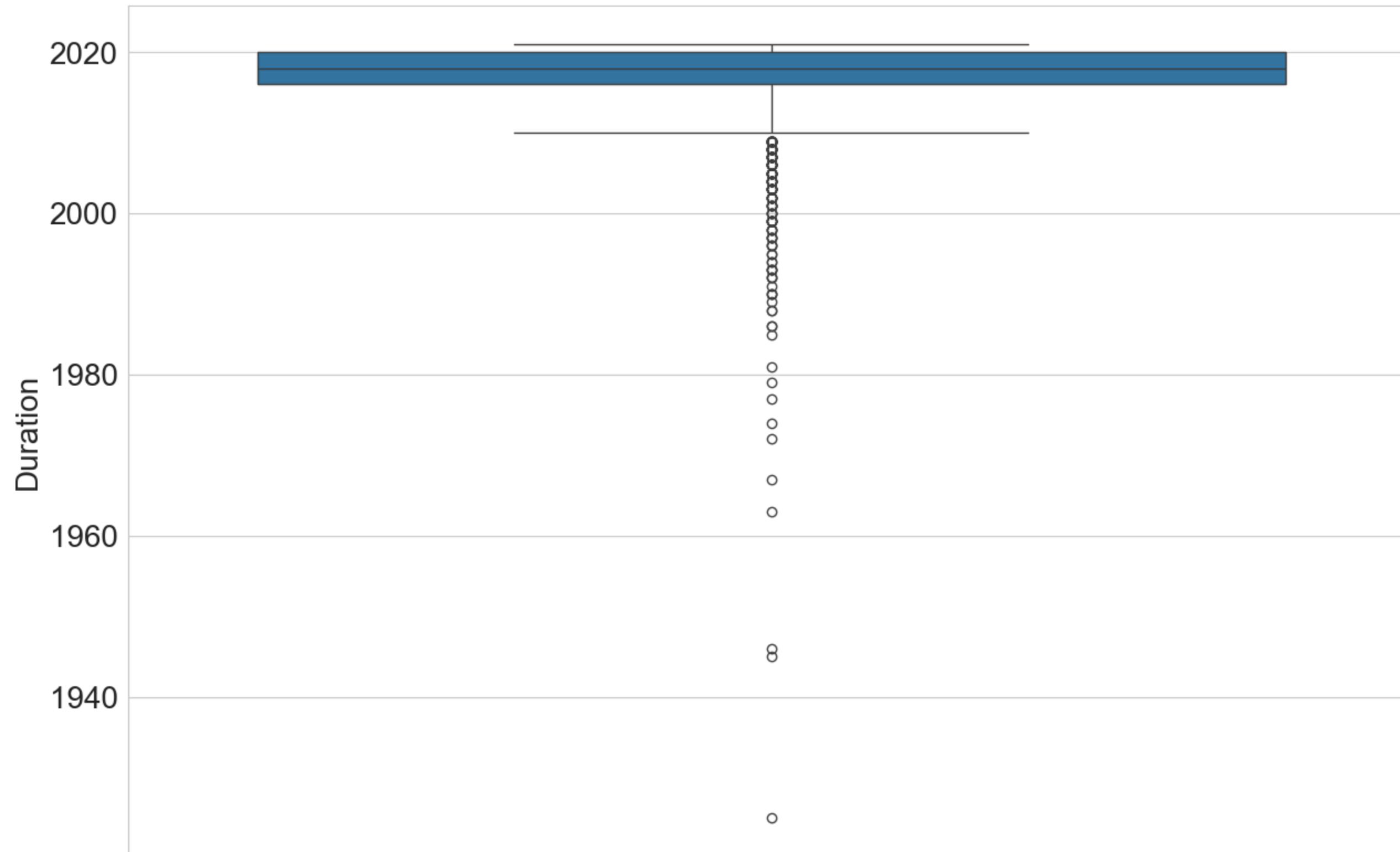
Out[445...]

Text(0.5, 1.0, 'Duration of TV Show')

### Duration of Movies



## Duration of TV Show



```
In [450]: movie_data["release_year"].value_counts()
```

```
Out[450...]: release_year  
2017    767  
2018    767  
2016    658  
2019    633  
2020    517  
...  
1966     1  
1961     1  
1946     1  
1963     1  
1947     1  
Name: count, Length: 73, dtype: int64
```

```
In [451...]: tvshow_data["release_year"].value_counts()
```

```
Out[451...]: release_year
2020    436
2019    397
2018    380
2021    315
2017    265
2016    244
2015    162
2014     88
2012     64
2013     63
2010     40
2011     40
2009     34
2008     23
2006     14
2007     14
2005     13
2003     10
2004      9
1999      7
2002      7
2001      5
1993      4
2000      4
1997      4
1998      4
1990      3
1996      3
1992      3
1995      2
1994      2
1988      2
1986      2
1989      1
1967      1
1985      1
1946      1
1981      1
1972      1
1979      1
1977      1
1991      1
1974      1
1925      1
1945      1
1963      1
Name: count, dtype: int64
```

```
In [452...]: #Most of the movies released in 2017 and 2018
```

```
In [453...]: #Most of the tv shows released in 2020
```

```
In [458...]: #Top 5 oldest Movies released
movie_data.sort_values('release_year', ascending=True)[['title', 'date_added', 'release_year']].head()
```

Out[458...]

	title	date_added	release_year
7790	Prelude to War	March 31, 2017	1942
8205	The Battle of Midway	March 31, 2017	1942
8660	Undercover: How to Operate Behind Enemy Lines	March 31, 2017	1943
8763	WWII: Report from the Aleutians	March 31, 2017	1943
8739	Why We Fight: The Battle of Russia	March 31, 2017	1943

In [459...]

```
#Top 5 Latest Movies released  
movie_data.sort_values('release_year', ascending=False)[['title', 'date_added', 'release_year']].head()
```

Out[459...]

	title	date_added	release_year
570	Dynasty Warriors	July 1, 2021	2021
770	Myriam Fares: The Journey	June 3, 2021	2021
766	Alan Saldaña: Locked Up	June 3, 2021	2021
765	Xtreme	June 4, 2021	2021
764	Trippin' with the Kandasamys	June 4, 2021	2021

In [460...]

```
#Top 5 oldest TV Shows released  
tvshow_data.sort_values('release_year', ascending=True)[['title', 'date_added', 'release_year']].head()
```

Out[460...]

	title	date_added	release_year
4250	Pioneers: First Women Filmmakers*	December 30, 2018	1925
1331	Five Came Back: The Reference Films	February 9, 2021	1945
7743	Pioneers of African-American Cinema	February 1, 2017	1946
8541	The Twilight Zone (Original Series)	July 1, 2017	1963
8189	The Andy Griffith Show	July 1, 2017	1967

In [461...]

```
#Top 5 latest TV Shows released  
tvshow_data.sort_values('release_year', ascending=False)[['title', 'date_added', 'release_year']].head()
```

Out[461...]

		title	date_added	release_year
1	Blood & Water	September 24, 2021	2021	
835	Eden	May 27, 2021	2021	
876	Alma Matters	May 14, 2021	2021	
874	Booba: Food Puzzle	May 15, 2021	2021	
862	Special	May 20, 2021	2021	

In [463...]

```
#Top 5 Oldest Movies/TV Shows Available on Netflix, data_added
data['year_added'] = pd.DatetimeIndex(data['date_added']).year
data.sort_values('year_added', ascending=True)[['title', 'type', 'date_added', 'release_year']].head()
```

Out[463...]

	title	type	date_added	release_year
5957	To and From New York	Movie	January 1, 2008	2006
6611	Dinner for Five	TV Show	February 4, 2008	2007
5956	Just Another Love Story	Movie	May 5, 2009	2007
5955	Splatter	Movie	November 18, 2009	2009
7370	Mad Ron's Previews from Hell	Movie	November 1, 2010	1987

In [464...]

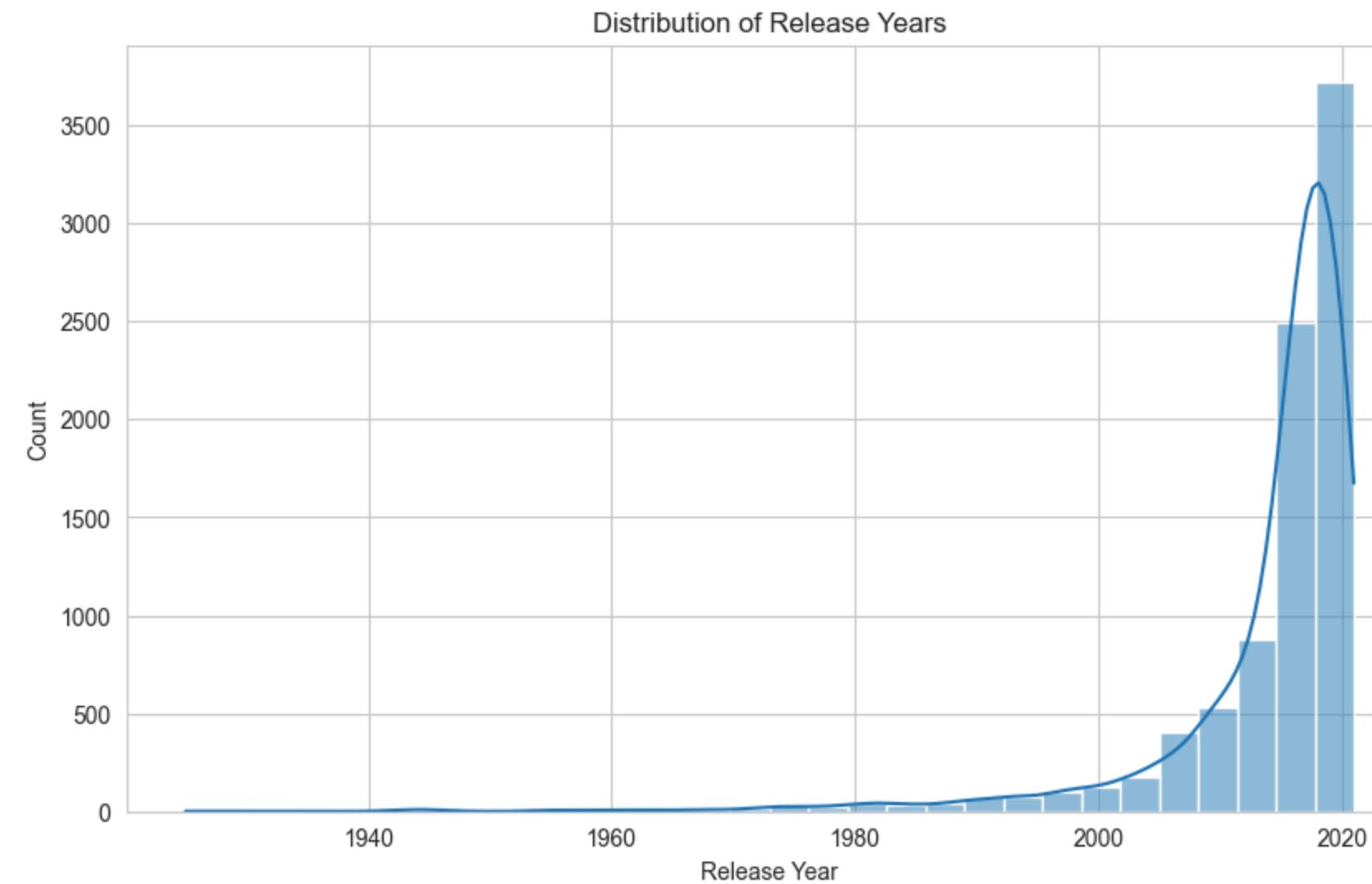
```
#Top 5 Latest Movies/TV Shows Available on Netflix, data_added
data['year_added'] = pd.DatetimeIndex(data['date_added']).year
data.sort_values('year_added', ascending=False)[['title', 'type', 'date_added', 'release_year']].head()
```

Out[464...]

	title	type	date_added	release_year
0	Dick Johnson Is Dead	Movie	September 25, 2021	2020
996	HOMUNCULUS	Movie	April 22, 2021	2021
1005	Keymon and Nani in Space Adventure	Movie	April 20, 2021	2013
1004	Izzy's Koala World	TV Show	April 20, 2021	2021
1003	Zero	TV Show	April 21, 2021	2021

In [465...]

```
#Distribution of release years
plt.figure(figsize=(10, 6))
sns.histplot(data['release_year'], bins=30, kde=True)
plt.title('Distribution of Release Years')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.show()
```

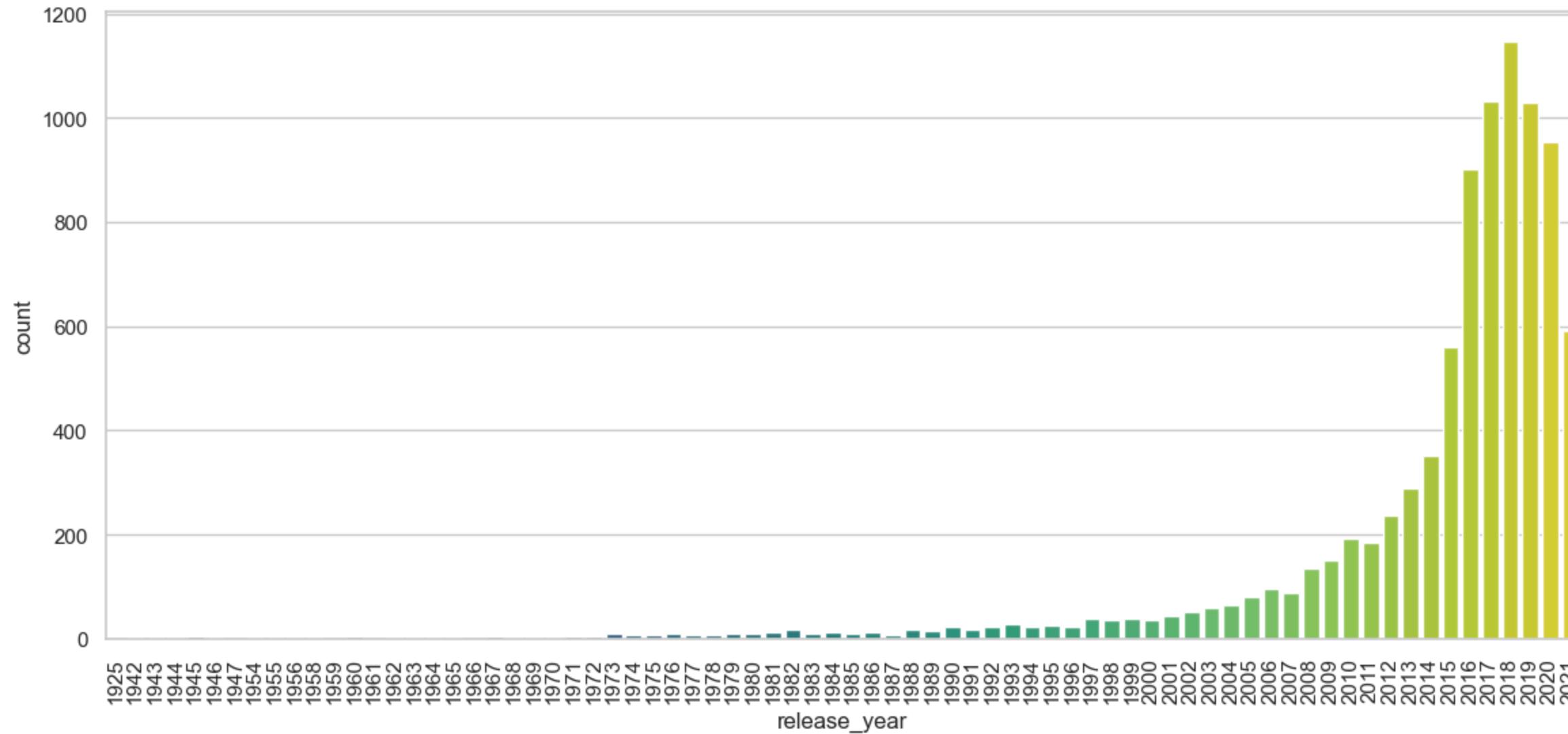


```
In [466...]: #intersection table between release_year and type
pd.crosstab(data.type, data.release_year)
```

	type																						
Movie	0	2	3	3	3	1	1	2	3	2	...	173	225	264	398	658	767	767	633	517	277		
TV Show	1	0	0	0	1	1	0	0	0	0	...	64	63	88	162	244	265	265	380	397	436	315	

2 rows × 74 columns

```
In [472...]: #Distribution of years in plot
data_temp=data['release_year'].sort_values()
sns.set(style="whitegrid")
plt.figure(figsize=(14, 6))
sns.countplot(x=data_temp, palette='viridis')
plt.xticks(rotation=90)
plt.show()
```



```
In [473]: type_and_date = data[['type', 'date_added']].copy()
```

```
type_and_date['date_added'] = pd.to_datetime(type_and_date['date_added'], format='mixed')  
type_and_date['year_added'] = pd.DatetimeIndex(type_and_date['date_added']).year  
type_and_date
```

```
Out[473...]
```

	type	date_added	year_added
0	Movie	2021-09-25	2021.0
1	TV Show	2021-09-24	2021.0
2	TV Show	2021-09-24	2021.0
3	TV Show	2021-09-24	2021.0
4	TV Show	2021-09-24	2021.0
...	...	...	...
8802	Movie	2019-11-20	2019.0
8803	TV Show	2019-07-01	2019.0
8804	Movie	2019-11-01	2019.0
8805	Movie	2020-01-11	2020.0
8806	Movie	2019-03-02	2019.0

8807 rows × 3 columns

```
In [474...]
```

```
type_and_date.isnull().sum()
```

```
Out[474...]
```

```
type      0
date_added 10
year_added 10
dtype: int64
```

```
In [475...]
```

```
#Dropping the null values as its very less in count
type_and_date.dropna(inplace=True)
type_and_date['year_added'].unique()
```

```
Out[475...]
```

```
array([2021., 2020., 2019., 2018., 2017., 2016., 2015., 2014., 2013.,
       2012., 2011., 2009., 2008., 2010.])
```

```
In [476...]
```

```
count = type_and_date.groupby(['year_added', 'type']).count().rename(columns={'date_added': 'count'})
count.reset_index(inplace=True)
count
```

Out[476...]

	year_added	type	count
0	2008.0	Movie	1
1	2008.0	TV Show	1
2	2009.0	Movie	2
3	2009.0	TV Show	0
4	2010.0	Movie	1
5	2010.0	TV Show	0
6	2011.0	Movie	13
7	2011.0	TV Show	0
8	2012.0	Movie	3
9	2012.0	TV Show	0
10	2013.0	Movie	6
11	2013.0	TV Show	5
12	2014.0	Movie	19
13	2014.0	TV Show	5
14	2015.0	Movie	56
15	2015.0	TV Show	26
16	2016.0	Movie	253
17	2016.0	TV Show	176
18	2017.0	Movie	839
19	2017.0	TV Show	349
20	2018.0	Movie	1237
21	2018.0	TV Show	412
22	2019.0	Movie	1424
23	2019.0	TV Show	592
24	2020.0	Movie	1284
25	2020.0	TV Show	595
26	2021.0	Movie	993
27	2021.0	TV Show	505

In [477...]

```
count_tvshow = count[count['type']=='TV Show']
count_movie = count[count['type']=='Movie']
```

In [478...]

count\_tvshow

Out[478...]

	year_added	type	count
1	2008.0	TV Show	1
3	2009.0	TV Show	0
5	2010.0	TV Show	0
7	2011.0	TV Show	0
9	2012.0	TV Show	0
11	2013.0	TV Show	5
13	2014.0	TV Show	5
15	2015.0	TV Show	26
17	2016.0	TV Show	176
19	2017.0	TV Show	349
21	2018.0	TV Show	412
23	2019.0	TV Show	592
25	2020.0	TV Show	595
27	2021.0	TV Show	505

In [479...]

count\_movie

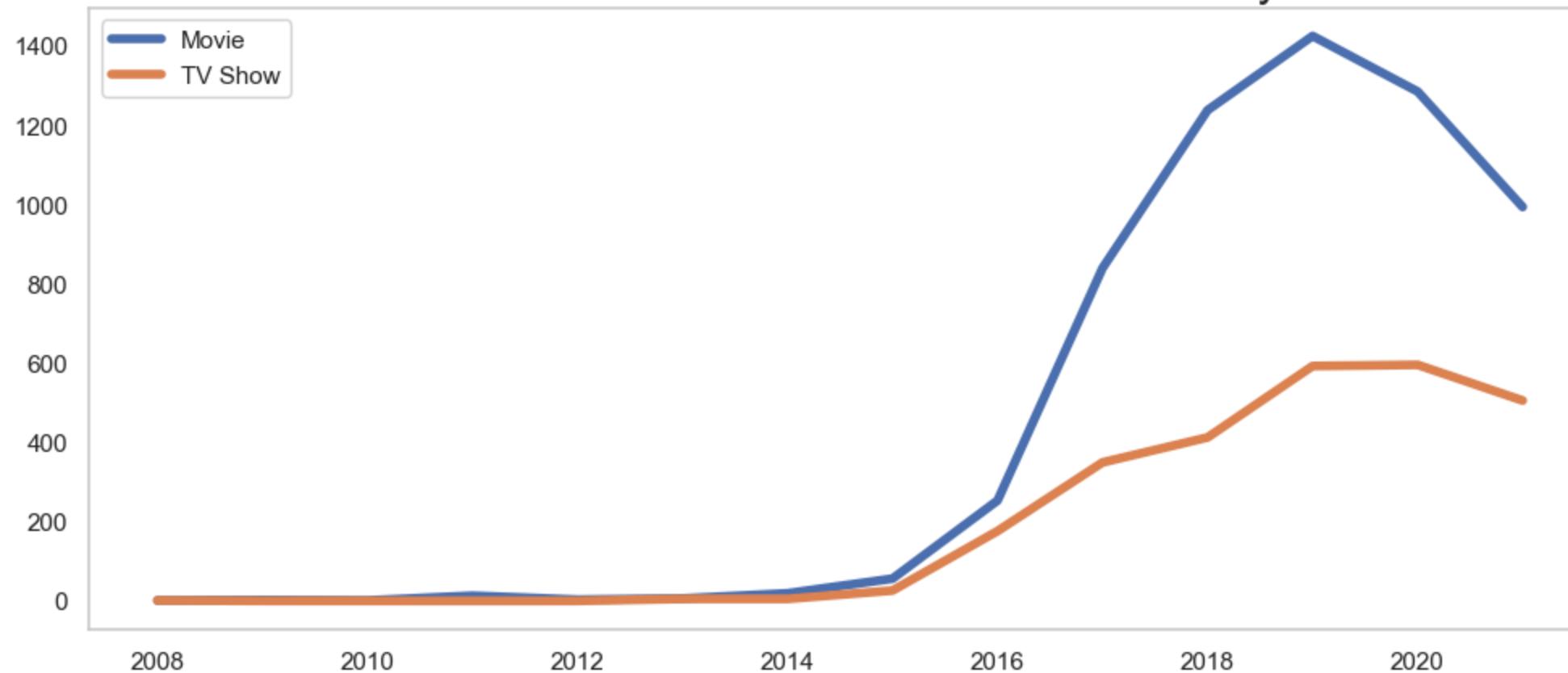
Out[479...]

	year_added	type	count
0	2008.0	Movie	1
2	2009.0	Movie	2
4	2010.0	Movie	1
6	2011.0	Movie	13
8	2012.0	Movie	3
10	2013.0	Movie	6
12	2014.0	Movie	19
14	2015.0	Movie	56
16	2016.0	Movie	253
18	2017.0	Movie	839
20	2018.0	Movie	1237
22	2019.0	Movie	1424
24	2020.0	Movie	1284
26	2021.0	Movie	993

In [480...]

```
plt.figure(figsize=(12,5))
plt.plot(count_movie['year_added'], count_movie['count'], lw=4, label='Movie')
plt.plot(count_tvshow['year_added'], count_tvshow['count'], lw=4, label='TV Show')
plt.legend()
plt.grid()
plt.title('Count of Movie and TV Show Based On The Year They Are Added', fontsize=20)
plt.show()
```

## Count of Movie and TV Show Based On The Year They Are Added



```
In [481]: cumulative_count_tvshow = count_tvshow.groupby('type').cumsum()  
cumulative_count_tvshow
```

Out[481...]

	year_added	count
1	2008.0	1
3	4017.0	1
5	6027.0	1
7	8038.0	1
9	10050.0	1
11	12063.0	6
13	14077.0	11
15	16092.0	37
17	18108.0	213
19	20125.0	562
21	22143.0	974
23	24162.0	1566
25	26182.0	2161
27	28203.0	2666

In [482...]

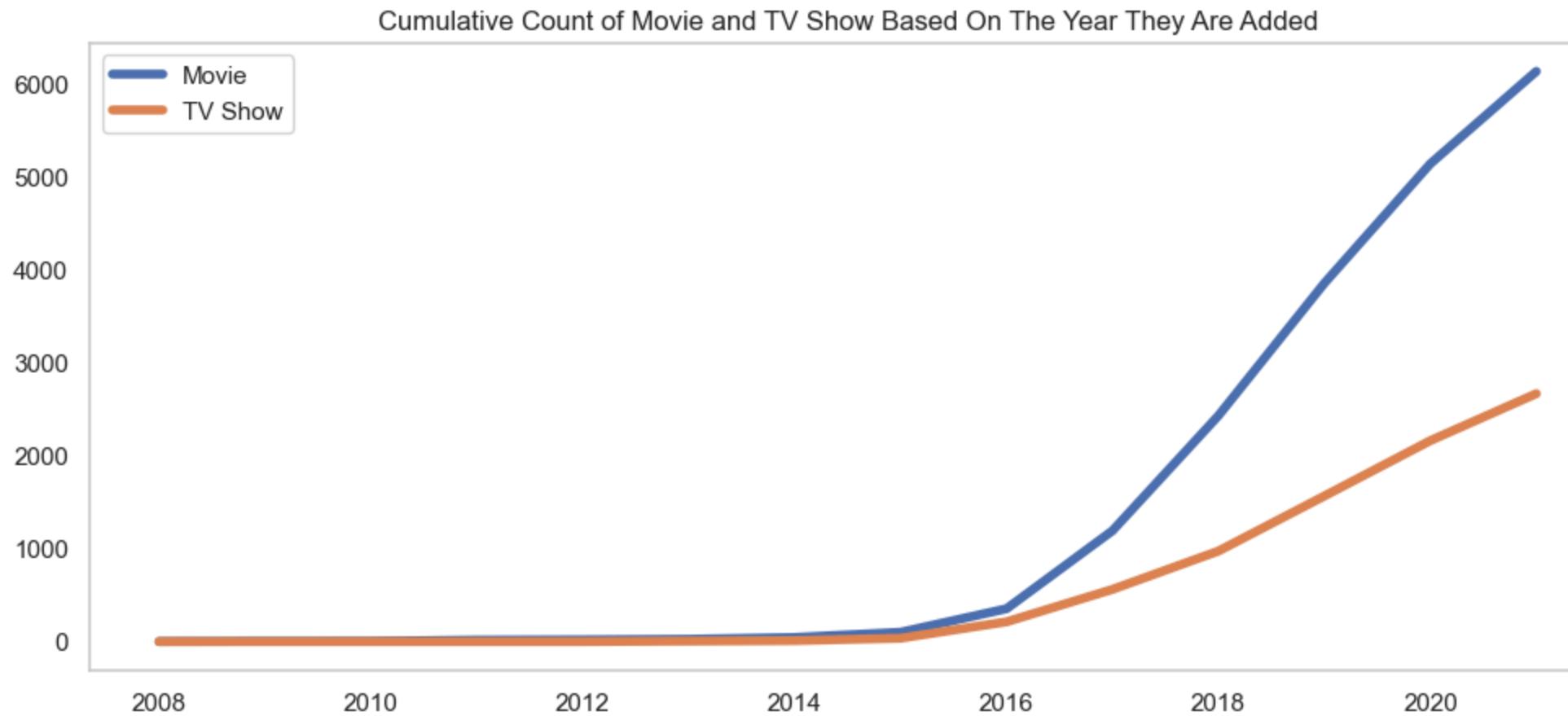
```
cumulative_count_movie = count_movie.groupby('type').cumsum()  
cumulative_count_movie
```

Out[482...]

	year_added	count
0	2008.0	1
2	4017.0	3
4	6027.0	4
6	8038.0	17
8	10050.0	20
10	12063.0	26
12	14077.0	45
14	16092.0	101
16	18108.0	354
18	20125.0	1193
20	22143.0	2430
22	24162.0	3854
24	26182.0	5138
26	28203.0	6131

In [483...]

```
plt.figure(figsize=(12,5))
plt.plot(count_movie['year_added'], cumulative_count_movie['count'], lw=4, label='Movie')
plt.plot(count_tvshow['year_added'], cumulative_count_tvshow['count'], lw=4, label='TV Show')
plt.legend()
plt.grid()
plt.title('Cumulative Count of Movie and TV Show Based On The Year They Are Added')
plt.show()
```



```
In [486]: data['country'].unique()
```

```
Out[486]: ['United States', 'South Africa', NaN, 'India', 'United States', 'Ghana', 'Burkina Faso', 'United Kingdom', ..., 'Russia', 'Spain', 'Croatia', 'Slovenia', 'Serbia', 'Montenegro', 'Japan', 'Canada', 'United States', 'France', 'South Korea', 'Indonesia', 'United Arab Emirates', 'Jordan']
Length: 749
Categories (748, object): [', France', 'Algeria', ', South Korea', 'Argentina', 'Argentina', 'Brazil', 'France', 'Poland', 'Germany', 'D...', ..., 'Venezuela', 'Colombia', 'Vietnam', 'West Germany', 'Zimbabwe']
```

```
In [488]: data["country"].value_counts()
```

```
Out[488]:
country
United States          2818
India                  972
United Kingdom         419
Japan                 245
South Korea            199
...
Ireland, Canada, Luxembourg, United States, United Kingdom, Philippines, India      1
Ireland, Canada, United Kingdom, United States                                     1
Ireland, Canada, United States, United Kingdom                                    1
Ireland, France, Iceland, United States, Mexico, Belgium, United Kingdom, Hong Kong    1
Zimbabwe                                         1
Name: count, Length: 748, dtype: int64
```

```
In [545]: data_country = data.copy()
data_country["country"] = data_country["country"].str.split(",")
data_country = data_country.explode("country")
data_country["country"].value_counts().reset_index()
```

```
Out[545...]
```

	country	count
0	United States	3211
1	India	1008
2	United Kingdom	628
3	United States	479
4	Canada	271
...	...	...
192	Ecuador	1
193	Iran	1
194	Cyprus	1
195	Mongolia	1
196	Montenegro	1

197 rows × 2 columns

```
In [546...]
```

```
# Count of unique values in the country column
data_country["country"].value_counts().head()
```

```
Out[546...]
```

```
country
United States    3211
India           1008
United Kingdom   628
United States     479
Canada            271
Name: count, dtype: int64
```

```
In [547...]
```

```
data_country["country"].isnull().sum()
```

```
Out[547...]
```

831

```
In [548...]
```

```
#There are a lot of missing values in the country, so removing them instead of assigning with mode value for better calculations
```

```
In [549...]
```

```
# Dropping the rows where there is null value in country
data_country.dropna(subset=['country'], inplace=True)
```

```
In [550...]
```

```
# Verifying the Null values
data_country["country"].isna().sum()
```

```
Out[550...]
```

0

```
In [551...]
```

```
data_country["country"].value_counts().head(10)
```

```
Out[551... country
United States    3211
India           1008
United Kingdom   628
United States     479
Canada          271
Japan            259
France           212
South Korea      211
Spain             181
France           181
Name: count, dtype: int64
```

```
In [552... data_country["country"] = data_country["country"].str.strip()
```

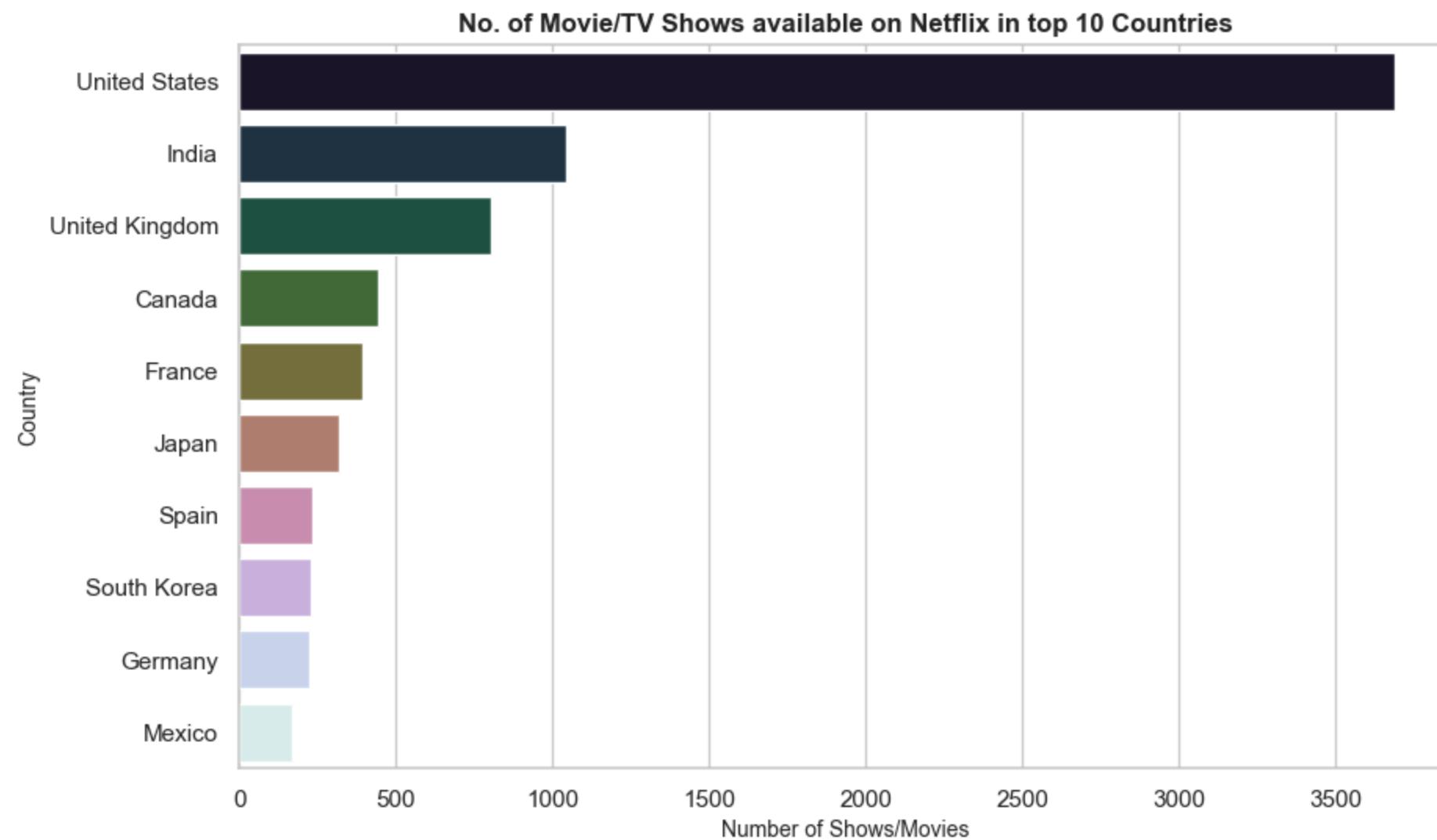
```
In [553... data_country["country"].value_counts().head(10)
```

```
Out[553... country
United States    3690
India           1046
United Kingdom   806
Canada          445
France           393
Japan            318
Spain             232
South Korea      231
Germany          226
Mexico            169
Name: count, dtype: int64
```

```
In [554... country_counts = data_country["country"].value_counts().head(10)
country_counts
```

```
Out[554... country
United States    3690
India           1046
United Kingdom   806
Canada          445
France           393
Japan            318
Spain             232
South Korea      231
Germany          226
Mexico            169
Name: count, dtype: int64
```

```
In [555... plt.figure(figsize=(10,6))
sns.barplot(x = country_counts.values, y = country_counts.index, palette = 'cubehelix')
plt.title('No. of Movie/TV Shows available on Netflix in top 10 Countries', fontweight = 'bold', fontsize = 12)
plt.xlabel('Number of Shows/Movies', fontsize = 10)
plt.ylabel('Country', fontsize = 10)
plt.show()
```



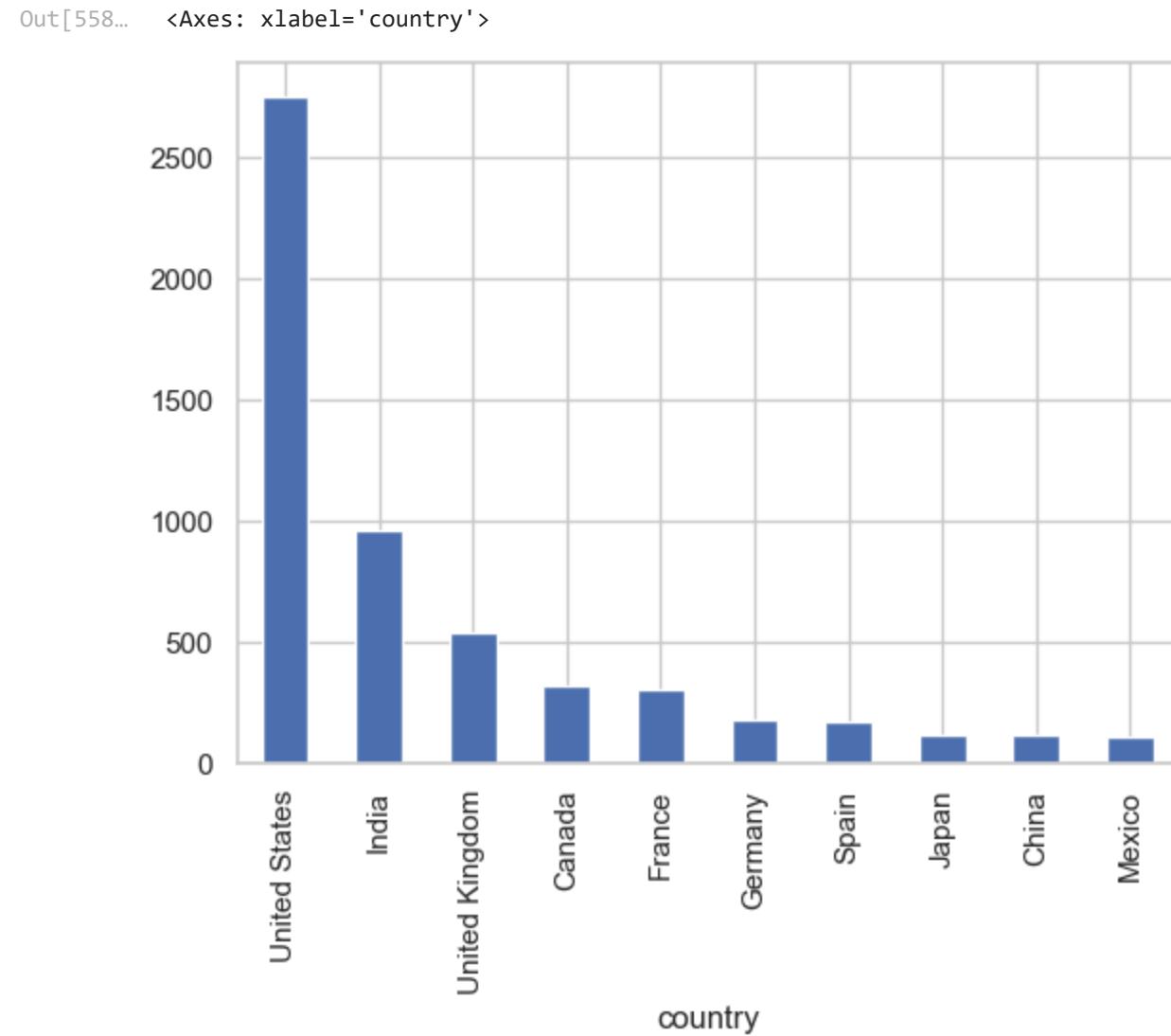
```
In [556]: #Countries with the highest number of Movie offers in the data
data_country[data_country['type']=='Movie']['country'].value_counts().head(10)
```

```
Out[556]: country
United States    2752
India            962
United Kingdom   534
Canada           319
France           303
Germany          182
Spain             171
Japan             119
China             114
Mexico            111
Name: count, dtype: int64
```

```
In [557]: #Countries with the highest number of TV Shows offers in the data
data_country[data_country['type']=='TV Show']['country'].value_counts().head(10)
```

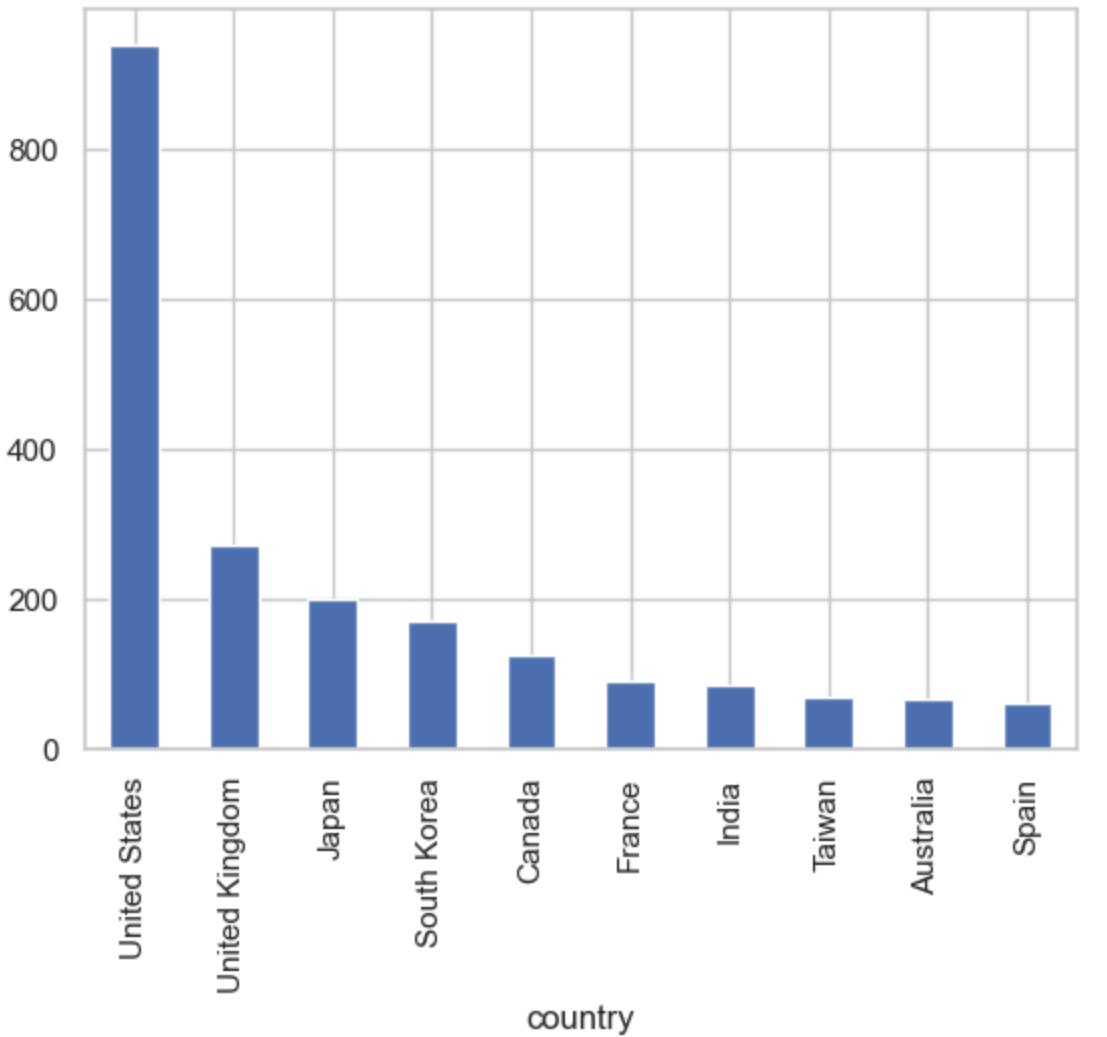
```
Out[557... country
United States    938
United Kingdom   272
Japan            199
South Korea     170
Canada           126
France           90
India             84
Taiwan            70
Australia         66
Spain             61
Name: count, dtype: int64
```

```
In [558... #The top ten countries producing Movies
data_country[data_country['type']=='Movie']['country'].value_counts().head(10).plot(kind='bar')
```



```
In [559... #The top ten countries producing TV Shows
data_country[data_country['type']=='TV Show']['country'].value_counts().head(10).plot(kind='bar')
```

Out[559... <Axes: xlabel='country'>



```
In [560]:
```

```
#TOP 3 Country in Movies
#1. United States
#2. India
#3. United Kingdom
```

```
#TOP 3 Country in TV Shows
#1. United States
#2. United Kingdom
#3. Japan
```

```
#TOP 3 Country in overall shows (Movies/TV Shows)
#1. United States
#2. India
#3. United Kingdom
```

```
In [561]:
```

```
#intersection table between country and type
pd.crosstab(data_country.type, data_country.country)
```

country	Afghanistan	Albania	Algeria	Angola	Argentina	Armenia	Australia	Austria	Azerbaijan	... Ukraine	United Arab Emirates	United Kingdom	United States	Uruguay	Vatican City	Venezuela	Vietnam	West Germany	Zimbabwe
type																			
Movie	6	1	1	3	1	71	1	94	11	0 ... 1	36	534	2752	13	1	4	7	3	3
TV Show	1	0	0	0	0	20	0	66	1	1 ... 2	1	272	938	1	0	0	0	2	0

2 rows × 123 columns

data_country.head()																			
	show_id	type	title	director	cast	country	date_added	release_year	rating	duration									
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Nan	United States	September 25, 2021	2020	PG-13	90			Documentaries	As her father nears the end of his life, filmm...		2021.0			
1	s2	TV Show	Blood & Water	Nan	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2		International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...		2021.0				
4	s5	TV Show	Kota Factory	Nan	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2	International TV Shows, Romantic TV Shows, TV ...		In a city of coaching centers known to train I...		2021.0				
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States	September 24, 2021	1993	TV-MA	125	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...		2021.0					
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	Ghana	September 24, 2021	1993	TV-MA	125	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...		2021.0					

data_country.isnull().sum()																					
show_id	0	type	0	title	0	director	2561	cast	852	country	0	date_added	9	release_year	0	rating	0	duration	0	listed_in	0
director		type		title		cast		country		date_added		release_year		rating		duration		listed_in			
cast		country		date_added		rating		duration		listed_in		description		year_added		dtype: int64		director			
country		date_added		rating		duration		listed_in		description		year_added		dtype: int64		type		title			
date_added		rating		duration		listed_in		description		year_added		dtype: int64		type		title		cast			
rating		duration		listed_in		description		year_added		dtype: int64		type		title		cast		country			
duration		listed_in		description		year_added		dtype: int64		type		title		cast		country		date_added			
listed_in		description		year_added		dtype: int64		type		title		cast		country		date_added		rating			
description		year_added		dtype: int64		type		title		cast		country		date_added		rating		duration			
year_added		dtype: int64		type		title		cast		country		date_added		rating		duration		listed_in			
dtype: int64		type		title		cast		country		date_added		rating		duration		listed_in		description			

```

In [566]: data_director = data.copy()
data_director["director"] = data_director["director"].str.split(",")
data_director = data_director.explode("director")
data_director["director"].value_counts().reset_index()

```

```
Out[566...]  
    director  count  
0      Rajiv Chilaka     22  
1      Raúl Campos      18  
2        Jan Suter      18  
3      Marcus Raboy      16  
4      Suhas Kadav      16  
...       ...      ...  
5115    Will Eisenberg      1  
5116  Marina Seresesky      1  
5117    Kenny Leon      1  
5118   James Dearden      1  
5119    Mozez Singh      1
```

5120 rows × 2 columns

```
In [567...]  
data.shape
```

```
Out[567...]  
(8807, 13)
```

```
In [568...]  
data_director.shape
```

```
Out[568...]  
(9612, 13)
```

```
In [569...]  
# Count of unique values in the director column  
data_director["director"].value_counts().head()
```

```
Out[569...]  
director  
Rajiv Chilaka     22  
Raúl Campos      18  
Jan Suter         18  
Marcus Raboy      16  
Suhas Kadav      16  
Name: count, dtype: int64
```

```
In [570...]  
data_director["director"].isnull().sum()
```

```
Out[570...]  
2634
```

```
In [571...]  
# Dropping the rows where there is null value in director  
data_director.dropna(subset=['director'], inplace=True)
```

```
In [573...]  
# Verifying the Null values  
data_director["director"].isna().sum()
```

```
Out[573...]: 0
```

```
In [574...]: data_director["director"].value_counts().head(10)
```

```
Out[574...]: director
Rajiv Chilaka      22
Raúl Campos        18
Jan Suter          18
Marcus Raboy       16
Suhas Kadav        16
Jay Karas          15
Cathy Garcia-Molina 13
Martin Scorsese     12
Jay Chapman         12
Youssef Chahine     12
Name: count, dtype: int64
```

```
In [575...]: data_director["director"] = data_director["director"].str.strip()
```

```
In [577...]: data_director["director"].value_counts().head(10)
```

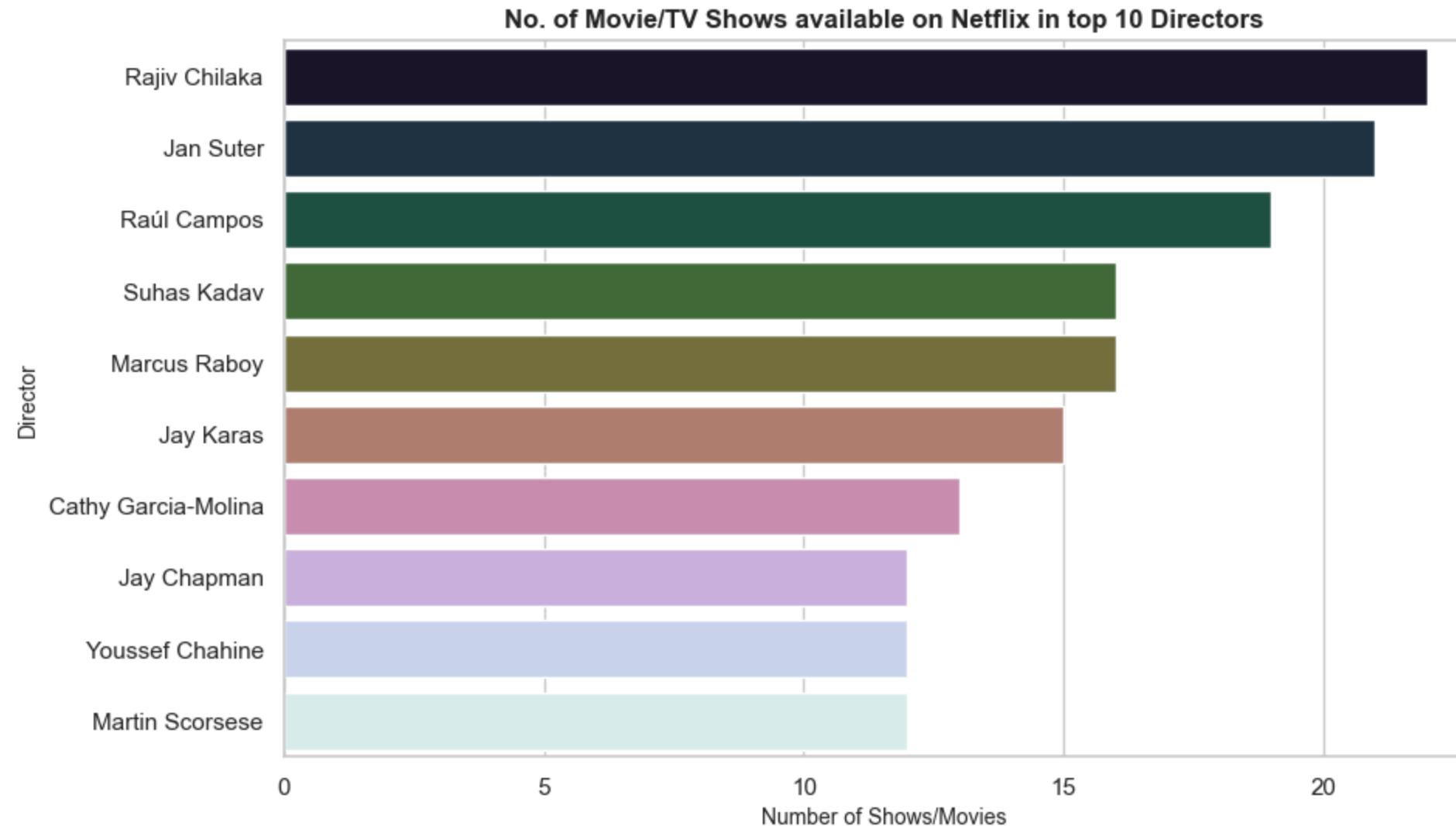
```
Out[577...]: director
Rajiv Chilaka      22
Jan Suter          21
Raúl Campos        19
Suhas Kadav        16
Marcus Raboy       16
Jay Karas          15
Cathy Garcia-Molina 13
Jay Chapman         12
Youssef Chahine     12
Martin Scorsese     12
Name: count, dtype: int64
```

```
In [578...]: director_counts = data_director["director"].value_counts().head(10)
director_counts
```

```
Out[578...]: director
Rajiv Chilaka      22
Jan Suter          21
Raúl Campos        19
Suhas Kadav        16
Marcus Raboy       16
Jay Karas          15
Cathy Garcia-Molina 13
Jay Chapman         12
Youssef Chahine     12
Martin Scorsese     12
Name: count, dtype: int64
```

```
In [579...]: plt.figure(figsize=(10,6))
sns.barplot(x = director_counts.values, y = director_counts.index, palette = 'cubehelix')
plt.title('No. of Movie/TV Shows available on Netflix in top 10 Directors', fontweight = 'bold', fontsize = 12)
plt.xlabel('Number of Shows/Movies', fontsize = 10)
```

```
plt.ylabel('Director', fontsize = 10)
plt.show()
```



```
In [583...]: #Directors with the highest number of Movie offers in the data
data_director[data_director['type']=='Movie']['director'].value_counts().head(10)
```

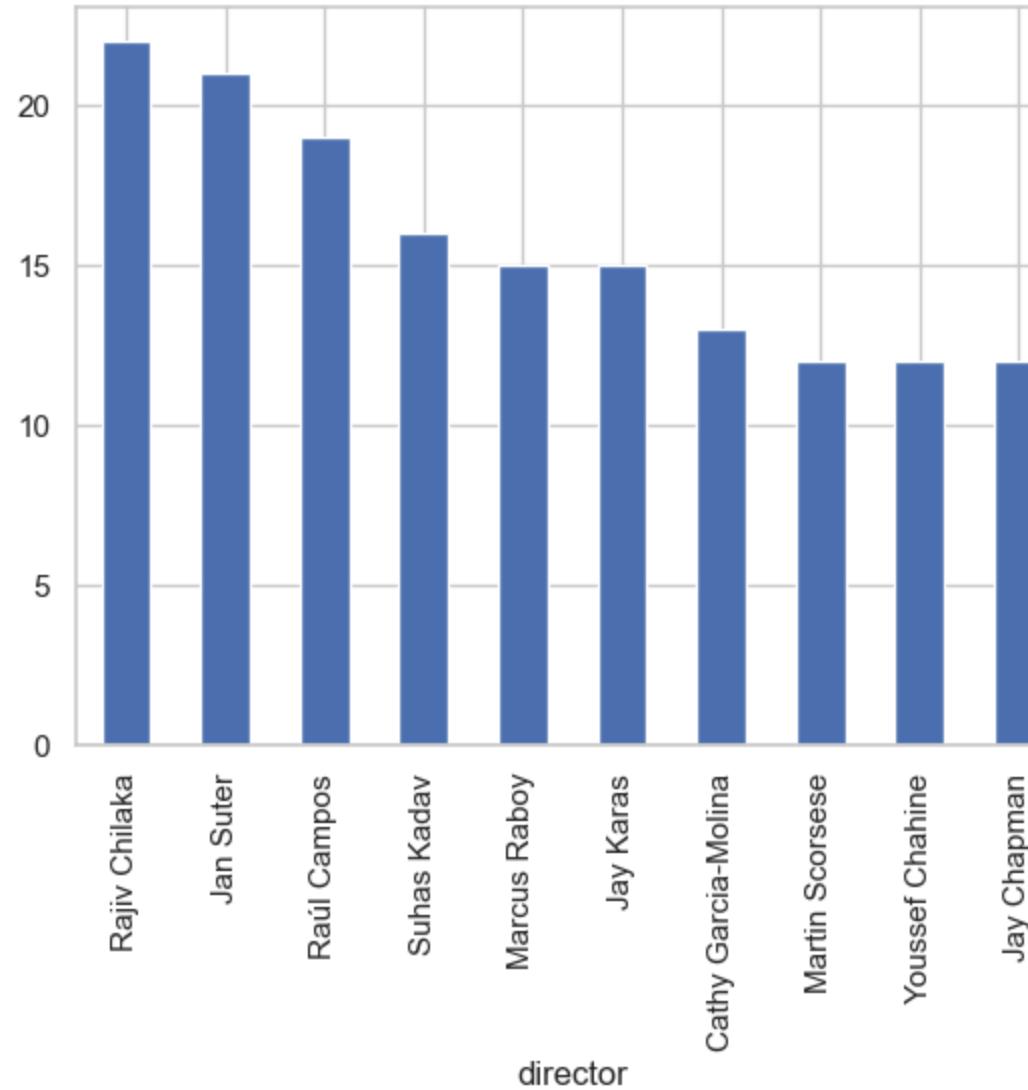
```
Out[583...]: director
Rajiv Chilaka    22
Jan Suter       21
Raúl Campos     19
Suhas Kadav     16
Marcus Raboy    15
Jay Karas       15
Cathy Garcia-Molina  13
Martin Scorsese   12
Youssef Chahine   12
Jay Chapman      12
Name: count, dtype: int64
```

```
In [584...]: #Directors with the highest number of TV Shows offers in the data
data_director[data_director['type']=='TV Show']['director'].value_counts().head(10)
```

```
Out[584... director
Alastair Fothergill      3
Ken Burns                 3
Jung-ah Im                2
Gautham Vasudev Menon    2
Iginio Straffi            2
Hsu Fu-chun               2
Stan Lathan                2
Joe Berlinger              2
Shin Won-ho                2
Lynn Novick                2
Name: count, dtype: int64
```

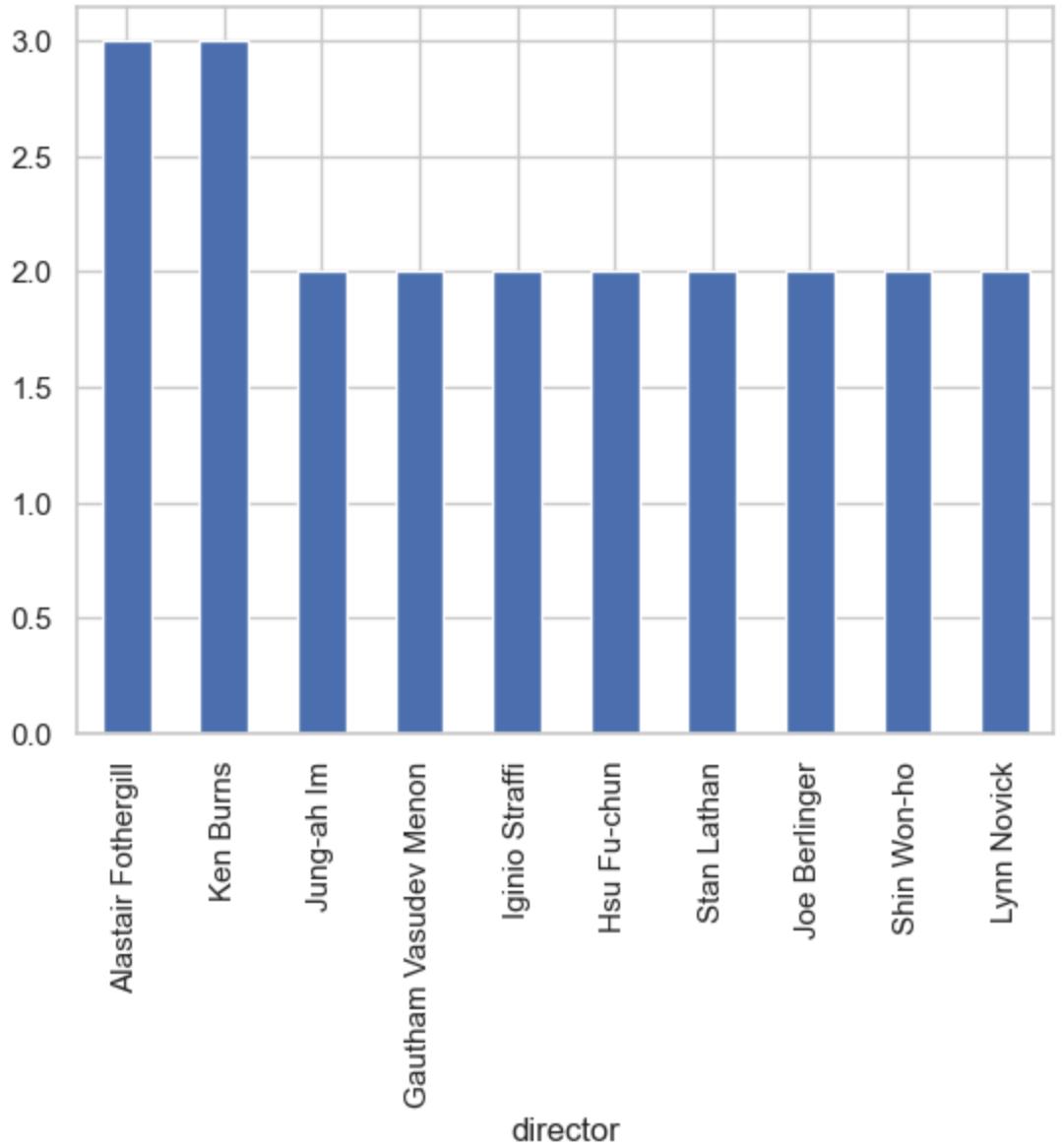
```
In [585... #The top ten Director producing Movies
data_director[data_director['type']=='Movie']['director'].value_counts().head(10).plot(kind='bar')
```

```
Out[585... <Axes: xlabel='director'>
```



```
In [586... #The top ten Director producing TV Shows
data_director[data_director['type']=='TV Show']['director'].value_counts().head(10).plot(kind='bar')
```

```
Out[586... <Axes: xlabel='director'>
```



```
In [587]: #TOP Director in Movies  
#1. Rajiv Chilaka
```

```
#TOP Director in TV Shows  
#1. Alastair Fothergill, Ken Burns
```

```
#TOP Director in overall shows (Movies/TV Shows)  
#1. Rajiv Chilaka
```

```
In [588]: data_cast = data.copy()  
data_cast["cast"] = data_cast["cast"].str.split(",")  
data_cast = data_cast.explode("cast")  
data_cast["cast"].value_counts().reset_index()
```

```
Out[588...]
```

	cast	count
0	Anupam Kher	39
1	Rupa Bhimani	31
2	Takahiro Sakurai	30
3	Julie Tejwani	28
4	Om Puri	27
...	...	...
39291	Vedika	1
39292	Tedros Teclebrhan	1
39293	Maryam Zaree	1
39294	Melanie Straub	1
39295	Chittaranjan Tripathy	1

39296 rows × 2 columns

```
In [589...]
```

```
# Count of unique values in the cast column  
data_cast["cast"].value_counts().head()
```

```
Out[589...]
```

```
cast  
Anupam Kher      39  
Rupa Bhimani    31  
Takahiro Sakurai 30  
Julie Tejwani    28  
Om Puri          27  
Name: count, dtype: int64
```

```
In [590...]
```

```
data_cast["cast"].isnull().sum()
```

```
Out[590...]
```

825

```
In [591...]
```

```
# Dropping the rows where there is null value in casr  
data_cast.dropna(subset=['cast'], inplace=True)
```

```
In [592...]
```

```
# Verifying the Null values  
data_cast["cast"].isna().sum()
```

```
Out[592...]
```

0

```
In [593...]
```

```
data_cast["cast"].value_counts().head(10)
```

```
Out[593...]: cast
Anupam Kher      39
Rupa Bhimani    31
Takahiro Sakurai 30
Julie Tejwani    28
Om Puri          27
Rajesh Kava      26
Shah Rukh Khan   26
Yuki Kaji         25
Paresh Rawal     25
Andrea Libman    25
Name: count, dtype: int64
```

```
In [594...]: data_cast["cast"] = data_cast["cast"].str.strip()
```

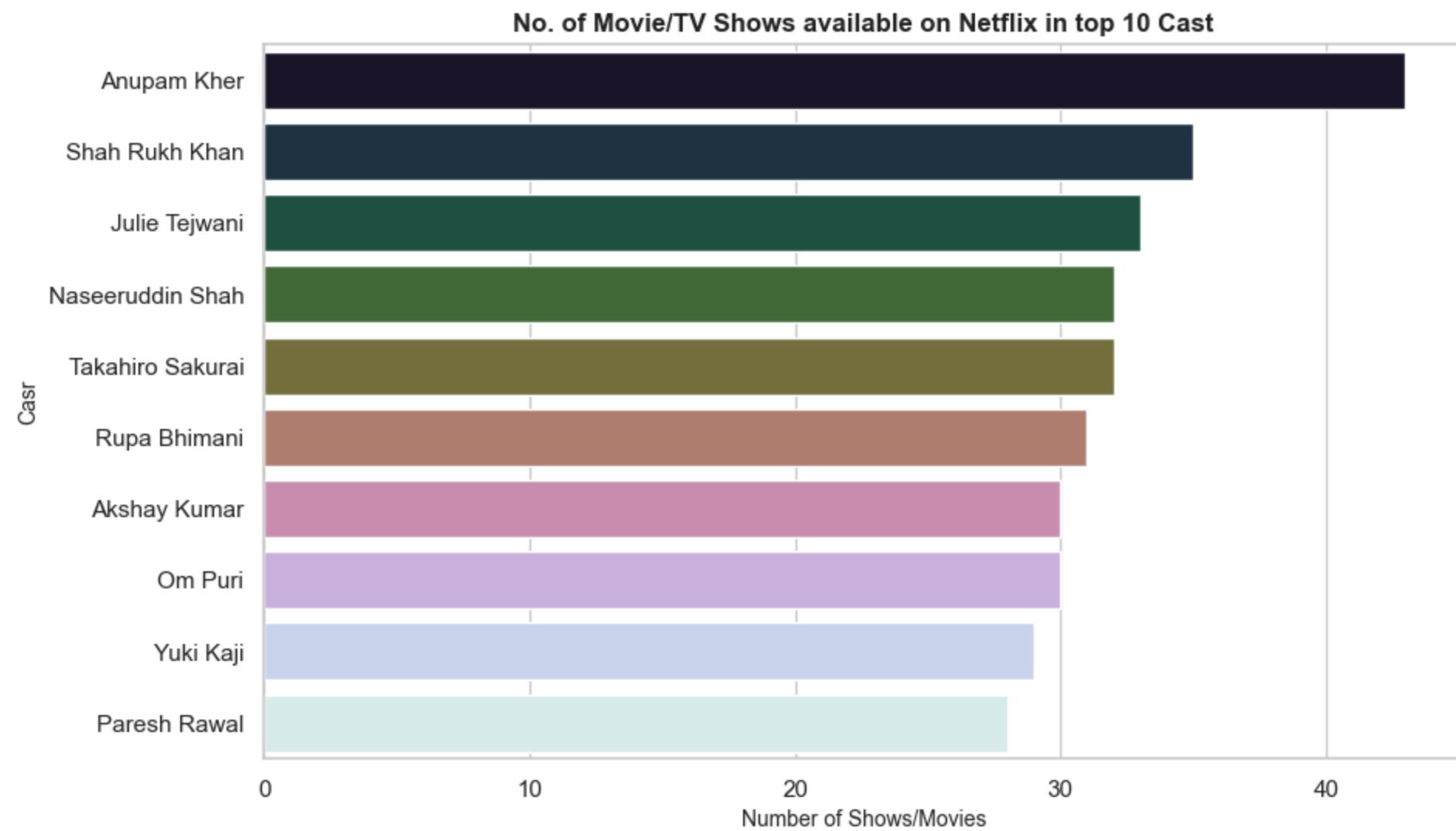
```
In [595...]: data_cast["cast"].value_counts().head(10)
```

```
Out[595...]: cast
Anupam Kher      43
Shah Rukh Khan   35
Julie Tejwani    33
Naseeruddin Shah 32
Takahiro Sakurai 32
Rupa Bhimani     31
Akshay Kumar      30
Om Puri          30
Yuki Kaji         29
Paresh Rawal     28
Name: count, dtype: int64
```

```
In [596...]: cast_counts = data_cast["cast"].value_counts().head(10)
cast_counts
```

```
Out[596...]: cast
Anupam Kher      43
Shah Rukh Khan   35
Julie Tejwani    33
Naseeruddin Shah 32
Takahiro Sakurai 32
Rupa Bhimani     31
Akshay Kumar      30
Om Puri          30
Yuki Kaji         29
Paresh Rawal     28
Name: count, dtype: int64
```

```
In [597...]: plt.figure(figsize=(10,6))
sns.barplot(x = cast_counts.values, y = cast_counts.index, palette = 'cubehelix')
plt.title('No. of Movie/TV Shows available on Netflix in top 10 Cast', fontweight = 'bold', fontsize = 12)
plt.xlabel('Number of Shows/Movies', fontsize = 10)
plt.ylabel('Cast', fontsize = 10)
plt.show()
```



```
In [598]: #Cast with the highest number of Movie offers in the data
data_cast[data_cast['type']=='Movie']['cast'].value_counts().head(10)
```

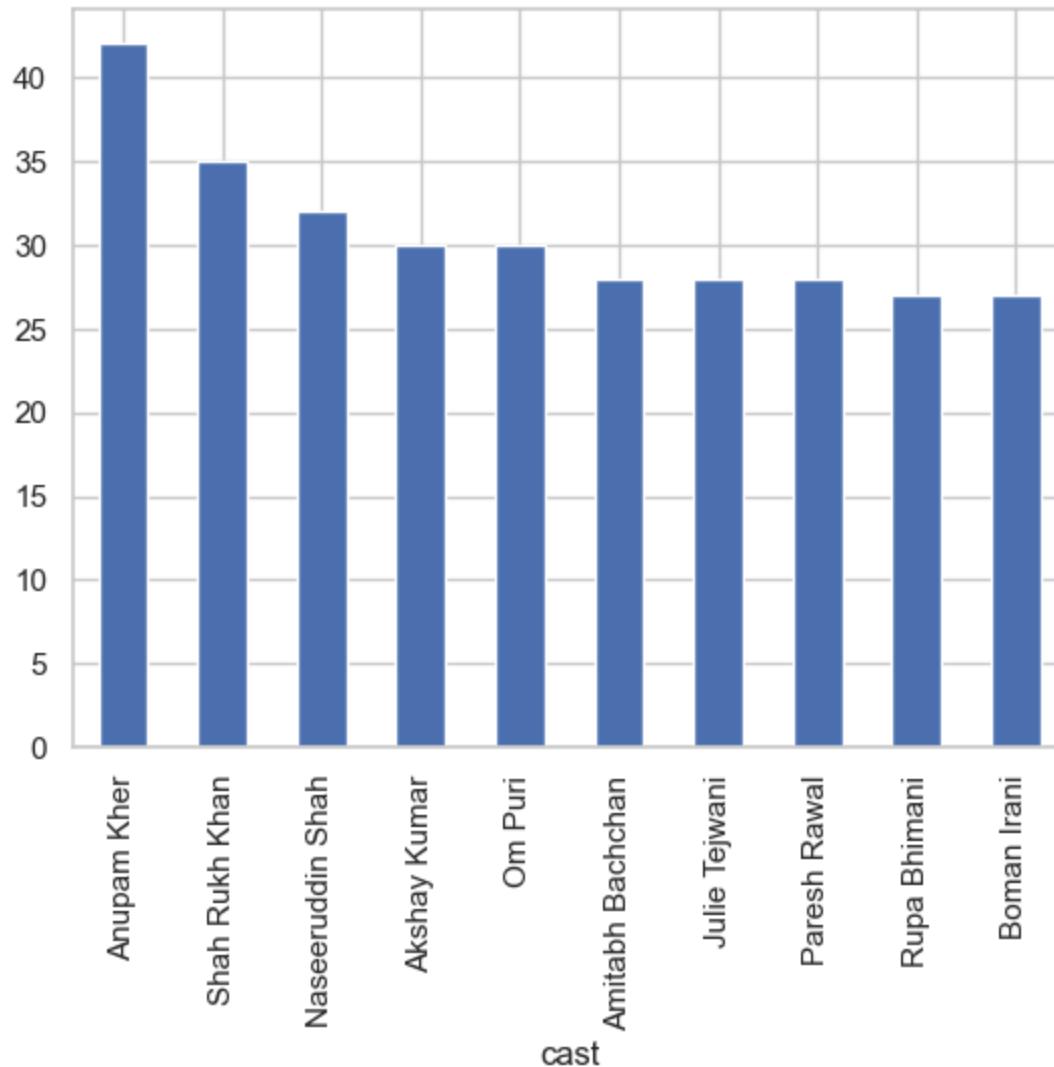
```
Out[598]: cast
Anupam Kher      42
Shah Rukh Khan   35
Naseeruddin Shah 32
Akshay Kumar      30
Om Puri          30
Amitabh Bachchan 28
Julie Tejwani     28
Paresh Rawal      28
Rupa Bhimani     27
Boman Irani       27
Name: count, dtype: int64
```

```
In [599]: #Cast with the highest number of TV Shows offers in the data
data_cast[data_cast['type']=='TV Show']['cast'].value_counts().head(10)
```

```
Out[599]: cast
Takahiro Sakurai      25
Yuki Kaji              19
Daisuke Ono            17
Junichi Suwabe          17
Ai Kayano              17
Yuichi Nakamura        16
Jun Fukuyama           15
Yoshimasa Hosoya       15
David Attenborough     14
Mamoru Miyano           13
Name: count, dtype: int64
```

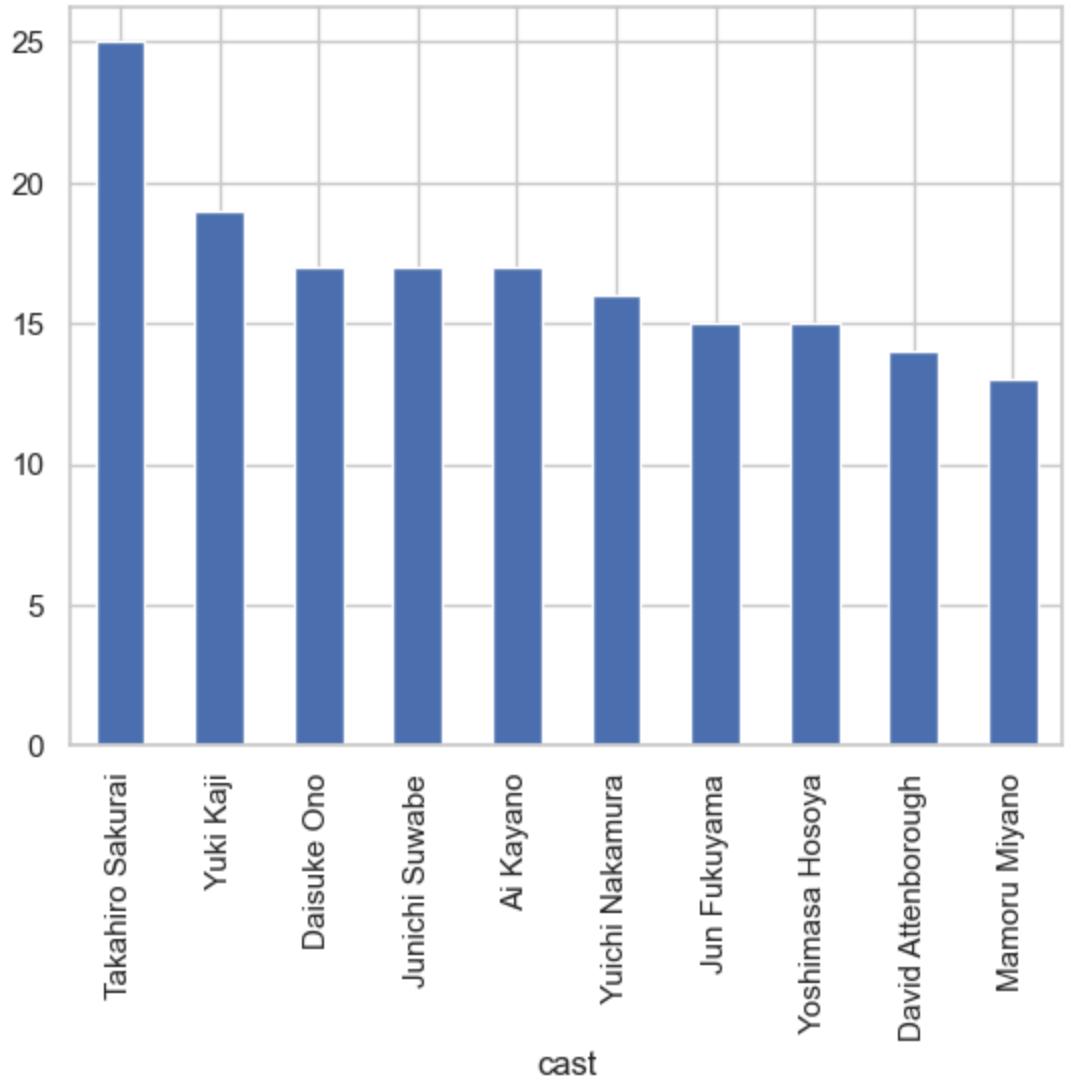
```
In [600]: #The top ten cast producing Movies
data_cast[data_cast['type']=='Movie']['cast'].value_counts().head(10).plot(kind='bar')
```

```
Out[600]: <Axes: xlabel='cast'>
```



```
In [601]: #The top ten cast producing TV Shows
data_cast[data_cast['type']=='TV Show']['cast'].value_counts().head(10).plot(kind='bar')
```

```
Out[601]: <Axes: xlabel='cast'>
```



```
In [602... #TOP Cast in Movies  
#1. Anupam Kher
```

```
#TOP Cast in TV Shows  
#1. Takahiro Sakurai
```

```
#TOP Cast in overall shows (Movies/TV Shows)  
#1. Anupam Kher
```

```
In [603... # Unique values in 'director' and 'cast' columns  
print(f"Unique Directors: {data_director['director'].nunique()}")  
print(f"Unique Cast members: {data_cast['cast'].nunique()}")
```

```
Unique Directors: 4993  
Unique Cast members: 36439
```

```
In [604... data_director.shape
```

```
Out[604... (6978, 13)
```

```
In [605... data_cast.shape
```

```
Out[605... (64126, 13)
```

```
In [607... data["listed_in"]
```

```
Out[607... 0 Documentaries
1 International TV Shows, TV Dramas, TV Mysteries
2 Crime TV Shows, International TV Shows, TV Act...
3 Docuseries, Reality TV
4 International TV Shows, Romantic TV Shows, TV ...
...
8802 Cult Movies, Dramas, Thrillers
8803 Kids' TV, Korean TV Shows, TV Comedies
8804 Comedies, Horror Movies
8805 Children & Family Movies, Comedies
8806 Dramas, International Movies, Music & Musicals
Name: listed_in, Length: 8807, dtype: category
Categories (514, object): ['Action & Adventure', 'Action & Adventure, Anime Features', 'Action & Adventure, Anime Features, Children ...', 'Action & Adventure, Anime Features, Classic M..., ..., 'TV Horror, Teen TV Shows', 'TV Sci-Fi & Fantasy, TV Thrillers', 'TV Shows', 'Thrillers']
```

```
In [608... data_listed_in = data.copy()
data_listed_in["listed_in"] = data_listed_in["listed_in"].str.split(",")
data_listed_in = data_listed_in.explode("listed_in")
data_listed_in["listed_in"].value_counts().reset_index()
```

```
Out[608...   listed_in  count
0 International Movies  2624
1 Dramas  1600
2 Comedies  1210
3 Action & Adventure  859
4 Documentaries  829
...
68 Romantic Movies  3
69 Spanish-Language TV Shows  2
70 LGBTQ Movies  1
71 TV Sci-Fi & Fantasy  1
72 Sports Movies  1
```

73 rows × 2 columns

```
In [609... # Count of unique values in the Listed_in column
data_listed_in["listed_in"].value_counts().head()
```

```
Out[609...]: listed_in  
    International Movies    2624  
    Dramas                 1600  
    Comedies                1210  
    Action & Adventure     859  
    Documentaries            829  
Name: count, dtype: int64
```

```
In [610...]: data_listed_in["listed_in"].isnull().sum()
```

```
Out[610...]: 0
```

```
In [611...]: data_listed_in["listed_in"].value_counts().head(10)
```

```
Out[611...]: listed_in  
    International Movies    2624  
    Dramas                 1600  
    Comedies                1210  
    Action & Adventure     859  
    Documentaries            829  
    Dramas                  827  
    International TV Shows   774  
    Independent Movies       736  
    TV Dramas                696  
    Romantic Movies           613  
Name: count, dtype: int64
```

```
In [612...]: data_listed_in["listed_in"] = data_listed_in["listed_in"].str.strip()
```

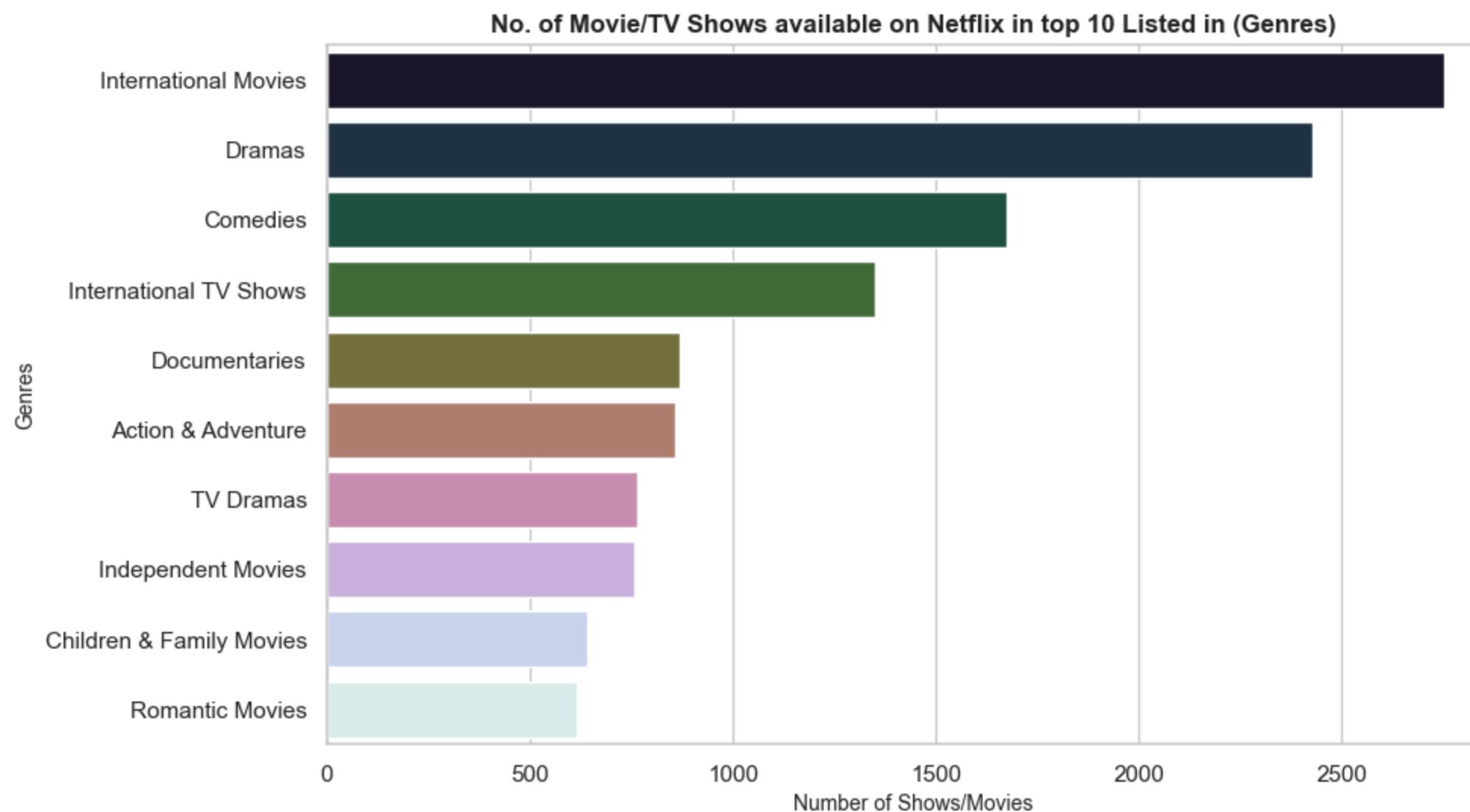
```
In [613...]: data_listed_in["listed_in"].value_counts().head(10)
```

```
Out[613...]: listed_in  
    International Movies    2752  
    Dramas                 2427  
    Comedies                1674  
    International TV Shows   1351  
    Documentaries             869  
    Action & Adventure      859  
    TV Dramas                 763  
    Independent Movies        756  
    Children & Family Movies   641  
    Romantic Movies              616  
Name: count, dtype: int64
```

```
In [614...]: listed_in_counts = data_listed_in["listed_in"].value_counts().head(10)  
listed_in_counts
```

```
Out[614...]: 
   listed_in
International Movies      2752
Dramas                  2427
Comedies                1674
International TV Shows   1351
Documentaries            869
Action & Adventure     859
TV Dramas                763
Independent Movies       756
Children & Family Movies 641
Romantic Movies           616
Name: count, dtype: int64
```

```
In [615...]: 
plt.figure(figsize=(10,6))
sns.barplot(x = listed_in_counts.values, y = listed_in_counts.index, palette = 'cubehelix')
plt.title('No. of Movie/TV Shows available on Netflix in top 10 Listed in (Genres)', fontweight = 'bold', fontsize = 12)
plt.xlabel('Number of Shows/Movies', fontsize = 10)
plt.ylabel('Genres', fontsize = 10)
plt.show()
```



```
In [616...]: 
#Genres with the highest number of Movie offers in the data
data_listed_in[data_listed_in['type']=='Movie']['listed_in'].value_counts().head(10)
```

```
Out[616...]:
```

listed_in	
International Movies	2752
Dramas	2427
Comedies	1674
Documentaries	869
Action & Adventure	859
Independent Movies	756
Children & Family Movies	641
Romantic Movies	616
Thrillers	577
Music & Musicals	375

Name: count, dtype: int64

```
In [617...]:
```

```
#Genres with the highest number of TV Shows offers in the data
data_listed_in[data_listed_in['type']=='TV Show']['listed_in'].value_counts().head(10)
```

```
Out[617...]:
```

listed_in	
International TV Shows	1351
TV Dramas	763
TV Comedies	581
Crime TV Shows	470
Kids' TV	451
Docuseries	395
Romantic TV Shows	370
Reality TV	255
British TV Shows	253
Anime Series	176

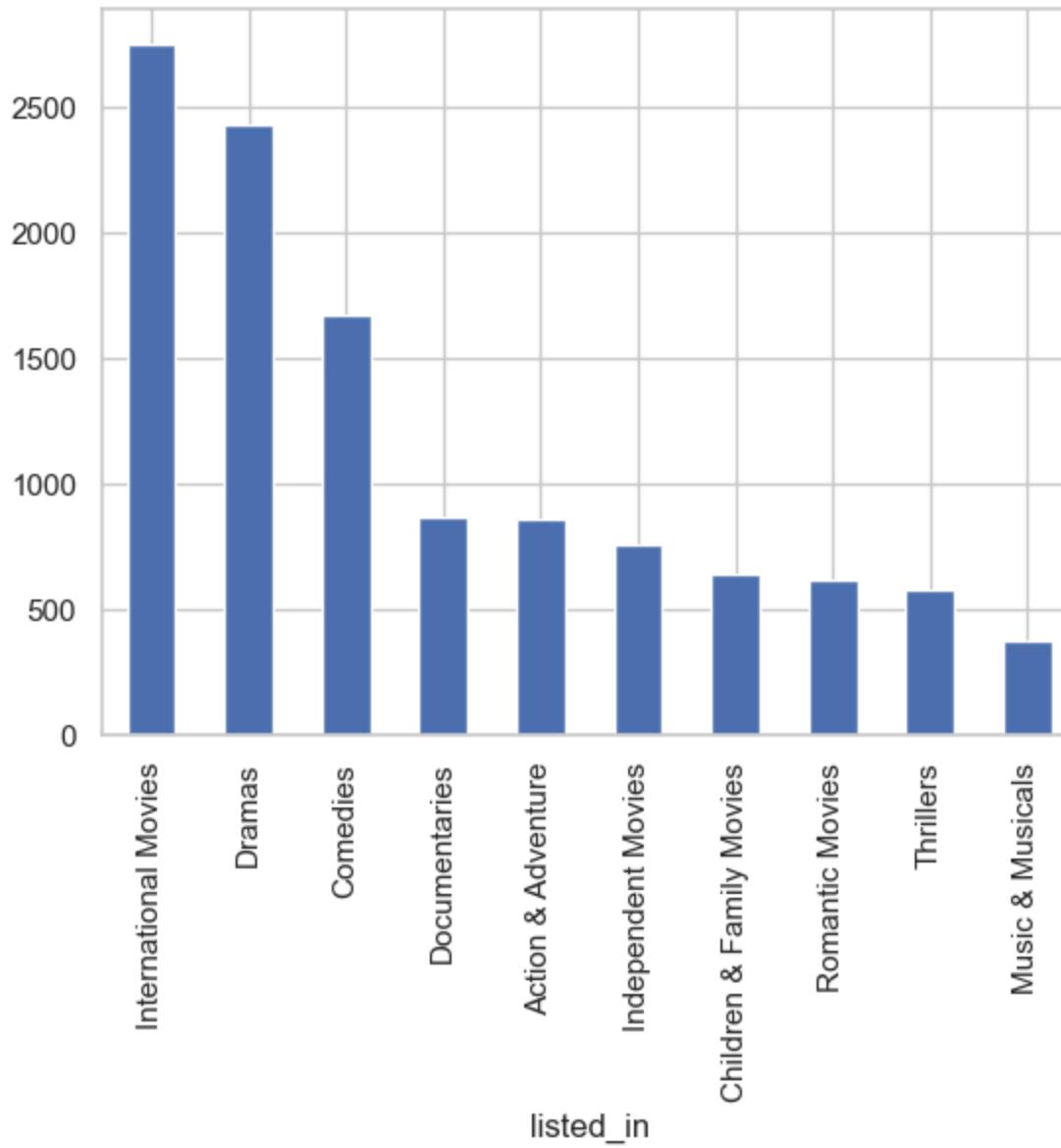
Name: count, dtype: int64

```
In [618...]:
```

```
#The top ten Genres producing Movies
data_listed_in[data_listed_in['type']=='Movie']['listed_in'].value_counts().head(10).plot(kind='bar')
```

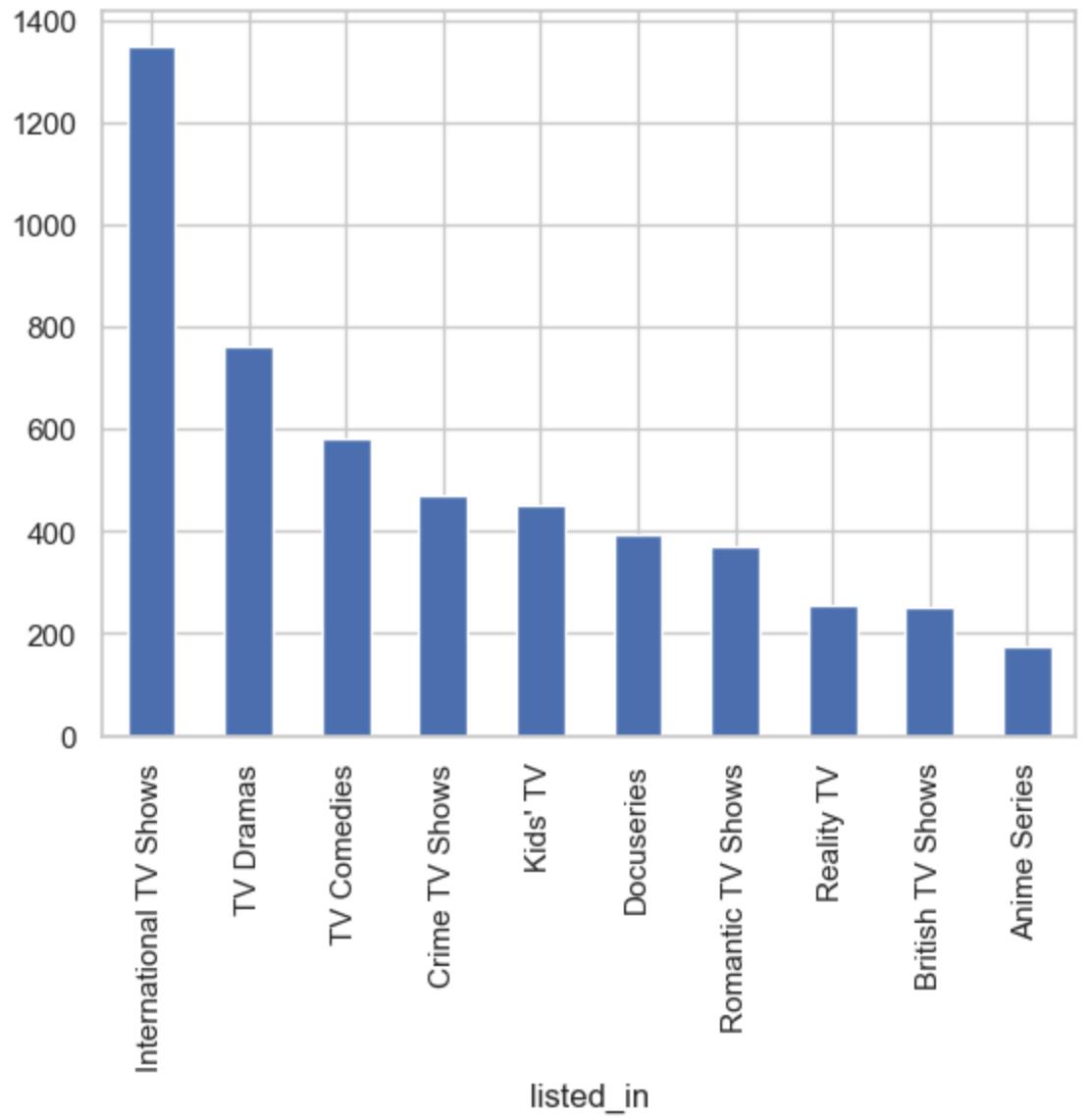
```
Out[618...]:
```

```
<Axes: xlabel='listed_in'>
```



```
In [619...]: #The top ten Genres producing TV Shows  
data_listed_in[data_listed_in['type']=='TV Show']['listed_in'].value_counts().head(10).plot(kind='bar')
```

```
Out[619...]: <Axes: xlabel='listed_in'>
```



```
In [620]:
```

```
#TOP 3 Genres in Movies
#1. International Movies
#2. Dramas
#3. Comedies
```

```
#TOP 3 Genres in TV Shows
#1. International TV Shows
#2. TV Dramas
#3. TV Comedies
```

```
#TOP 3 Genres in overall shows (Movies/TV Shows)
#1. International Movies
#2. Dramas
#3. Comedies
```

```
In [621]:
```

```
# Count of unique values in the date_added column
data["date_added"].value_counts().head(3)
```

```
Out[621...]: date_added
January 1, 2020      109
November 1, 2019     89
March 1, 2018        75
Name: count, dtype: int64

In [622...]: data["date_added"].isnull().sum()

Out[622...]: 10

In [623...]: data_date_added = data.copy()

In [625...]: # Dropping the rows where there is null value in data_date_added
data_date_added.dropna(subset=['date_added'], inplace=True)

In [626...]: data_date_added["date_added"].isnull().sum()

Out[626...]: 0

In [627...]: # Removing (,) commas appearing in the date_added column
data_date_added["date_added"] = data_date_added["date_added"].str.replace(",","")

In [628...]: data_date_added["date_added"]

Out[628...]: 0      September 25 2021
1      September 24 2021
2      September 24 2021
3      September 24 2021
4      September 24 2021
...
8802    November 20 2019
8803      July 1 2019
8804    November 1 2019
8805    January 11 2020
8806      March 2 2019
Name: date_added, Length: 8797, dtype: object

In [631...]: # Changing the date_added column to DateTime Format
data_date_added["date_added"] = pd.to_datetime(data_date_added["date_added"], format="mixed")

In [632...]: # Extracting Year from date_added column
data_date_added["year"] = data_date_added["date_added"].dt.year

In [633...]: # Extracting Month Names from date_added column
data_date_added["month"] = data_date_added["date_added"].dt.month_name()

In [634...]: # Extracting Date from date_added column
data_date_added["date"] = data_date_added["date_added"].dt.day

In [635...]: data_date_added.head()
```

Out[635...]	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	year_added	year	month	date
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90	Documentaries	As her father nears the end of his life, filmm...	2021.0	2021	September	25
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	2021.0	2021	September	24
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	2021.0	2021	September	24
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	1	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...	2021.0	2021	September	24
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...	2021.0	2021	September	24

In [637...]: `data_date_added["year"].value_counts().head(3)`

Out[637...]:

year	count
2019	2016
2020	1879
2018	1649

Name: count, dtype: int64

In [638...]: `data_date_added["month"].value_counts()`

Out[638...]:

month	count
July	827
December	813
September	770
April	764
October	760
August	755
March	742
January	738
June	728
November	705
May	632
February	563

Name: count, dtype: int64

In [640...]: `data_date_added["date"].value_counts().head(5)`

Out[640...]:

date	count
1	2212
15	687
2	325
16	289
31	274

Name: count, dtype: int64

```
In [641... x = data_date_added.groupby('date_added')[['show_id']].count()
x['show_day'] = x.index.day_name()
x
```

```
Out[641...      show_id  show_day
```

date_added	show_id	show_day
2008-01-01	1	Tuesday
2008-02-04	1	Monday
2009-05-05	1	Tuesday
2009-11-18	1	Wednesday
2010-11-01	1	Monday
...	...	...
2021-09-21	5	Tuesday
2021-09-22	9	Wednesday
2021-09-23	2	Thursday
2021-09-24	10	Friday
2021-09-25	1	Saturday

1714 rows × 2 columns

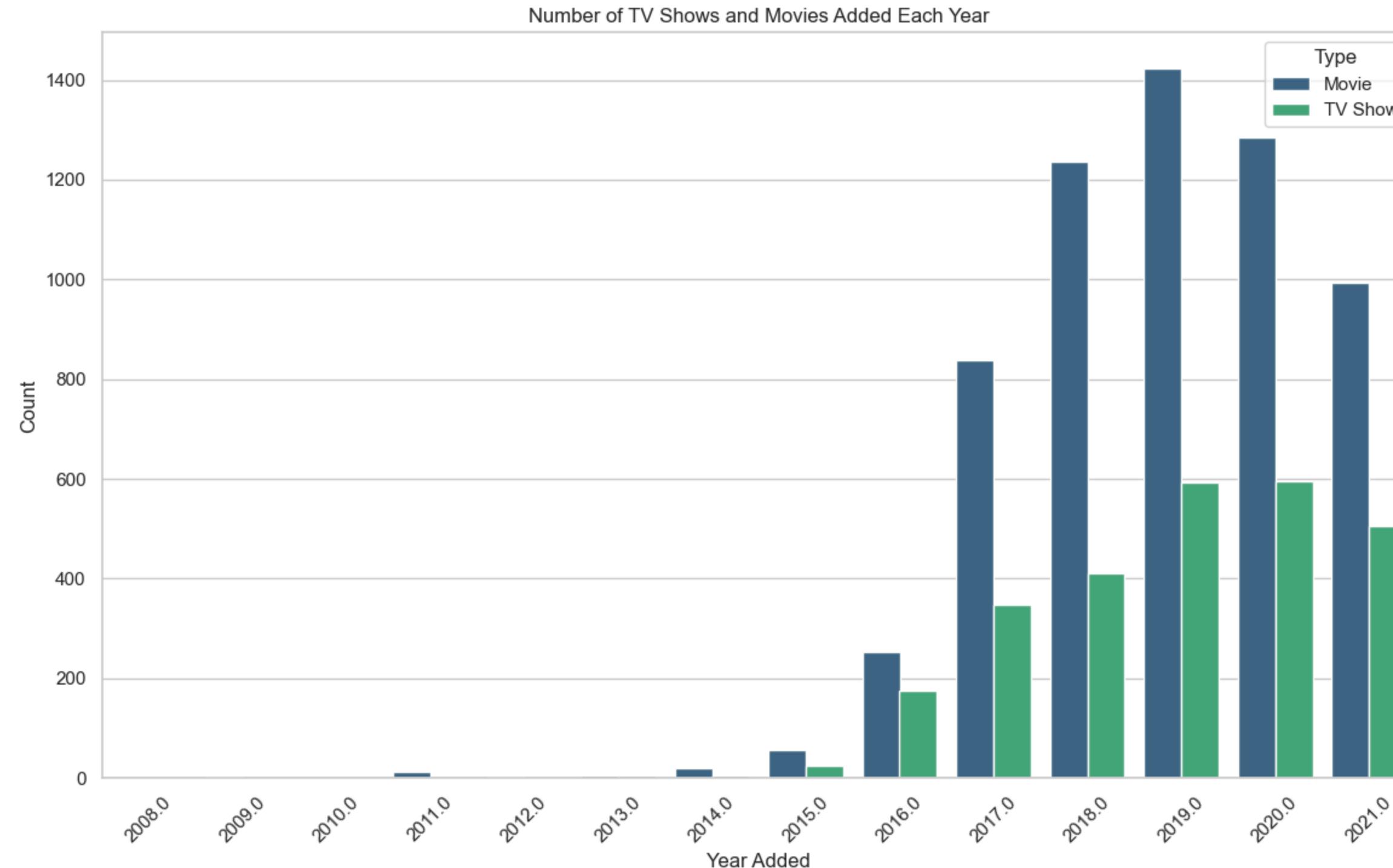
```
In [642... y = x.groupby('show_day')[['show_id']].sum()
y
```

```
Out[642...      show_id
```

show_day	show_id
Friday	2498
Monday	851
Saturday	816
Sunday	751
Thursday	1396
Tuesday	1197
Wednesday	1288

```
In [645... # Plot the number of TV shows and movies added each year
plt.figure(figsize=(14, 8))
sns.countplot(data=data_date_added, x='year_added', hue='type', palette='viridis')
plt.title('Number of TV Shows and Movies Added Each Year')
plt.xlabel('Year Added')
```

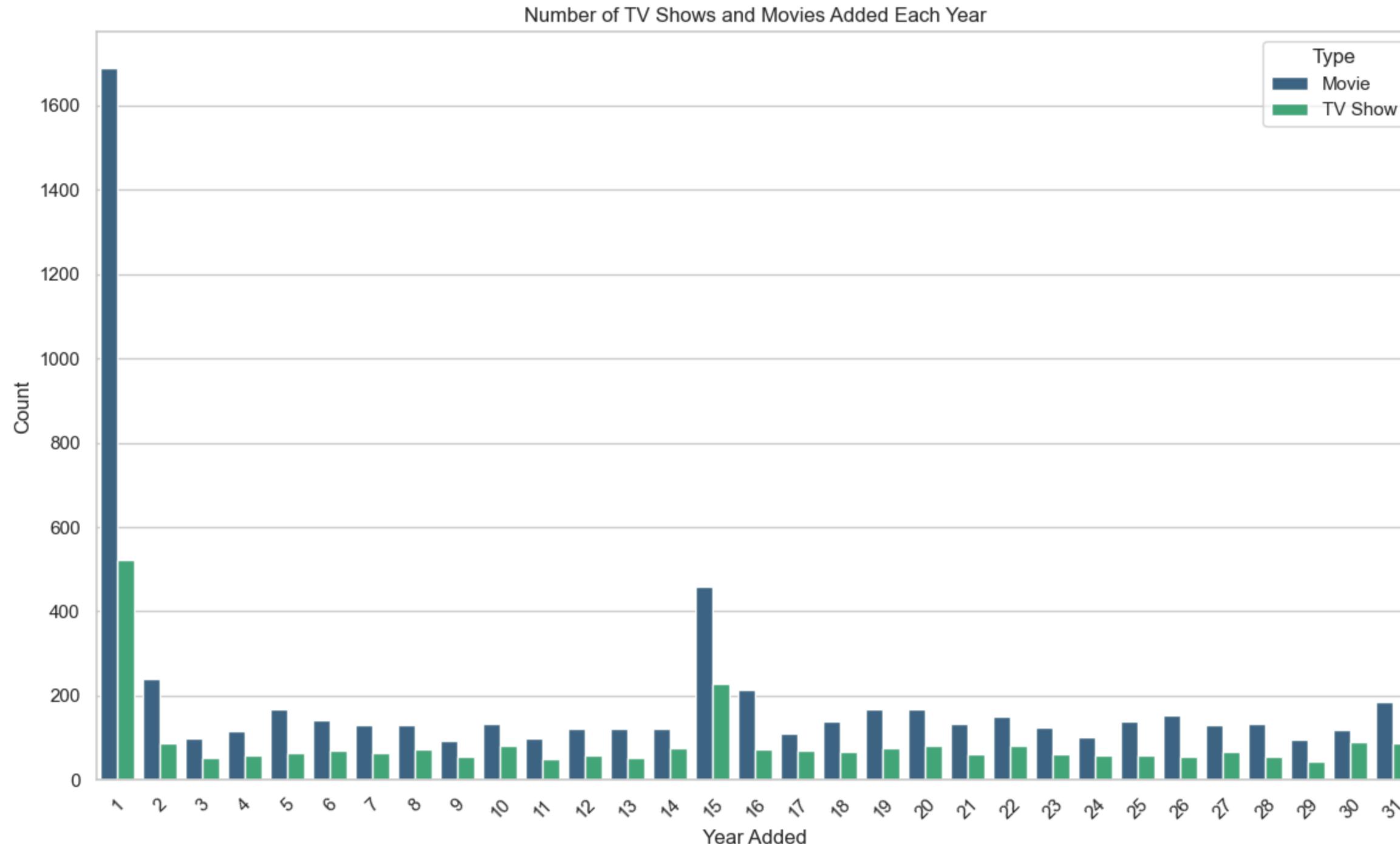
```
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Type')
plt.show()
```



In [646]:

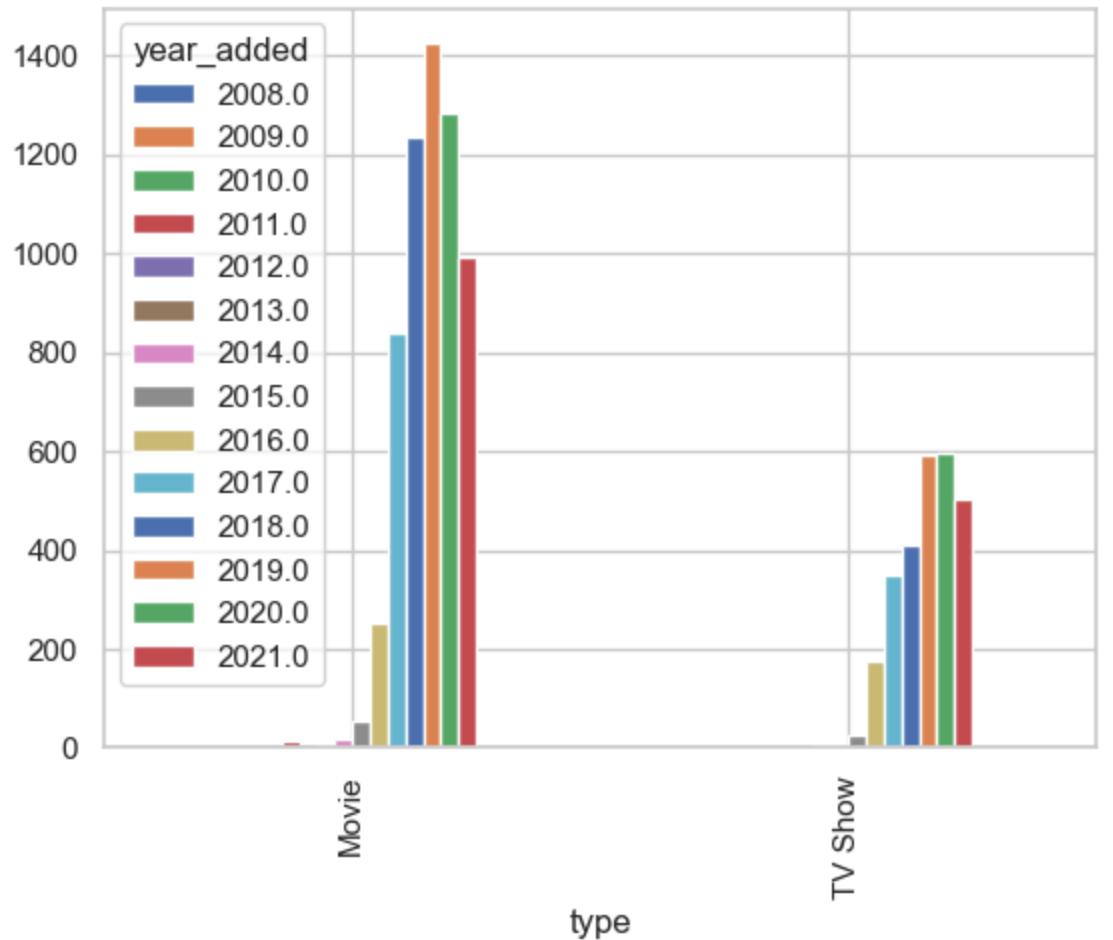
```
# Plot the number of TV shows and movies added each year
plt.figure(figsize=(14, 8))
sns.countplot(data=data_date_added, x='date', hue='type', palette='viridis')
plt.title('Number of TV Shows and Movies Added Each Year')
plt.xlabel('Year Added')
plt.ylabel('Count')
plt.xticks(rotation=45)
```

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



```
In [650]: pd.crosstab(data.type ,data.year_added).plot(kind='bar')
```

```
Out[650]: <Axes: xlabel='type'>
```



```
In [795]: #Most of the Movies/TV Shows added in the 2019
#Most of the Movies/TV Shows are added in 1st of 15th of the every month
#Friday is the most releases added in the Netflix
```

```
In [660]: data_country.groupby(["country", "listed_in"])[["country", "listed_in"]].value_counts()
```

```
Out[660]: country    listed_in
Action & Adventure                      0
Action & Adventure, Anime Features          0
Action & Adventure, Anime Features, Children & Family Movies 0
Action & Adventure, Anime Features, Classic Movies   0
Action & Adventure, Anime Features, Horror Movies   0
..                                           ..
Zimbabwe   TV Horror, TV Mysteries, Teen TV Shows      0
           TV Horror, Teen TV Shows                  0
           TV Sci-Fi & Fantasy, TV Thrillers        0
           TV Shows                                0
           Thrillers                               0
Name: count, Length: 63222, dtype: int64
```

```
In [665]: # Group by 'country' and 'type' and count the occurrences
content_by_country = data_country.groupby(['country', 'type']).size().reset_index(name='count')
content_by_country
```

Out[665...]

	country	type	count
0		Movie	6
1		TV Show	1
2	Afghanistan	Movie	1
3	Afghanistan	TV Show	0
4	Albania	Movie	1
...	...	...	...
241	Vietnam	TV Show	0
242	West Germany	Movie	3
243	West Germany	TV Show	2
244	Zimbabwe	Movie	3
245	Zimbabwe	TV Show	0

246 rows × 3 columns

In [668...]

```
# Pivot the data for better visualization
pivot_df = content_by_country.pivot(index='country', columns='type', values='count').fillna(0)
pivot_df
```

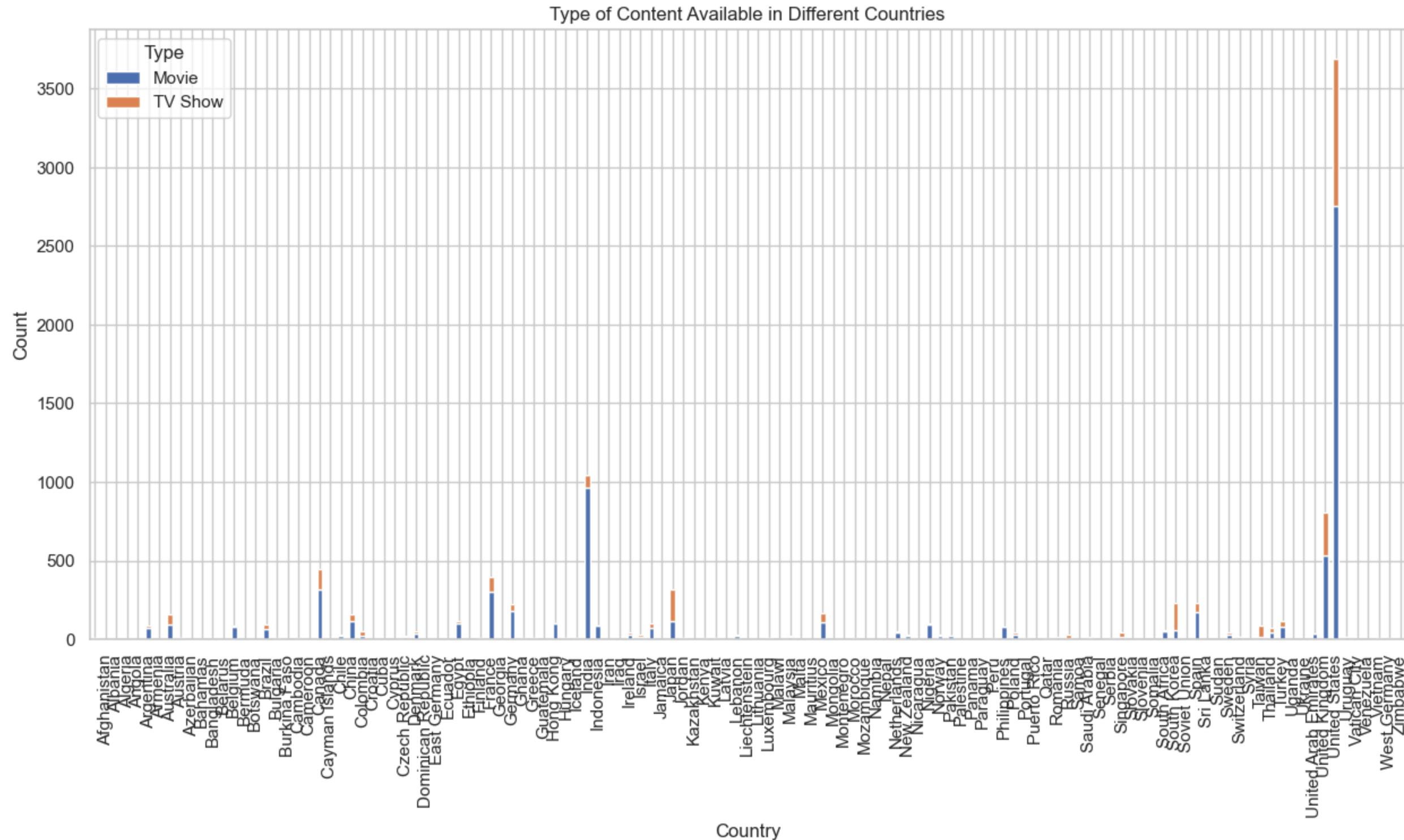
Out[668...]

country	type	Movie	TV Show
	6	1	
<b>Afghanistan</b>	1	0	
<b>Albania</b>	1	0	
<b>Algeria</b>	3	0	
<b>Angola</b>	1	0	
...	...	...	
<b>Vatican City</b>	1	0	
<b>Venezuela</b>	4	0	
<b>Vietnam</b>	7	0	
<b>West Germany</b>	3	2	
<b>Zimbabwe</b>	3	0	

123 rows × 2 columns

In [669...]

```
# Plot the data
pivot_df.plot(kind='bar', stacked=True, figsize=(15, 7))
plt.title('Type of Content Available in Different Countries')
plt.xlabel('Country')
plt.ylabel('Count')
plt.legend(title='Type')
plt.show()
```



In [673...]

```
# Group by 'country' and count the total content
total_content_by_country = data_country['country'].value_counts().reset_index()
```

```
total_content_by_country.columns = ['country', 'type']

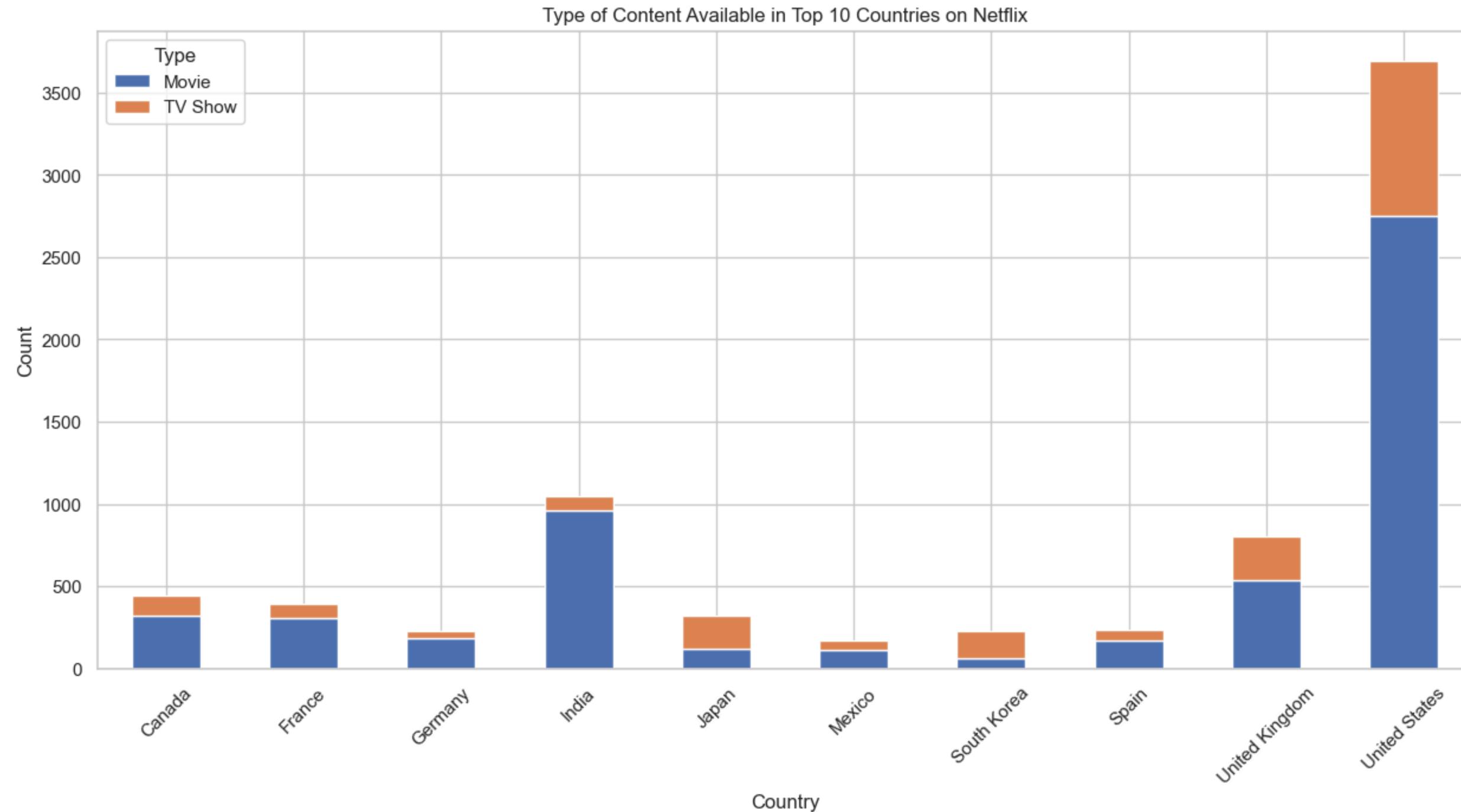
In [674...]: # Sort and select the top 10 countries
top_10_countries = total_content_by_country.head(10)

In [676...]: # Filter the original DataFrame to include only the top 10 countries
top_10_countries_list = top_10_countries['country'].tolist()
filtered_df = data_country[data_country['country'].isin(top_10_countries_list)]

In [677...]: # Group by 'country' and 'type' and count the occurrences
content_by_country_type = filtered_df.groupby(['country', 'type']).size().reset_index(name='count')

In [678...]: # Pivot the data for better visualization
pivot_df = content_by_country_type.pivot(index='country', columns='type', values='count').fillna(0)

In [679...]: # Plot the data
pivot_df.plot(kind='bar', stacked=True, figsize=(15, 7))
plt.title('Type of Content Available in Top 10 Countries on Netflix')
plt.xlabel('Country')
plt.ylabel('Count')
plt.legend(title='Type')
plt.xticks(rotation=45)
plt.show()
```



```
In [680]: #United States is having more content of Movies/TV Shows  
#India is having second top content of Movies/TV Shows
```

```
In [684]: movie_data.head()
```

Out[684...]

	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	Documentaries	As her father nears the end of his life, filmm...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	Nan	September 24, 2021	2021	PG	91	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	2021	PG-13	104	Comedies, Dramas	A woman adjusting to life after a loss contend...
12	s13	Movie	Je Suis Karl	Christian Schwochow	Luna Wedler, Jannis Niewöhner, Milan Peschel, ...	Germany, Czech Republic	September 23, 2021	2021	TV-MA	127	Dramas, International Movies	After most of her family is murdered in a terr...

In [687...]

```
# Extract the relevant years (Last 30 years)
current_year = pd.to_datetime('now').year
start_year = current_year - 30
filtered_movies_df = movie_data[movie_data['release_year'] >= start_year]
filtered_movies_df.head()
```

Out[687...]

	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	Documentaries	As her father nears the end of his life, filmm...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	Nan	September 24, 2021	2021	PG	91	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	2021	PG-13	104	Comedies, Dramas	A woman adjusting to life after a loss contend...
12	s13	Movie	Je Suis Karl	Christian Schwochow	Luna Wedler, Jannis Niewöhner, Milan Peschel, ...	Germany, Czech Republic	September 23, 2021	2021	TV-MA	127	Dramas, International Movies	After most of her family is murdered in a terr...
13	s14	Movie	Confessions of an Invisible Girl	Bruno Garotti	Klara Castanho, Lucca Picon, Júlia Gomes, Marc...	Nan	September 22, 2021	2021	TV-PG	91	Children & Family Movies, Comedies	When the clever but socially-awkward Tetê join...

In [689...]

```
# Group by 'release_year' and count the number of movies released each year
movies_per_year = filtered_movies_df.groupby('release_year').size().reset_index(name='count')
movies_per_year
```

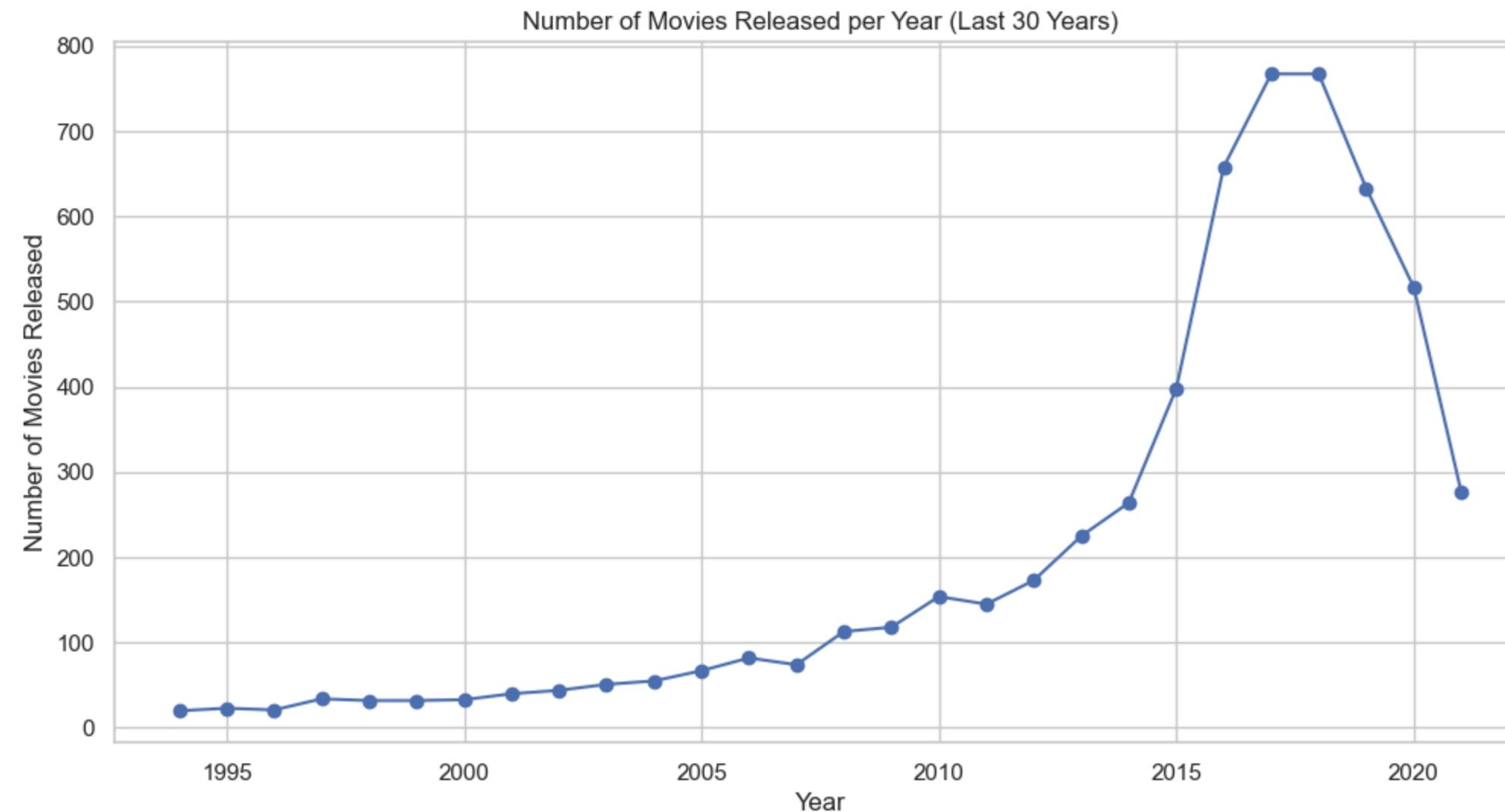
Out[689...]

	release_year	count
0	1994	20
1	1995	23
2	1996	21
3	1997	34
4	1998	32
5	1999	32
6	2000	33
7	2001	40
8	2002	44
9	2003	51
10	2004	55
11	2005	67
12	2006	82
13	2007	74
14	2008	113
15	2009	118
16	2010	154
17	2011	145
18	2012	173
19	2013	225
20	2014	264
21	2015	398
22	2016	658
23	2017	767
24	2018	767
25	2019	633
26	2020	517
27	2021	277

In [690...]

```
# Plot the data
plt.figure(figsize=(12, 6))
plt.plot(movies_per_year['release_year'], movies_per_year['count'], marker='o', linestyle='-' )
```

```
plt.title('Number of Movies Released per Year (Last 30 Years)')
plt.xlabel('Year')
plt.ylabel('Number of Movies Released')
plt.grid(True)
plt.show()
```



In [691]: #Number of Movies were keep on increasing till the year 2019, then its slitly getting reduced

In [692]: tvshow\_data.head()

Out[692...]

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	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	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	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	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	September 24, 2021	2021	TV-MA	1	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...

In [693...]

```
# Extract the relevant years (Last 30 years)
current_year = pd.to_datetime('now').year
start_year = current_year - 30
filtered_tvshow_df = tvshow_data[tvshow_data['release_year'] >= start_year]
filtered_tvshow_df.head()
```

Out[693...]

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2	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	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	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	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	September 24, 2021	2021	TV-MA	1	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...

In [694...]

```
# Group by 'release_year' and count the number of TV Show released each year
tvshow_per_year = filtered_tvshow_df.groupby('release_year').size().reset_index(name='count')
tvshow_per_year
```

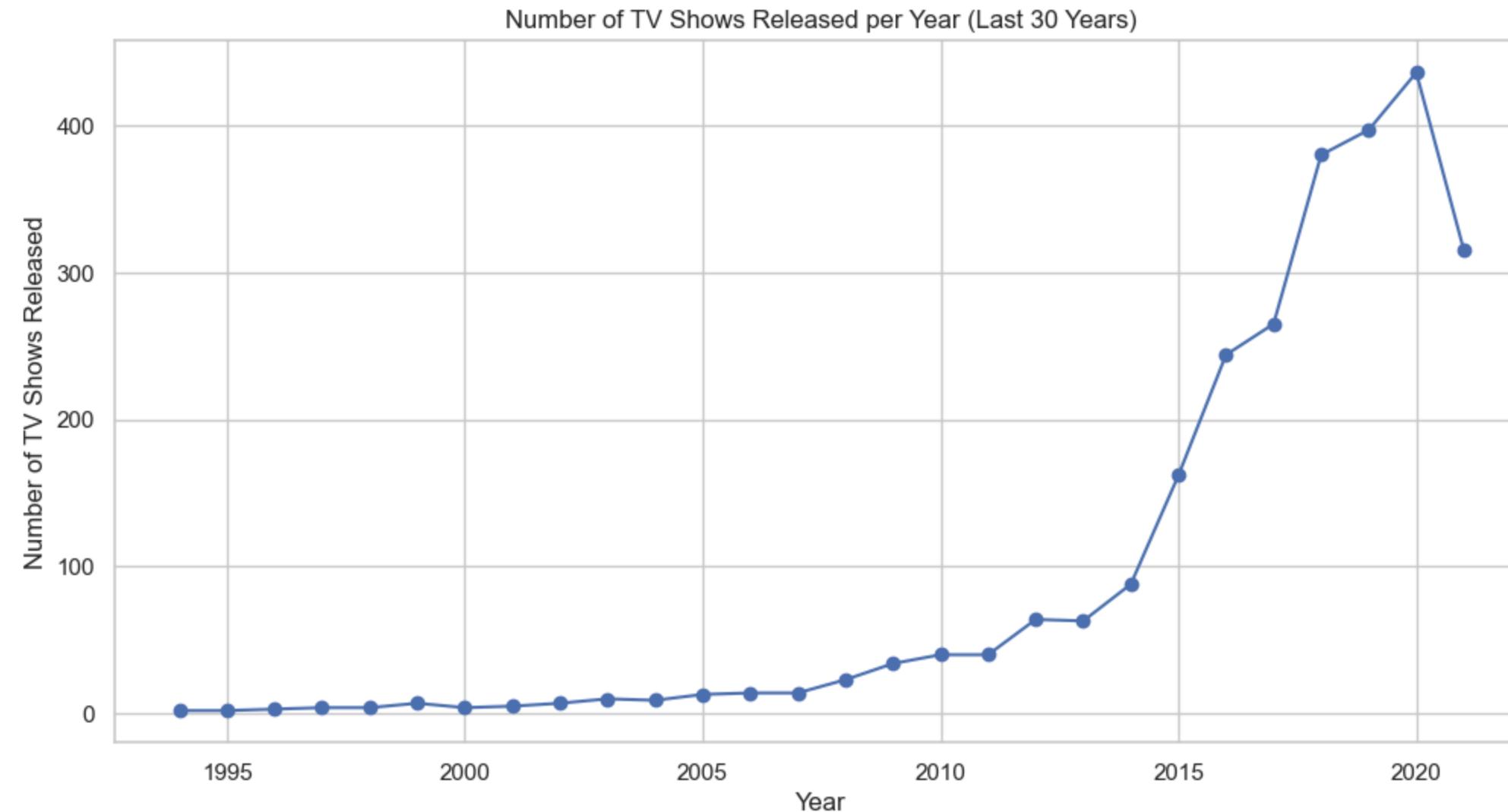
Out[694...]

	release_year	count
0	1994	2
1	1995	2
2	1996	3
3	1997	4
4	1998	4
5	1999	7
6	2000	4
7	2001	5
8	2002	7
9	2003	10
10	2004	9
11	2005	13
12	2006	14
13	2007	14
14	2008	23
15	2009	34
16	2010	40
17	2011	40
18	2012	64
19	2013	63
20	2014	88
21	2015	162
22	2016	244
23	2017	265
24	2018	380
25	2019	397
26	2020	436
27	2021	315

In [695...]

```
# Plot the data
plt.figure(figsize=(12, 6))
plt.plot(tvshow_per_year['release_year'], tvshow_per_year['count'], marker='o', linestyle='-' )
```

```
plt.title('Number of TV Shows Released per Year (Last 30 Years)')
plt.xlabel('Year')
plt.ylabel('Number of TV Shows Released')
plt.grid(True)
plt.show()
```



In [696...]: #Number of TV Shows were keep on increasing till the year 2020, then its slitly getting reduced

In [697...]: data\_date\_added.head()

Out[69]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	year_added	year	month	date
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90	Documentaries	As her father nears the end of his life, filmm...	2021.0	2021	September	25
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	2021.0	2021	September	24
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	2021.0	2021	September	24
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	1	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...	2021.0	2021	September	24
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...	2021.0	2021	September	24

In [710]:

```
data_date_added["type"].value_counts()
```

Out[710]:

type	count
Movie	6131
TV Show	2666
Name: count, dtype: int64	

In [711]:

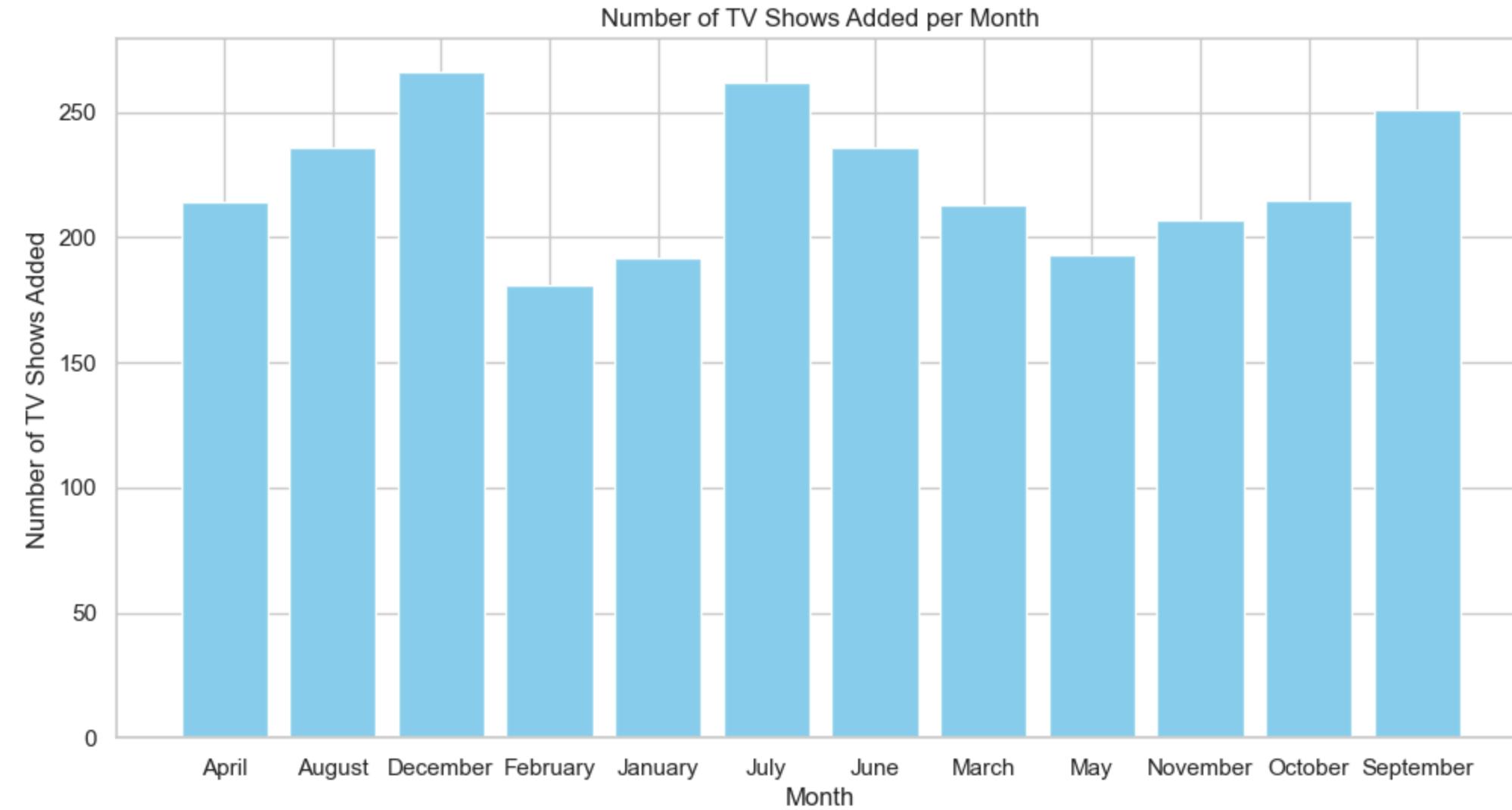
```
# Group by month and count the number of TV shows added
tv_shows_per_month = data_date_added[data_date_added["type"] == "TV Show"].groupby('month').size().reset_index(name='count')
```

In [712]:

```
# Group by month and count the number of Movie added
movies_per_month = data_date_added[data_date_added["type"] == "Movie"].groupby('month').size().reset_index(name='count')
```

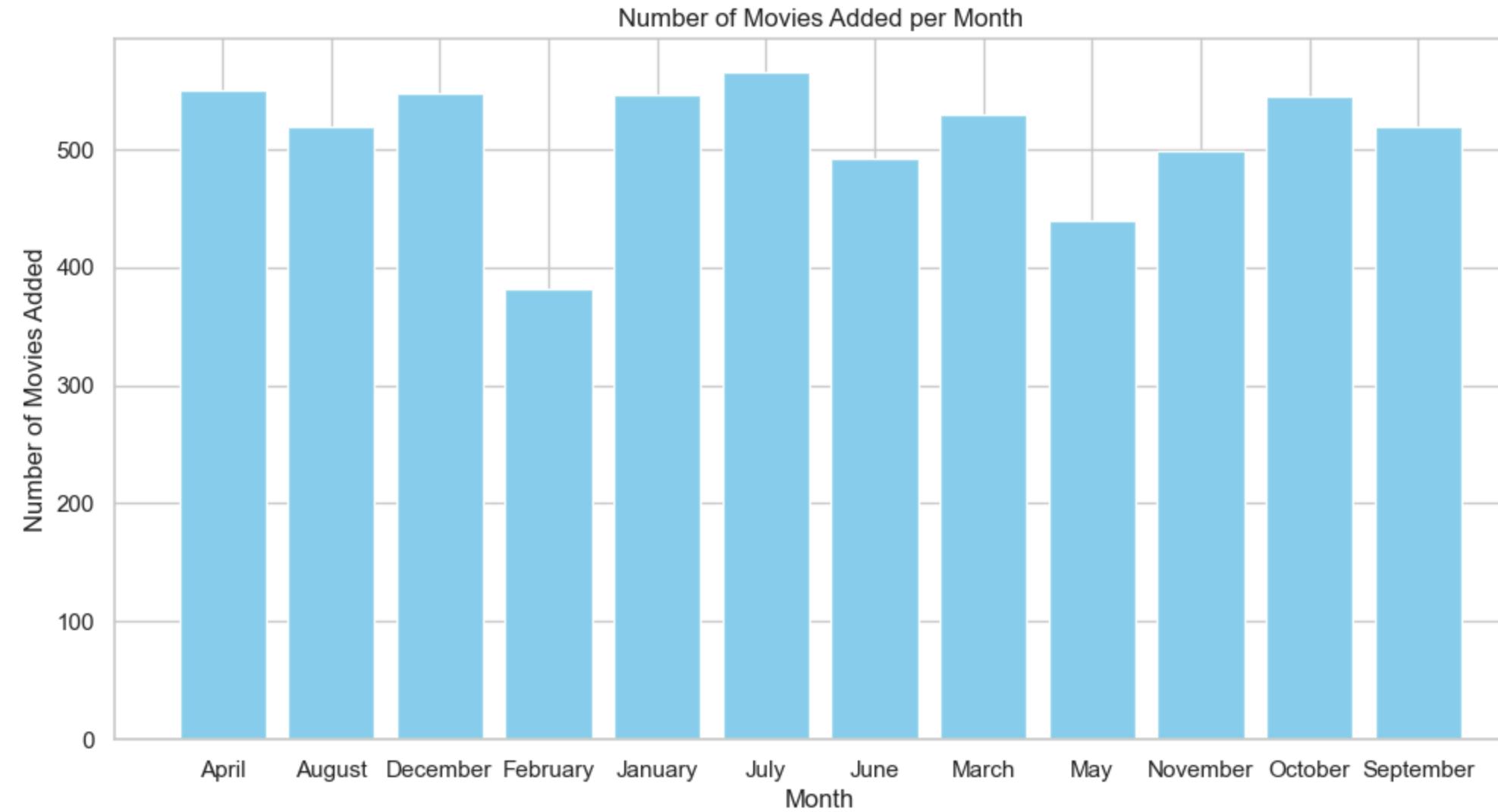
In [713]:

```
# Plot the data
plt.figure(figsize=(12, 6))
plt.bar(tv_shows_per_month['month'], tv_shows_per_month['count'], color='skyblue')
plt.title('Number of TV Shows Added per Month')
plt.xlabel('Month')
plt.ylabel('Number of TV Shows Added')
plt.grid(True)
plt.show()
```



In [714...]

```
# Plot the data
plt.figure(figsize=(12, 6))
plt.bar(movies_per_month['month'], movies_per_month['count'], color='skyblue')
plt.title('Number of Movies Added per Month')
plt.xlabel('Month')
plt.ylabel('Number of Movies Added')
plt.grid(True)
plt.show()
```



```
In [796...]: #Added year in Movies/TV Show with count are almost equal but,
#Most of the TV Shows added in Dec, July, Sept
#Most of the Movies added in April, Dec, Jan, July, Oct
```

```
In [716...]: data.shape
```

```
Out[716...]: (8807, 13)
```

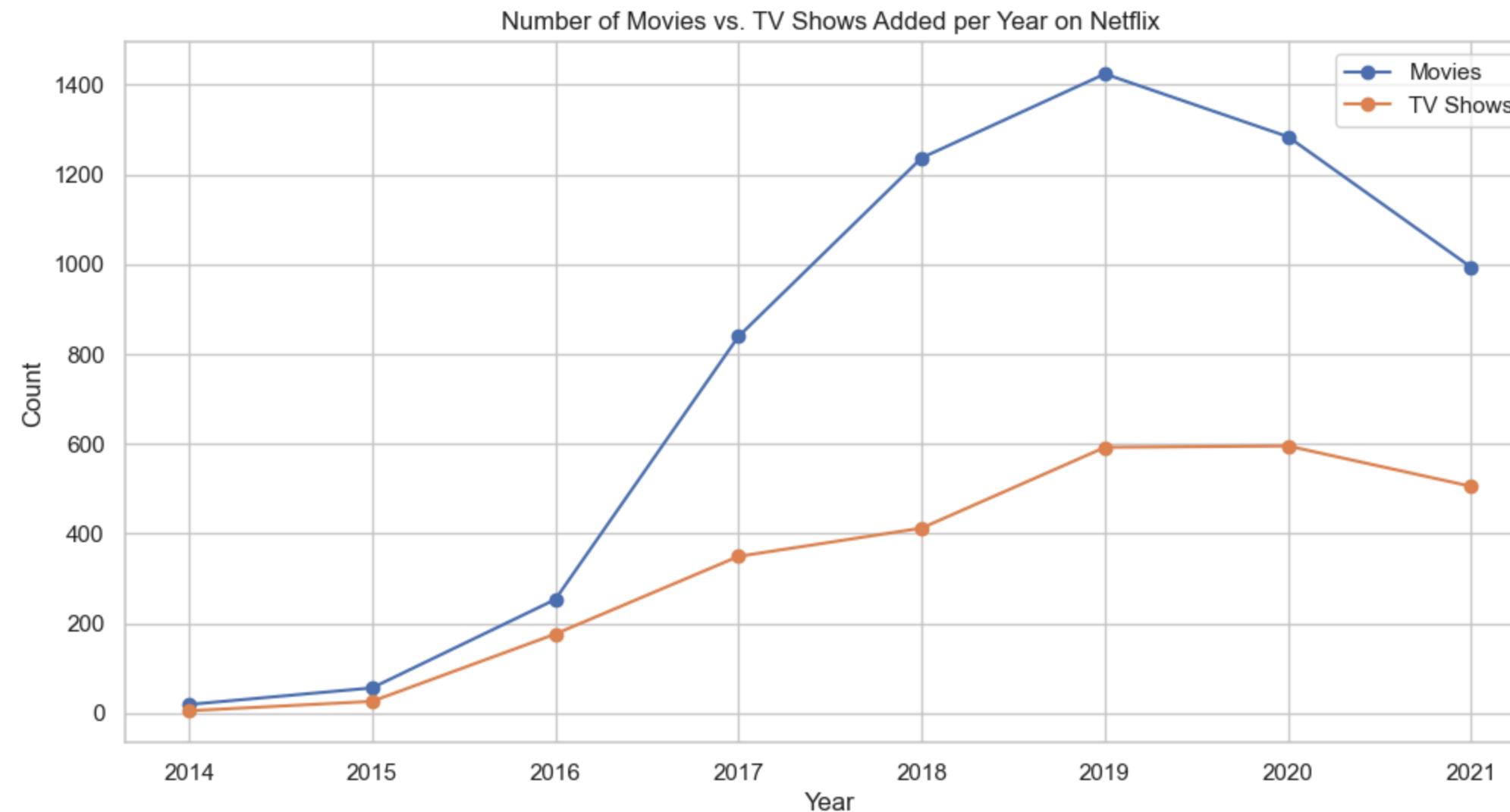
```
In [717...]: # Filter the data for the last few years (e.g., last 10 years)
recent_years_df = data[data['year_added'] >= (pd.to_datetime('now').year - 10)]
```

```
In [718...]: # Group by 'year_added' and 'type' and count the occurrences
content_by_year_type = recent_years_df.groupby(['year_added', 'type']).size().reset_index(name='count')
```

```
In [719...]: # Pivot the data for better visualization
pivot_df = content_by_year_type.pivot(index='year_added', columns='type', values='count').fillna(0)
```

```
In [720...]: # Plot the data
plt.figure(figsize=(12, 6))
plt.plot(pivot_df.index, pivot_df['Movie'], marker='o', linestyle='-', label='Movies')
plt.plot(pivot_df.index, pivot_df['TV Show'], marker='o', linestyle='-', label='TV Shows')
```

```
plt.title('Number of Movies vs. TV Shows Added per Year on Netflix')
plt.xlabel('Year')
plt.ylabel('Count')
plt.legend()
plt.grid(True)
plt.show()
```



```
In [721...]: #Yes, Netflix has more focus on TV Shows than movies in recent years
```

```
In [724...]: data_date_added.shape
```

```
Out[724...]: (8797, 16)
```

```
In [726...]: correlation = data_date_added[["release_year", "year_added"]]
```

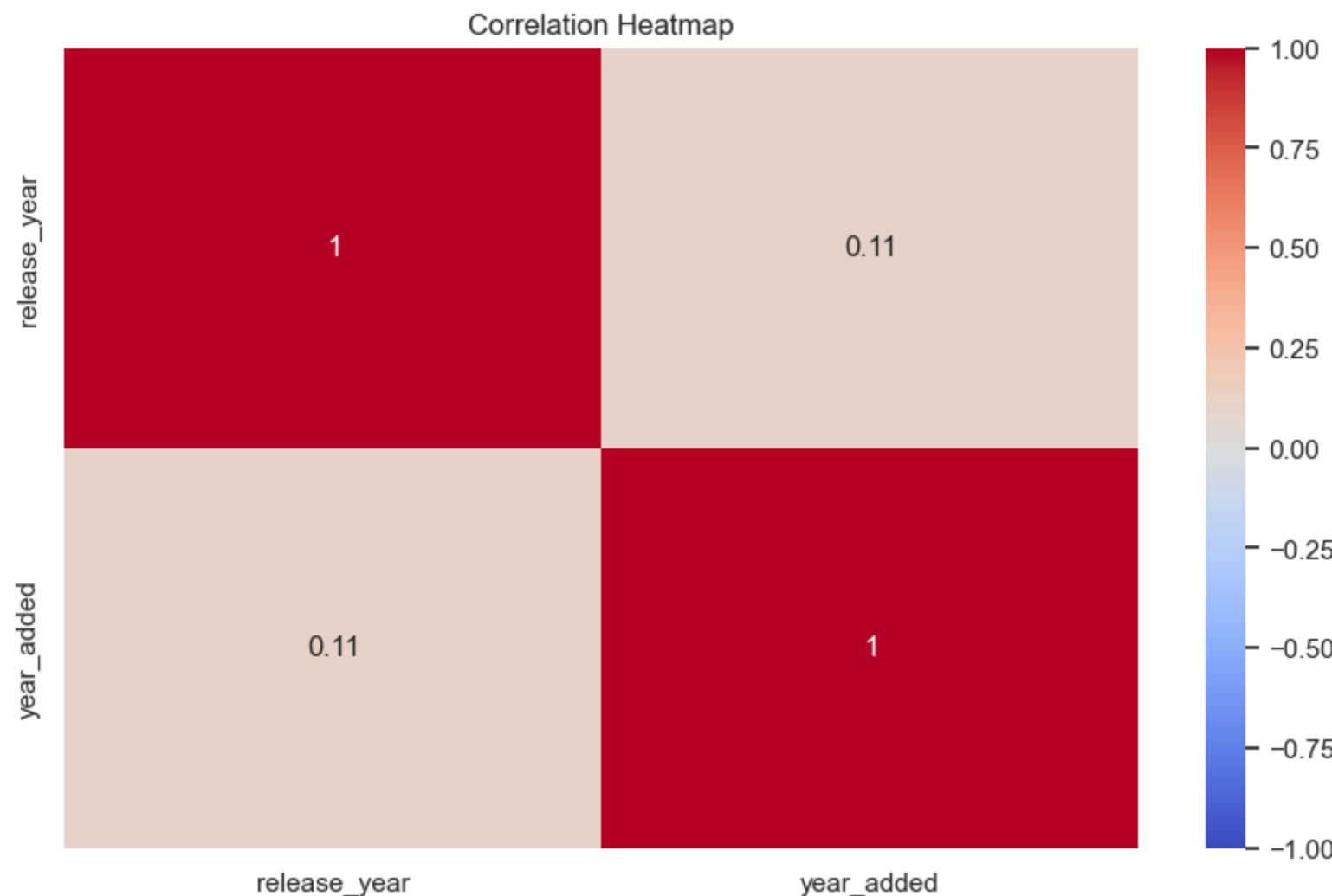
```
In [728...]: # Calculate the correlation matrix
correlation_matrix = correlation.corr()
correlation_matrix
```

Out[728...]

	release_year	year_added
release_year	1.000000	0.111531
year_added	0.111531	1.000000

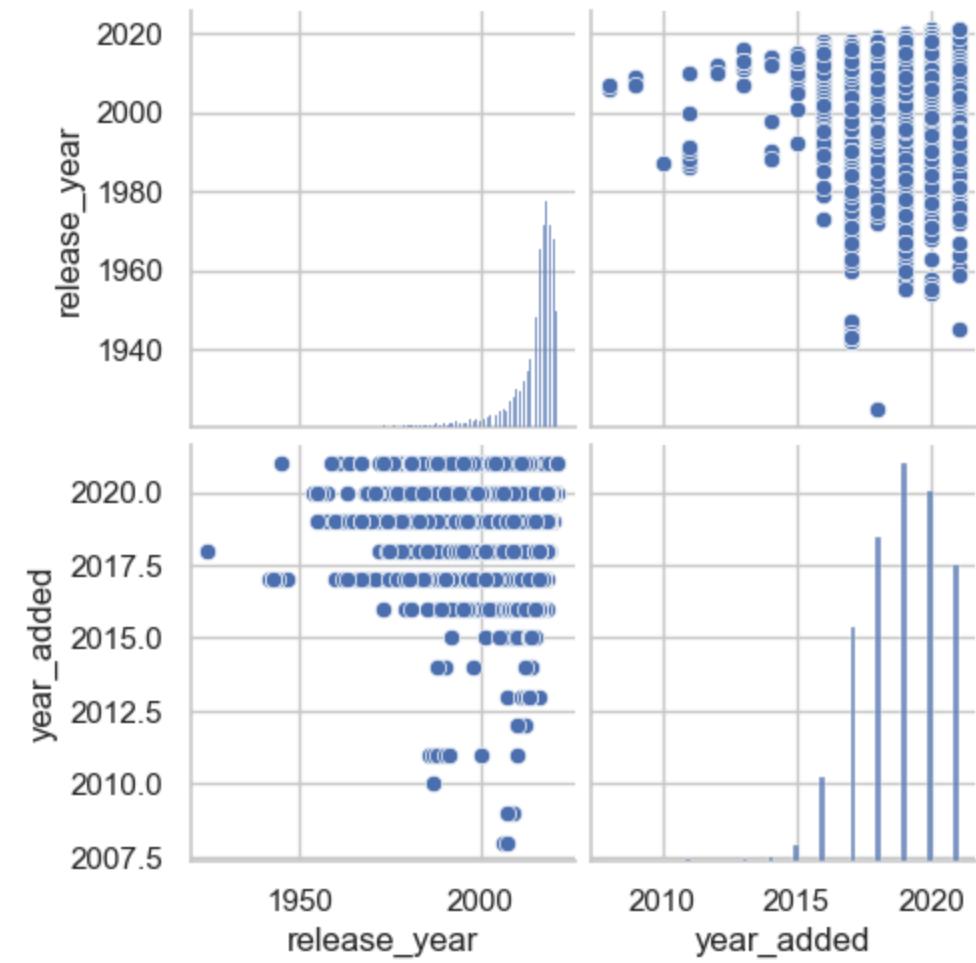
In [729...]

```
# Plot the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap')
plt.show()
```



In [731...]

```
# Plot pair plots
sns.pairplot(correlation)
plt.show()
```



In [732...]

correlation

Out[732...]

release\_year year\_added

<b>0</b>	2020	2021.0
<b>1</b>	2021	2021.0
<b>2</b>	2021	2021.0
<b>3</b>	2021	2021.0
<b>4</b>	2021	2021.0
...	...	...
<b>8802</b>	2007	2019.0
<b>8803</b>	2018	2019.0
<b>8804</b>	2009	2019.0
<b>8805</b>	2006	2020.0
<b>8806</b>	2015	2019.0

8797 rows × 2 columns

```
In [738... correlation_temp = correlation.copy()
```

```
In [739... correlation_temp["added_release"] = correlation_temp["year_added"] - correlation_temp["release_year"]
correlation_temp["added_release"]
```

```
Out[739... 0      1.0
1      0.0
2      0.0
3      0.0
4      0.0
...
8802    12.0
8803    1.0
8804    10.0
8805    14.0
8806    4.0
Name: added_release, Length: 8797, dtype: float64
```

```
In [742... correlation_temp["added_release"].value_counts().head(10)
```

```
Out[742... added_release
0.0    3241
1.0    1585
2.0    714
3.0    491
4.0    367
5.0    261
6.0    251
7.0    187
8.0    185
9.0    161
Name: count, dtype: int64
```

```
In [743... correlation_temp.shape
```

```
Out[743... (8797, 3)
```

```
In [744... #Out of 8797 Movies/TV Shows 3241 added in the netflix in the same year it released
#1585 added after 1 year of release
#714 added after 2 years of release
```

```
In [748... #Word Cloud
from wordcloud import WordCloud
plt.subplots(figsize=(25,15))
wordcloud = WordCloud(
    background_color='White',
    width=1920,
    height=1080
).generate(' '.join(data.description))

plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



#These are some common words in the description,

In [797]:

#Solution in the Netflix Data

```
#1. Uploaded the Netflix.csv into a DataFrame and Analysed the Columns, DType, Shape
#2. Null values, Duplicates and invalid entries are analysed and treated based on the needs
#3. Converted some columns into category type
#4. Visualised the Missing values
```

#5. Count, Unique, Min, Max, Mean, Median, Mode are analysed  
#6. Modified the DF bases on the usecases and plotted the graphs  
#7. Analysed and noted the Business Insights and Recommendations

In [798... #Business Insights

#1. Netflix hosts 6,131 movies and 2,676 TV shows, totaling 8,807 titles. Movies constitute 30.38% of the content, while TV shows make up 69.62%.

#2. There are missing values in the fields for director, cast, country, date added, rating, and duration. These are addressed by either removal or replacement with mode values, depending on the use

#3. The DataFrame contains no duplicate entries.

#4. The most frequent rating is TV-MA, followed by TV-14. Ratings like NC-17 and UR are the least common. The distribution is similar across types, but ratings like TV-Y7 and TV-Y are more prevalent

#5. There are 14 movie rating categories and 9 TV show rating categories.

#6. Most movies have a duration of 90-110 minutes. The shortest movie is 3 minutes long, the longest is 312 minutes, and the mean length is approximately 99.57 minutes.

#7. Most TV shows have 1-2 seasons. The shortest TV show has 1 season, the longest has 17 seasons, and the average length is approximately 1.76 seasons.

#8. The majority of movies were released in 2017 and 2018, while most TV shows were released in 2020.

#9. The top three countries for movies are the United States, India, and the United Kingdom. For TV shows, the top three are the United States, the United Kingdom, and Japan. Overall, the top three

#10. The top director for movies is Rajiv Chilaka, The top directors for TV shows are Alastair Fothergill and Ken Burns. Overall, the top director in overall shows (Movies/TV Shows) is Rajiv Chilak

#11. The top cast member for movies is Anupam Kher. For TV shows, it is Takahiro Sakurai. Overall, the top cast member is Anupam Kher

#12. The top three genres for movies are International Movies, Dramas, and Comedies. For TV shows, they are International TV Shows, TV Dramas, and TV Comedies. Overall, the top three genres are Int

#13. Most movies and TV shows were added in 2019. The majority are added on the 1st or 15th of each month, with Friday being the most common day for new releases.

#14. The United States has the most content, followed by India.

#15. The number of movies increased until 2019, after which it slightly declined. The number of TV shows increased until 2020, then slightly decreased.

#16. The number of movies and TV shows added each year is almost equal, but most TV shows are added in December, July, and September, while most movies are added in April, December, January, July,

#17. There is a noticeable trend in the number of TV shows and movies added to Netflix over the years. It indicates a strategic focus on TV shows. Netflix may be investing more in TV shows to cater

#18. The number of TV shows and movies added each month shows peaks during certain times of the year, such as around holidays or summer months. Netflix likely schedules major releases during period

#19. Different countries have varying amounts of content available, with some countries having a higher number of TV shows or movies. Netflix tailors its content library based on regional preference

#20. The number of movies released per year has shown increasing over the last 20-30 years, with certain periods experiencing higher releases. Out of 8797 Movies/TV Shows 3241 added in the netflix

#21. Correlation analysis between features 'release\_year', 'year\_added' reveal patterns such as older movies are added to the platform during specific times.

In [799... #Recommendations

#1. Continue investing in high-quality TV shows, as they appear to drive higher engagement and viewer retention.

#2. Plan major content releases around holidays and summer months when viewership is likely to be higher. This can help maximize audience engagement and attract new subscribers.

#3. Expand the content library based on regional preferences to increase global market penetration.

- #4. Maintain a balanced mix of new releases and older content. This ensures that there is always something new for regular viewers while also attracting those interested in classic or older movies
- #5. Promote good directors and cast in releases through targeted campaigns to generate and attract more viewers.

In [ ]: