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

```
import pandas as pd
from google.colab import files
```

```
# Uploading  files of data set here
uploaded = files.upload()
# Read the CSV file
import io
data = pd.read_csv(io.BytesIO(uploaded['amazon_prime_titles.csv']))
# Display the DataFrame
data
```

Choose Files amazon_prime_titles.csv

- **amazon_prime_titles.csv**(text/csv) - 3972416 bytes, last modified: 10/12/2021 - 100% done

```
Saving amazon_prime_titles.csv to amazon_prime_titles.csv
```

	show_id	type	title	director	cast	country	date_added	release_yea
0	s1	Movie	The Grand Seduction	Don McKellar	Brendan Gleeson, Taylor Kitsch, Gordon Pinsent	Canada	March 30, 2021	201
1	s2	Movie	Take Care Good Night	Girish Joshi	Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar	India	March 30, 2021	201
2	s3	Movie	Secrets of Deception	Josh Webber	Tom Sizemore, Lorenzo Lamas, Robert LaSardo, R...	United States	March 30, 2021	201
3	s4	Movie	Pink: Staying True	Sonia Anderson	Interviews with: Pink, Adele, Beyoncé, Britney...	United States	March 30, 2021	201
4	s5	Movie	Monster Maker	Giles Foster	Harry Dean Stanton, Kieran O'Brien, George Cos...	United Kingdom	March 30, 2021	198
...	...	...	...	...	...	...	...	.
9663	s9664	Movie	Pride Of The Bowery	Joseph H. Lewis	Leo Gorcey, Bobby Jordan	NaN	NaN	194
9664	s9665	TV Show	Planet Patrol	NaN	DICK VOSBURGH, RONNIE STEVENS, LIBBY MORRIS, M...	NaN	NaN	201
9665	s9666	Movie	Outpost	Steve Barker	Ray Stevenson, Julian Wadham, Richard Brake, M...	NaN	NaN	200
9666	s9667	TV Show	Maradona: Blessed Dream	NaN	Esteban Recagno, Ezequiel Stremiz, Luciano Vit...	NaN	NaN	202
9667	s9668	Movie	Harry Brown	Daniel Barber	Michael Caine, Emily Mortimer, Joseph Gilgun, ...	NaN	NaN	201

9668 rows × 12 columns

Next steps: View recommended plots

```
data.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

```
data.describe().round(2)
```

	release_year
count	9668.00
mean	2008.34
std	18.92
min	1920.00
25%	2007.00
50%	2016.00
75%	2019.00
max	2021.00

```
data.duplicated().sum()
```

```
0
```

```
numeric_columns = data.select_dtypes(include=['number']).columns

# Fill NaN values with mean for numeric columns
data[numeric_columns] = data[numeric_columns].fillna(data[numeric_columns].mean())
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9668 entries, 0 to 9667
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       9668 non-null   object
 1   type          9668 non-null   object
 2   title         9668 non-null   object
 3   director      7585 non-null   object
 4   cast          8435 non-null   object
 5   country       672 non-null    object
 6   date_added    155 non-null    object
 7   release_year  9668 non-null   int64
 8   rating        9331 non-null   object
 9   duration      9668 non-null   object
 10  listed_in     9668 non-null   object
 11  description   9668 non-null   object
dtypes: int64(1), object(11)
memory usage: 906.5+ KB
```

```
data.isna().sum()
```

```
show_id           0
type              0
title             0
```

```
director        2083
cast            1233
country         8996
date_added      9513
release_year       0
rating           337
duration           0
listed_in          0
description        0
dtype: int64
```

```
data['cast'].fillna("Unknown", inplace=True)
```

```
data['director'].fillna("Unknown", inplace=True)
```

```
most_common_rating = data['rating'].mode()[0]
data['rating'].fillna(most_common_rating, inplace=True)
```

```
data_remove=['description']
data=data.drop(columns=data_remove)
```

```
data.isna().sum()
```

```
show_id            0
type               0
title              0
director           0
cast               0
country         8996
date_added      9513
release_year       0
rating             0
duration           0
listed_in          0
dtype: int64
```

```
data.sample(10)
```

	show_id	type	title	director	cast	country	date_added	release_year	r
7195	s7196	Movie	Preetam	Sijo Rocky	Pranav Raorane, Upendra Limaye, Nakshatra Medheka	NaN	NaN	2021	
9417	s9418	Movie	The Tall Texan	Elmo Williams	Luther Adler, Lloyd Bridges, Marie Windsor, Le...	NaN	NaN	2021	
3275	s3276	TV Show	American Playboy	Unknown	Hugh Hefner, Matt Whelan	United States	NaN	2017	
4214	s4215	Movie	Wild Horses	Robert Duvall	Robert Duvall, James Franco, Josh Hartnett	NaN	NaN	2015	
6888	s6889	Movie	Ice Agent	Ray O'Neill	Ray O'Neill, Michael Madsen, Joanna Pacula, Ra...	NaN	NaN	2013	
7739	s7740	Movie	Green-Eyed Monster	Jane Prowse	Fay Ripley, Emma Fielding, Hugo Speer	NaN	NaN	2001	
2027	s2028	Movie	Jim Gaffigan: Noble Ape	Jeannie Gaffigan	Jim Gaffigan	NaN	NaN	2018	
2029	s2030	Movie	Jill-Michele Meleán: White / Latina	Alex Ferrari	Jill-Michele Melean	NaN	NaN	2019	
8065	s8066	Movie	Wah Wah	Richard E. Grant	Nicholas Hoult, Emily Watson, Gabriel Byrne, J...	NaN	NaN	2006	
7361	s7362	Movie	Drunk, Stoned, Brilliant, Dead: The Story of t...	Douglas Tirola	Chevy Chase, Judd Apatow, Henry Beard, PJ O'Ro...	NaN	NaN	2015	

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

```
array(['13+', 'ALL', '18+', 'R', 'TV-Y', 'TV-Y7', 'NR', '16+', 'TV-PG',
       '7+', 'TV-14', 'TV-NR', 'TV-G', 'PG-13', 'TV-MA', 'G', 'PG',
       'NC-17', 'UNRATED', '16', 'AGES_16_', 'AGES_18_', 'ALL_AGES',
       'NOT_RATE'], dtype=object)
```

```
rating_counts = data['rating'].value_counts()
rating_counts
```

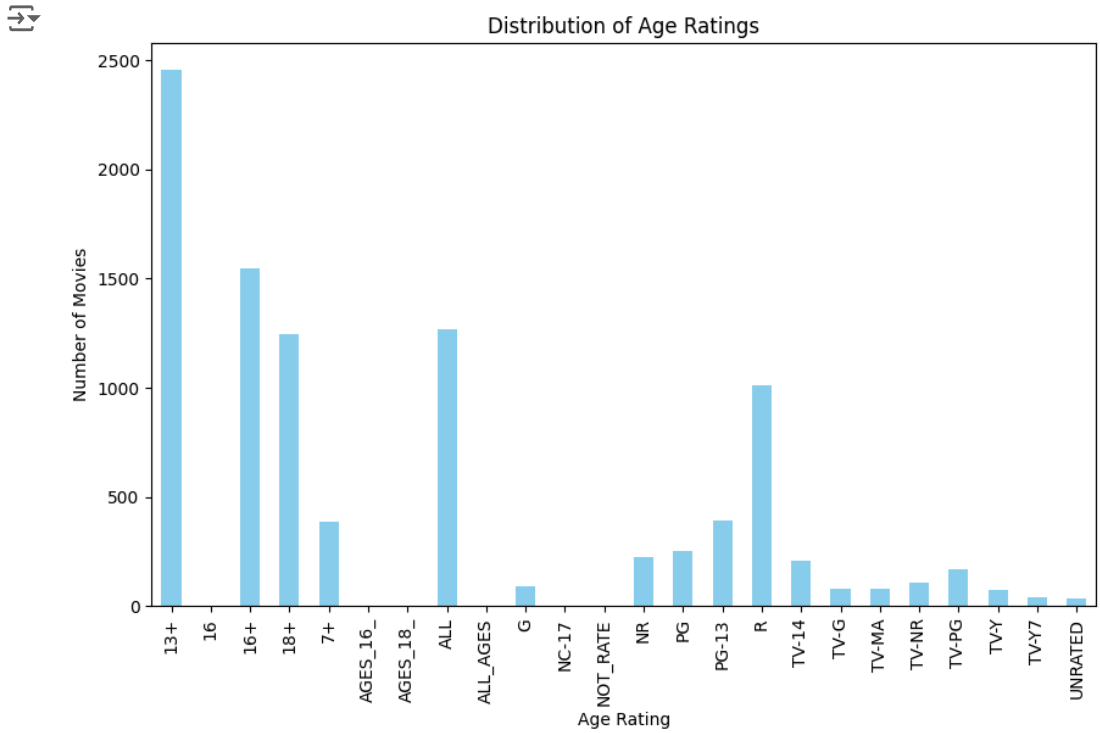
```
rating
13+         2454
16+         1547
ALL         1268
18+         1243
R           1010
PG-13        393
7+           385
PG           253
NR           223
TV-14        208
TV-PG        169
TV-NR        105
G             93
TV-G          81
TV-MA         77
TV-Y          74
TV-Y7         39
UNRATED       33
NC-17          3
AGES_18_       3
NOT_RATE       3
AGES_16_       2
16             1
ALL_AGES       1
Name: count, dtype: int64
```

```
# Exclude rows where the director is "Unknown"
filtered_directors = data[data['director'] != 'Unknown']
top_directors = filtered_directors['director'].value_counts().head(10)
print("Top 10 Prolific Directors:")
print(top_directors)

# Exclude rows where the cast is "Unknown"
filtered_actors = data[data['cast'] != 'Unknown']
top_actors = filtered_actors['cast'].value_counts().head(10)
print("\nTop 10 Prolific Actors:")
print(top_actors)
```

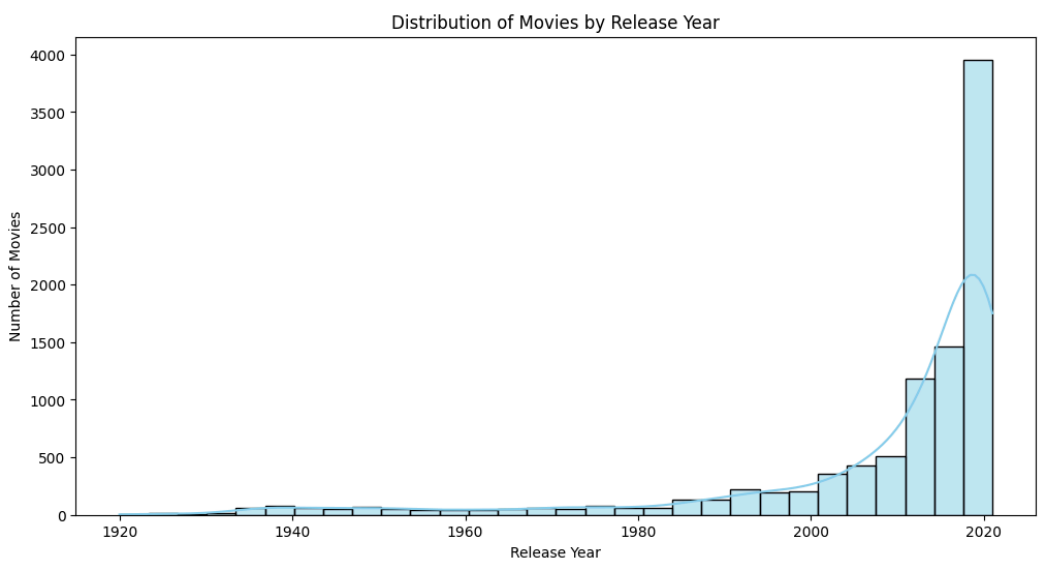
```
Top 10 Prolific Directors:
director
Mark Knight              113
Cannis Holder             61
Moonbug Entertainment     37
Jay Chapman               34
Arthur van Merwijk        30
Manny Rodriguez           22
John English              20
1                         16
Brian Volk-Weiss          15
Baeble Music              14
Name: count, dtype: int64

Top 10 Prolific Actors:
cast
Maggie Binkley                      56
1                                   34
Anne-Marie Newland                  24
Cassandra Peterson                  21
```

```
Grace Tamayo, Erin Webbs            17
Gene Autry, Champion, Gail Davis    12
Stevin John                         11
Gallagher                            9
LB, Aaron Michael                    9
Eddie Izzard                         9
Name: count, dtype: int64
```

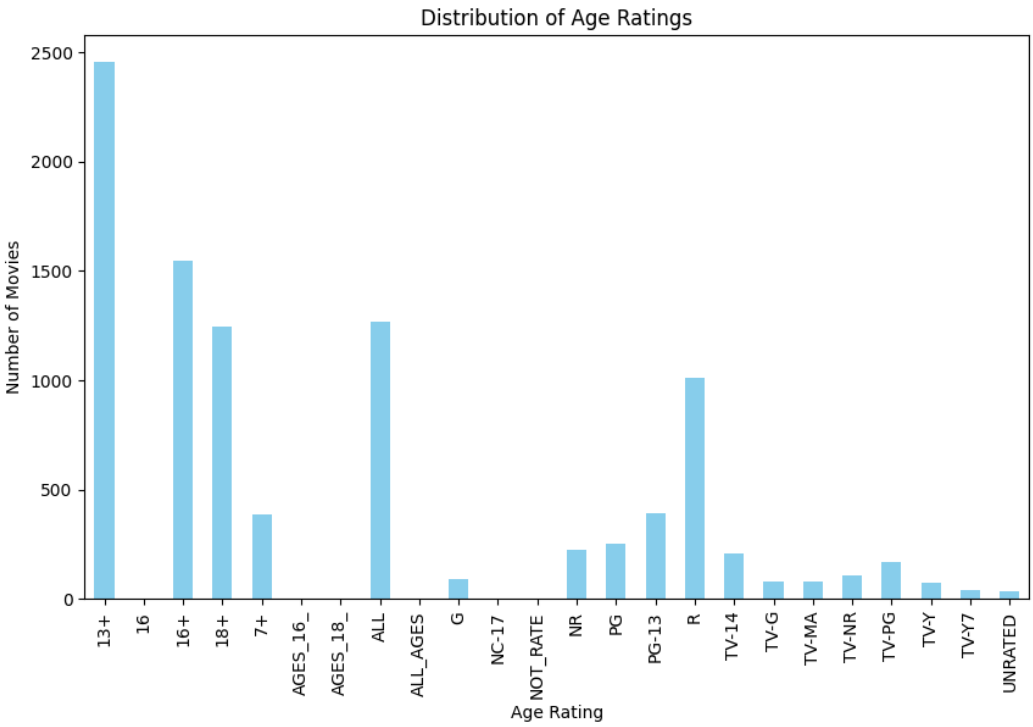
```
plt.figure(figsize=(10, 6))
rating_counts.sort_index().plot(kind='bar', color='skyblue')
plt.title('Distribution of Age Ratings')
plt.xlabel('Age Rating')
plt.ylabel('Number of Movies')
plt.show()
```




```
# Distribution of movies by release year
plt.figure(figsize=(12, 6))
sns.histplot(data['release_year'], bins=30, kde=True, color='skyblue')
plt.title('Distribution of Movies by Release Year')
plt.xlabel('Release Year')
plt.ylabel('Number of Movies')
plt.show()
```

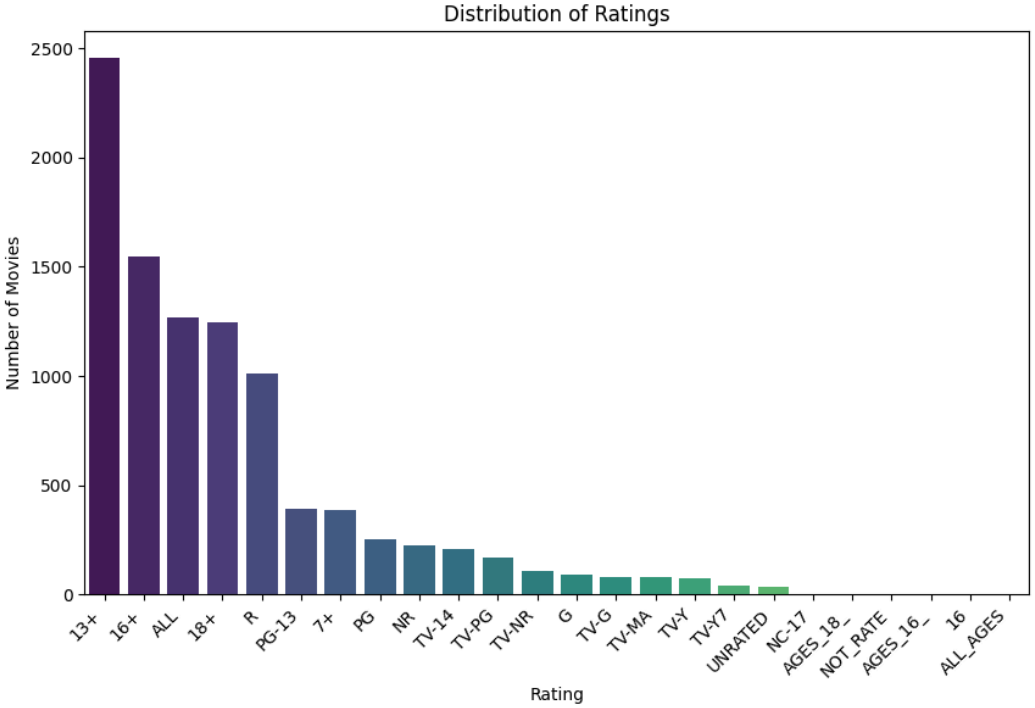



```
plt.figure(figsize=(10, 6))
rating_counts.sort_index().plot(kind='bar', color='skyblue')
plt.title('Distribution of Age Ratings')
plt.xlabel('Age Rating')
plt.ylabel('Number of Movies')
plt.show()
```

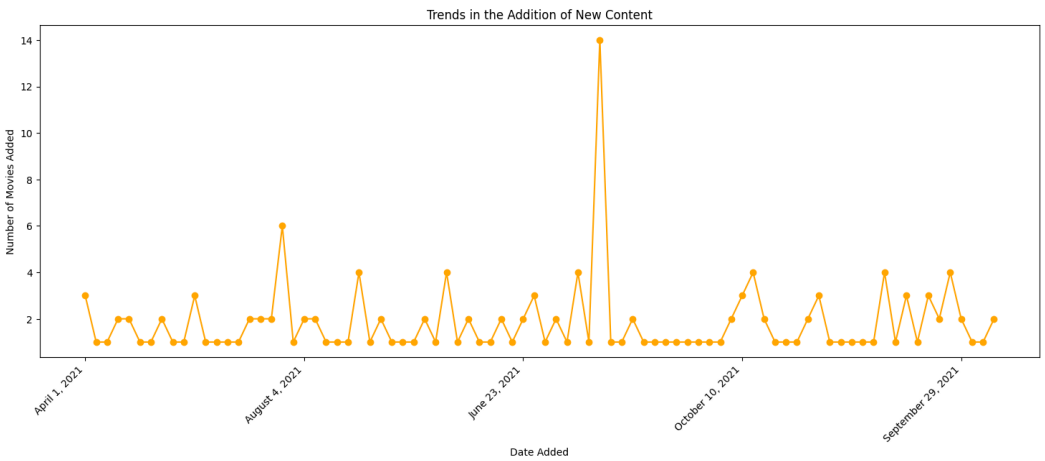



```
# Distribution of ratings
plt.figure(figsize=(10, 6))
sns.countplot(x='rating', data=data, order=data['rating'].value_counts().index, palette='viridis')
plt.title('Distribution of Ratings')
plt.xlabel('Rating')
plt.ylabel('Number of Movies')
plt.xticks(rotation=45, ha='right')
plt.show()
```

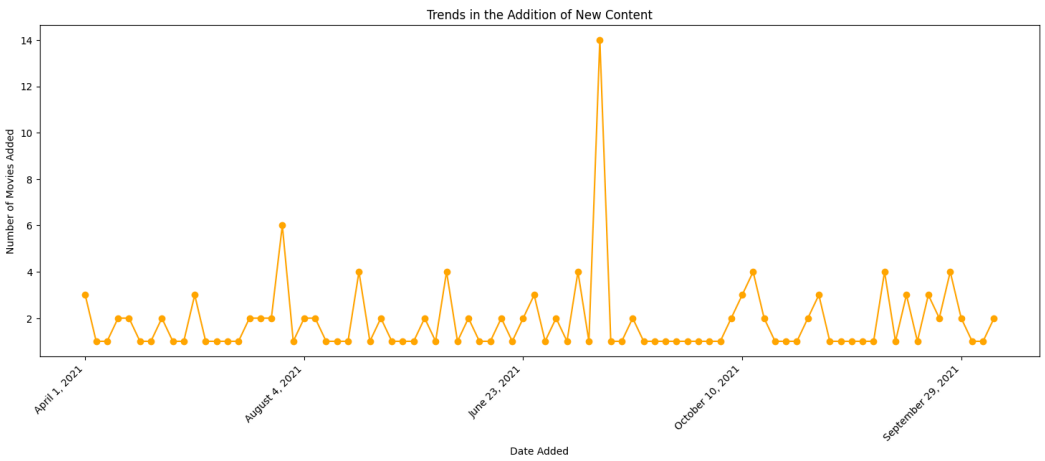
```
<ipython-input-25-5dd9c1739a4b>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.countplot(x='rating', data=data, order=data['rating'].value_counts().index, palet
```




```
# Trends or patterns in the addition of new content
date_added_trends = data.groupby('date_added')['show_id'].count()
plt.figure(figsize=(18, 6))
date_added_trends.plot(marker='o', linestyle='-', color='orange')
plt.title('Trends in the Addition of New Content')
plt.xlabel('Date Added')
plt.ylabel('Number of Movies Added')
plt.xticks(rotation=45, ha='right')
plt.show()
```

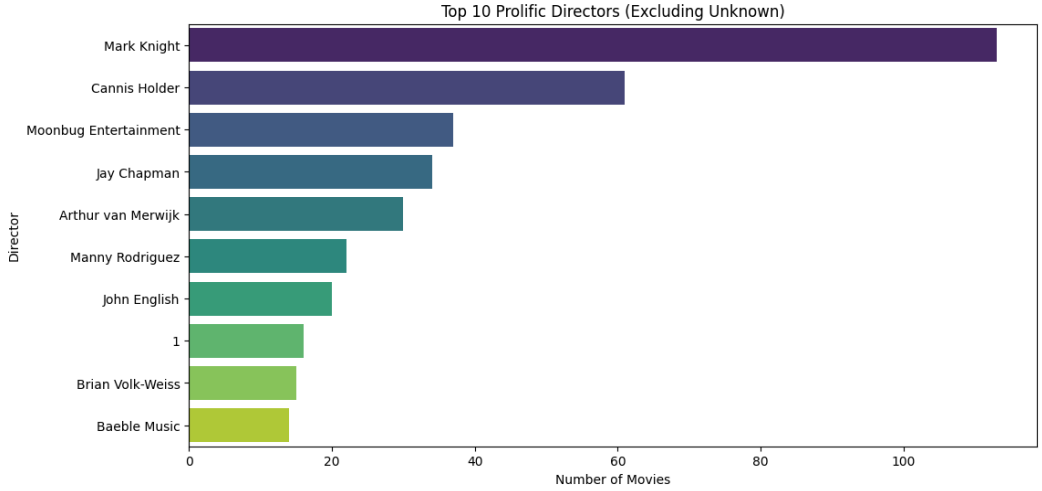



```
# Trends or patterns in the addition of new content
date_added_trends = data.groupby('date_added')['show_id'].count()
plt.figure(figsize=(18, 6))
date_added_trends.plot(marker='o', linestyle='-', color='orange')
plt.title('Trends in the Addition of New Content')
plt.xlabel('Date Added')
plt.ylabel('Number of Movies Added')
plt.xticks(rotation=45, ha='right')
plt.show()
```



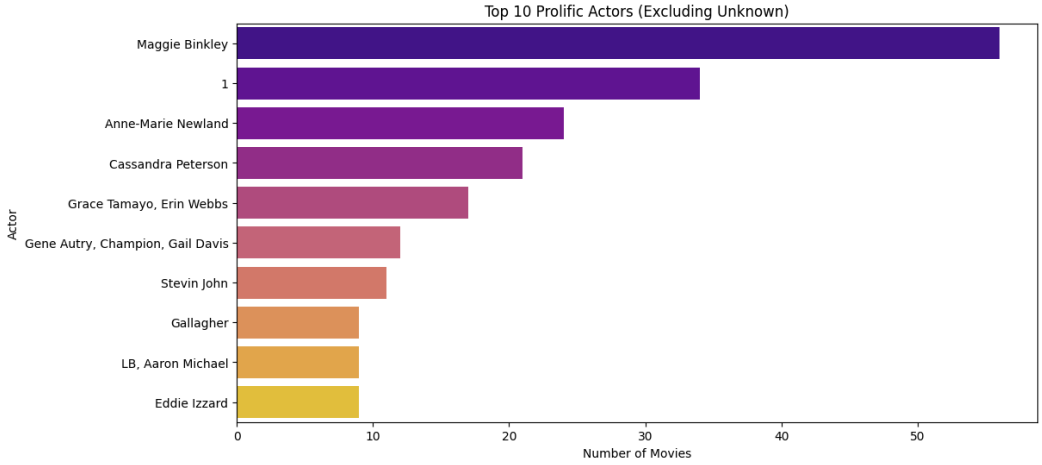

```
plt.figure(figsize=(12, 6))
sns.barplot(x=top_directors.values, y=top_directors.index, palette='viridis')
plt.title('Top 10 Prolific Directors (Excluding Unknown)')
plt.xlabel('Number of Movies')
plt.ylabel('Director')
plt.show()
```

```
<ipython-input-28-f094f57e32a7>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x=top_directors.values, y=top_directors.index, palette='viridis')
```




```
plt.figure(figsize=(12, 6))
sns.barplot(x=top_actors.values, y=top_actors.index, palette='plasma')
plt.title('Top 10 Prolific Actors (Excluding Unknown)')
plt.xlabel('Number of Movies')
plt.ylabel('Actor')
plt.show()
```

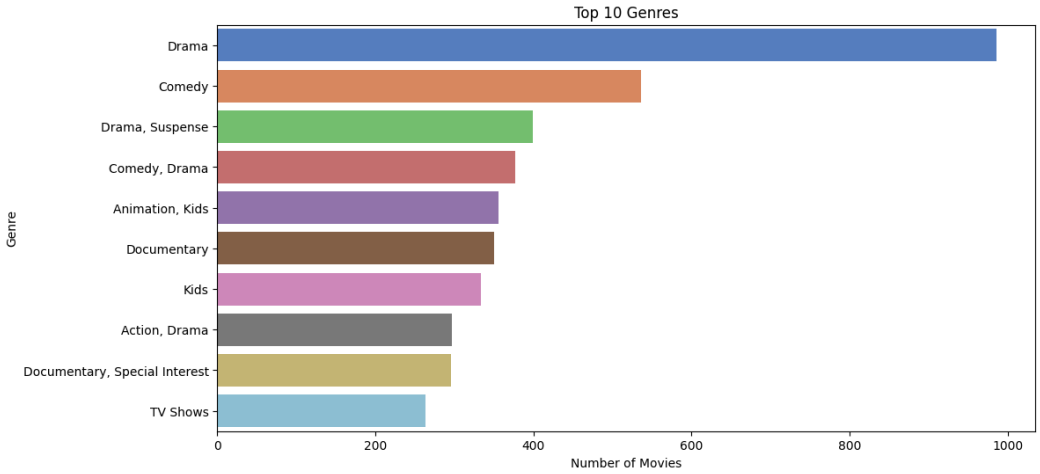
```
<ipython-input-29-4475e29578e2>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x=top_actors.values, y=top_actors.index, palette='plasma')
```




```
plt.figure(figsize=(12, 6))
sns.barplot(x=top_actors.values, y=top_actors.index, palette='plasma')
plt.title('Top 10 Prolific Actors (Excluding Unknown)')
plt.xlabel('Number of Movies')
plt.ylabel('Actor')
plt.show()
```

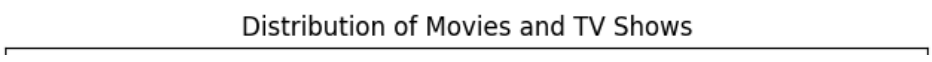
```
# Top genres
top_genres = data['listed_in'].value_counts().head(10)

# Plot Top Genres
plt.figure(figsize=(12, 6))
sns.barplot(x=top_genres.values, y=top_genres.index, palette='muted')
plt.title('Top 10 Genres')
plt.xlabel('Number of Movies')
plt.ylabel('Genre')
plt.show()
```

```
<ipython-input-31-da567dc681db>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x=top_genres.values, y=top_genres.index, palette='muted')
```



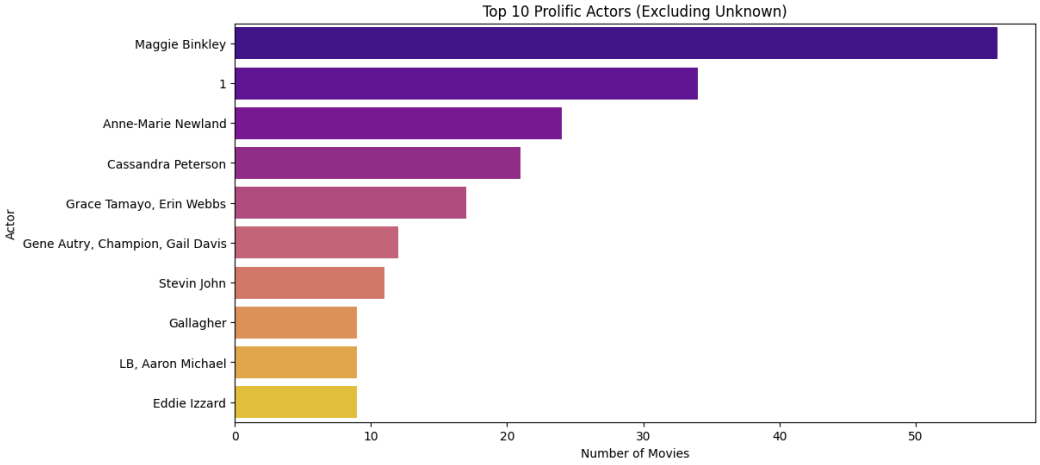

```
# Plotting the distribution of movies and TV shows
plt.figure(figsize=(8, 6))
sns.countplot(x='type', data=data, palette='viridis')
plt.title('Distribution of Movies and TV Shows')
plt.xlabel('Type')
plt.ylabel('Number of Titles')
plt.show()
```

```
<ipython-input-32-02474a7a4108>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.countplot(x='type', data=data, palette='viridis')
```



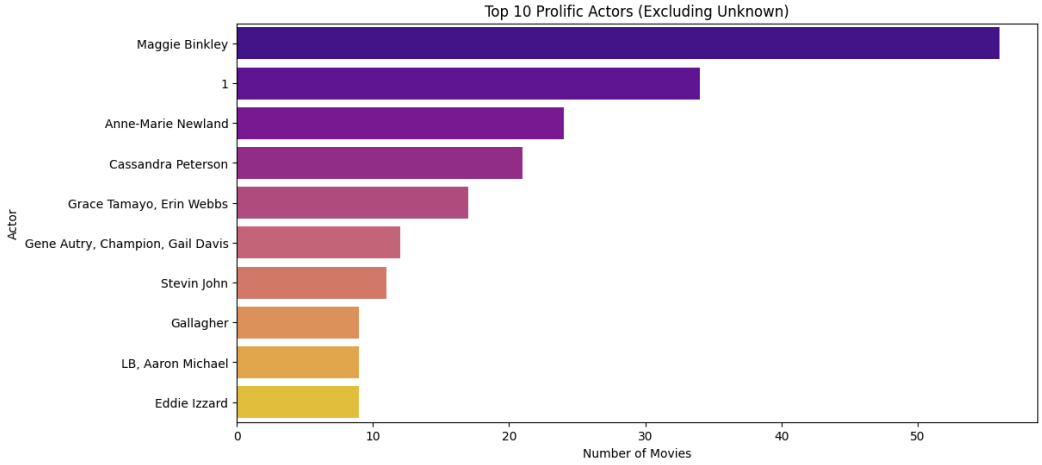
```
plt.figure(figsize=(12, 6))
sns.barplot(x=top_actors.values, y=top_actors.index, palette='plasma')
plt.title('Top 10 Prolific Actors (Excluding Unknown)')
plt.xlabel('Number of Movies')
plt.ylabel('Actor')
plt.show()
```

```
<ipython-input-29-4475e29578e2>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x=top_actors.values, y=top_actors.index, palette='plasma')
```



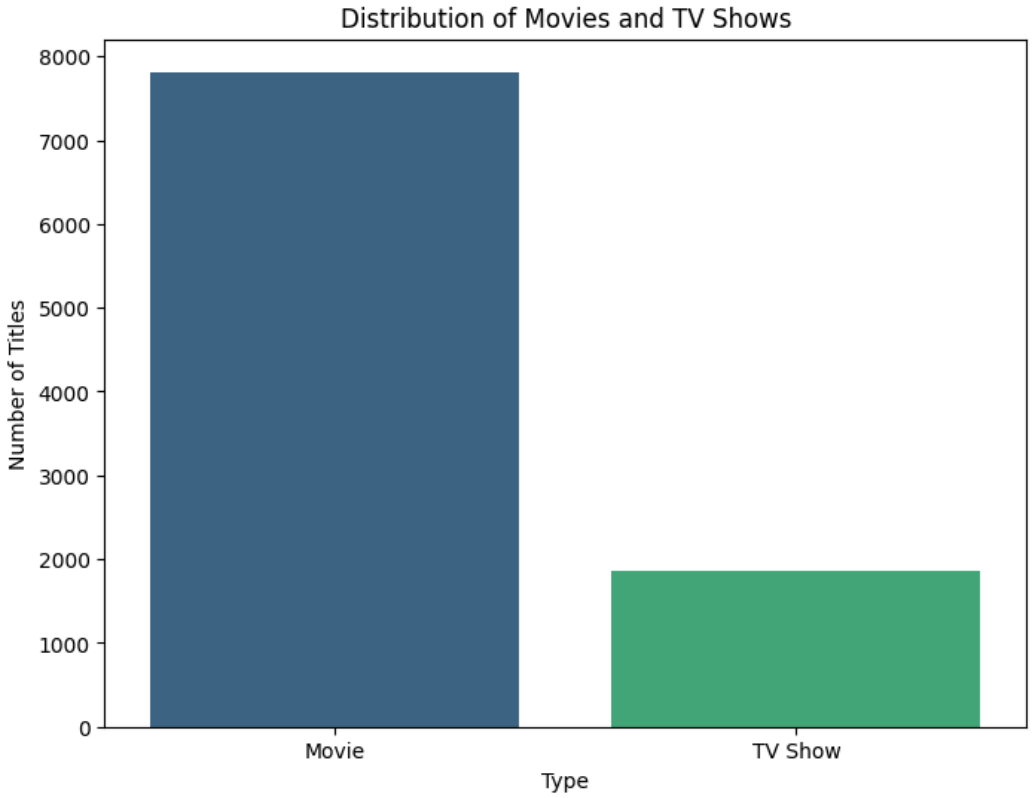

```
plt.figure(figsize=(12, 6))
sns.barplot(x=top_actors.values, y=top_actors.index, palette='plasma')
plt.title('Top 10 Prolific Actors (Excluding Unknown)')
plt.xlabel('Number of Movies')
plt.ylabel('Actor')
plt.show()
```

```
<ipython-input-30-4475e29578e2>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x=top_actors.values, y=top_actors.index, palette='plasma')
```

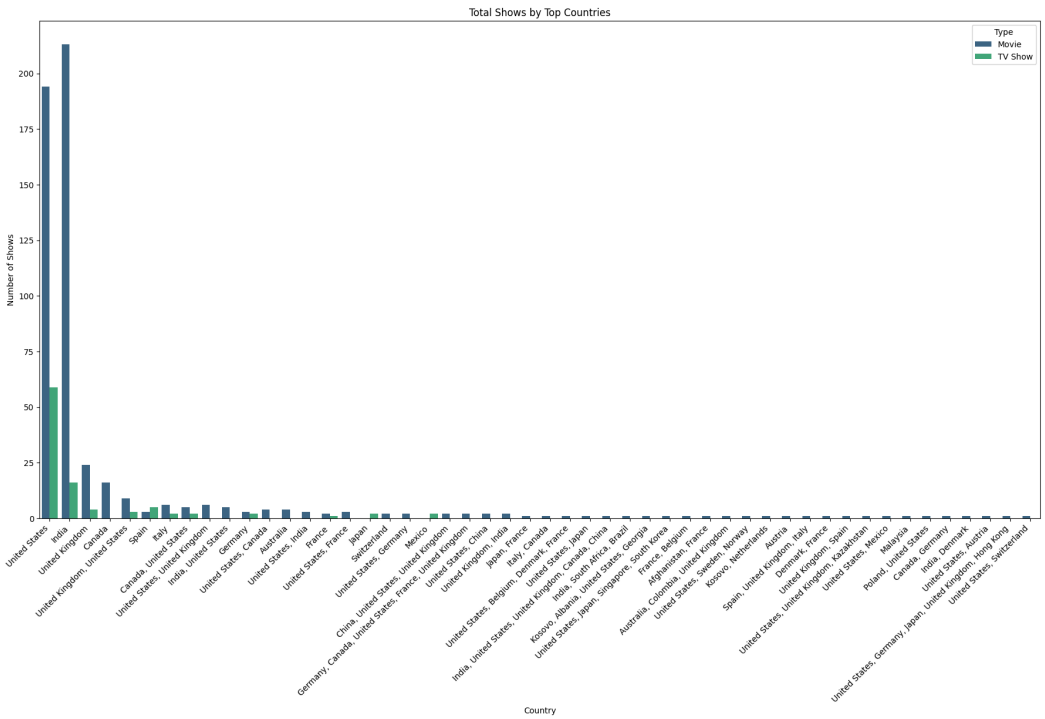



```
# Plotting the distribution of movies and TV shows
plt.figure(figsize=(8, 6))
sns.countplot(x='type', data=data, palette='viridis')
plt.title('Distribution of Movies and TV Shows')
plt.xlabel('Type')
plt.ylabel('Number of Titles')
plt.show()
```

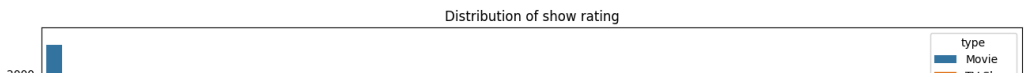
```
<ipython-input-32-02474a7a4108>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.countplot(x='type', data=data, palette='viridis')
```




```
# Top N countries
top_countries = data['country'].value_counts().head(50).index

# Filter the data for the top countries
filtered_data = data[data['country'].isin(top_countries)]

# Visualize the total shows by country for the top countries
plt.figure(figsize=(22, 11))
sns.countplot(x='country', data=filtered_data, hue='type', order=top_countries, palette='viridis')
plt.title('Total Shows by Top Countries')
plt.xlabel('Country')
plt.ylabel('Number of Shows')
plt.legend(title='Type', loc='upper right')
plt.xticks(rotation=45, ha='right')
plt.show()
```




```
data_count1=data['rating'].value_counts().reset_index()
plt.figure(figsize=(16,6))
sns.countplot(x='rating',data=data,hue='type',order=data['rating'].value_counts().index)
plt.xticks(rotation=90)
plt.title('Distribution of show rating')
plt.xlabel('Rating')
plt.ylabel('Number of Shows')
plt.show()
```




```
import matplotlib.pyplot as plt

def plot_end_bar_chart():
    # Define the data
    labels = ['T', 'h', 'e', ' ', 'E', 'N', 'D']
    values = [20, 8, 5, 10, 15, 10, 12]  # You can adjust these values for different heights

    # Create the bar chart
    plt.bar(labels, values, color='blue')
```