

NETFLIX

Analysis of Netflix Shows

-By Ankita Sarkar



What We Will Talk About

1

Introduction

2

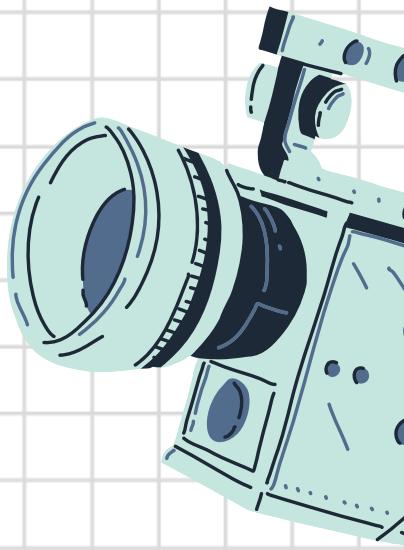
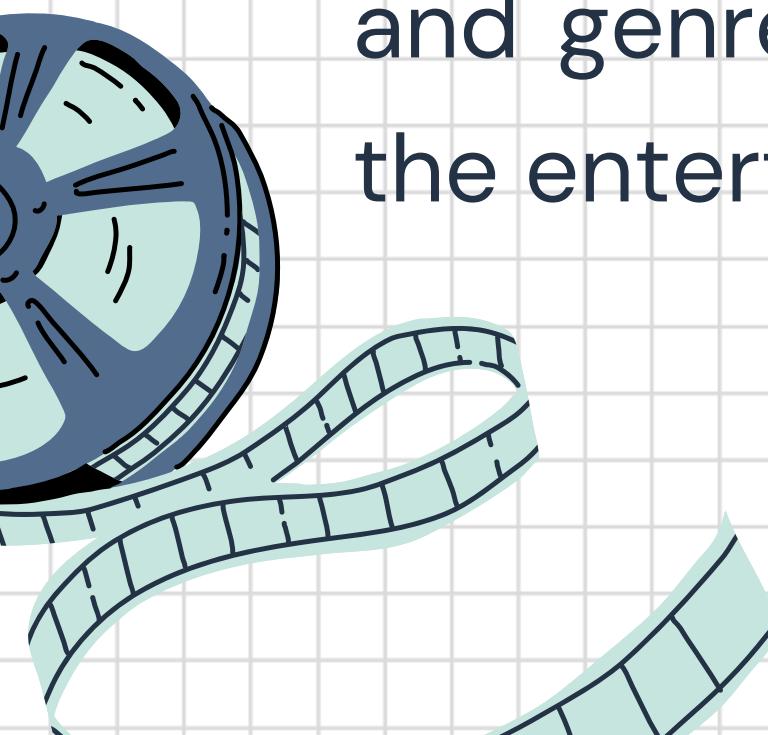
Objective Of
The Analysis

3

Conclusion

Introduction

This project aims to analyze the dataset containing information about movies and TV shows, focusing on trends, distributions, and insights. By exploring attributes such as content type, duration, release year, genres, and directors, the analysis provides a comprehensive understanding of the data. Visualizations are used to highlight patterns, such as the distribution of TV show seasons, the growth of content over time, and the most frequent directors and genres. This analysis supports data-driven decision-making in the entertainment domain.



Tools Used For Analysis

Python

Libraries:

Pandas: For data manipulation and preprocessing.

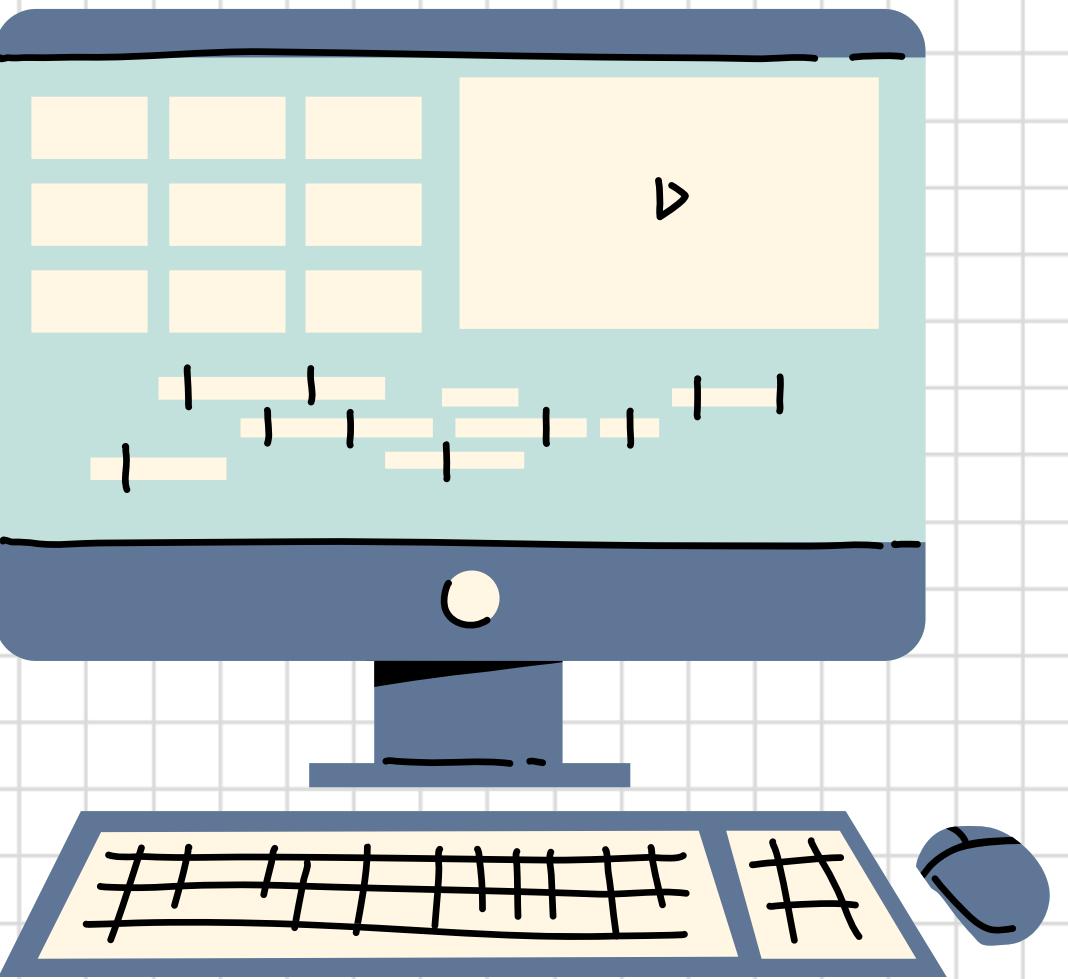
Matplotlib: For creating visualizations.

Seaborn: For enhancing visualizations and styling.

Jupyter Notebook

An interactive environment for writing and running

Python code.



Loading and Previewing the Netflix Dataset

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

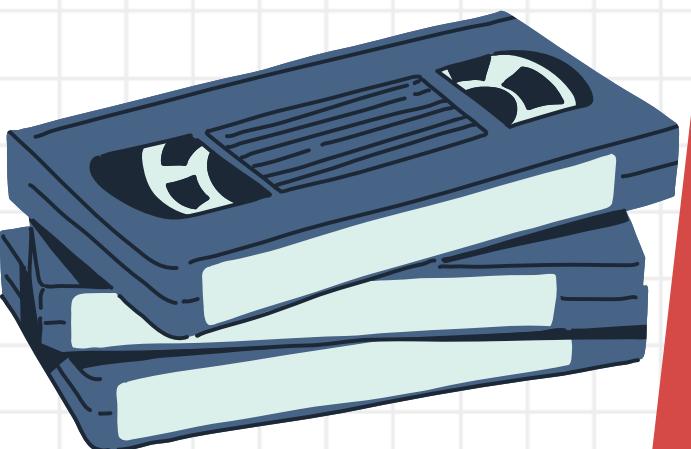
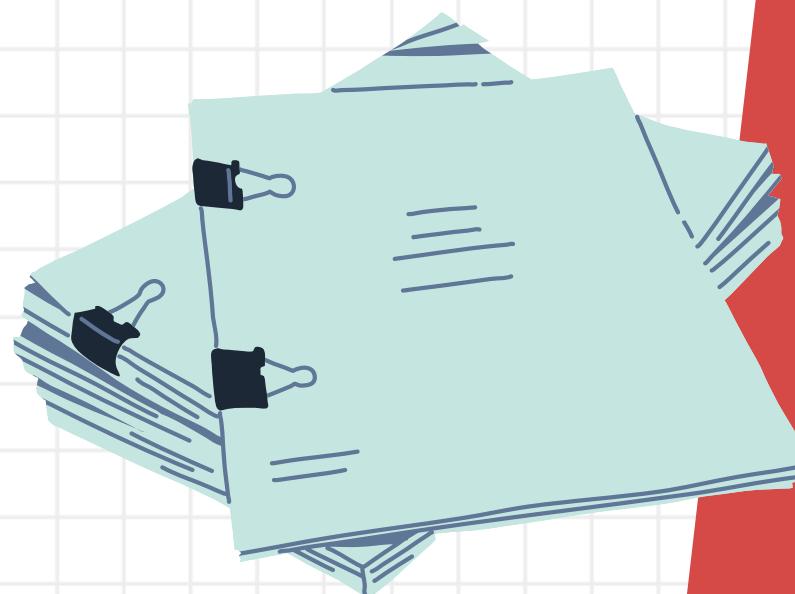
df=pd.read_csv("netflix1 (2).csv")

print(df.head())

  show_id      type              title        director \
0      s1    Movie     Dick Johnson Is Dead  Kirsten Johnson
1      s3   TV Show          Ganglands  Julien Leclercq
2      s6   TV Show       Midnight Mass  Mike Flanagan
3     s14    Movie Confessions of an Invisible Girl  Bruno Garotti
4      s8    Movie            Sankofa  Haile Gerima

  country date_added release_year rating duration \
0  United States  9/25/2021        2020  PG-13    90 min
1      France  9/24/2021        2021  TV-MA  1 Season
2  United States  9/24/2021        2021  TV-MA  1 Season
3      Brazil  9/22/2021        2021  TV-PG    91 min
4  United States  9/24/2021        1993  TV-MA   125 min

  listed_in
0      Documentaries
1 Crime TV Shows, International TV Shows, TV Act...
2      TV Dramas, TV Horror, TV Mysteries
3 Children & Family Movies, Comedies
4      Dramas, Independent Movies, International Movies
```



Distribution Of Content Types

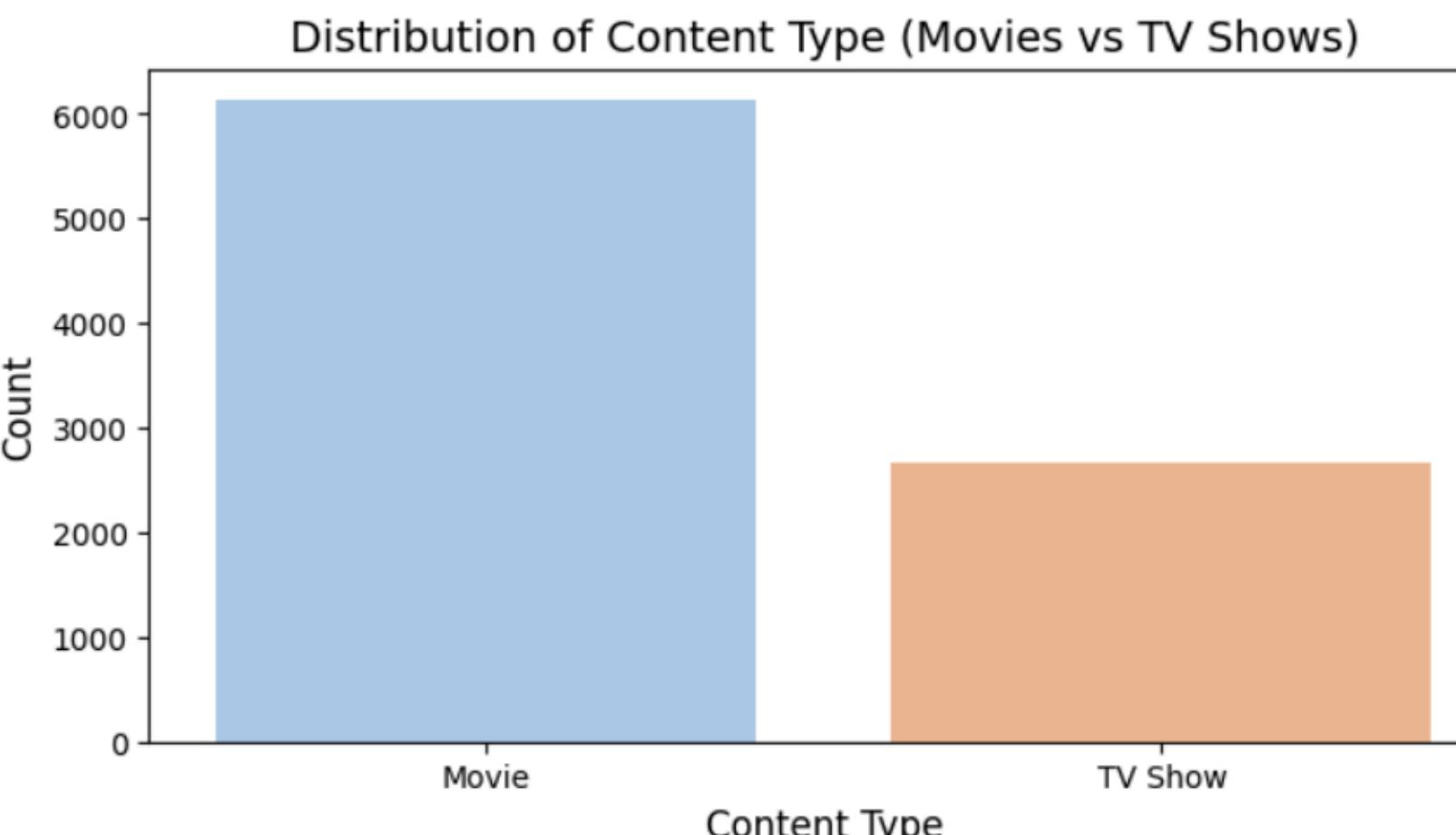
```
# Count of content type
type_counts = df['type'].value_counts()
print("\nCount of Content Type:\n")
print(type_counts)
```

Count of Content Type:

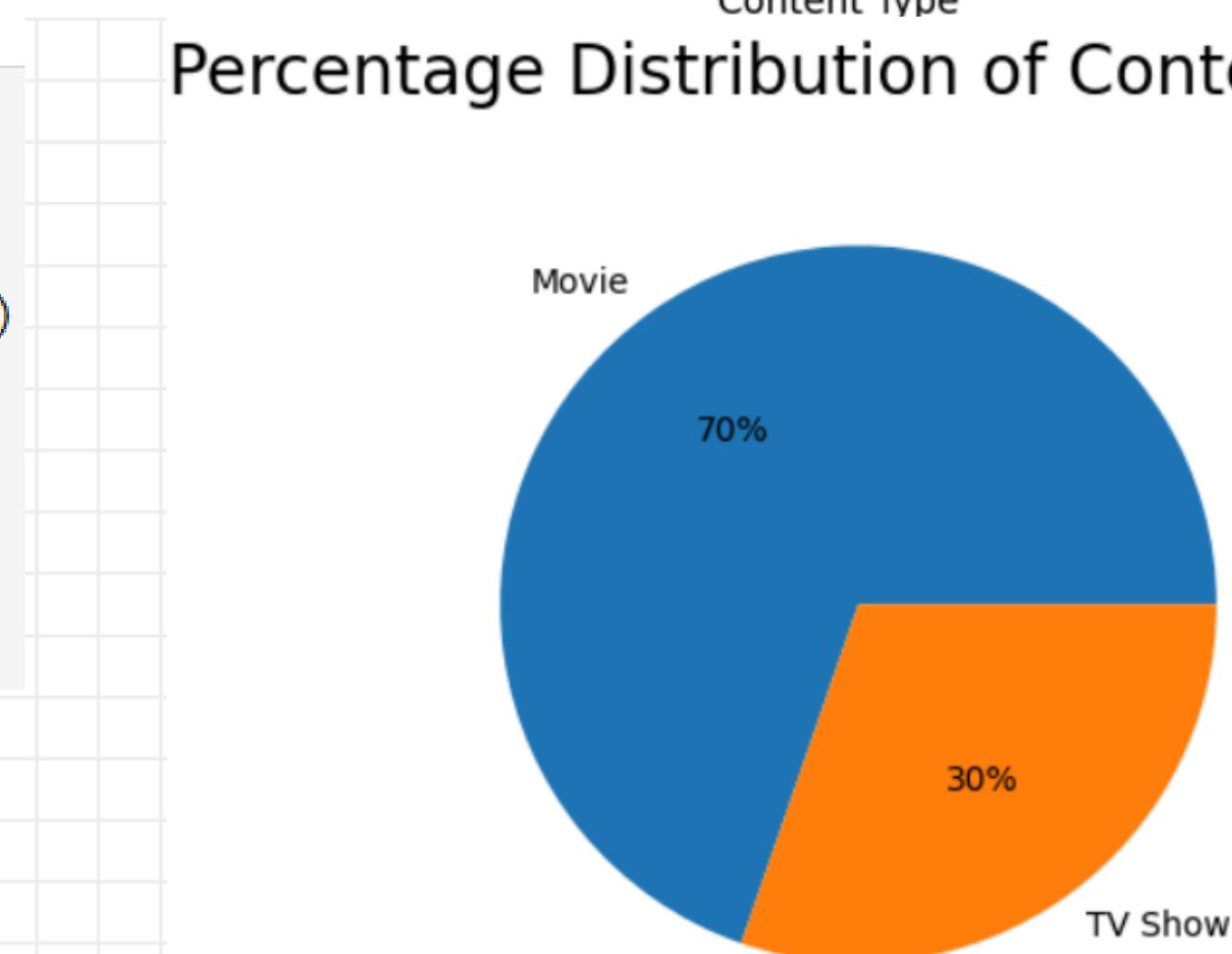
```
type
Movie      6126
TV Show    2664
Name: count, dtype: int64
```

```
# Distribution of content type
plt.figure(figsize=(8, 4))
sns.countplot(x='type', data=df, palette='pastel')
plt.title('Distribution of Content Type (Movies vs TV Shows)', fontsize=14)
plt.xlabel('Content Type', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.show()

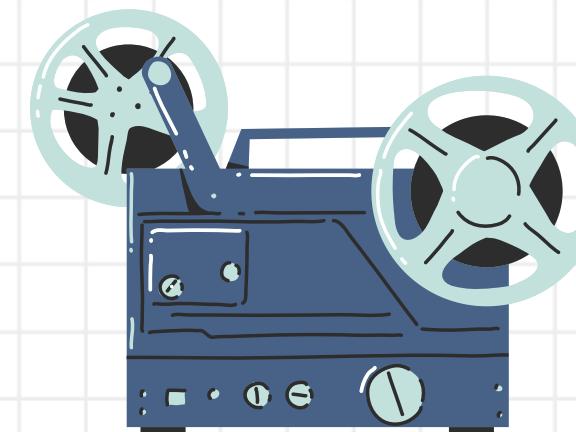
plt.pie(type_counts, labels=['Movie', 'TV Show'], autopct='%.0f%%')
plt.suptitle('Percentage Distribution of Content Type', fontsize=20)
```



Percentage Distribution of Content Type



Top 10 Countries With Most Content

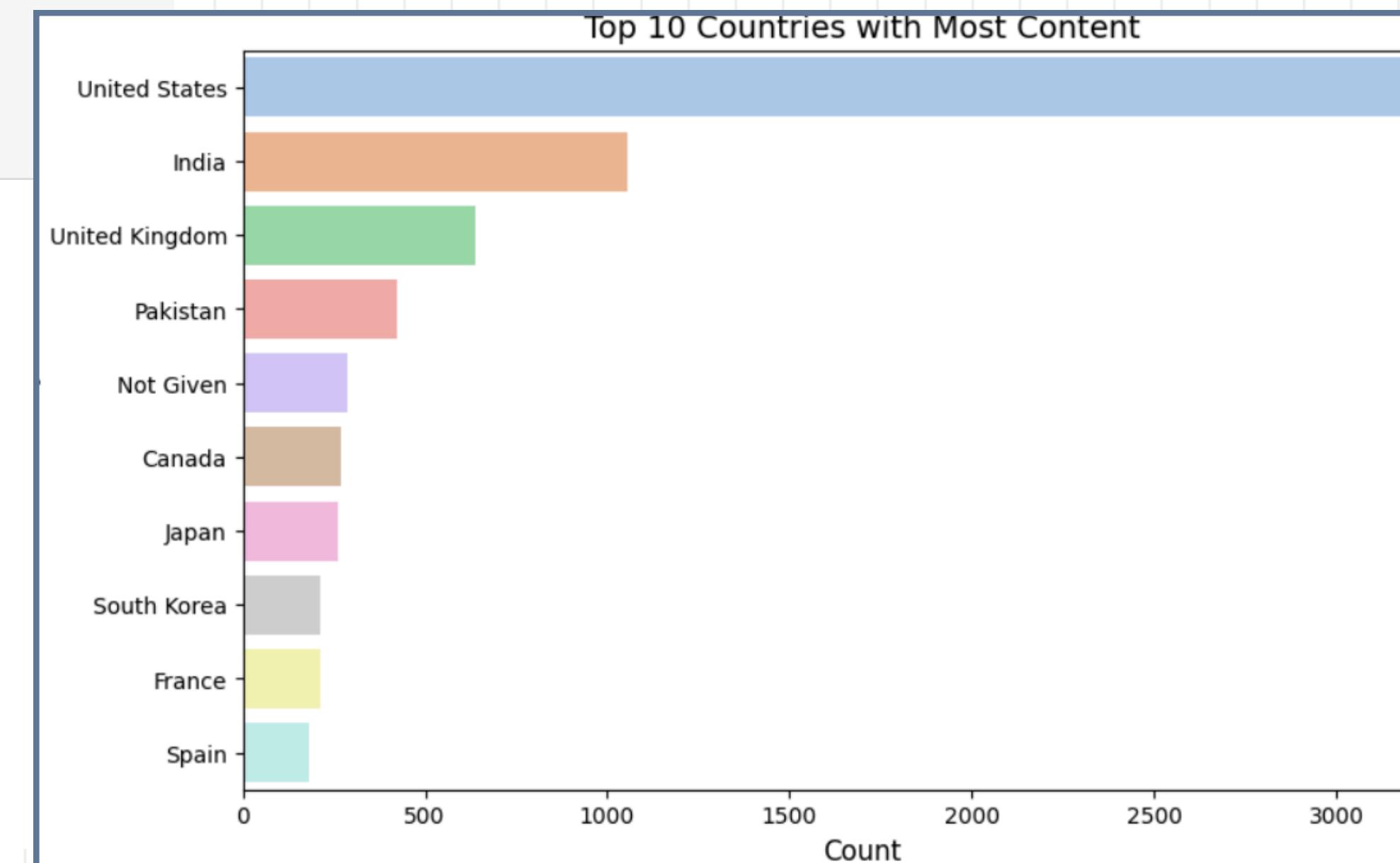


```
# Top 10 countries with most content
country_counts = df['country'].value_counts().head(10)
print("\nTop 10 Countries with Most Content:\n")
country_table = country_counts.reset_index()
country_table.columns = ['Country', 'Count']
print(country_table)

plt.figure(figsize=(10, 6))
sns.barplot(x='Count', y='Country', data=country_table, palette='pastel')
plt.title('Top 10 Countries with Most Content', fontsize=14)
plt.xlabel('Count', fontsize=12)
plt.ylabel('Country', fontsize=12)
plt.show()
```

Top 10 Countries with Most Content:

	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

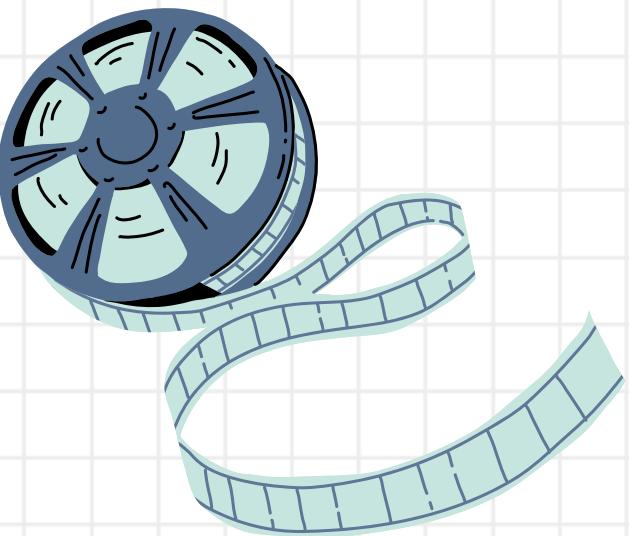


Mothly Releases Of Movies & TV Shows

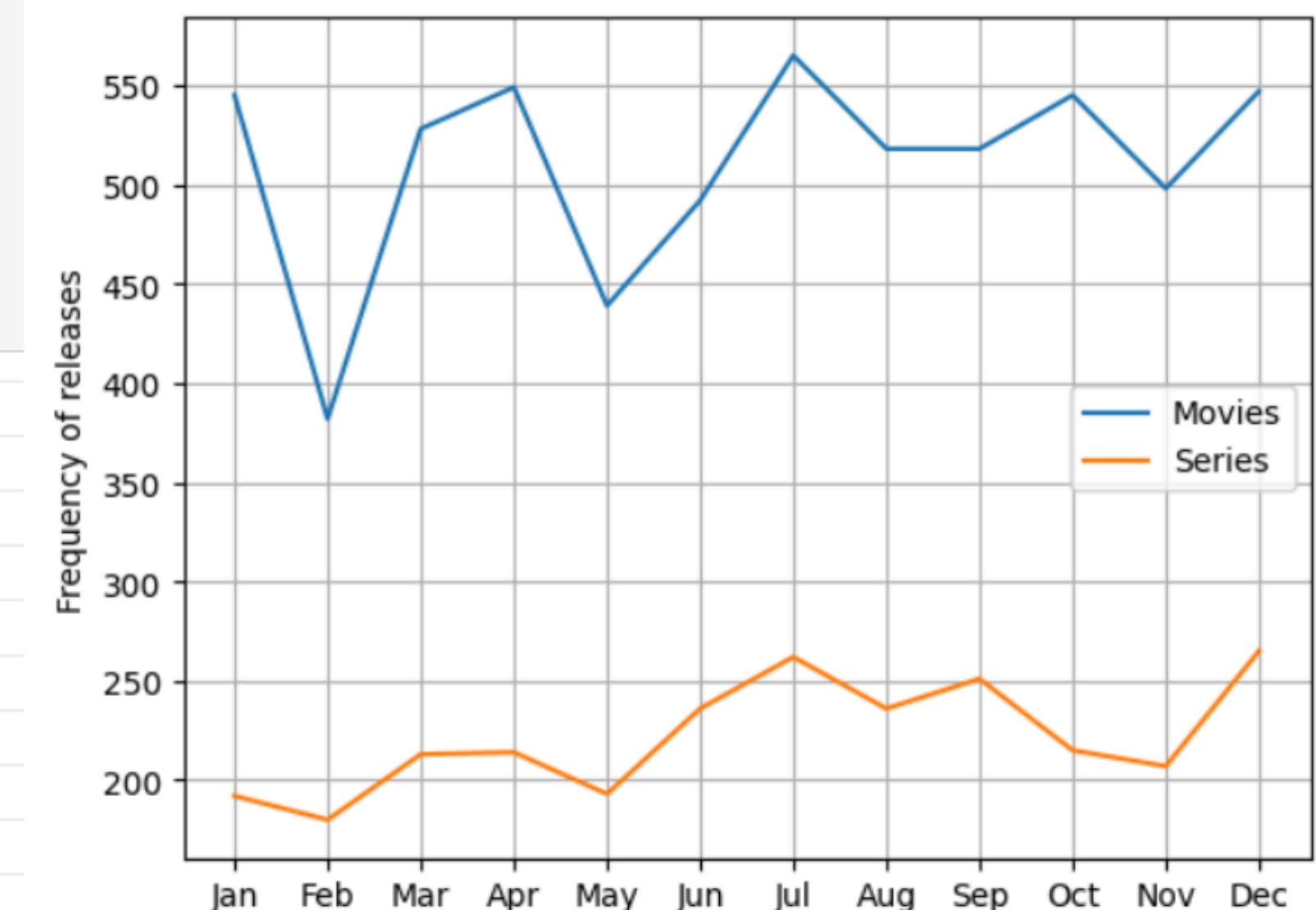
```
#Monthly releases of Movies and TV shows on Netflix
```

```
df['year']=df['date_added'].dt.year  
df['month']=df['date_added'].dt.month  
df['day']=df['date_added'].dt.day
```

```
monthly_movie_release=df[df['type']=='Movie']['month'].value_counts().sort_index()  
monthly_series_release=df[df['type']=='TV Show']['month'].value_counts().sort_index()  
plt.plot(monthly_movie_release.index,monthly_movie_release.values, label='Movies')  
plt.plot(monthly_series_release.index,monthly_series_release.values, label='Series')  
plt.xlabel("Months")  
plt.ylabel("Frequency of releases")  
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May',  
'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])  
plt.legend()  
plt.grid(True)  
plt.suptitle("Monthly releases of Movies and TV shows on Netflix")  
plt.show()
```

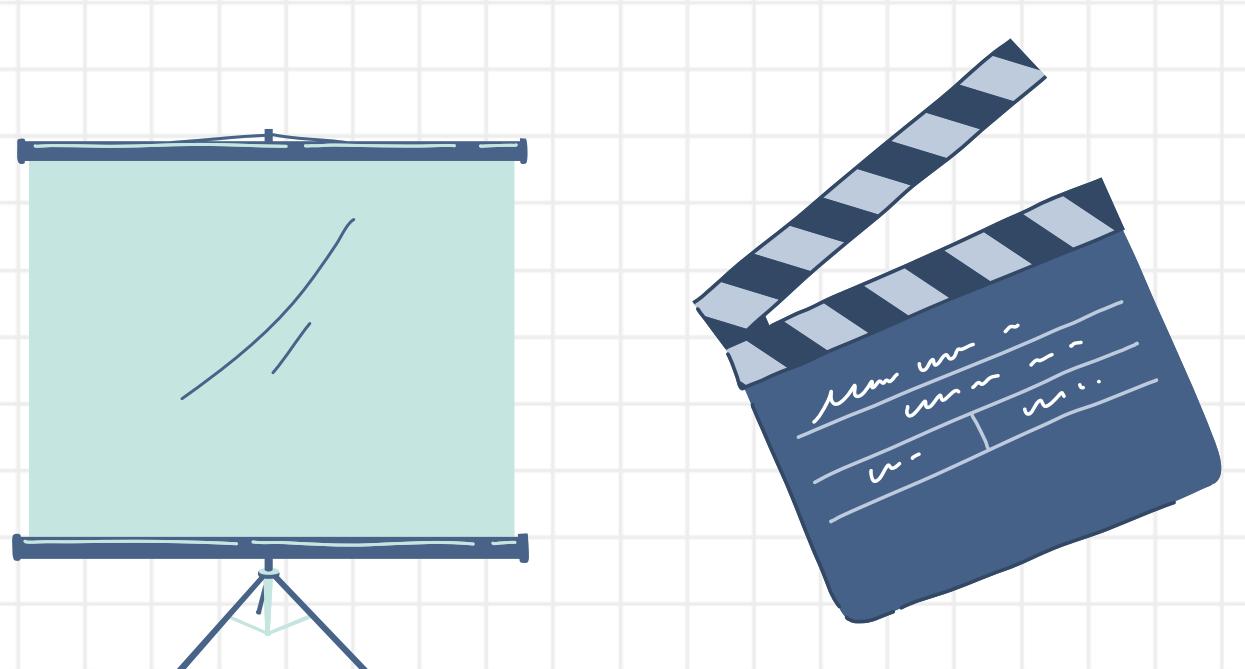


Monthly releases of Movies and TV shows on Netflix

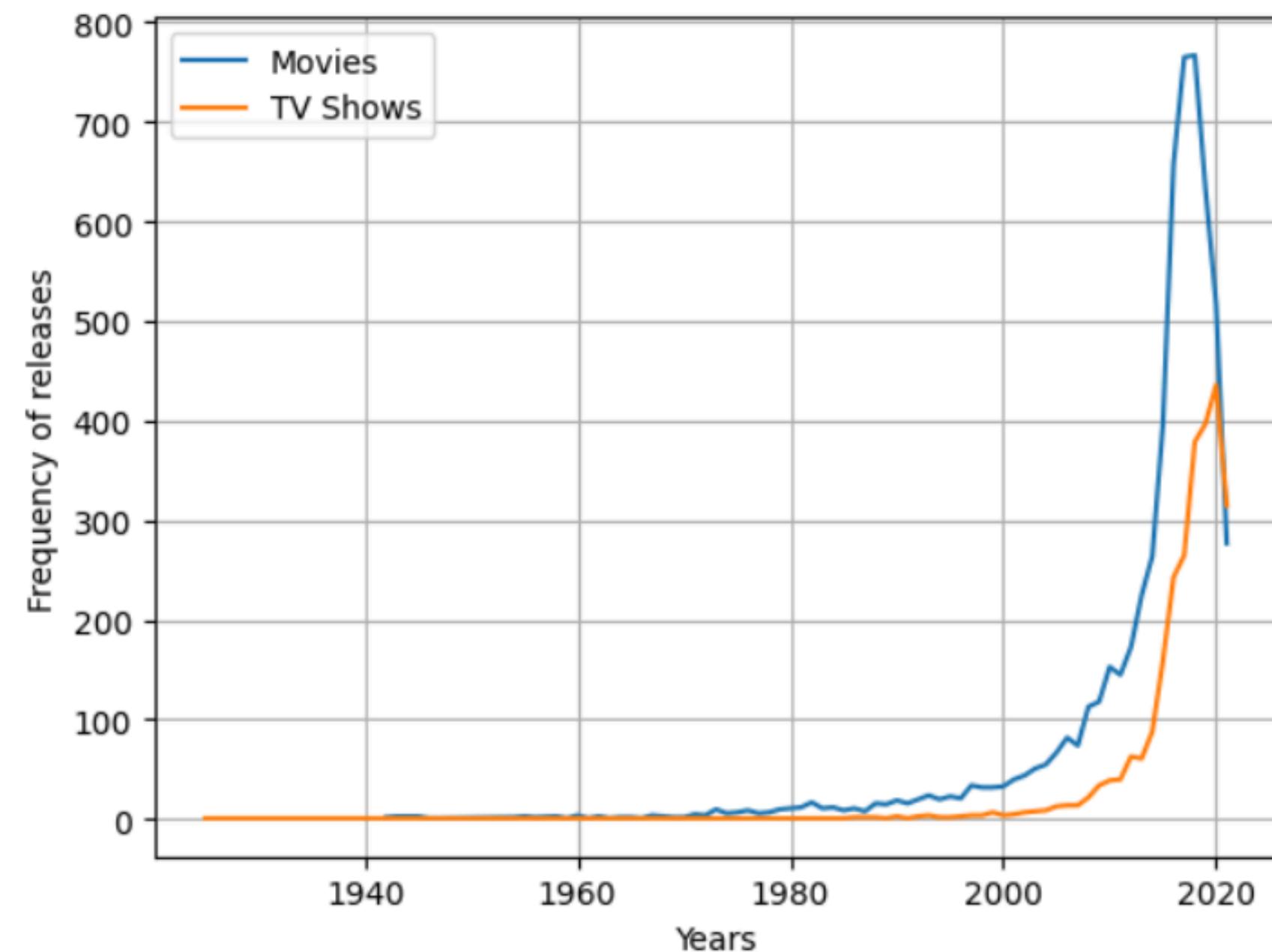


Yearly Releases Of Movies & TV Shows

```
yearly_movie_releases=df[df['type']=='Movie']['release_year'].value_counts().sort_index()  
yearly_series_releases=df[df['type']=='TV Show']['release_year'].value_counts().sort_index()  
plt.plot(yearly_movie_releases.index,yearly_movie_releases.values, label='Movies')  
plt.plot(yearly_series_releases.index,yearly_series_releases.values, label='TV Shows')  
plt.xlabel("Years")  
plt.ylabel("Frequency of releases")  
plt.grid(True)  
plt.suptitle("Yearly releases of Movies and TV Shows on Netflix")  
plt.legend()
```



Yearly releases of Movies and TV Shows on Netflix



Ratings On Netflix

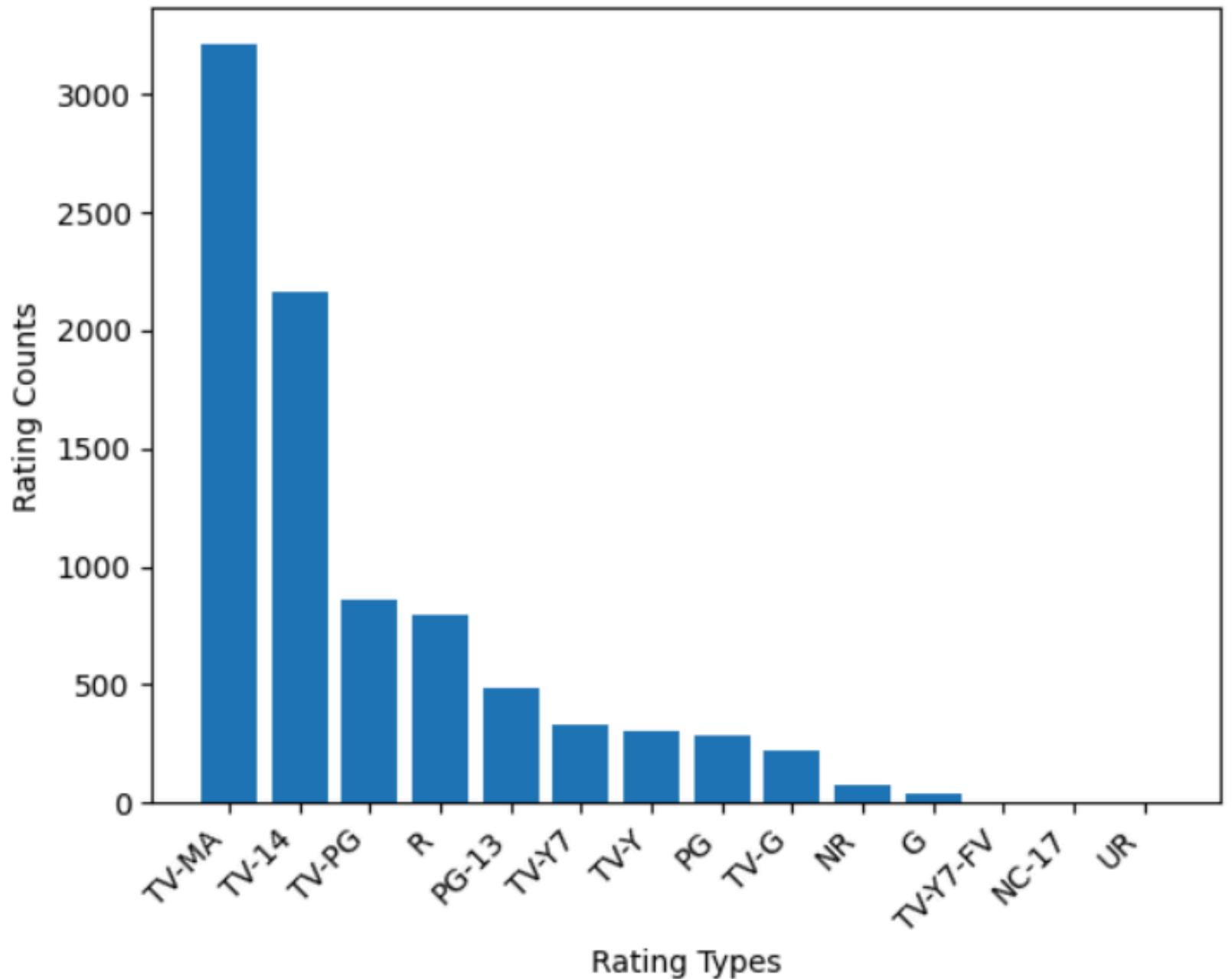
```
df['rating'].value_counts()
```

```
rating
TV-MA      3205
TV-14      2157
TV-PG      861
R          799
PG-13      490
TV-Y7      333
TV-Y       306
PG          287
TV-G       220
NR          79
G           41
TV-Y7-FV     6
NC-17        3
UR          3
Name: count, dtype: int64
```

```
ratings=df['rating'].value_counts().reset_index().sort_values(by='count', ascending=False)

plt.bar(ratings['rating'], ratings['count'])
plt.xticks(rotation=45, ha='right')
plt.xlabel("Rating Types")
plt.ylabel("Rating Counts")
plt.suptitle('Ratings on Netflix', fontsize=20)
```

Ratings on Netflix



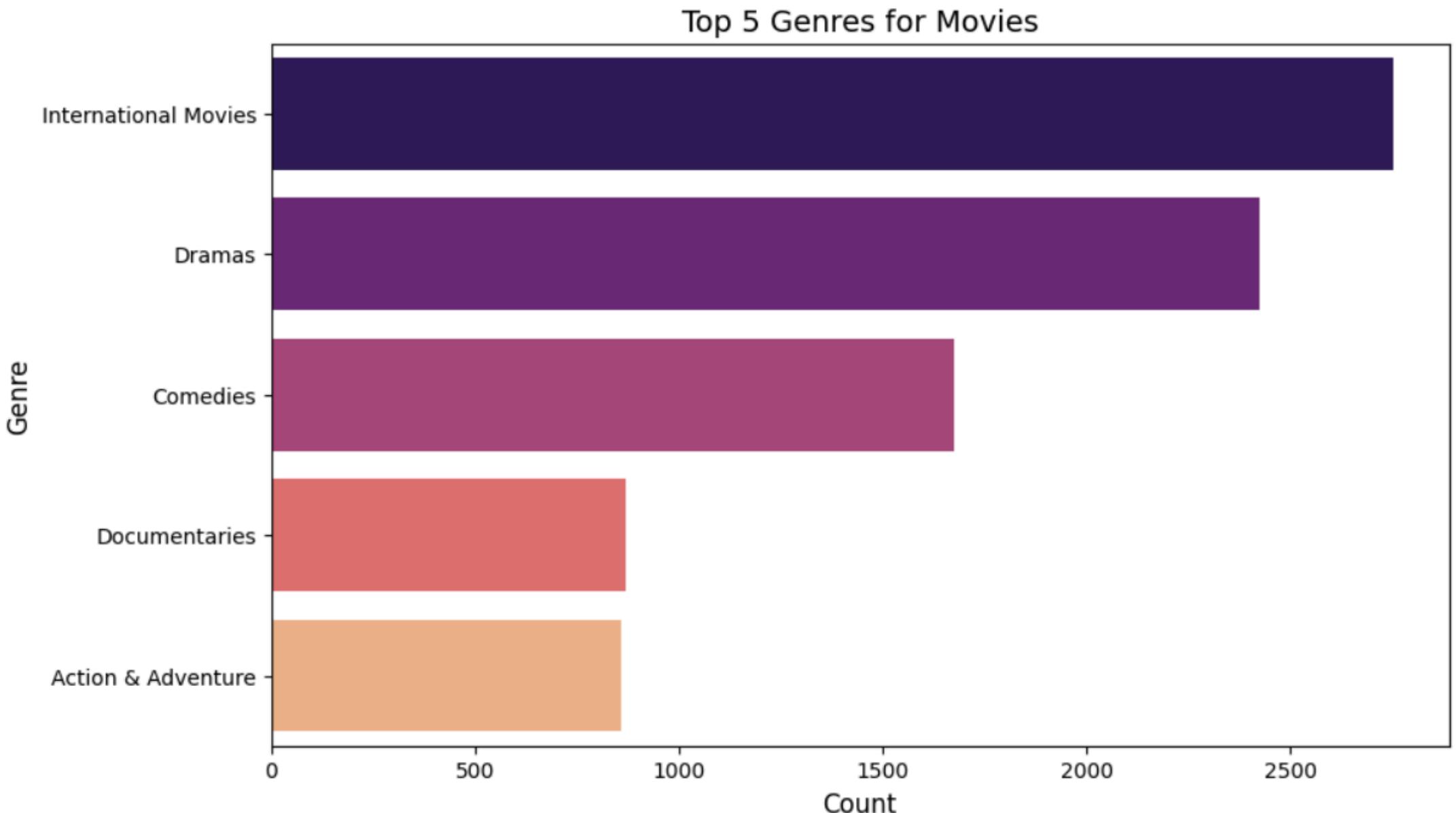
Top 5 Genres For Movies

```
#Top 5 genres for movies
df['listed_in'] = df['listed_in'].fillna('Unknown')
movie_genres = df[df['type'] == 'Movie']['listed_in'].str.split(', ').explode()
top_genres = movie_genres.value_counts().head(5)
print("\nTop 5 Genres for Movies:\n")
genre_table = top_genres.reset_index()
genre_table.columns = ['Genre', 'Count']
print(genre_table)

plt.figure(figsize=(10, 6))
sns.barplot(x='Count', y='Genre', data=genre_table, palette='magma')
plt.title('Top 5 Genres for Movies', fontsize=14)
plt.xlabel('Count', fontsize=12)
plt.ylabel('Genre', fontsize=12)
plt.show()
```

Top 5 Genres for Movies:

	Genre	Count
0	International Movies	2752
1	Dramas	2426
2	Comedies	1674
3	Documentaries	869
4	Action & Adventure	859



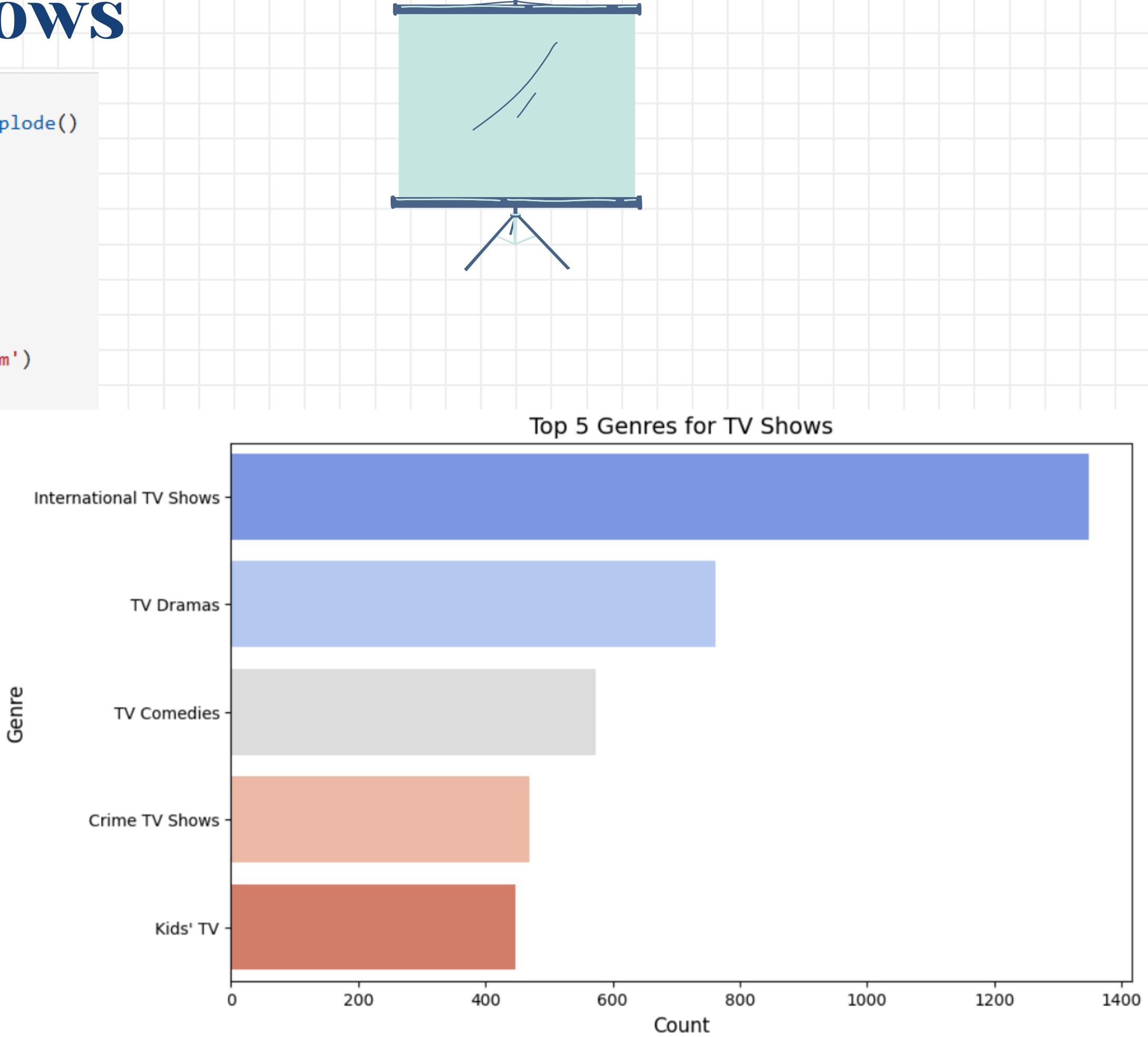
Top 5 Genres For TV Shows

```
# Top 5 genres for TV shows
tv_genres = df[df['type'] == 'TV Show']['listed_in'].str.split(', ').explode()
top_tv_genres = tv_genres.value_counts().head(5)
print("\nTop 5 Genres for TV Shows:\n")
tv_genre_table = top_tv_genres.reset_index()
tv_genre_table.columns = ['Genre', 'Count']
print(tv_genre_table)

plt.figure(figsize=(10, 6))
sns.barplot(x='Count', y='Genre', data=tv_genre_table, palette='coolwarm')
plt.title('Top 5 Genres for TV Shows', fontsize=14)
plt.xlabel('Count', fontsize=12)
plt.ylabel('Genre', fontsize=12)
plt.show()
```

Top 5 Genres for TV Shows:

	Genre	Count
0	International TV Shows	1349
1	TV Dramas	762
2	TV Comedies	573
3	Crime TV Shows	469
4	Kids' TV	448



Top 10 Directors by Number Of TV Shows

```
# Remove rows where the 'director' column is NaN or contains "Not Given"
tv_shows_filtered = tv_shows[tv_shows['director'].notna()] # Filters out NaN values

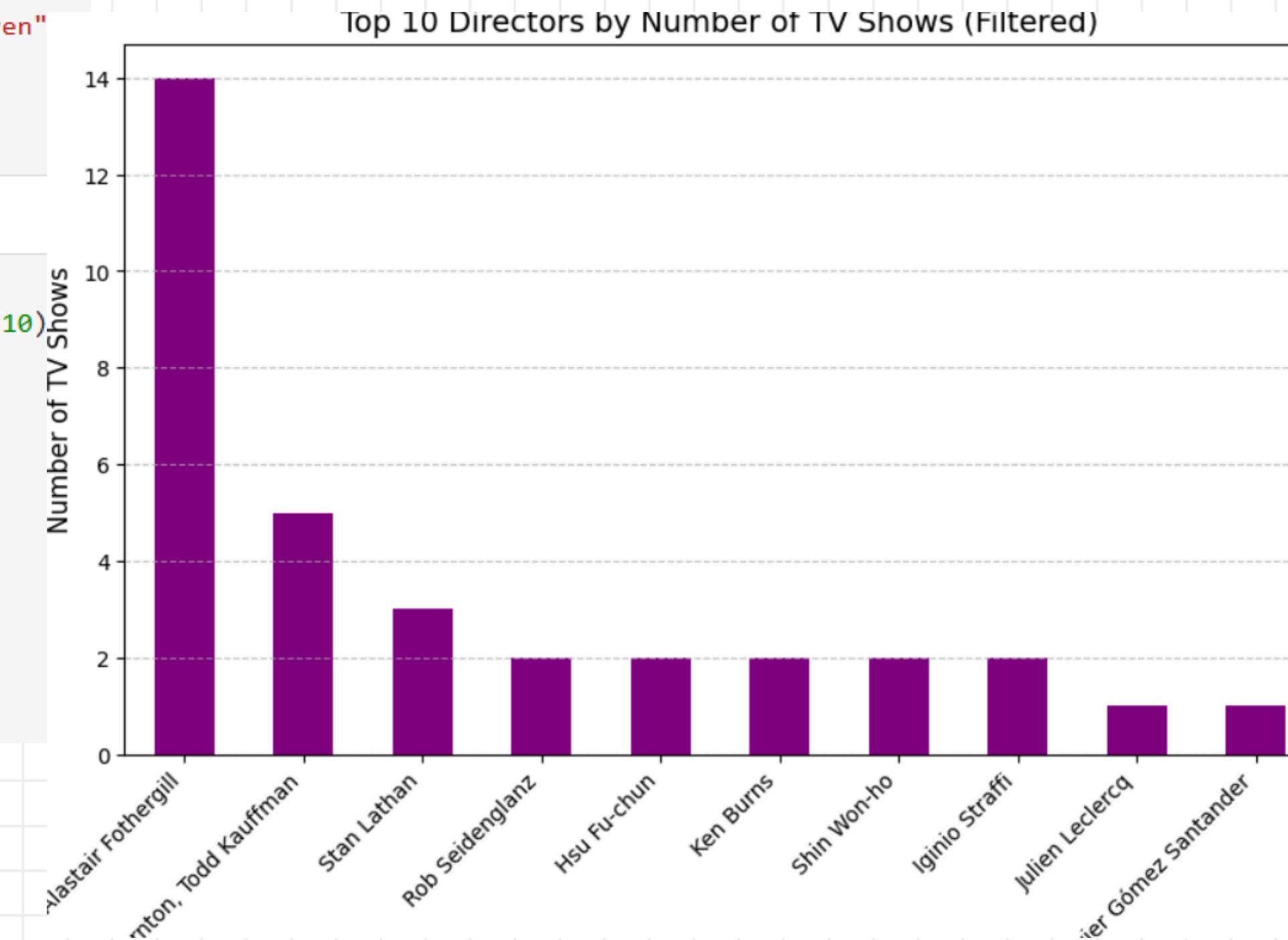
# Alternatively, if "Not Given" is a placeholder in the director column:
tv_shows_filtered = tv_shows_filtered[tv_shows_filtered['director'] != "Not Given"]

# Verify the filtering
print(tv_shows_filtered['director'].isnull().sum()) # Should print 0

0

# Count the frequency of TV shows by director after filtering
tv_show_directors_filtered = tv_shows_filtered['director'].value_counts().head(10)

# Plot the results
plt.figure(figsize=(10, 6))
tv_show_directors_filtered.plot(kind='bar', color='purple')
plt.title('Top 10 Directors by Number of TV Shows (Filtered)', fontsize=14)
plt.xlabel('Director', fontsize=12)
plt.ylabel('Number of TV Shows', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



Top 10 Directors By Number of Movies

```
# Remove rows where the 'director' column is NaN or contains "Not Given"
movies_filtered = movies[movies['director'].notna()] # Filters out NaN values

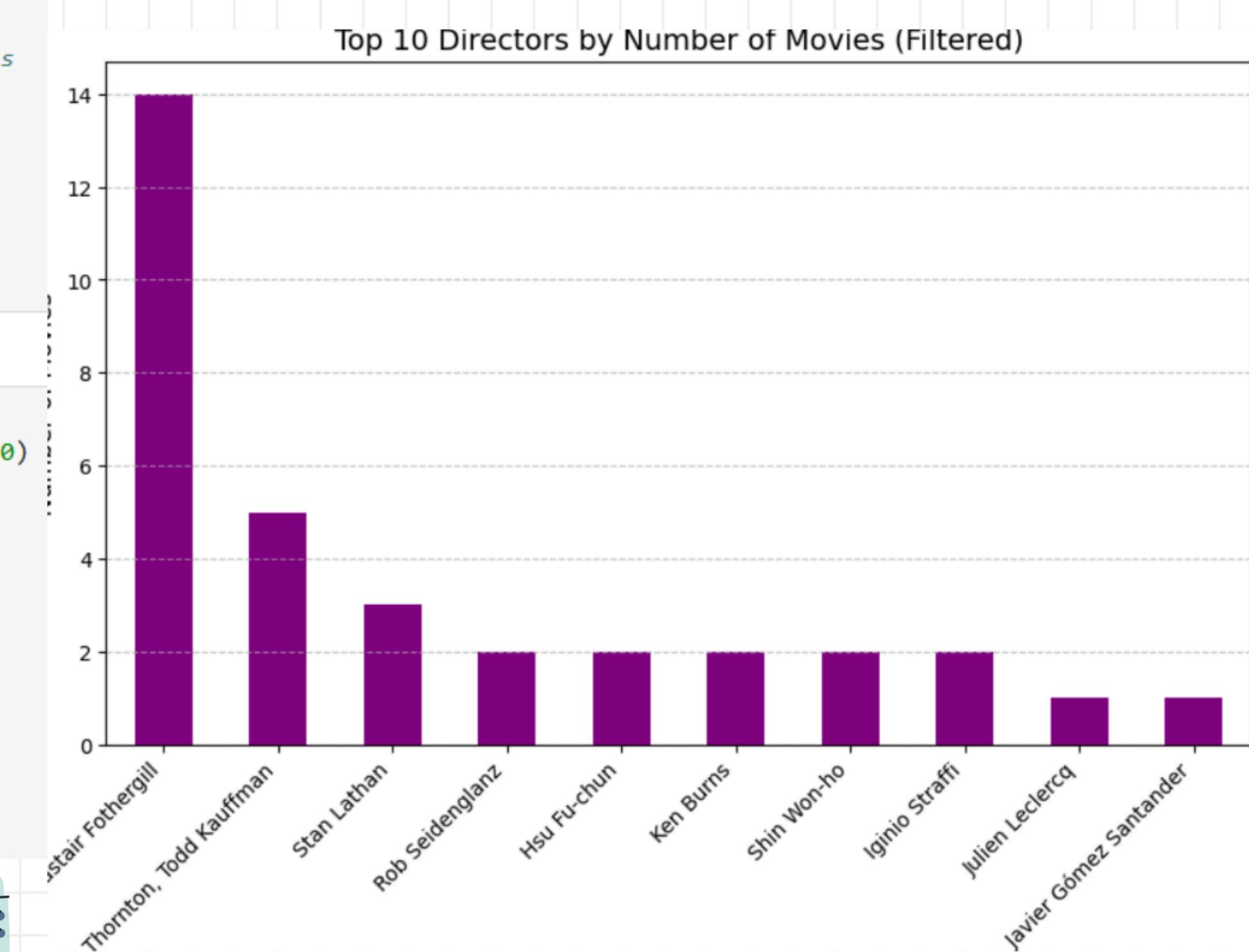
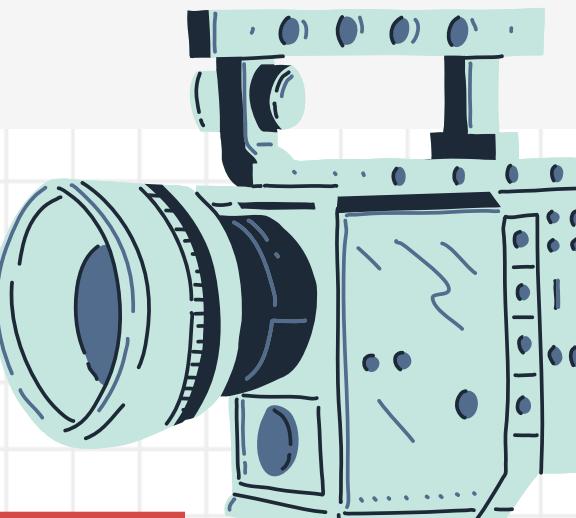
# Alternatively, if "Not Given" is a placeholder in the director column:
movies_filtered = movies_filtered[movies_filtered['director'] != "Not Given"]

# Verify the filtering
print(movies_filtered['director'].isnull().sum()) # Should print 0

0

# Count the frequency of TV shows by director after filtering
movies_directors_filtered = movies_filtered['director'].value_counts().head(10)

# Plot the results
plt.figure(figsize=(10, 6))
tv_show_directors_filtered.plot(kind='bar', color='purple')
plt.title('Top 10 Directors by Number of Movies (Filtered)', fontsize=14)
plt.xlabel('Director', fontsize=12)
plt.ylabel('Number of Movies', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



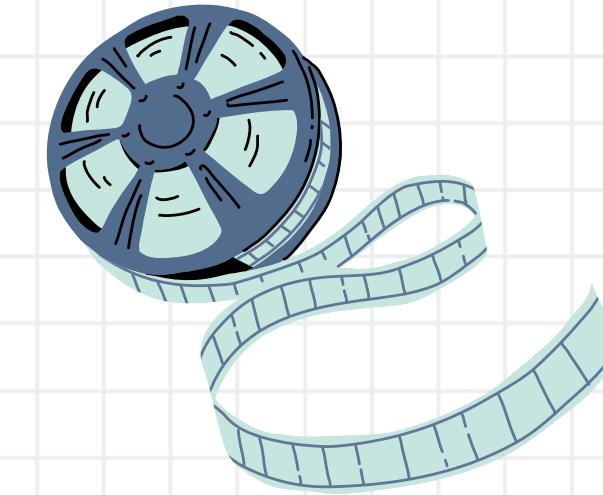
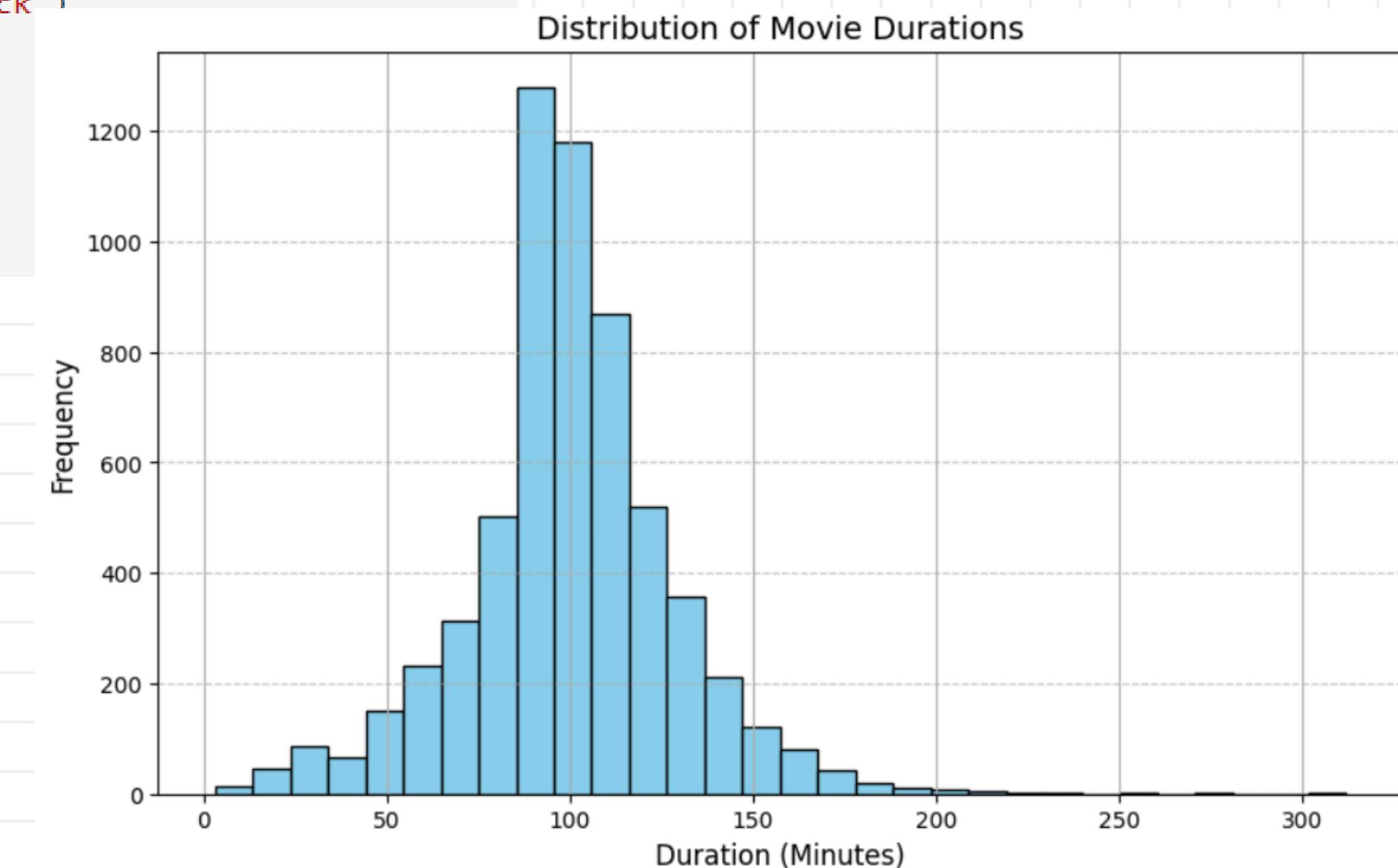
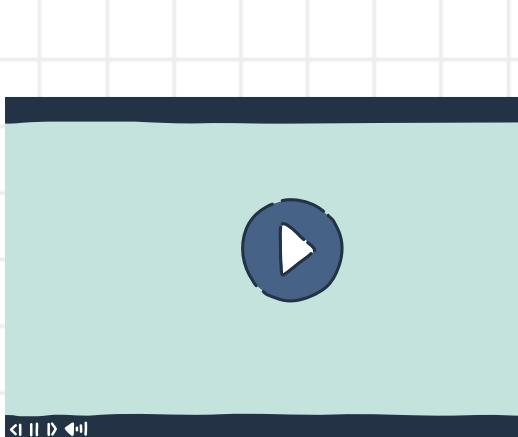
Duration Of Movies

```
# Filter data for movies
movies = df[df['type'] == 'Movie']

# Extract numeric values from the 'duration' column for movies
movies['duration_numeric'] = movies['duration'].str.replace(' min', '', regex=False).astype(float)

# Plot the distribution of movie durations
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
movies['duration_numeric'].hist(bins=30, color='skyblue', edgecolor='black')
plt.title('Distribution of Movie Durations', fontsize=14)
plt.xlabel('Duration (Minutes)', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

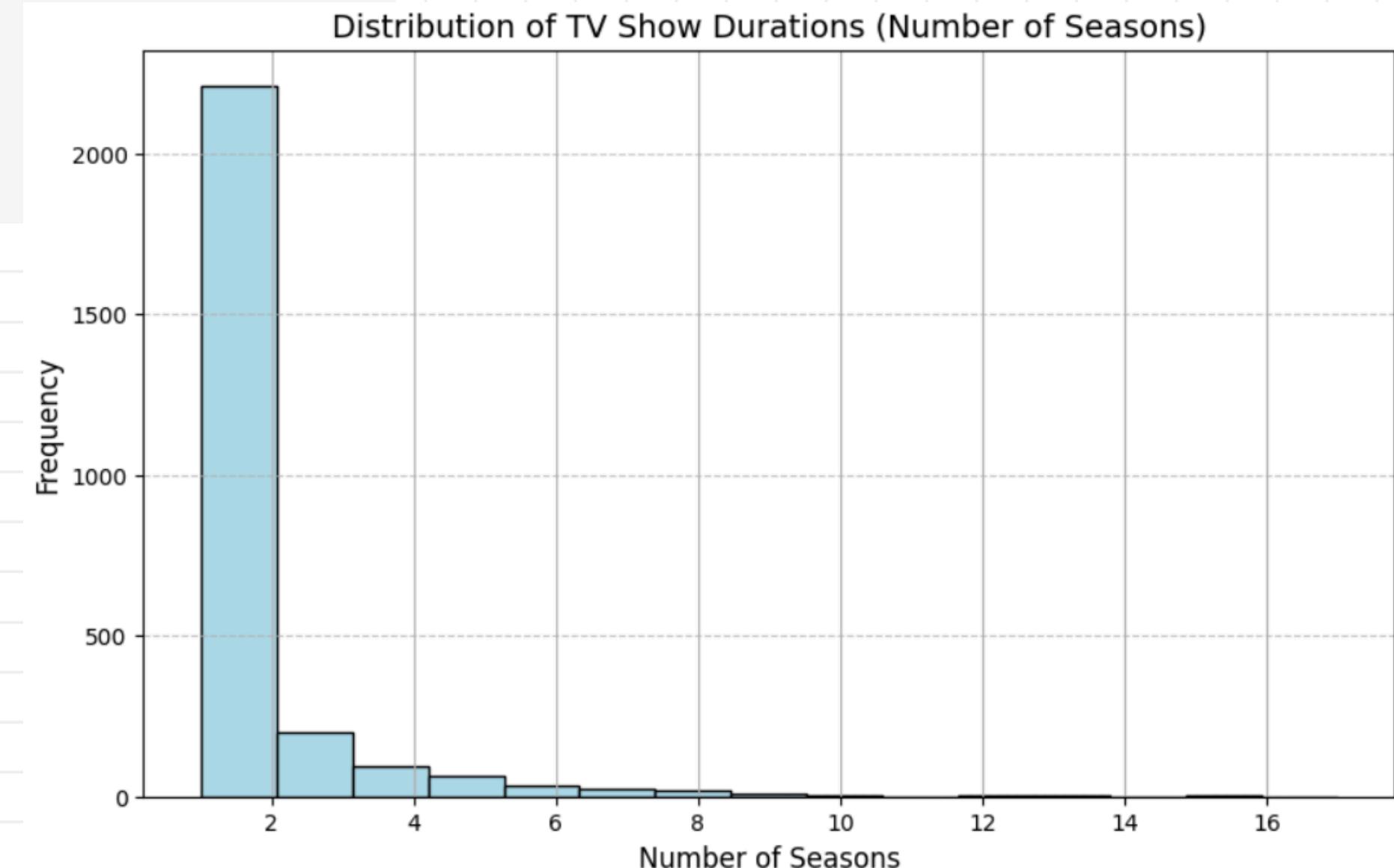
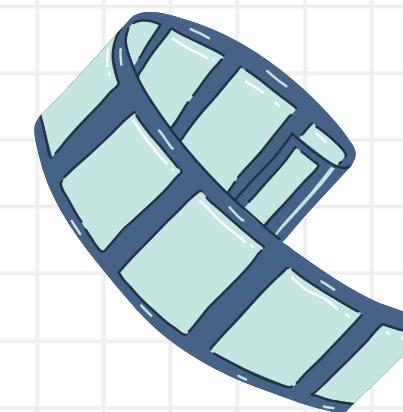
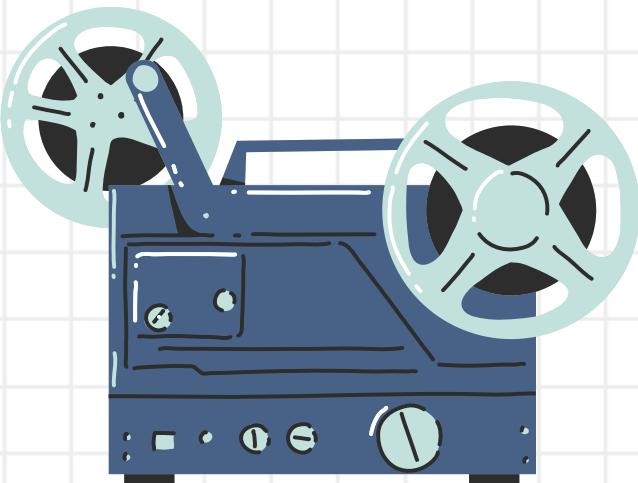


Number Of Seasons Of TV Shows

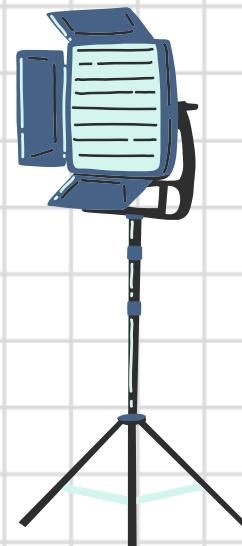
```
# Filter data for TV shows
tv_shows = df[df['type'] == 'TV Show']

# Extract numeric values from the 'duration' column
tv_shows['num_seasons'] = tv_shows['duration'].str.extract('(\d+)').astype(float)

# Plot the distribution of the number of seasons
plt.figure(figsize=(10, 6))
tv_shows['num_seasons'].hist(bins=15, color='lightblue', edgecolor='black')
plt.title('Distribution of TV Show Durations (Number of Seasons)', fontsize=14)
plt.xlabel('Number of Seasons', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



Insights



1. Content Type Distribution:

- Movies dominate with approximately 70%, while TV Shows contribute 30%.

2. Top 10 Countries with Most Content:

- The leading contributors include the USA (~30%) and India (~10%), United Kingdom (~7%) followed by other countries with decreasing percentages.

3. Year-wise Content Addition:

- Steady growth observed, with a significant spike around 2018-2020, contributing ~50% of the total content.

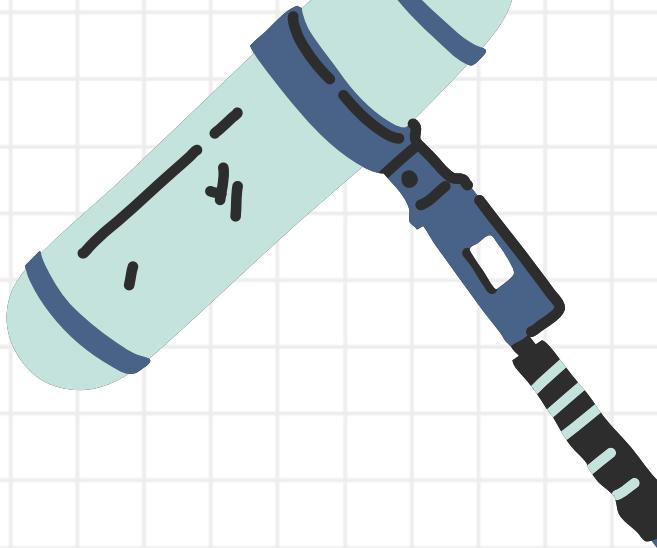
4. Top Genres:

- Dominated by genres like Drama (40%), Comedy (30%), and others trailing behind.

5. Content Ratings:

- Majority rated TV-MA (30%), followed by TV-14 (25%), showcasing a focus on mature audiences.

Insights



6. Duration of Movies:

- Average movie duration is around 100 minutes, with most ranging between 80-120 minutes.

7. TV Show Seasons:

- Most TV Shows have 1-2 seasons, with longer series being less common.

8. Top Directors:

- Directors like A. Fothergill and W.M.Thornton & T. Fauffman, with each contributing to ~5-10% of the total.

9. Content by Release Date:

- A clear rise in content added post-2015, highlighting Netflix's growth phase.

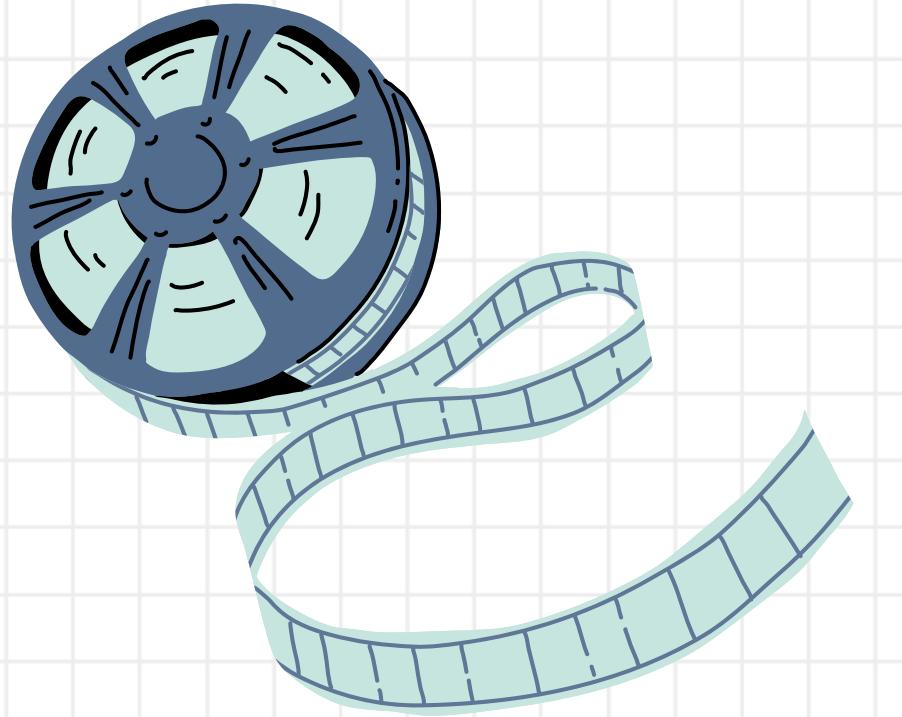
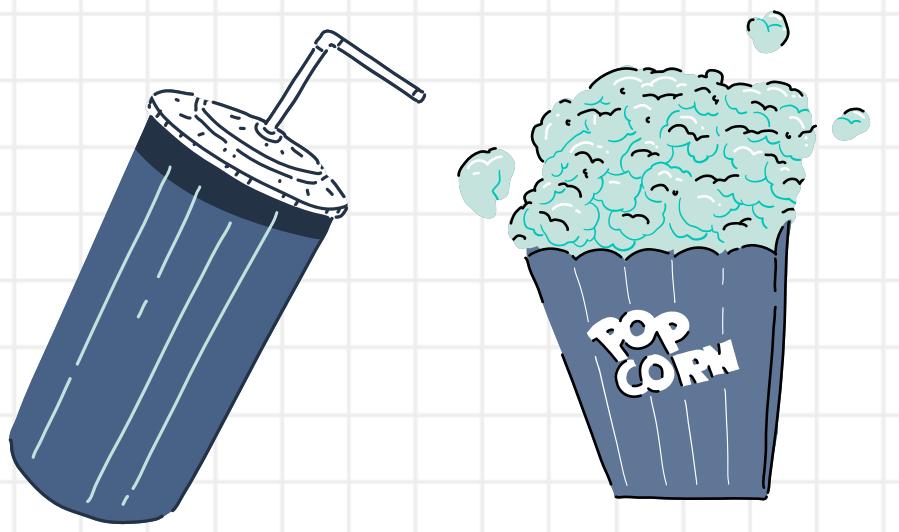


Conclusion

This project provides key insights into Netflix's content library, including distribution by type, genre, country, and trends over time. Through data analysis and visualization, we highlighted Netflix's growth, content diversity, and audience focus, showcasing its evolving platform dynamics.



NETFLIX



Thank You

