Netflix

Data Cleaning, Analysis and Visualization

Objective:

- Import and clean the Netflix data set using pandas in python.
- Analyze data trends and distributions using summary statistics and visualizations in python.
- Create charts like bar charts pie charts using Matplotlib and Seaborn.
- Extract key trends and patterns, such as top and most common genre.

Importing Libraries:

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

Data Analysis:

```
[22]  data.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
```

```
[26]
#Prints first 5 rows of the dataset
data.head()
```

	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	9/25/2021	2020	PG-13	90 min	Documentaries
1	s3	TV Show	Ganglands	Julien Leclercq	France	9/24/2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...
2	s6	TV Show	Midnight Mass	Mike Flanagan	United States	9/24/2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries
3	s14	Movie	Confessions of an Invisible Girl	Bruno Garotti	Brazil	9/22/2021	2021	TV-PG	91 min	Children & Family Movies, Comedies
4	s8	Movie	Sankofa	Haile Gerima	United States	9/24/2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies

Data Analysis:

```
[27]
#Gives the Top Ratings
print("Top Ratings Counts:")
data.rating.value_counts()
```

```
Top Ratings 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
```

Data Cleaning:
Adjusting Date Format

```
[28]
    # Convert 'date_added' to datetime
    data['date_added'] = pd.to_datetime(data['date_added'], errors='coerce')

    # Print the first few rows of the DataFrame to check the conversion
    print(data[['date_added']].head())
```

```
...   date_added
0 2021-09-25
1 2021-09-24
2 2021-09-24
3 2021-09-22
4 2021-09-24
```

Data Cleaning:
Checking for Null Values

```
[23]  data.size

...   87900
```

```
[24]  #Counts the null values in the dataset
      data.isnull().value_counts()

...   show_id  type   title  director  country  date_added  release_year  rating  duration  listed_in
      False    False  False  False     False    False       False         False   False     False       8790
      Name: count, dtype: int64
```

Data Cleaning:
Removing Duplicates

```
#Dropping duplicates
data=data.drop_duplicates()
data
```

[25]

	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	9/25/2021	2020	PG-13	90 min	Documentaries
1	s3	TV Show	Ganglands	Julien Leclercq	France	9/24/2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...
2	s6	TV Show	Midnight Mass	Mike Flanagan	United States	9/24/2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries
3	s14	Movie	Confessions of an Invisible Girl	Bruno Garotti	Brazil	9/22/2021	2021	TV-PG	91 min	Children & Family Movies, Comedies
4	s8	Movie	Sankofa	Haile Gerima	United States	9/24/2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies
...	...	...	...	...	...	...	...	...	...	...
8785	s8797	TV Show	Yunus Emre	Not Given	Turkey	1/17/2017	2016	TV-PG	2 Seasons	International TV Shows, TV Dramas
8786	s8798	TV Show	Zak Storm	Not Given	United States	9/13/2018	2016	TV-Y7	3 Seasons	Kids' TV
8787	s8801	TV Show	Zindagi Gulzar Hai	Not Given	Pakistan	12/15/2016	2012	TV-PG	1 Season	International TV Shows, Romantic TV Shows, TV ...
8788	s8784	TV Show	Yoko	Not Given	Pakistan	6/23/2018	2016	TV-Y	1 Season	Kids' TV
8789	s8786	TV Show	YOM	Not Given	Pakistan	06-07-2018	2016	TV-Y7	1 Season	Kids' TV

8790 rows × 10 columns

Data Cleaning:

```
[29]
# Data Cleaning
print("Missing Values Before Cleaning:")
data.isnull().sum()
```

```
Missing Values Before Cleaning:

show_id            0
type               0
title              0
director           0
country            0
date_added      4386
release_year       0
rating             0
duration           0
listed_in          0
dtype: int64
```

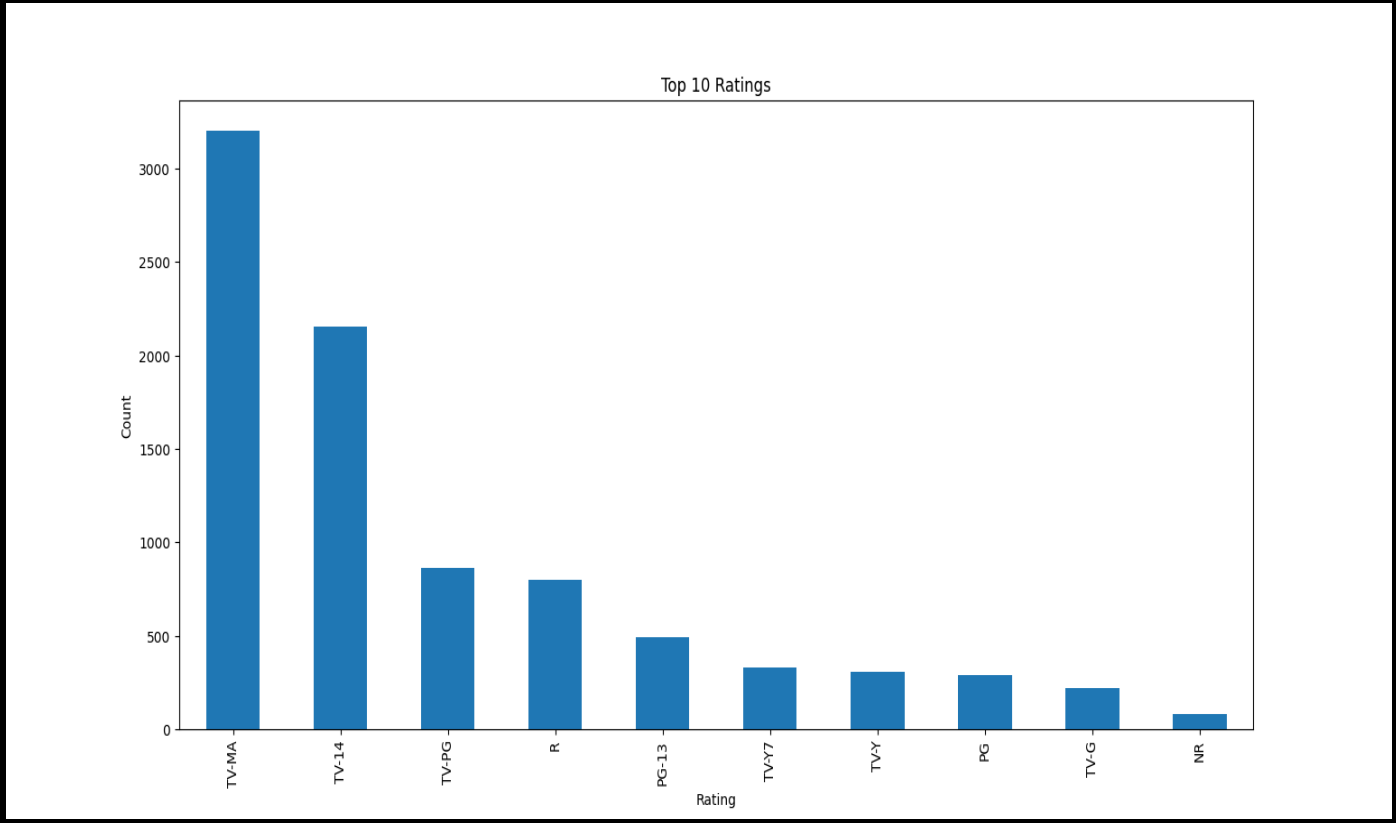
```
[30]
data.drop_duplicates(inplace=True)
data['date_added'] = data['date_added'].fillna(pd.to_datetime('today'))
print("Missing Values After Cleaning:")
data.isnull().sum()
```

```
Missing Values After Cleaning:

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
```

Data Visualization:

Top 10 ratings

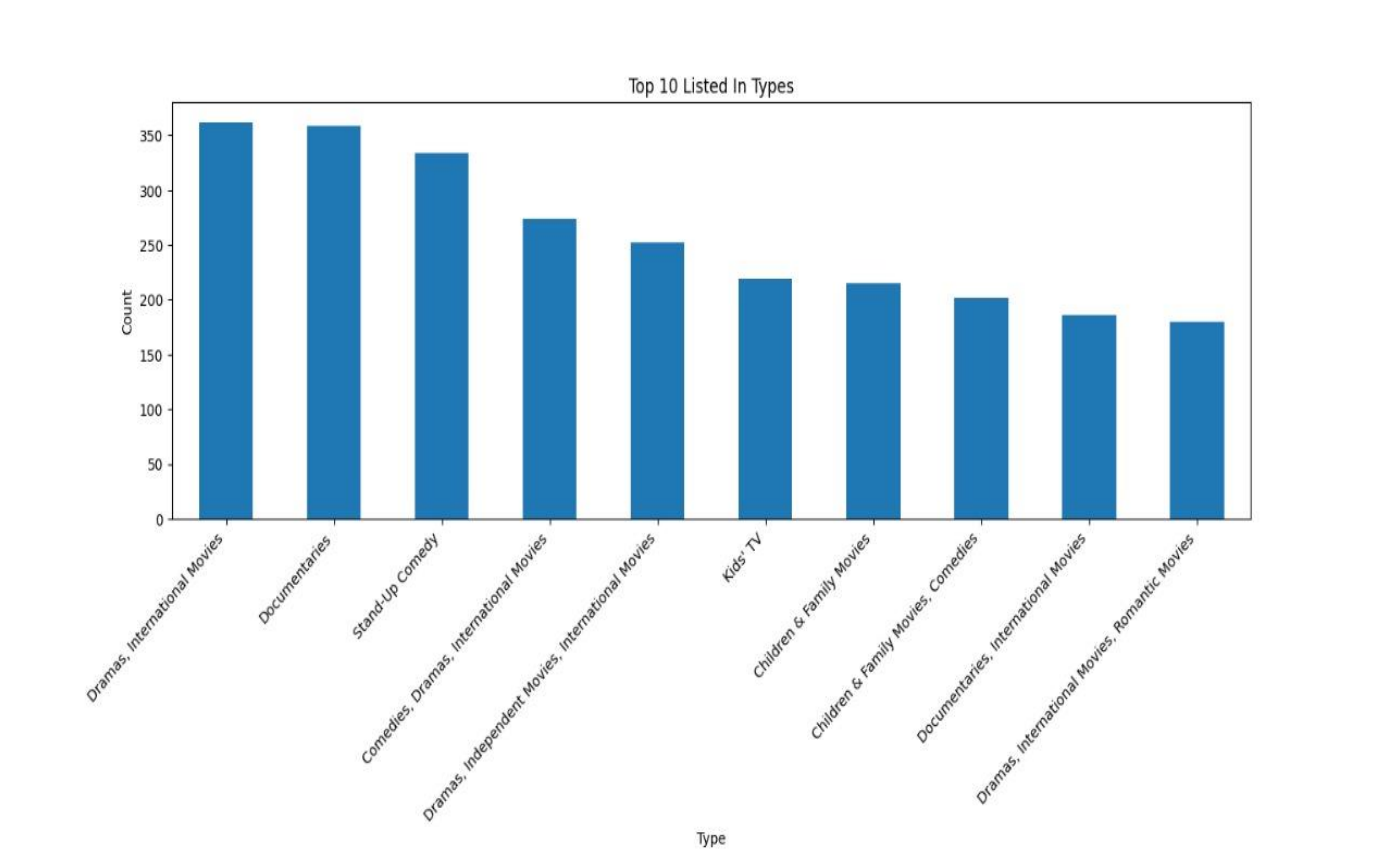
```
# Keep 'rating' as a categorical variable
data['rating'] = data['rating'].astype(str)
data['Years']=data['release_year']
year = data['rating'].value_counts()
year[:10].plot(kind='bar')
plt.title('Top 10 Ratings')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.show()
```



Data Visualization:

Top 10 Listed in Genre Types

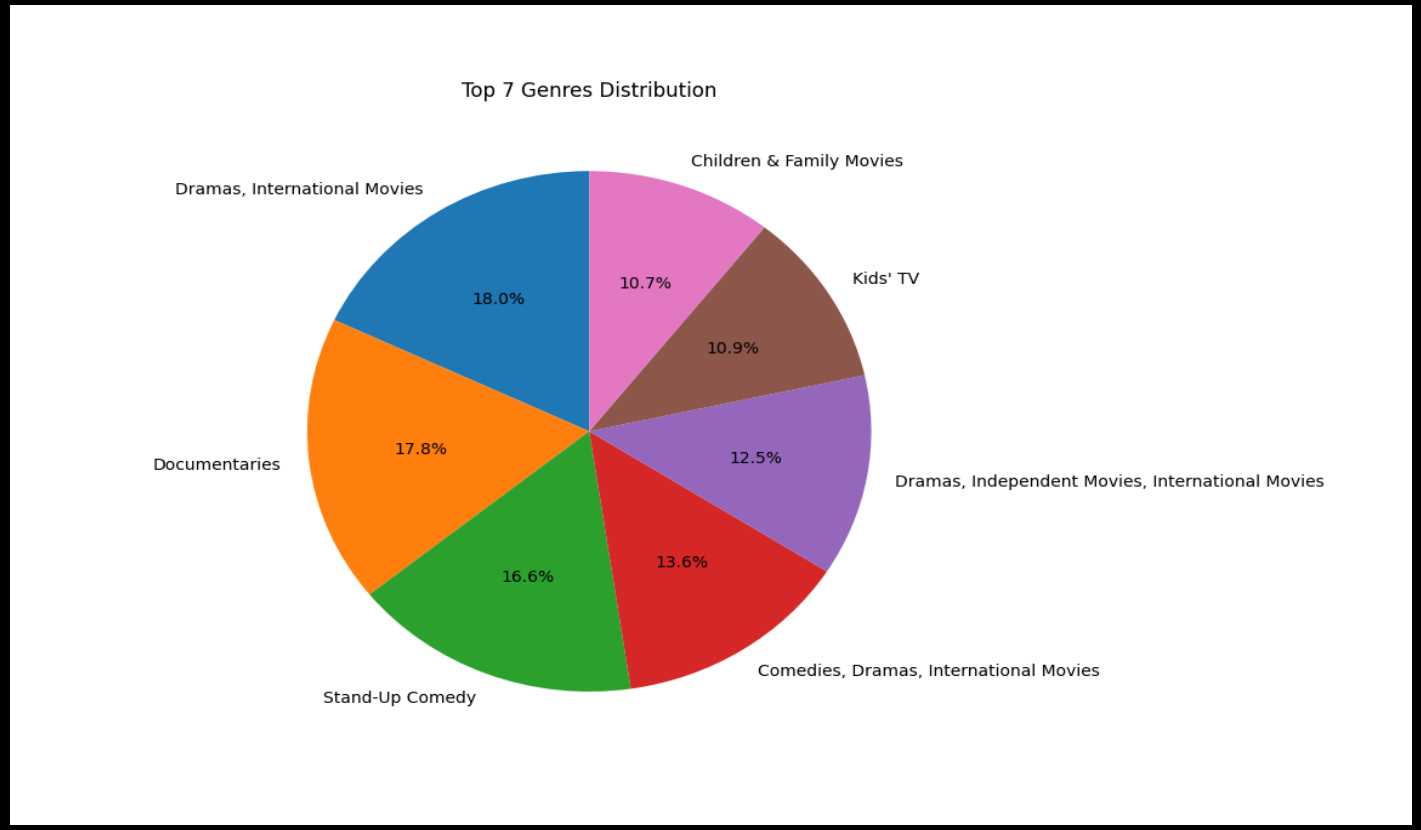
```
#Top 10 Genre Lists
top_10_types = data['listed_in'].value_counts()
top_10_types[:10].plot(kind='bar')
plt.title('Top 10 Listed In Genre Types')
plt.xlabel('Type of Genres')
plt.xticks(rotation=45, ha='right')
plt.ylabel('Count')
plt.show()
```



Data Visualization:

Top 7 Genres Distribution

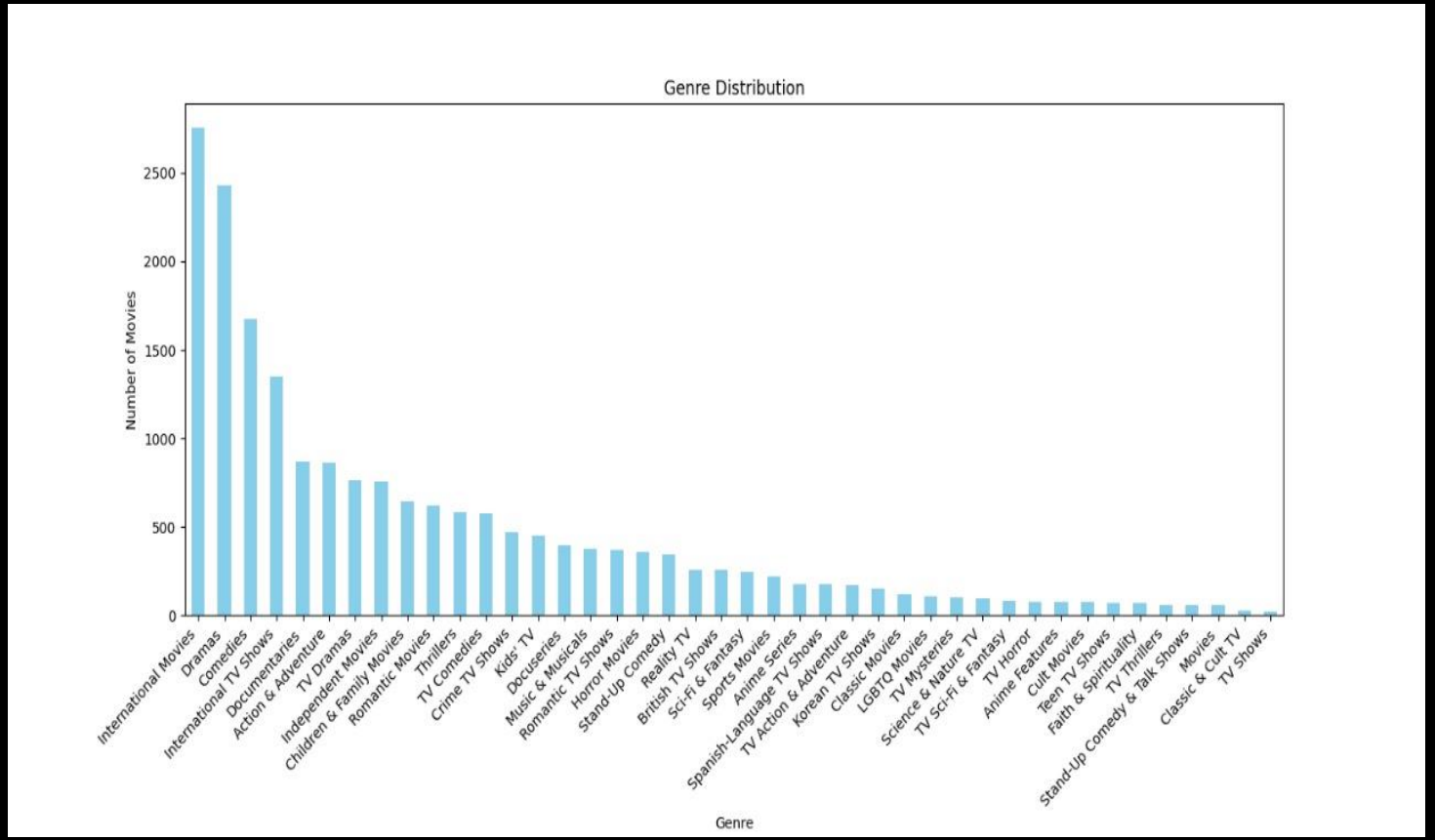
```
#Top 7 Genre Distribution
genre_list = data['listed_in'].value_counts()
genre_list[:7].plot(kind='pie', autopct='%1.1f%%', startangle=90)
plt.title('Top 7 Genres Distribution')
plt.ylabel('')
plt.show()
```



Data Visualization:

Genre Distribution

```
# Count the occurrences of each genre
genre_counts = genres_series.value_counts()

# Plot a bar chart for the genre distribution
genre_counts.plot(kind='bar', figsize=(10, 6), color='skyblue')
plt.title('Genre Distribution')
plt.xlabel('Genre')
plt.ylabel('Number of Movies')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for
plt.show()
```



Data Visualization:

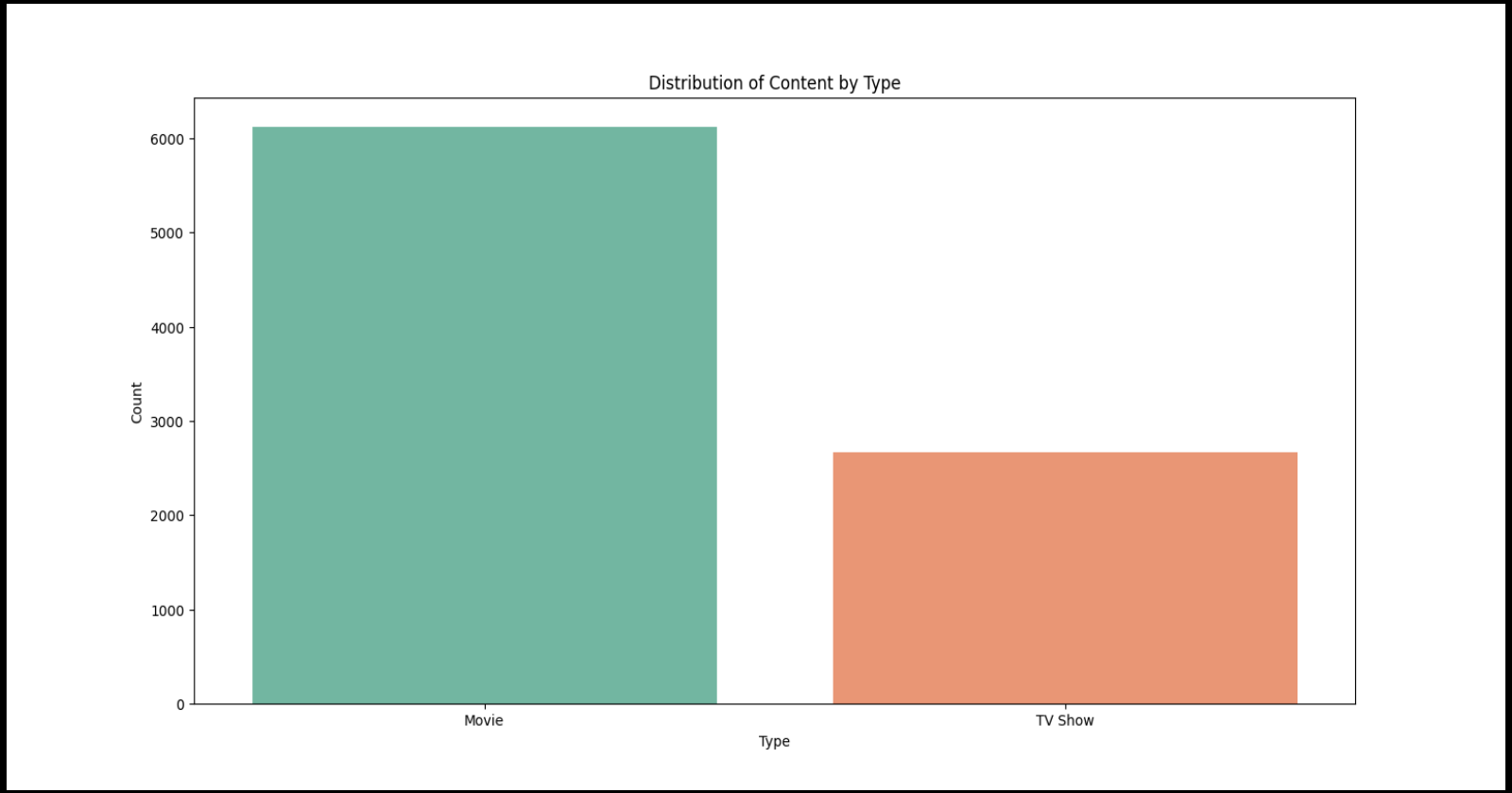
Tree Map of Genre Distribution

```
# Count the occurrences of each genre
genre_counts = genres_series.value_counts().reset_index()
genre_counts.columns = ['Genre', 'Count']
fig = px.treemap(genre_counts, path=['Genre'], values='Count', title='Genre Distribution')
fig.show()
```

Data Visualization:

Distribution of Content by Type

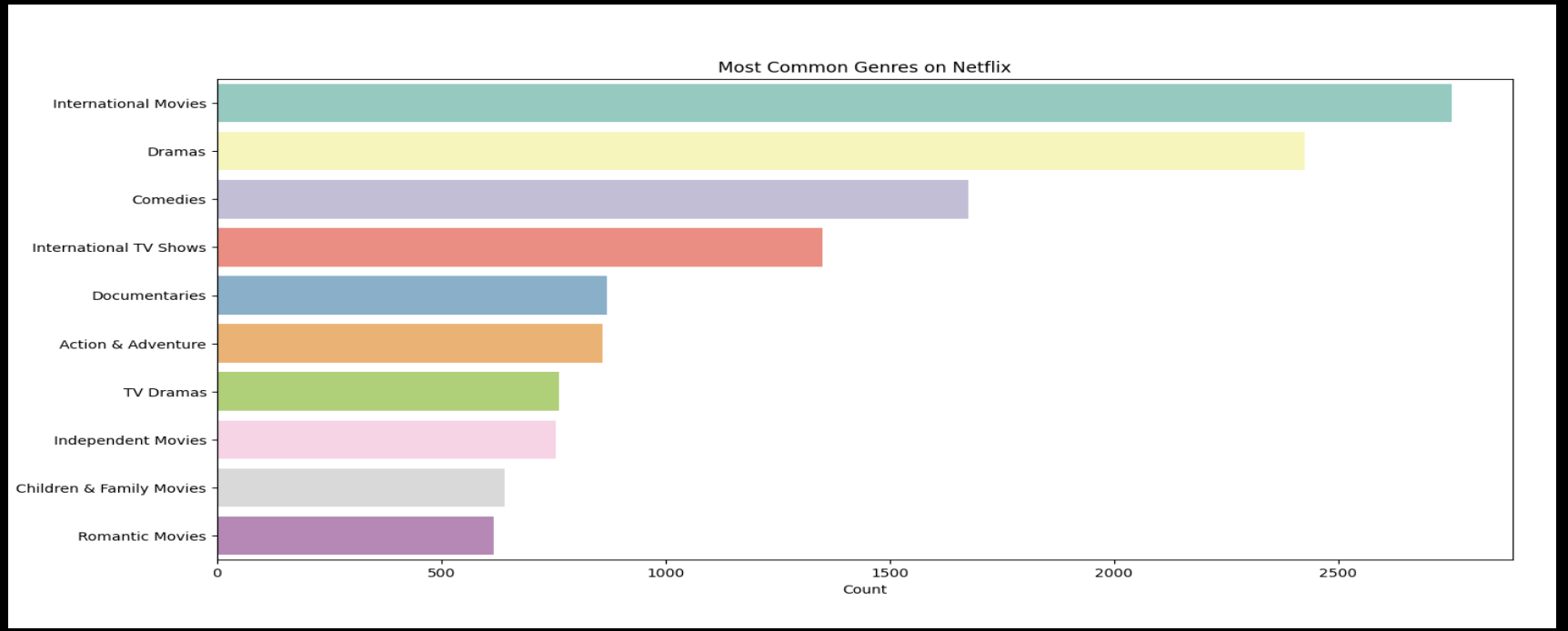
```
# Exploratory Data Analysis (EDA)
plt.figure(figsize=(8, 6))
# Explicitly assign 'x' variable to 'hue' and set legend=False to remove the legend
sns.barplot(x=data['type'].value_counts().index,
            y=data['type'].value_counts().values,
            hue=data['type'].value_counts().index,  # Assign 'x' variable to 'hue'
            palette='Set2',
            dodge=False)  # No need for dodge if using one group
# Add title and labels
plt.title('Distribution of Content by Type')
plt.xlabel('Type')
plt.ylabel('Count')
# Remove the legend as we're using hue, and the legend is unnecessary
plt.legend([], [], frameon=False)
plt.show()
```



Data Visualization:

Most common Genres on Netflix

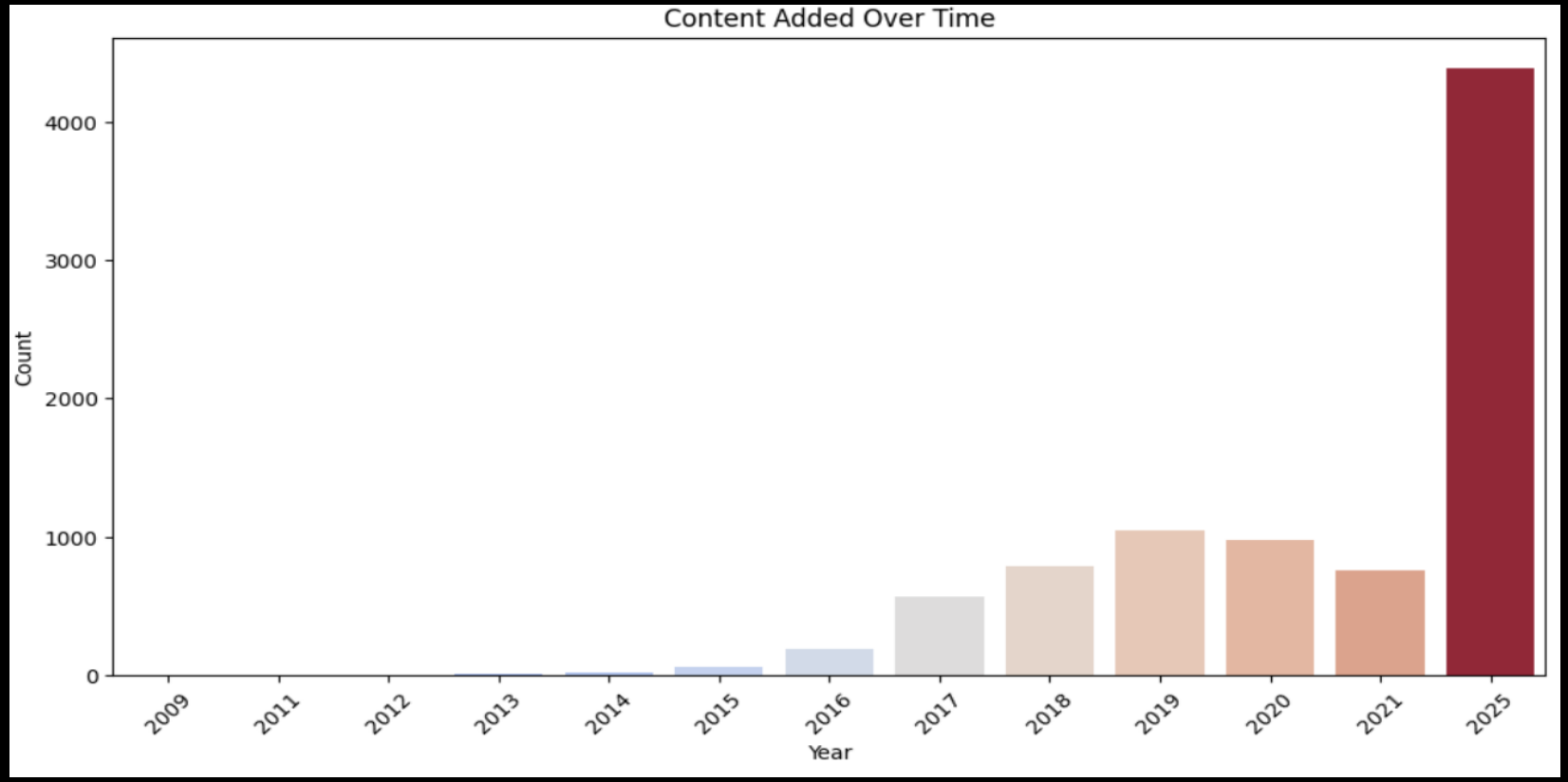
```
# Most Common Genres
data['genres'] = data['listed_in'].apply(lambda x: x.split(', '))
all_genres = sum(data['genres'], [])
genre_counts = pd.Series(all_genres).value_counts().head(10)

# Plotting
plt.figure(figsize=(10, 6))

# Explicitly assigning 'y' to 'hue' to avoid the deprecation warning
sns.barplot(x=genre_counts.values,
            y=genre_counts.index,
            hue=genre_counts.values,  # Assign 'y' variable to 'hue'
            palette='Set3',
            dodge=False)

# Add title and labels
plt.title('Most Common Genres on Netflix')
plt.xlabel('Count')
plt.ylabel('Genre')

# Remove the legend since it's unnecessary in this case
plt.legend([], [], frameon=False)

plt.show()
```



Data Visualization:

Content Added Over Time

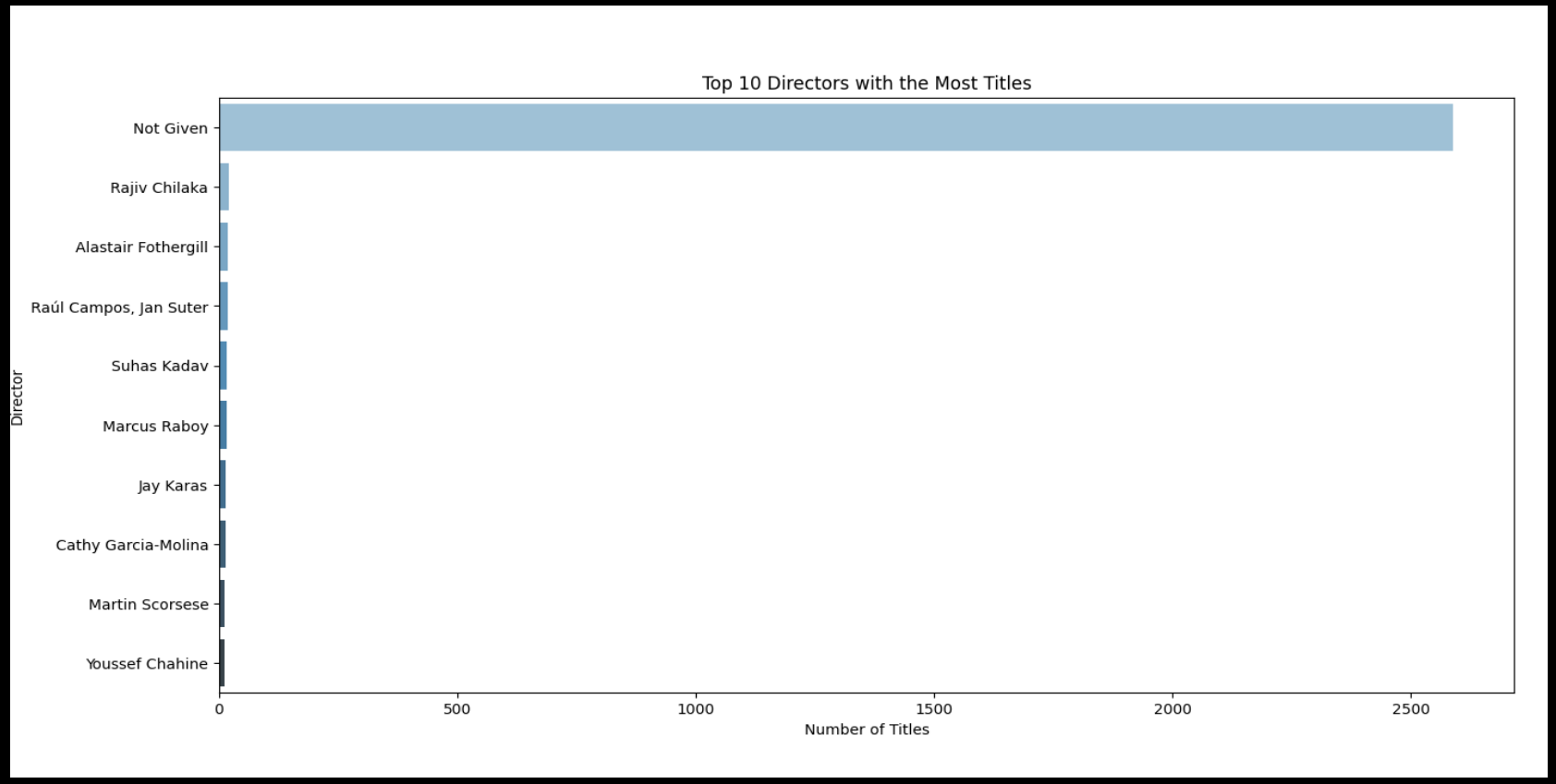
```
# Content Added Over Time
plt.figure(figsize=(12, 6))
# Explicitly assigning 'x' to 'hue' to avoid the deprecation warning
sns.countplot(x='year_added', data=data, palette='coolwarm', hue='year_added')
# Add title and labels
plt.title('Content Added Over Time')
plt.xlabel('Year')
plt.ylabel('Count')
plt.xticks(rotation=45)
# Remove the legend since it's unnecessary in this case
plt.legend([], [], frameon=False)
plt.show()
```



Data Visualization:

Top 10 Directors with Most Titles

```
# Top 10 Directors with the Most Titles
plt.figure(figsize=(10, 6))
# Plotting the bar chart without hue
sns.barplot(x=top_directors.values, y=top_directors.index)
# Add title and labels
plt.title('Top 10 Directors with the Most Titles')
plt.xlabel('Number of Titles')
plt.ylabel('Director')
plt.show()
```



Data Visualization:

Word Cloud of Movie Titles

```python
# Word Cloud of Movie Titles
movie_titles = data[data['type'] == 'Movie']['title']
wordcloud = WordCloud(width=800, height=400,
    background_color='black').generate(' '.join(movie_titles))
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Key Insights:

- Most Netflix content falls under a few dominant genres such as Drama, Comedy, Documentaries.
- Netflix adds content at varying rates over the years, highlighting trends in expansion.
- Movies dominate over other TV shows in Netflix’s content library.
- The top directors, Rajiv Chilaka, Alastair Fothergill, Raul Campos, Jan Sutas, and others, with the most Netflix content were identified.