Import Needed Libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Read Data and Get Some Information

In [5]:

```
df = pd.read_csv("E://Project//netflix1.csv")
df
```

Out[5]:

	show_id	type	title	director	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	9/25/2021	2020	PG-13
1	s3	TV Show	Ganglands	Julien Leclercq	France	9/24/2021	2021	TV-MA
2	s6	TV Show	Midnight Mass	Mike Flanagan	United States	9/24/2021	2021	TV-MA
3	s14	Movie	Confessions of an Invisible Girl	Bruno Garotti	Brazil	9/22/2021	2021	TV-PG
4	s8	Movie	Sankofa	Haile Gerima	United States	9/24/2021	1993	TV-MA
...	...	...	...	...	...	...	...	...
8785	s8797	TV Show	Yunus Emre	Not Given	Turkey	1/17/2017	2016	TV-PG
8786	s8798	TV Show	Zak Storm	Not Given	United States	9/13/2018	2016	TV-Y7
8787	s8801	TV Show	Zindagi Gulzar Hai	Not Given	Pakistan	12/15/2016	2012	TV-PG
8788	s8784	TV Show	Yoko	Not Given	Pakistan	6/23/2018	2016	TV-Y
8789	s8786	TV Show	YOM	Not Given	Pakistan	6/7/2018	2016	TV-Y7

8790 rows × 10 columns

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8790 non-null   object
 1   type          8790 non-null   object
 2   title         8790 non-null   object
 3   director      8790 non-null   object
 4   country       8790 non-null   object
 5   date_added    8790 non-null   object
 6   release_year  8790 non-null   int64 
 7   rating        8790 non-null   object
 8   duration      8790 non-null   object
 9   listed_in     8790 non-null   object
dtypes: int64(1), object(9)
memory usage: 686.8+ KB
```

In [9]:

```
missing_values = df.isnull().sum()
missing_values
```

Out[9]:

```
show_id         0
type            0
title           0
director        0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
dtype: int64
```

In [11]:

```
duplicate = df.duplicated().sum()
duplicate
```

Out[11]:

```
0
```

In [13]:

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

Out[13]:

	type	title	director	country	date_added	release_year	rating	duration	
0	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	9/25/2021	2020	PG-13	90 min	D
1	TV Show	Ganglands	Julien Leclercq	France	9/24/2021	2021	TV-MA	1 Season	
2	TV Show	Midnight Mass	Mike Flanagan	United States	9/24/2021	2021	TV-MA	1 Season	T
3	Movie	Confessions of an Invisible Girl	Bruno Garotti	Brazil	9/22/2021	2021	TV-PG	91 min	F
4	Movie	Sankofa	Haile Gerima	United States	9/24/2021	1993	TV-MA	125 min	

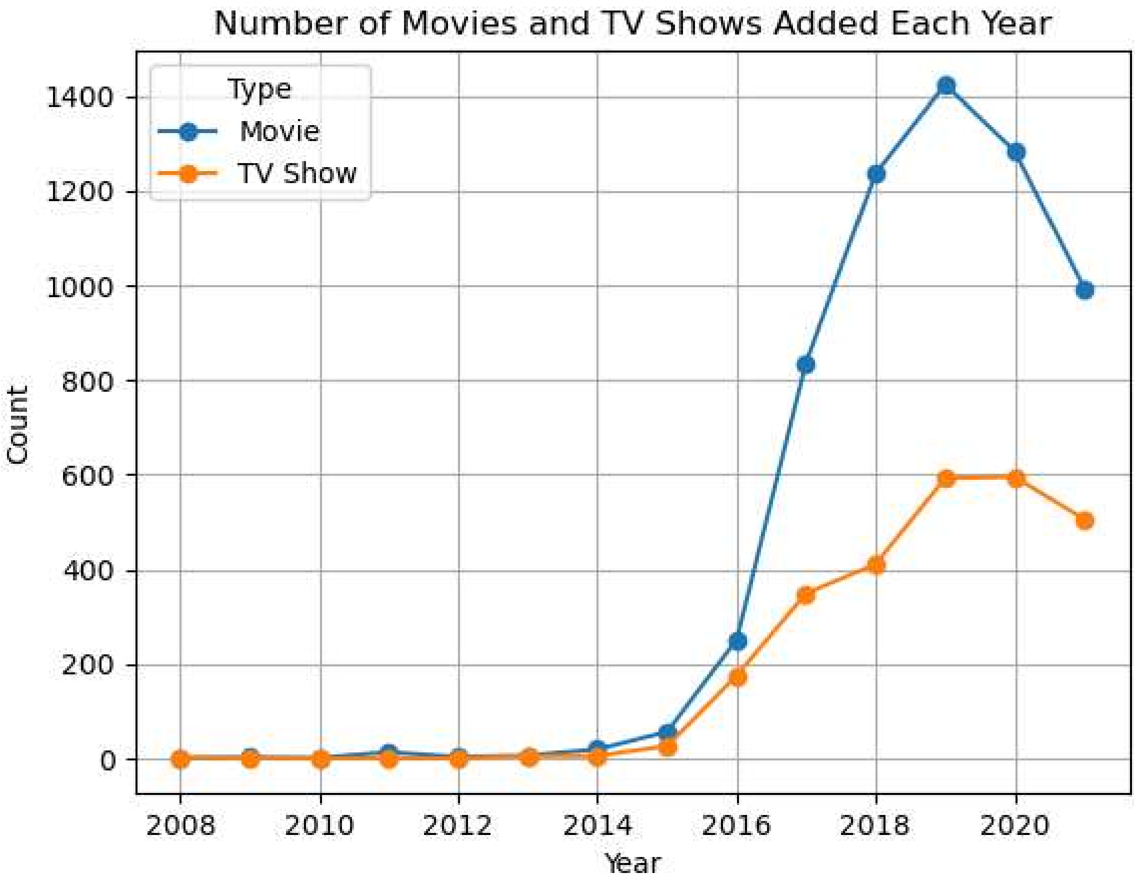
Visualizations

Compare Between Movies and TV Shows By Time

In [15]:

```
df['date_added'] = pd.to_datetime(df['date_added'])
df['year_added'] = df['date_added'].dt.year
content_per_year = df.groupby(['year_added', 'type']).size().unstack().fillna(0)

plt.figure(figsize=(14, 7))
content_per_year.plot(kind='line', marker='o')
plt.title('Number of Movies and TV Shows Added Each Year')
plt.xlabel('Year')
plt.ylabel('Count')
plt.legend(title='Type')
plt.grid(True)
plt.show()
```

```
<Figure size 1400x700 with 0 Axes>
```

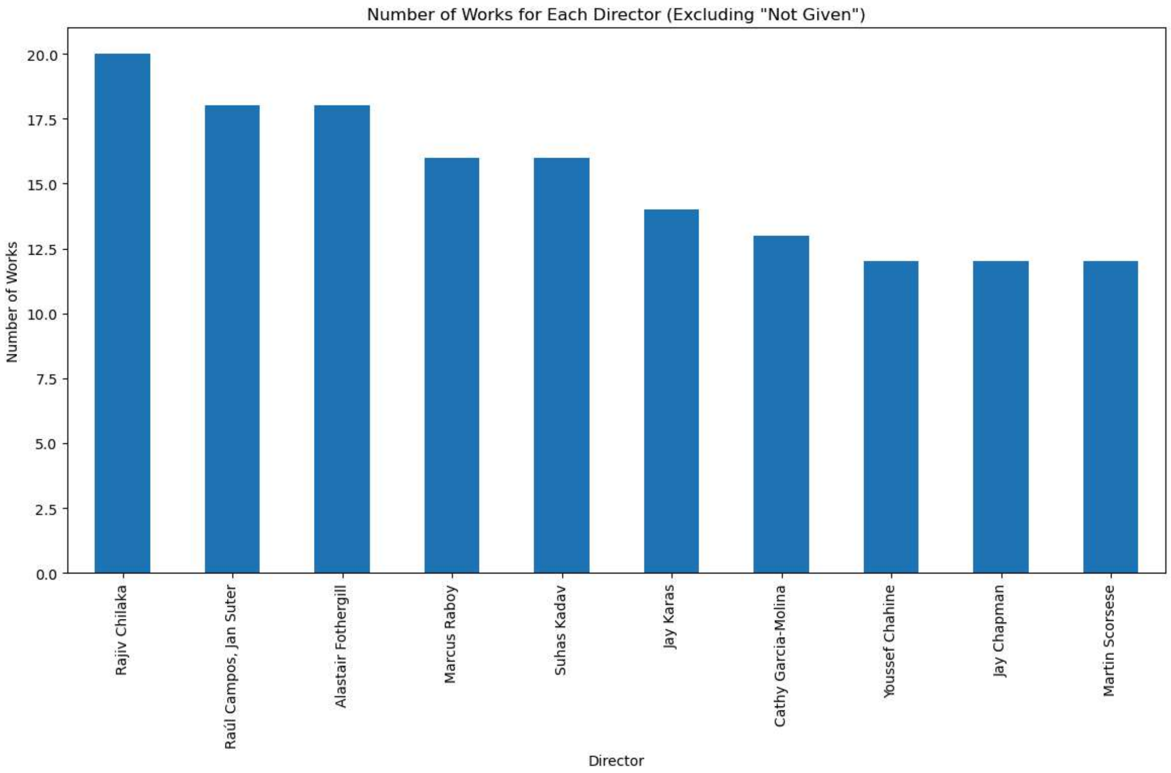


Know The Top 10 Directors

```
In [17]: director_counts = df['director'].value_counts()
         director_counts
```

```
Out[17]: director
         Not Given                          2588
         Rajiv Chilaka                        20
         Alastair Fothergill                  18
         Raúl Campos, Jan Suter               18
         Suhas Kadav                          16
                                            ... 
         Matt D'Avella                         1
         Parthiban                             1
         Scott McAboy                          1
         Raymie Muzquiz, Stu Livingston        1
         Mozez Singh                           1
         Name: count, Length: 4528, dtype: int64
```

```
In [19]: director_filtered = df[df['director'] != 'Not Given']
         director_counts = director_filtered['director'].value_counts().head(10)

         plt.figure(figsize=(14, 7))
         director_counts.plot(kind='bar')
         plt.title('Number of Works for Each Director (Excluding "Not Given")')
         plt.xlabel('Director')
         plt.ylabel('Number of Works')
         plt.xticks(rotation=90)
         plt.show()
```



The percentages between Movies and TV shows

In [21]:

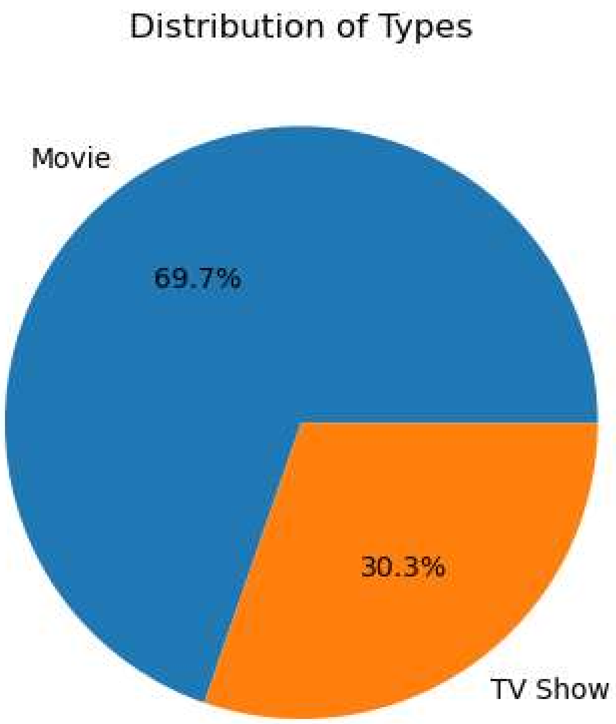
```
type_counts = df['type'].value_counts()
percentages = (type_counts / type_counts.sum()) * 100
percentages
```

Out[21]:

```
type
Movie      69.692833
TV Show    30.307167
Name: count, dtype: float64
```

In [23]:

```
type_counts = df['type'].value_counts()

plt.pie(type_counts, labels=type_counts.index, autopct='%1.1f%%')
plt.title('Distribution of Types')
plt.show()
```



The Best 50 Countries with Number of Movies and TV Shows

In [25]:

```
country_counts = df['country'].value_counts()
country_df = country_counts.reset_index()
country_df.columns = ['country', 'count']
country_df = country_df.sort_values(by='count', ascending=False).head(50)
country_df
```

Out[25]:

	country	count
0	United States	3240
1	India	1057
2	United Kingdom	638
3	Pakistan	421
4	Not Given	287
5	Canada	271
6	Japan	259
7	South Korea	214
8	France	213
9	Spain	182
10	Mexico	138
11	Egypt	123
12	Australia	114
13	Turkey	112
14	Nigeria	105
15	Germany	104
16	China	100
17	Brazil	88
18	Taiwan	86
19	Indonesia	86
20	Philippines	83
21	Hong Kong	79
22	Argentina	76
23	Italy	66
24	Thailand	66
25	Colombia	42
26	South Africa	42
27	Denmark	30
28	Poland	30
29	Netherlands	28
30	Ireland	28
31	Singapore	28
32	Lebanon	24

	country	count
33	Malaysia	23
34	Israel	22
35	Chile	22
36	Norway	21
37	United Arab Emirates	21
38	Russia	19
39	Sweden	19
40	Belgium	18
41	New Zealand	17
42	Saudi Arabia	12
43	Romania	10
44	Austria	9
45	Uruguay	9
46	Switzerland	8
47	Vietnam	7
48	Kuwait	7
49	Peru	6

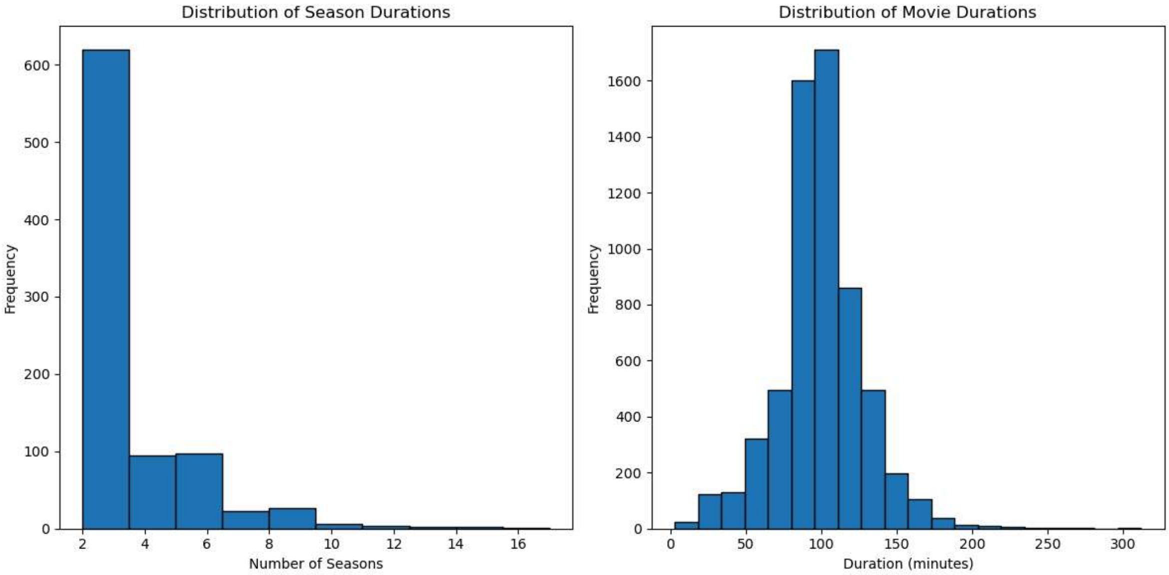
In [ ]:

```
plt.figure(figsize=(20, 6))
plt.plot(country_df['country'], country_df['count'], marker='o', color='purple',
plt.xticks(rotation=90)
plt.title('Top 50 Countries by Count')
plt.xlabel('Country')
plt.ylabel('Count')
#plt.grid(True)
plt.tight_layout()
plt.show()
```

Distribution of Durations

In [27]:

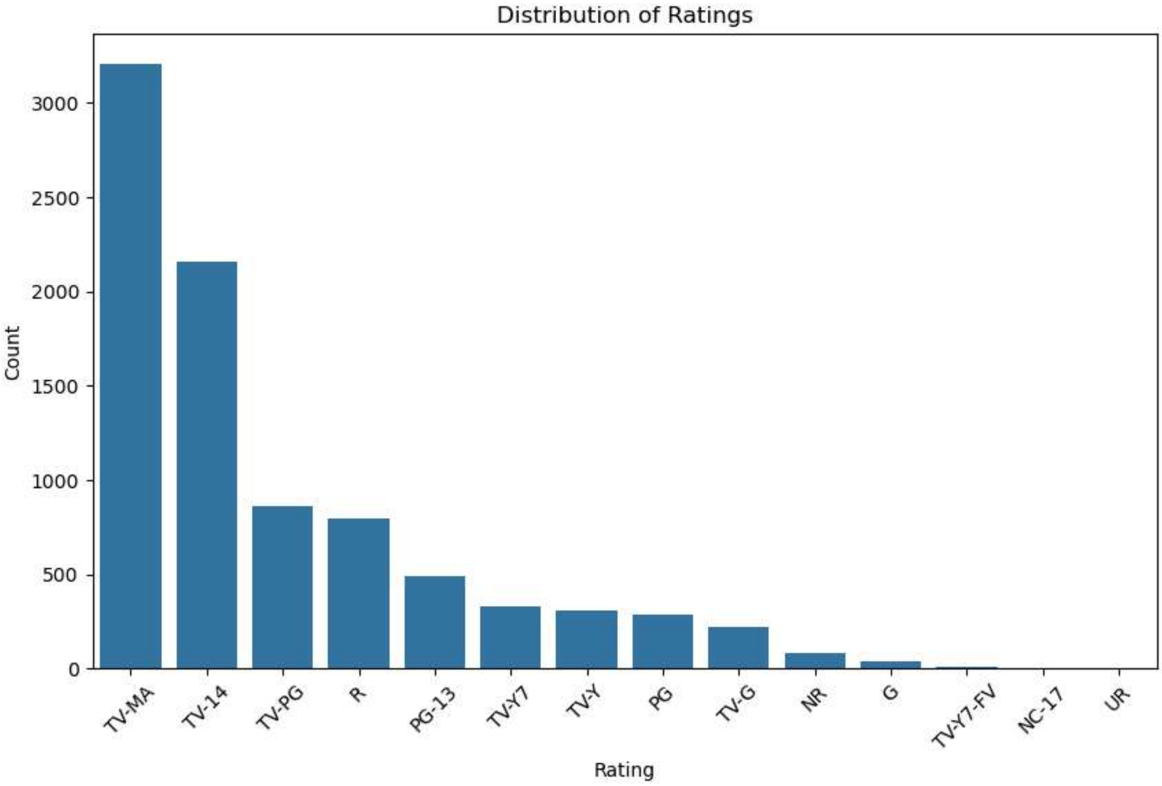
```
df[['value', 'type']] = df['duration'].str.split(' ', expand=True)
df['value'] = df['value'].astype(int)

plt.figure(figsize=(12, 6))

# Plot for Seasons
plt.subplot(1, 2, 1)
plt.hist(df[df['type'] == 'Seasons']['value'], bins=10, edgecolor='black')
plt.title('Distribution of Season Durations')
plt.xlabel('Number of Seasons')
plt.ylabel('Frequency')

# Plot for Movies
plt.subplot(1, 2, 2)
plt.hist(df[df['type'] == 'min']['value'], bins=20, edgecolor='black')
plt.title('Distribution of Movie Durations')
plt.xlabel('Duration (minutes)')
plt.ylabel('Frequency')

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



Rating Figure Using Seaborn

In [29]:

```
plt.figure(figsize=(10, 6))
sns.countplot(x='rating', data=df, order=df['rating'].value_counts().index)
plt.title('Distribution of Ratings')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Categories of Movies and TV Shows by Plotly

```
In [ ]: df['listed_in'] = df['listed_in'].str.split(', ')
        df_exploded = df.explode('listed_in')
        genre_counts = df_exploded['listed_in'].value_counts()

        fig = px.bar(genre_counts, x=genre_counts.index, y=genre_counts.values,
                     labels={'x': 'listed_in', 'y': 'Count'},
                     title='Categories')
        fig.show()
```