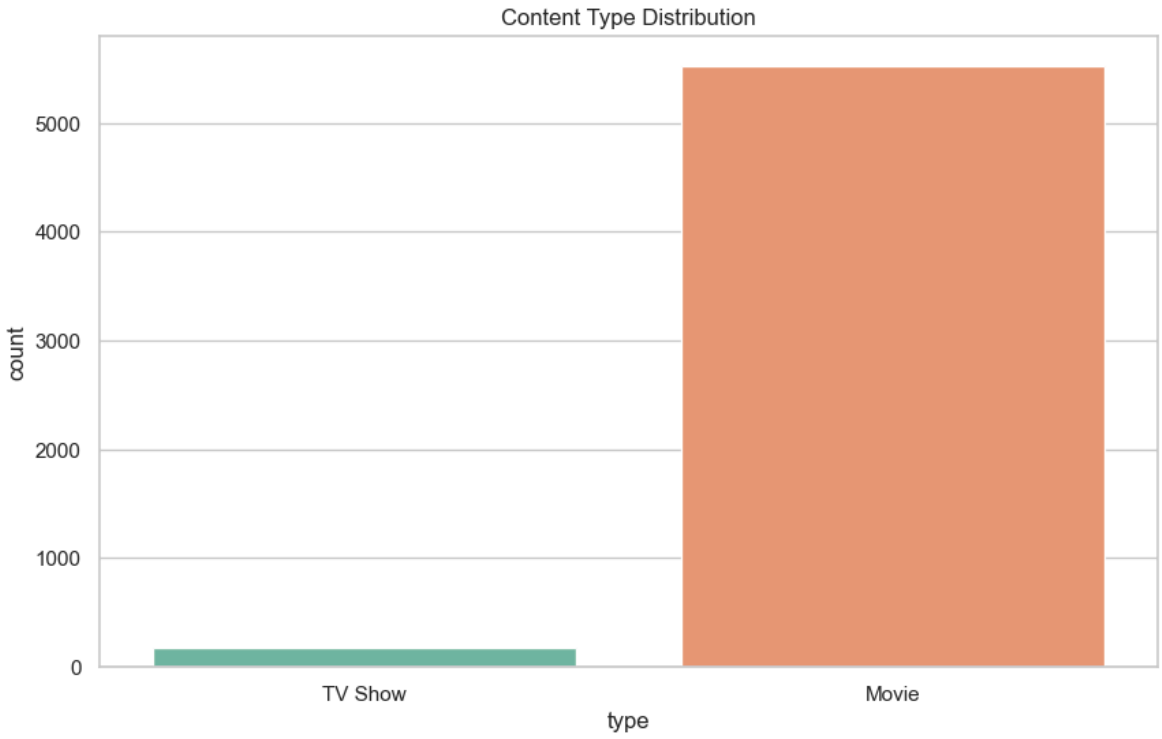
In [ ]:

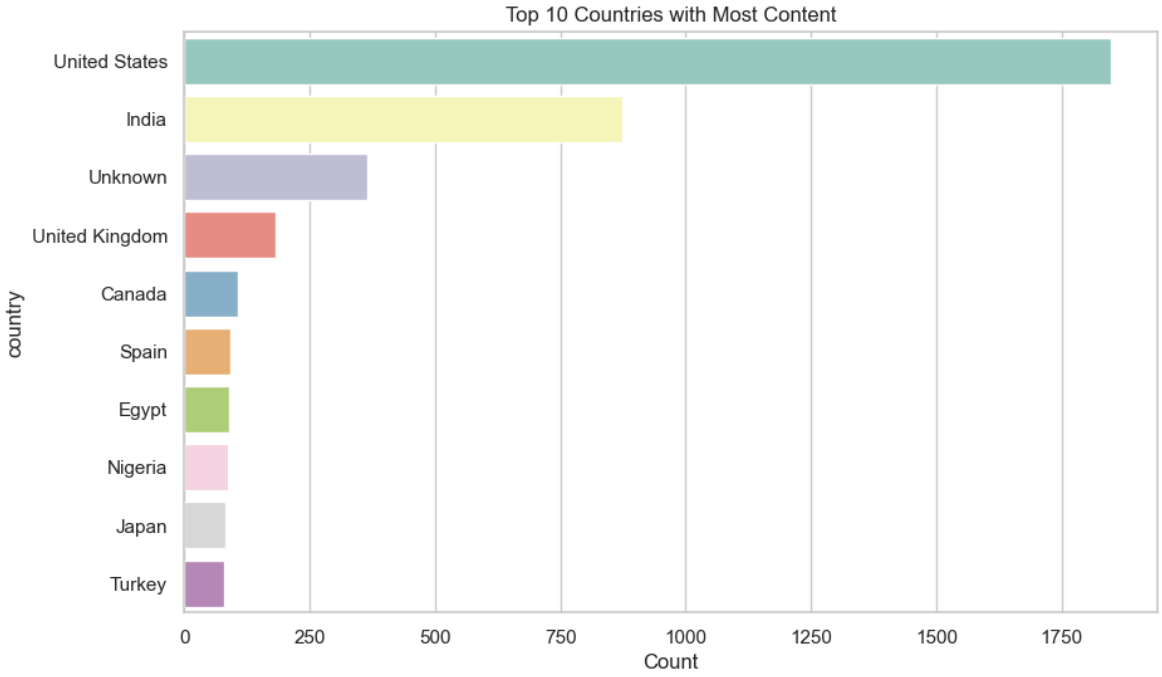
```
# Netflix EDA Project
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Settings
sns.set(style="whitegrid")
plt.rcParams['figure.figsize'] = (10, 6)

# Load Data
df = pd.read_csv(r"C:\Users\Dhams\Downloads\archive (2)\netflix_titles.csv")

# 1. Quick Overview
print("Shape:", df.shape)
print("\nData Types:\n", df.dtypes)
print("\nMissing Values:\n", df.isnull().sum())

# 2. Data Cleaning
df['date_added'] = pd.to_datetime(df['date_added'].str.strip(), errors='coerce')
df['year_added'] = df['date_added'].dt.year
df['month_added'] = df['date_added'].dt.month
df['country'] = df['country'].fillna("Unknown")
df['rating'] = df['rating'].fillna("Not Rated")
df['duration'] = df['duration'].fillna("Unknown")
df.dropna(subset=['cast', 'director'], inplace=True)

# 3. Univariate Analysis
# Content Type
sns.countplot(data=df, x='type', palette='Set2')
plt.title("Content Type Distribution")
plt.show()

# Top 10 Countries
top_countries = df['country'].value_counts().head(10)
sns.barplot(y=top_countries.index, x=top_countries.values, palette="Set3")
plt.title("Top 10 Countries with Most Content")
plt.xlabel("Count")
plt.show()

# Content added per year
df['year_added'].value_counts().sort_index().plot(kind='bar')
plt.title("Content Added per Year")
plt.xlabel("Year")
plt.ylabel("Number of Titles")
plt.show()

# Rating distribution
sns.countplot(data=df, y='rating', order=df['rating'].value_counts().index, pale
plt.title("Distribution of Ratings")
plt.show()

# 4. Bivariate Analysis
# TV Show vs Movie over time
tv_movie_by_year = df.groupby(['year_added', 'type']).size().unstack().fillna(0)
tv_movie_by_year.plot(kind='bar', stacked=True, colormap='Accent')
plt.title("Movies vs TV Shows Added Over Years")
```

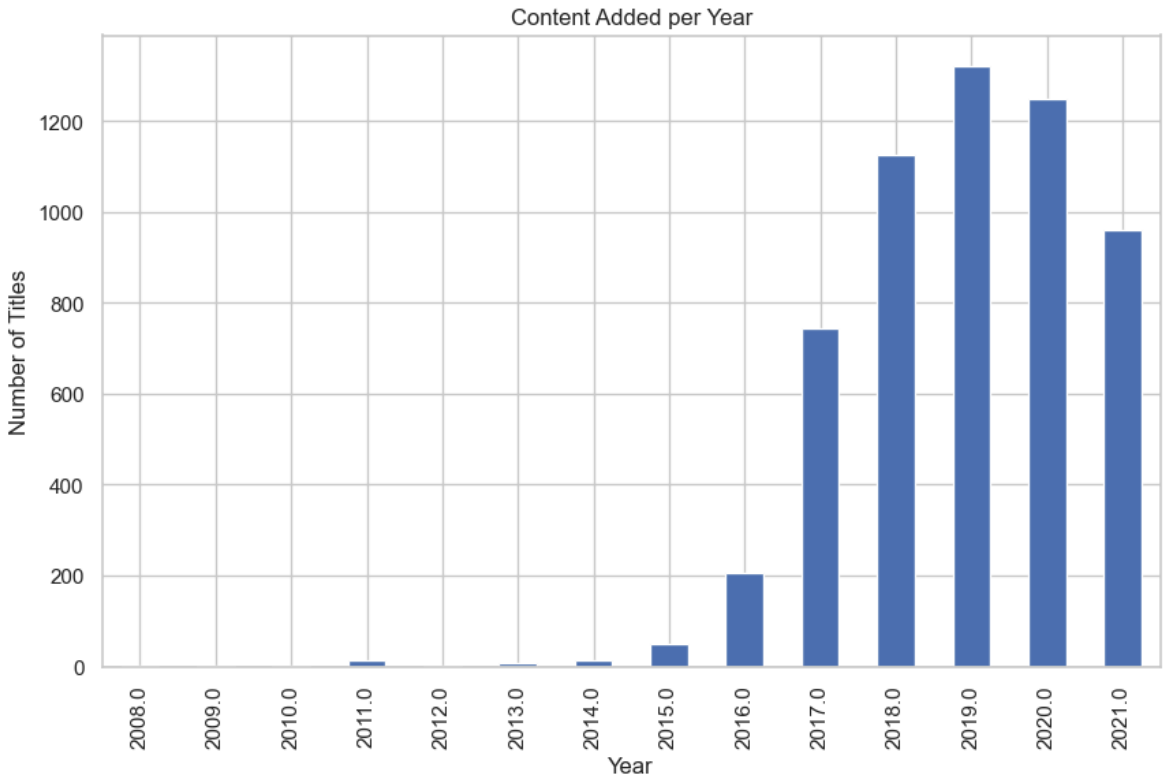
```
plt.ylabel("Count")
plt.show()

# Duration distribution for Movies
movie_df = df[df['type'] == 'Movie'].copy()
movie_df['duration_minutes'] = movie_df['duration'].str.extract('(\d+)').astype(

sns.histplot(movie_df['duration_minutes'], bins=30, kde=True, color="coral")
plt.title("Movie Duration Distribution")
plt.xlabel("Duration (Minutes)")
plt.show()

# 5. Top Directors & Actors
top_directors = df['director'].value_counts().head(10)
top_actors = pd.Series(", ".join(df['cast'].dropna()).split(", ")).value_counts(

# Directors
sns.barplot(y=top_directors.index, x=top_directors.values, palette="viridis")
plt.title("Top 10 Directors")
plt.xlabel("Count")
plt.show()

# Actors
sns.barplot(y=top_actors.index, x=top_actors.values, palette="magma")
plt.title("Top 10 Most Frequent Actors")
plt.xlabel("Count")
plt.show()

# 6. Interactive Plot (Optional)
fig = px.treemap(df, path=['country', 'type'], title='Content Distribution by Co
fig.show()

# 7. Word Cloud (Optional for cast or title)
from wordcloud import WordCloud, STOPWORDS

text = ' '.join(df['title'].dropna())
wordcloud = WordCloud(stopwords=STOPWORDS, background_color='white', colormap='R
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Most Common Words in Titles")
plt.show()
```

```
Shape: (8807, 12)

Data Types:
 show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object

Missing Values:
 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
<ipython-input-2-1ad0c0931c82>:31: FutureWarning:



Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.
```

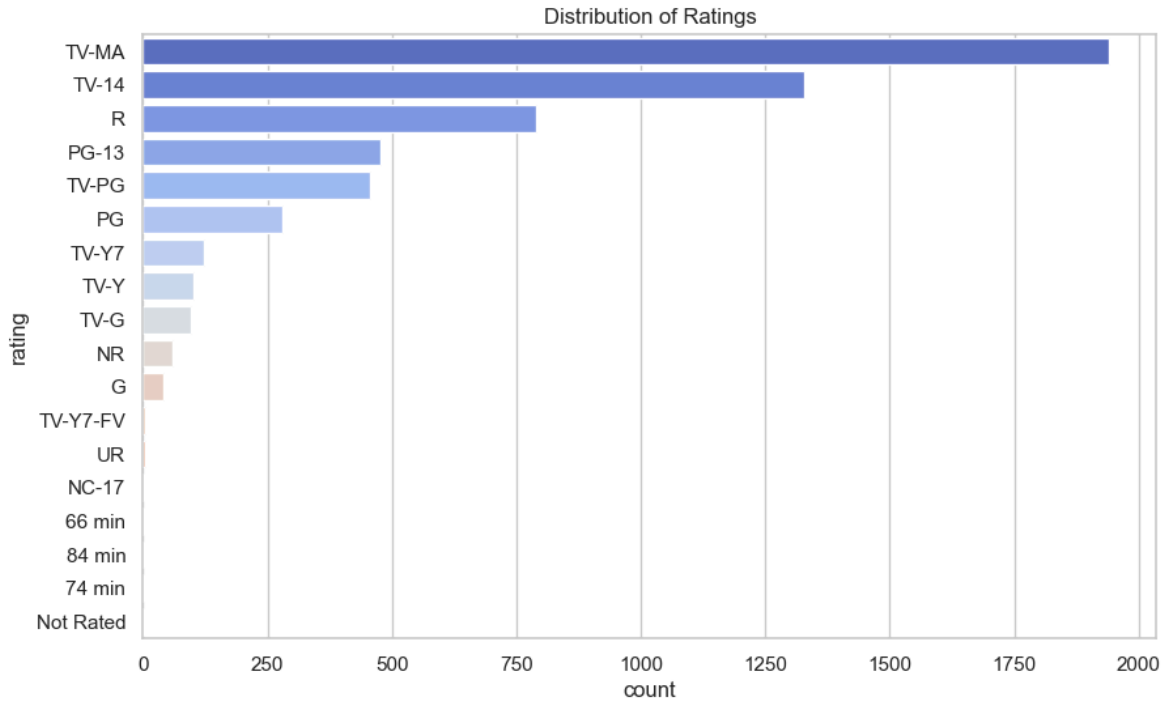


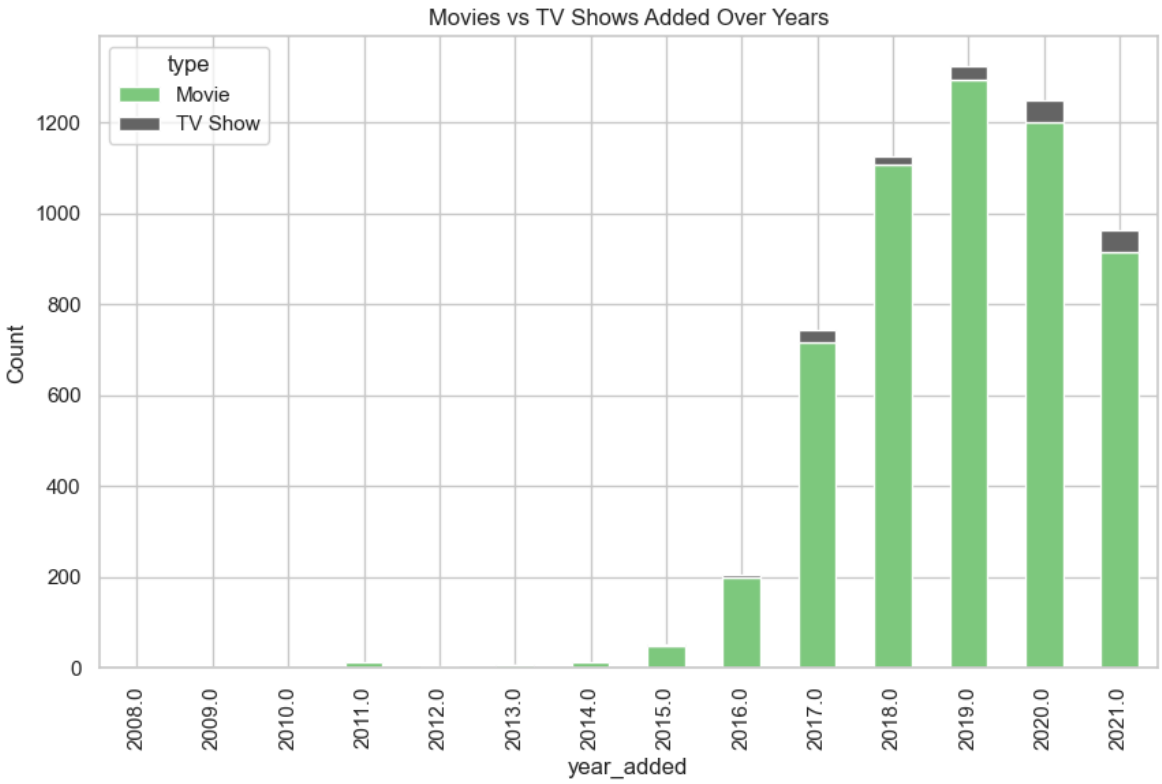

```
<ipython-input-2-1ad0c0931c82>:37: FutureWarning:


Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.
```

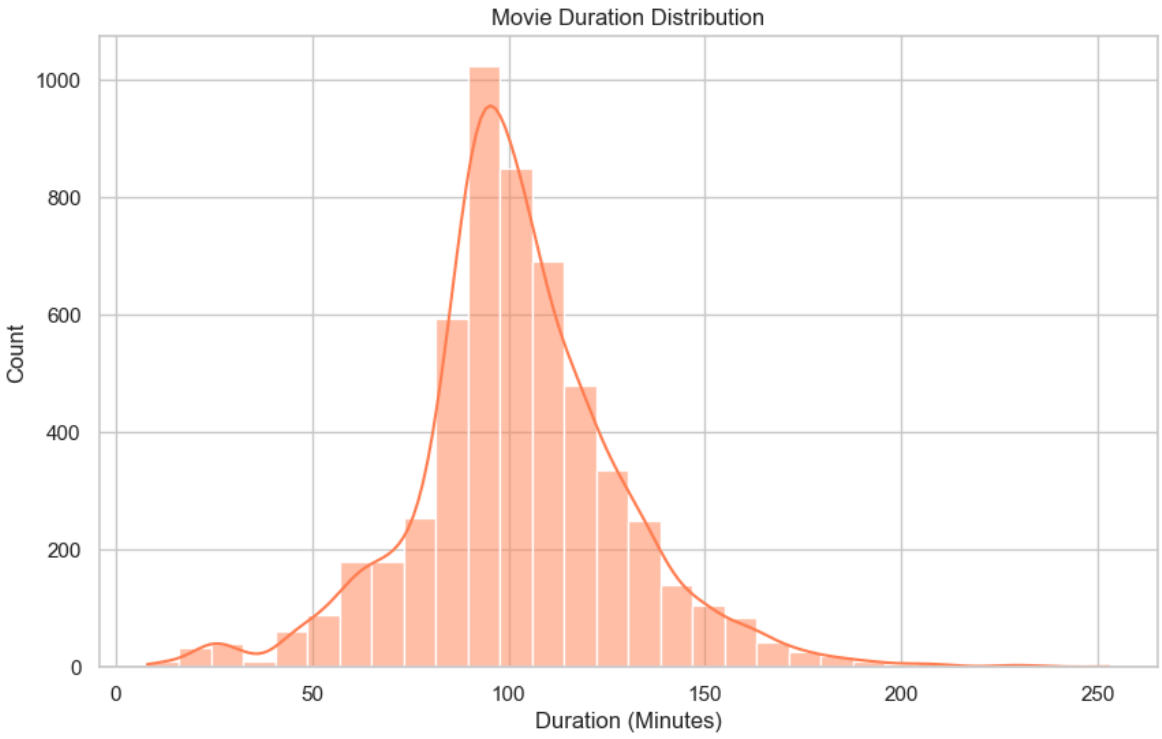




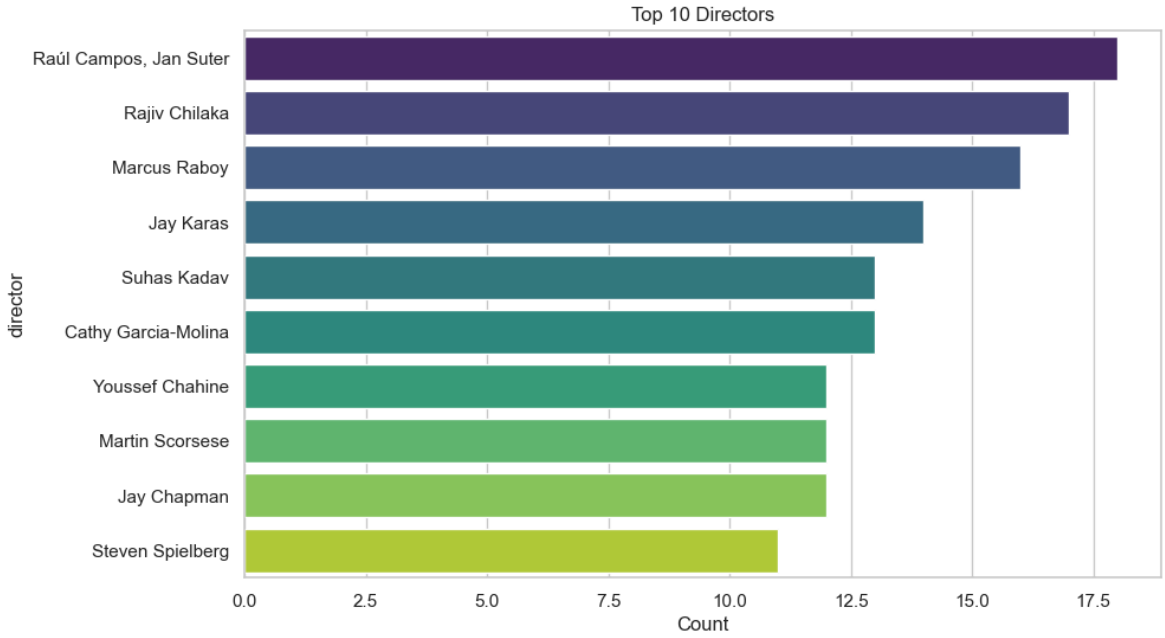

```
<ipython-input-2-1ad0c0931c82>:50: FutureWarning:


Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.
```



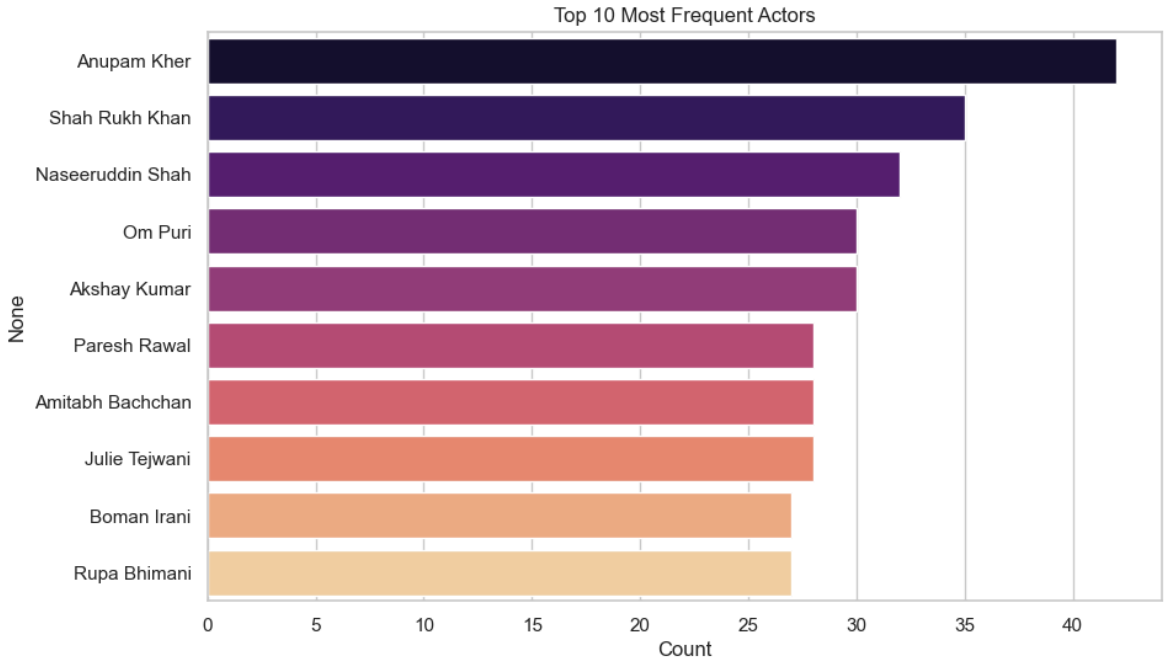





```
<ipython-input-2-1ad0c0931c82>:76: FutureWarning:


Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.
```




```
<ipython-input-2-1ad0c0931c82>:82: FutureWarning:


Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.
```



Most Common Words in Titles