NETFLIX

Data Analysis

By :- Subodh Kumar Yadav

Netflix Data: Cleaning, Analysis, and Visualization

Streaming platforms like Netflix generate vast amounts of data that, when analyzed, can uncover valuable insights about content trends and user preferences. This project focuses on cleaning, analyzing, and visualizing Netflix's content dataset from 2008 to 2021. Utilizing Python and its powerful libraries—Pandas, NumPy, Seaborn, and Matplotlib—this analysis dives into data exploration, uncovering patterns such as content distribution, popular genres, and temporal trends. These insights demonstrate the significance of effective data analysis in the evolving entertainment industry.

Objectives

- Clean the Netflix dataset by handling missing values and duplicates.
- Analyze trends such as content type distribution, popular genres, and release patterns.
- Visualize findings using Seaborn and Matplotlib for better understanding.

Scope

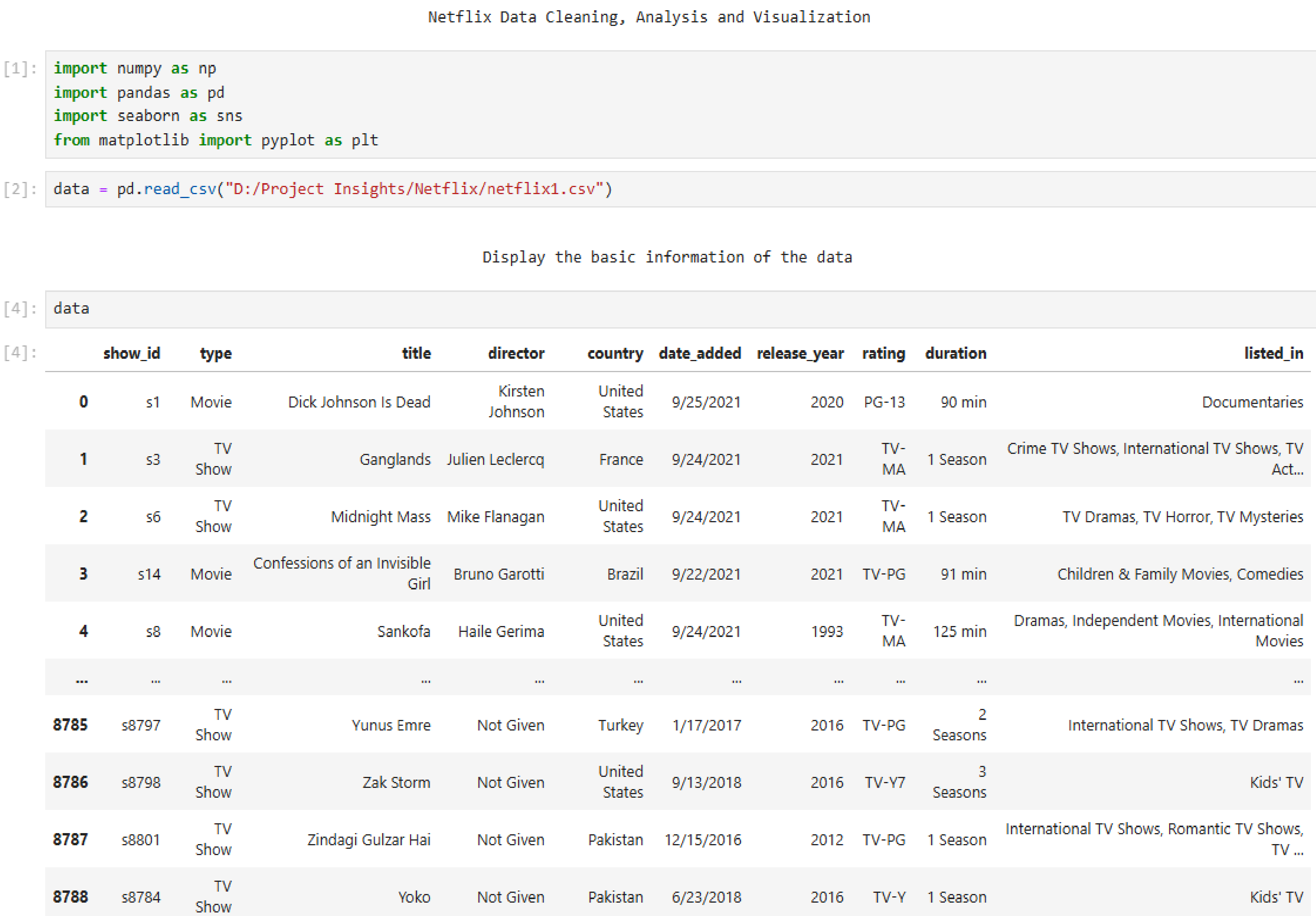
- Focuses on Netflix content data from 2008 to 2021.
- Uses Python libraries: Pandas, NumPy, Seaborn, and Matplotlib.
- Provides insights into content trends and prepares data for further analysis.

NETFLIX

Let’s begin with our basic Fundamental Analysis

Data Import and Data Display

Import necessary Python Libraries for Data Analysis.



Netflix Data Cleaning, Analysis and Visualization

[1]:

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

[2]:

```python
data = pd.read_csv("D:/Project Insights/Netflix/netflix1.csv")
```

Display the basic information of the data

[4]:

```python
data
```

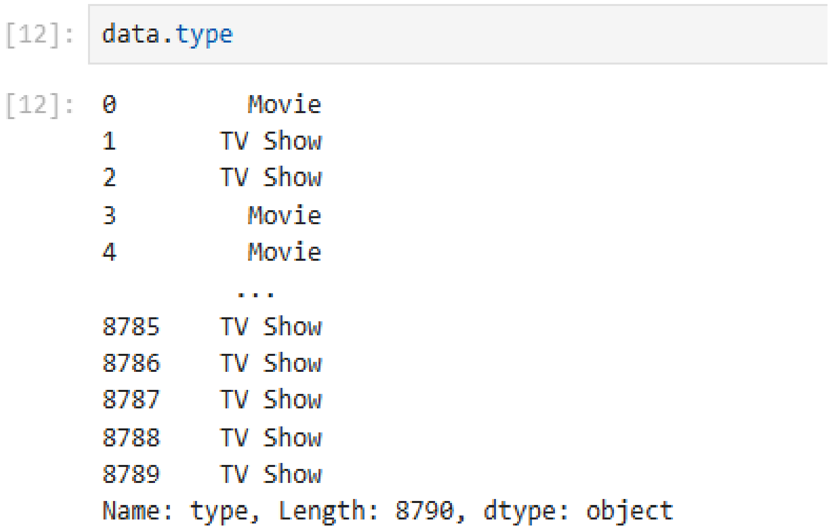
[4]:

	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

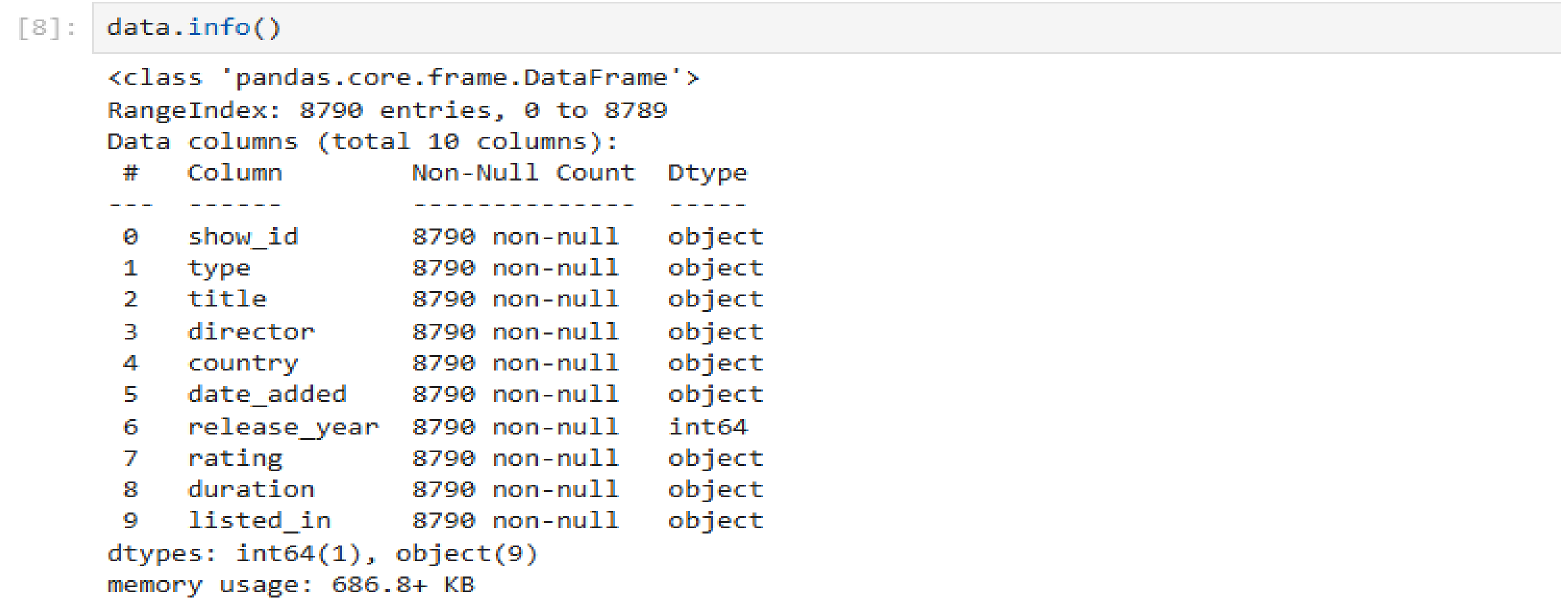
Display basic information about the Data

Basic and fundamental Information about the Data like.

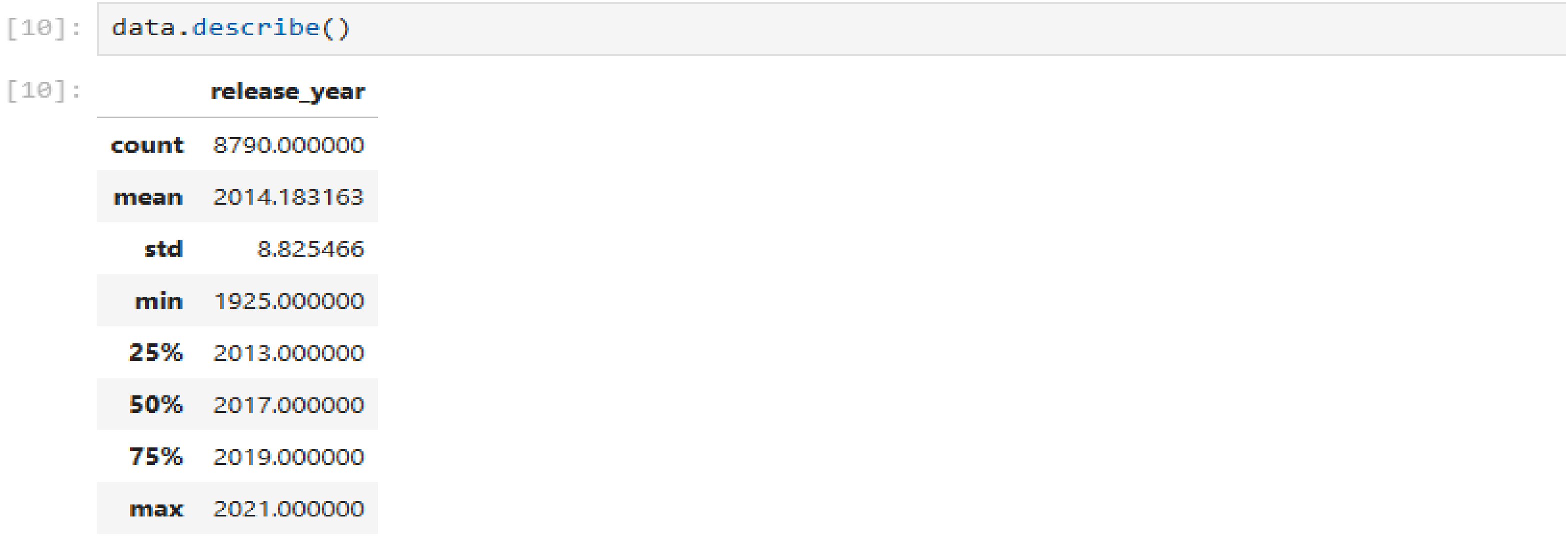
Information, Statistics, Data Type.



```
[12]: data.type

[12]: 0          Movie
      1        TV Show
      2        TV Show
      3          Movie
      4          Movie
                ...
      8785     TV Show
      8786     TV Show
      8787     TV Show
      8788     TV Show
      8789     TV Show
      Name: type, Length: 8790, dtype: object
```



```
[8]: 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
```



```
[10]: data.describe()
```

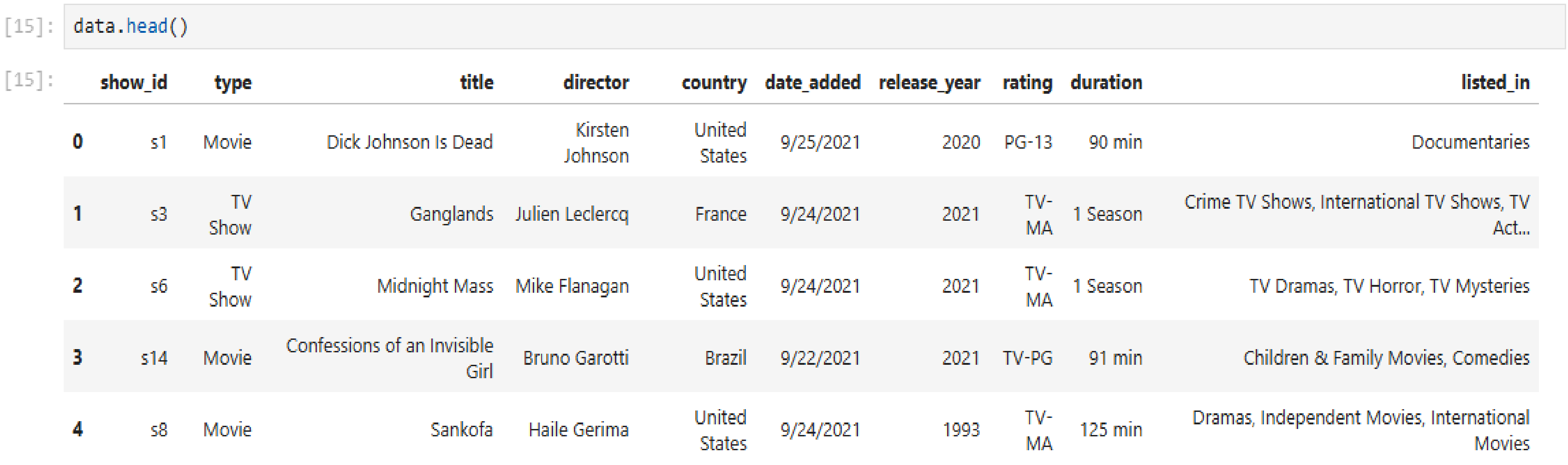
[10]:

	release_year
count	8790.000000
mean	2014.183163
std	8.825466
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

Let’s Begin with Data cleaning process

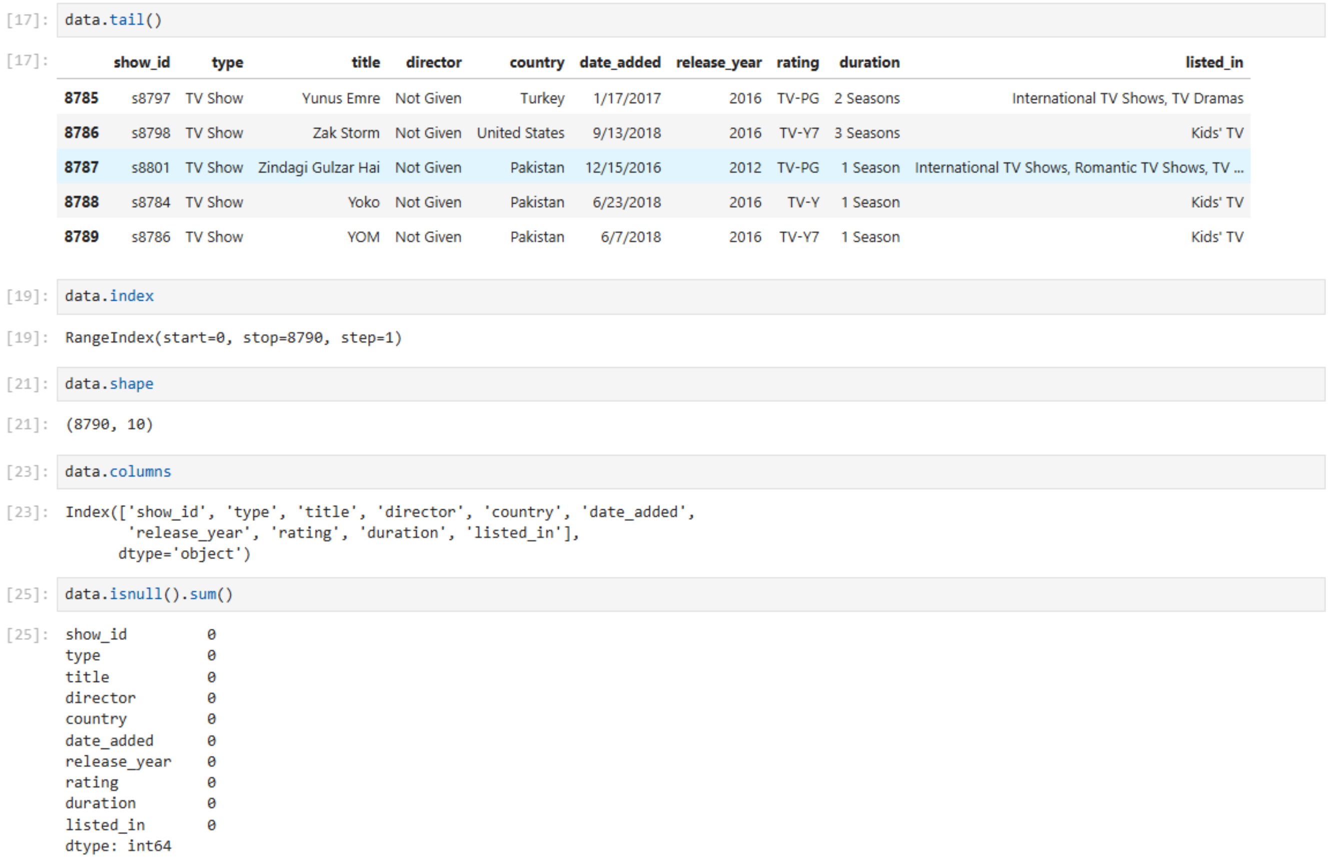
Cleaning the data is one of the important step in Data Analysis.

There are few steps for cleaning data as given:-



```
[15]: data.head()
[15]:
```

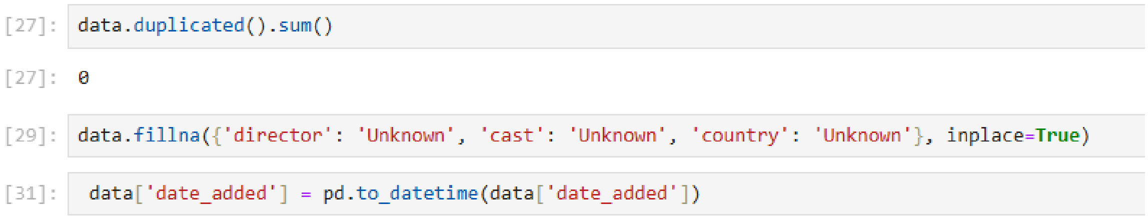
	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	9/25/2021	2020	PG-13	90 min	Documentaries
1	s3	TV Show	Ganglands	Julien Leclercq	France	9/24/2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, 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



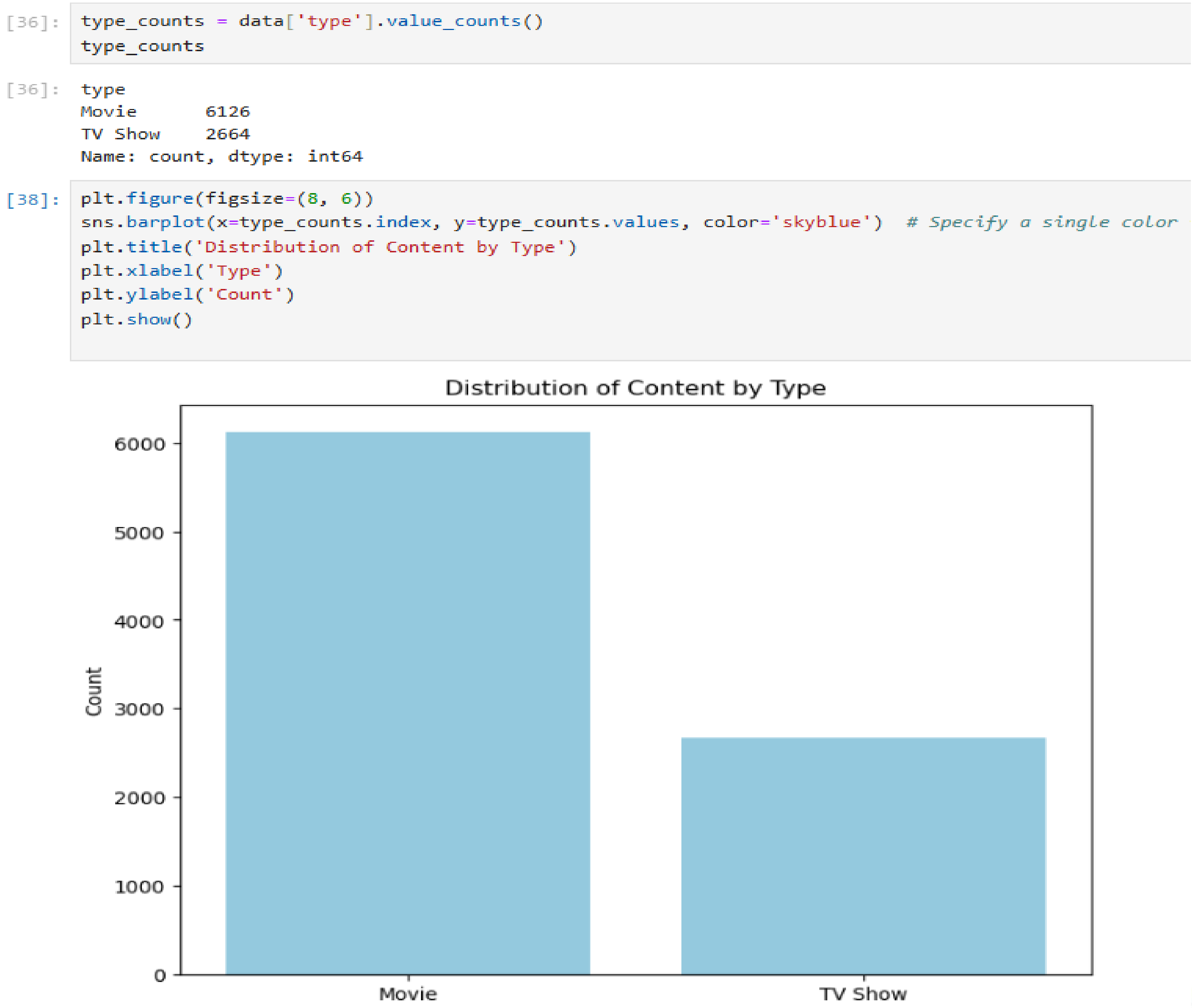
```
[17]: data.tail()
[17]:
```

	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in
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	6/7/2018	2016	TV-Y7	1 Season	Kids' TV

```
[19]: data.index

[19]: RangeIndex(start=0, stop=8790, step=1)

[21]: data.shape

[21]: (8790, 10)

[23]: data.columns

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

[25]: data.isnull().sum()

[25]: 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
```



```
[27]: data.duplicated().sum()

[27]: 0

[29]: data.fillna({'director': 'Unknown', 'cast': 'Unknown', 'country': 'Unknown'}, inplace=True)

[31]:  data['date_added'] = pd.to_datetime(data['date_added'])
```



```
[36]: type_counts = data['type'].value_counts()
      type_counts
```

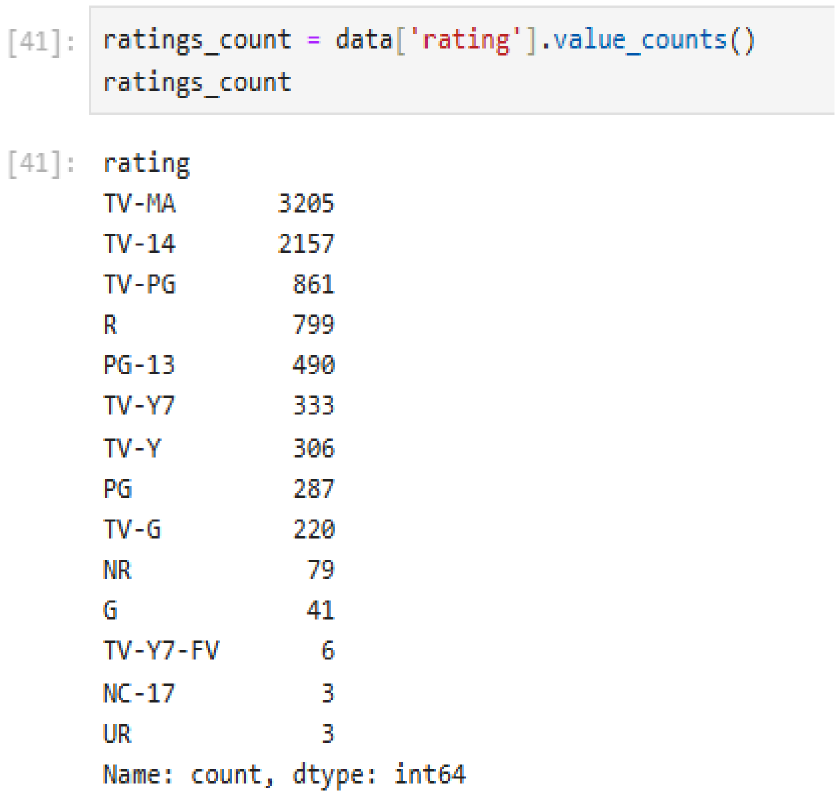
```
[36]: type
      Movie      6126
      TV Show    2664
      Name: count, dtype: int64
```

```
[38]: plt.figure(figsize=(8, 6))
      sns.barplot(x=type_counts.index, y=type_counts.values, color='skyblue')  # Specify a single color
      plt.title('Distribution of Content by Type')
      plt.xlabel('Type')
      plt.ylabel('Count')
      plt.show()
```

Distribution of content on basis of it’s type

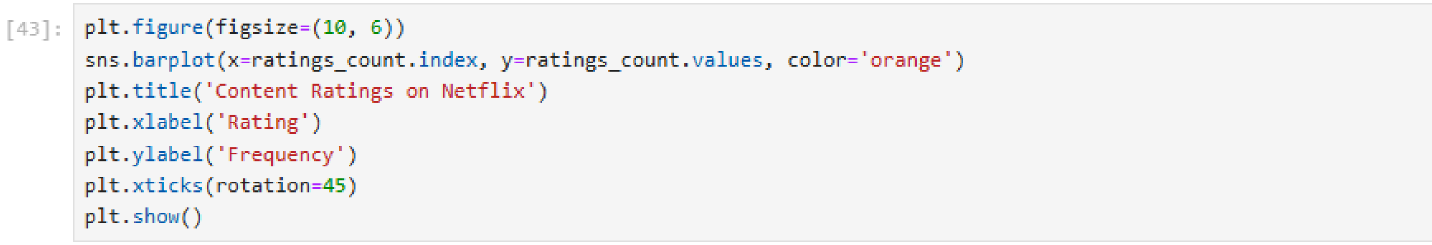
NETFLIX

Content Ratings on Netflix

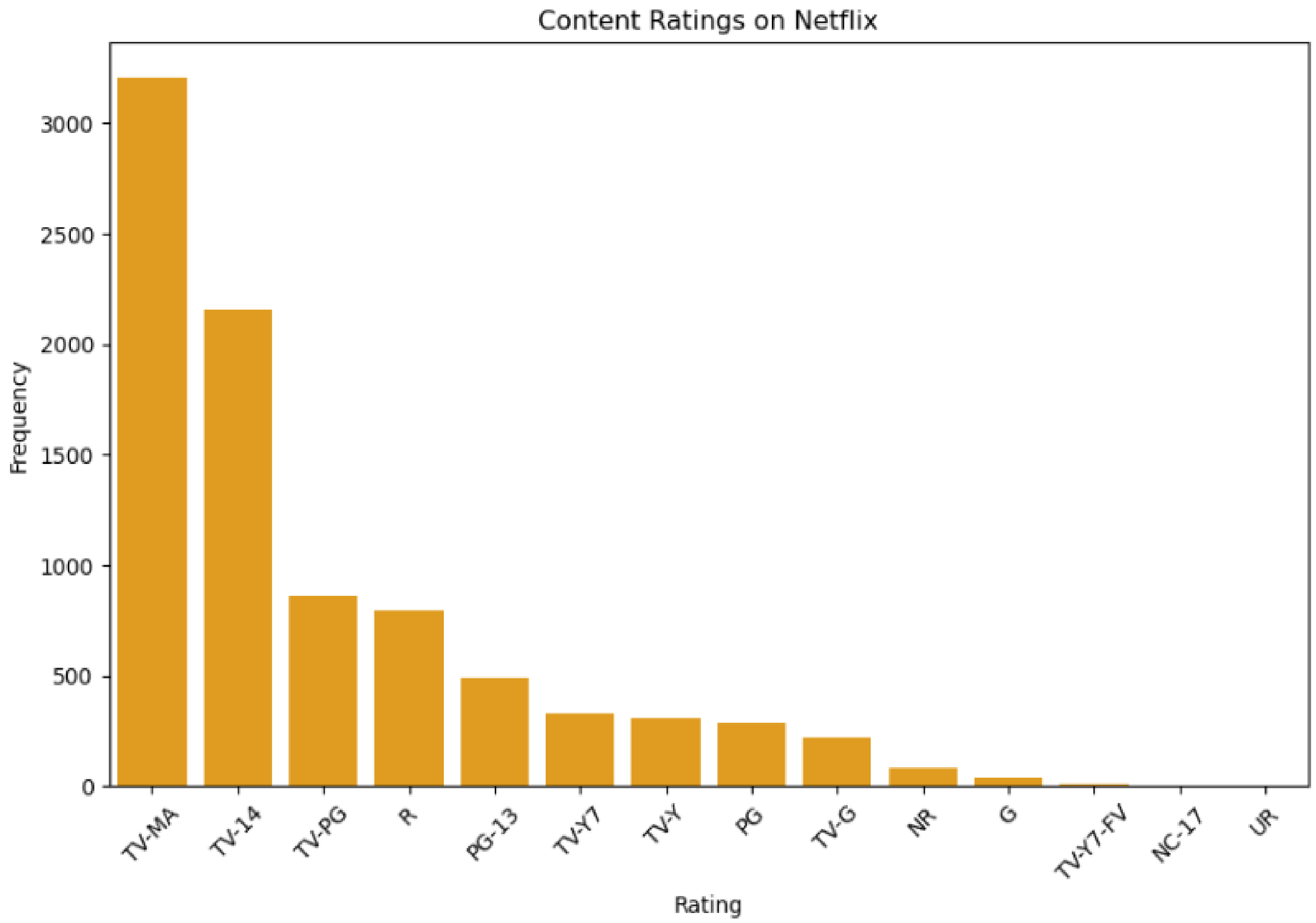


```
[41]: ratings_count = data['rating'].value_counts()
      ratings_count
```

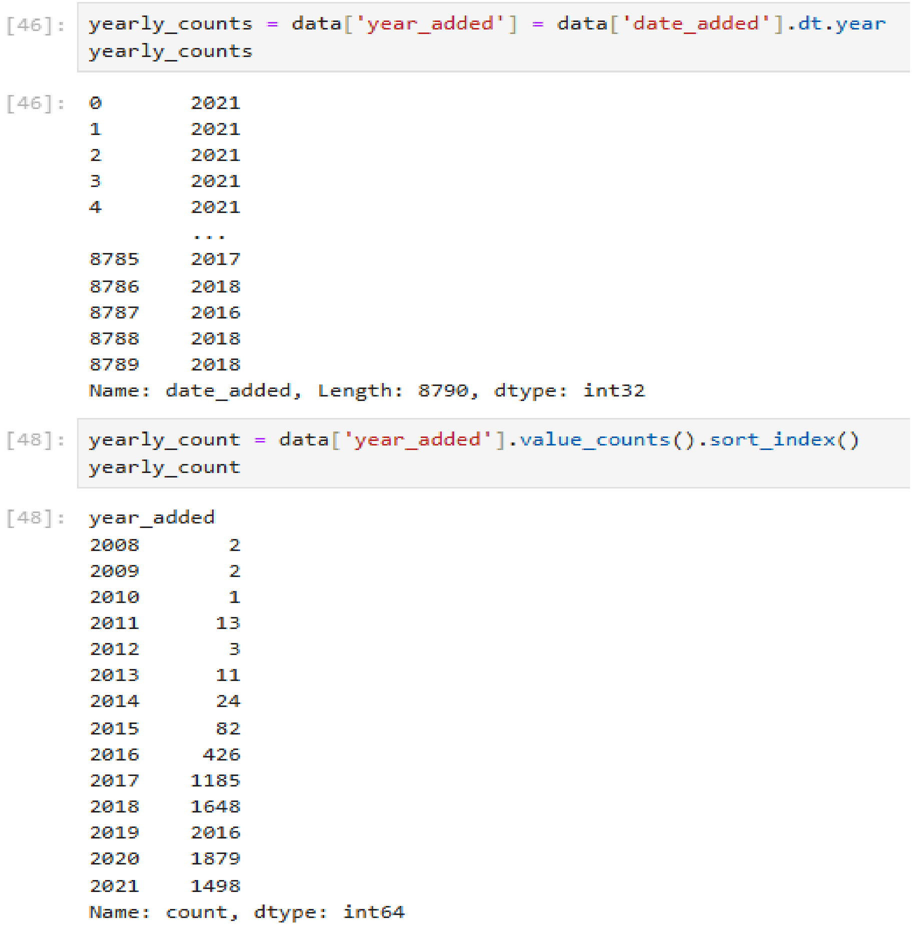
```
[41]: rating
      TV-MA       3205
      TV-14       2157
      TV-PG        861
      R            799
      PG-13        490
      TV-Y7        333
      TV-Y         306
      PG           287
      TV-G         220
      NR            79
      G             41
      TV-Y7-FV       6
      NC-17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



```
[43]: plt.figure(figsize=(10, 6))
      sns.barplot(x=ratings_count.index, y=ratings_count.values, color='orange')
      plt.title('Content Ratings on Netflix')
      plt.xlabel('Rating')
      plt.ylabel('Frequency')
      plt.xticks(rotation=45)
      plt.show()
```



Yearly trends in adding Content

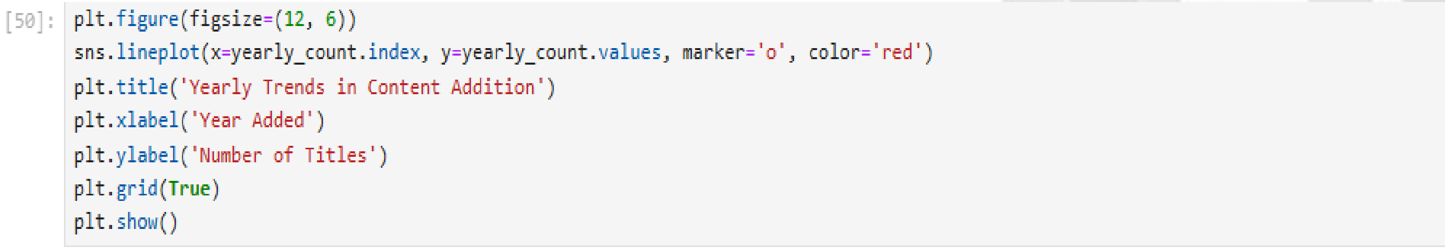


```
[46]: yearly_counts = data['year_added'] = data['date_added'].dt.year
      yearly_counts
```

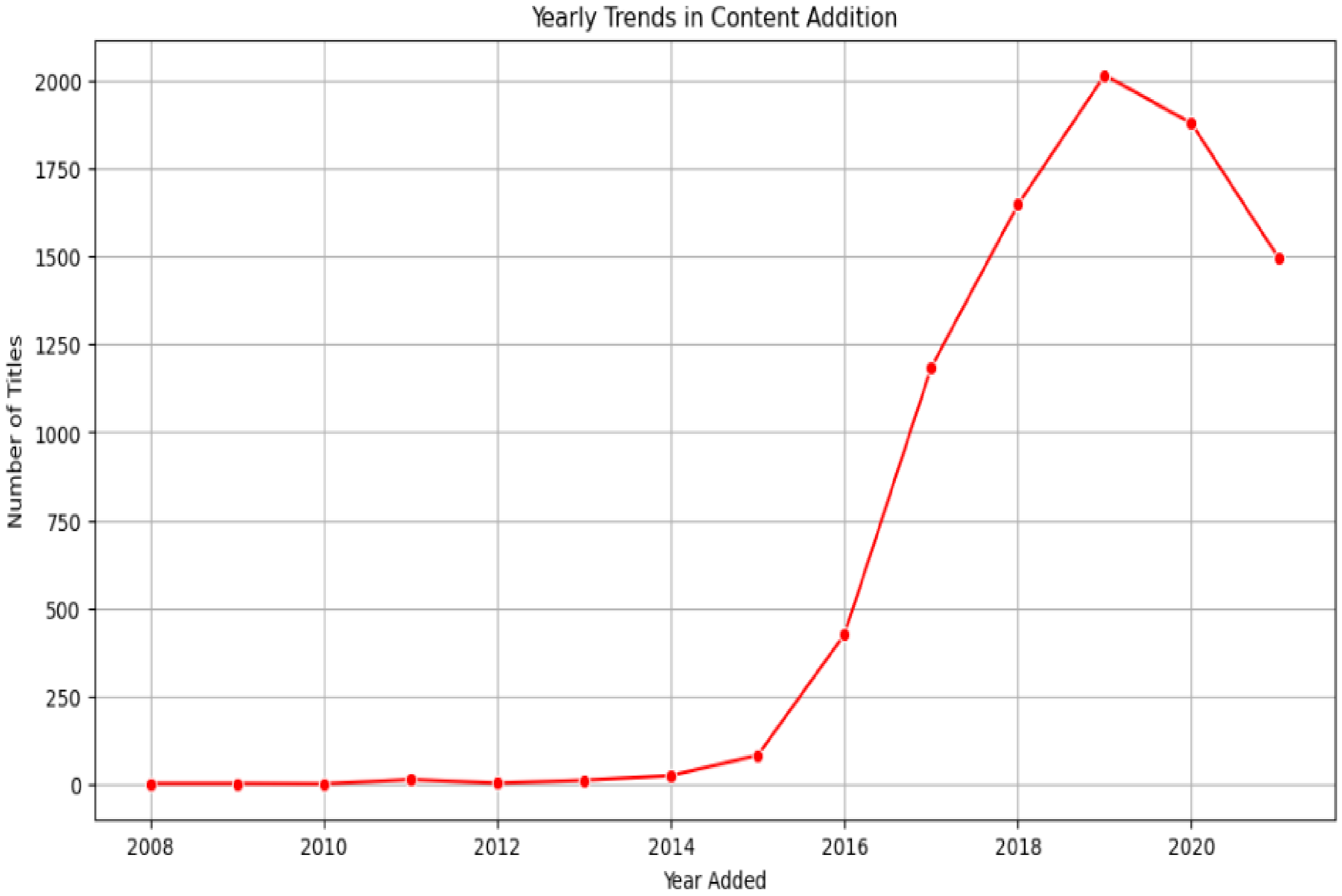
```
[46]: 0       2021
      1       2021
      2       2021
      3       2021
      4       2021
              ...
      8785    2017
      8786    2018
      8787    2016
      8788    2018
      8789    2018
      Name: date_added, Length: 8790, dtype: int32
```

```
[48]: yearly_count = data['year_added'].value_counts().sort_index()
      yearly_count
```

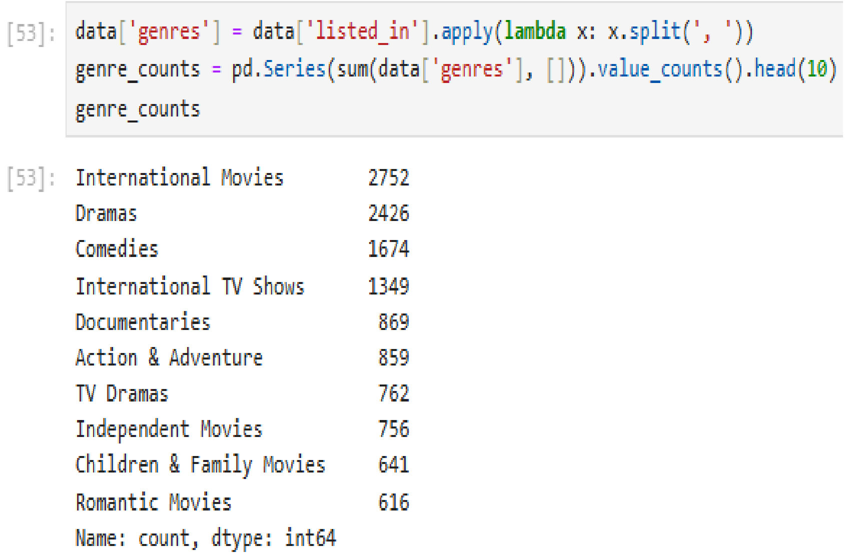
```
[48]: year_added
      2008       2
      2009       2
      2010       1
      2011      13
      2012       3
      2013      11
      2014      24
      2015      82
      2016     426
      2017    1185
      2018    1648
      2019    2016
      2020    1879
      2021    1498
      Name: count, dtype: int64
```



```
[50]: plt.figure(figsize=(12, 6))
      sns.lineplot(x=yearly_count.index, y=yearly_count.values, marker='o', color='red')
      plt.title('Yearly Trends in Content Addition')
      plt.xlabel('Year Added')
      plt.ylabel('Number of Titles')
      plt.grid(True)
      plt.show()
```

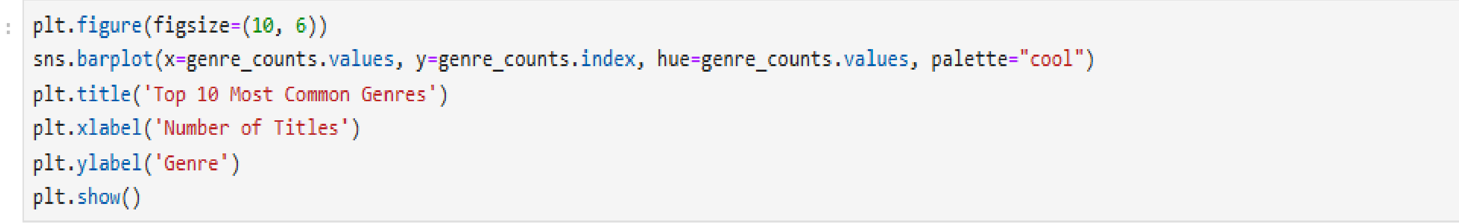


Top 10 Most common Genres

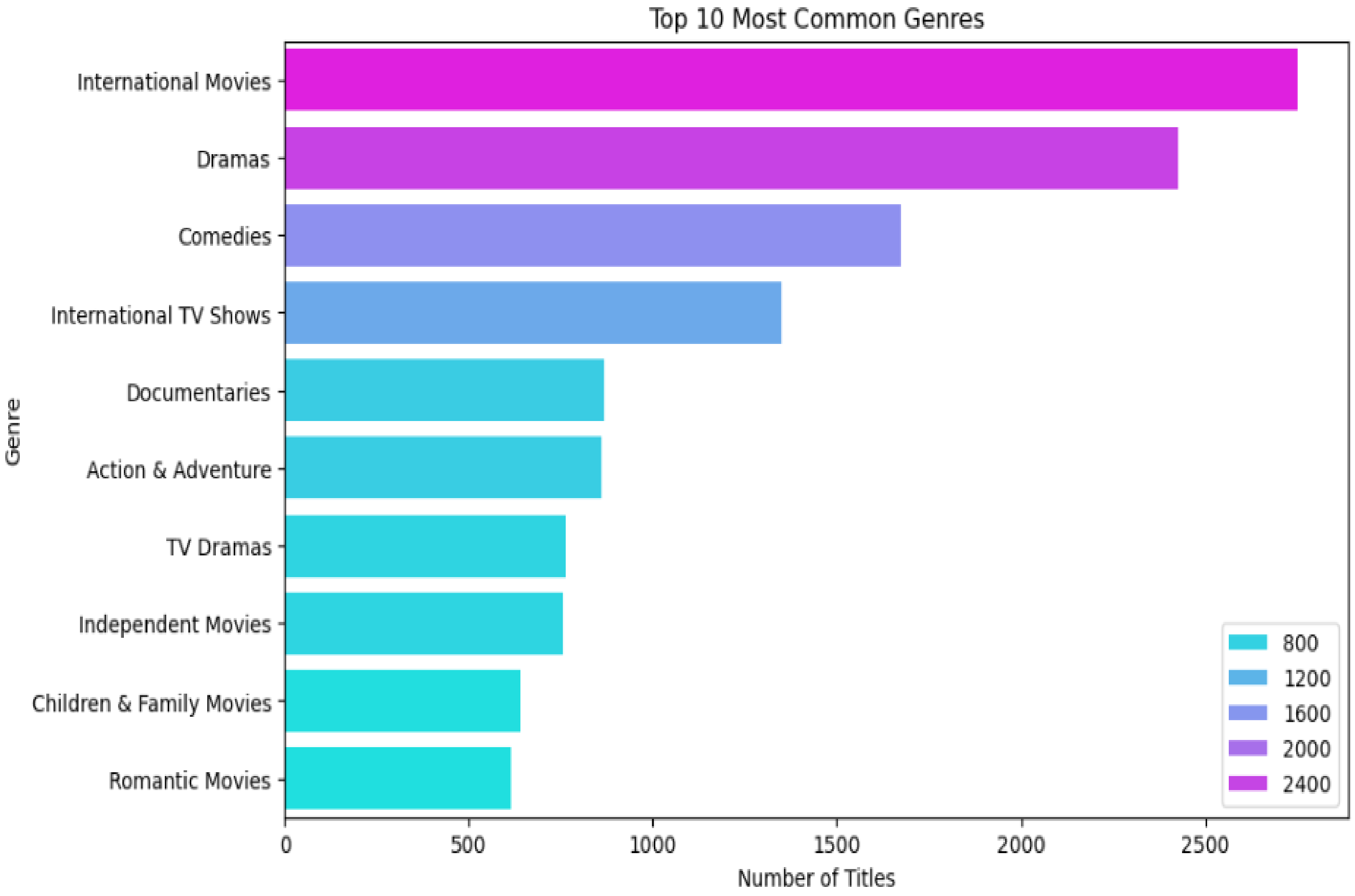


```
[53]: data['genres'] = data['listed_in'].apply(lambda x: x.split(', '))
      genre_counts = pd.Series(sum(data['genres'], [])).value_counts().head(10)
      genre_counts
```

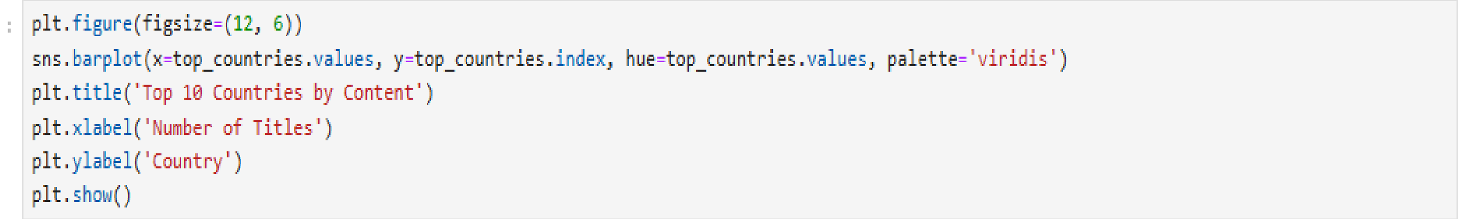
```
[53]: International Movies        2752
      Dramas                      2426
      Comedies                    1674
      International TV Shows      1349
      Documentaries                869
      Action & Adventure           859
      TV Dramas                    762
      Independent Movies           756
      Children & Family Movies     641
      Romantic Movies              616
      Name: count, dtype: int64
```



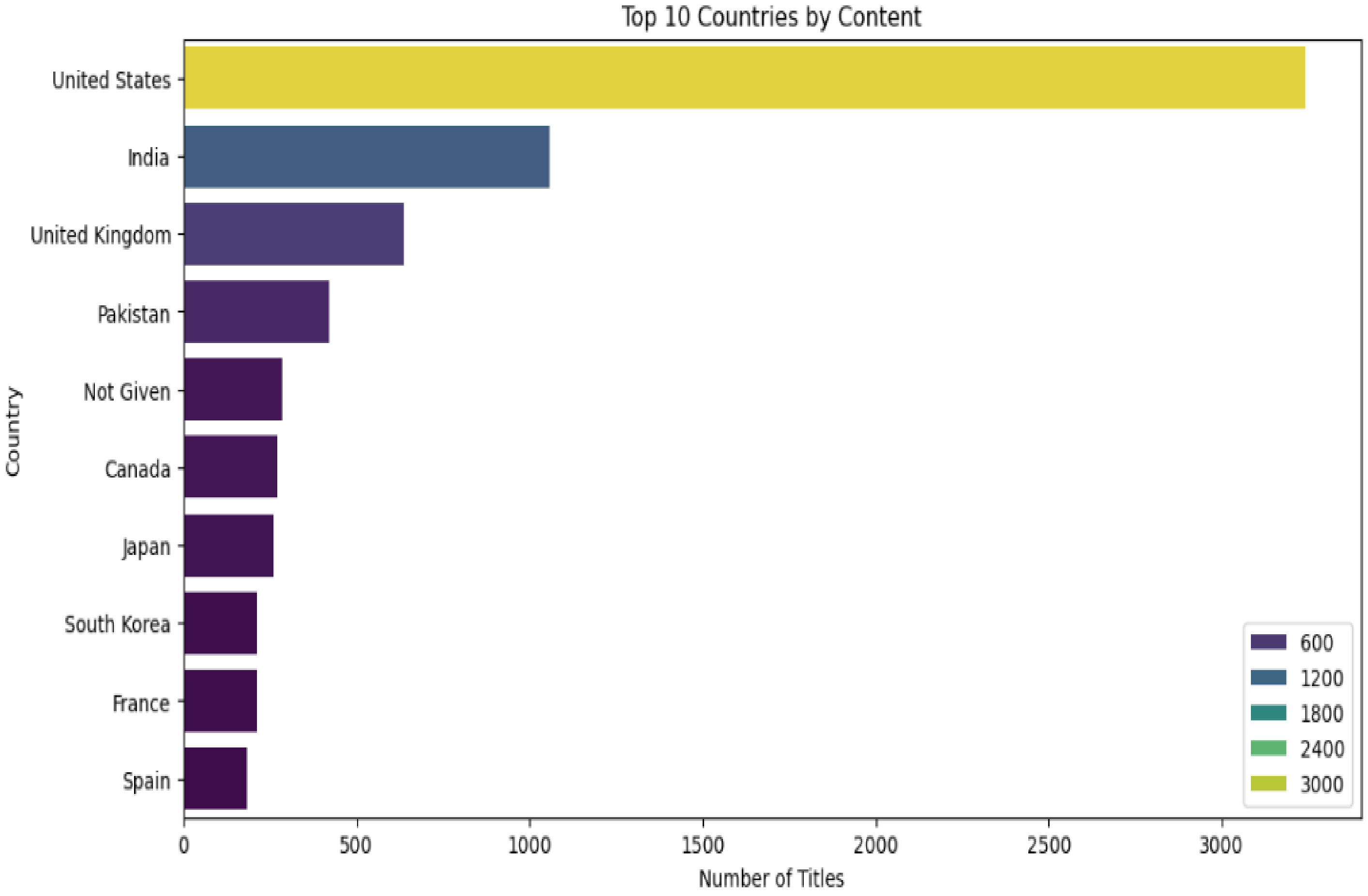
```
plt.figure(figsize=(10, 6))
sns.barplot(x=genre_counts.values, y=genre_counts.index, hue=genre_counts.values, palette="cool")
plt.title('Top 10 Most Common Genres')
plt.xlabel('Number of Titles')
plt.ylabel('Genre')
plt.show()
```

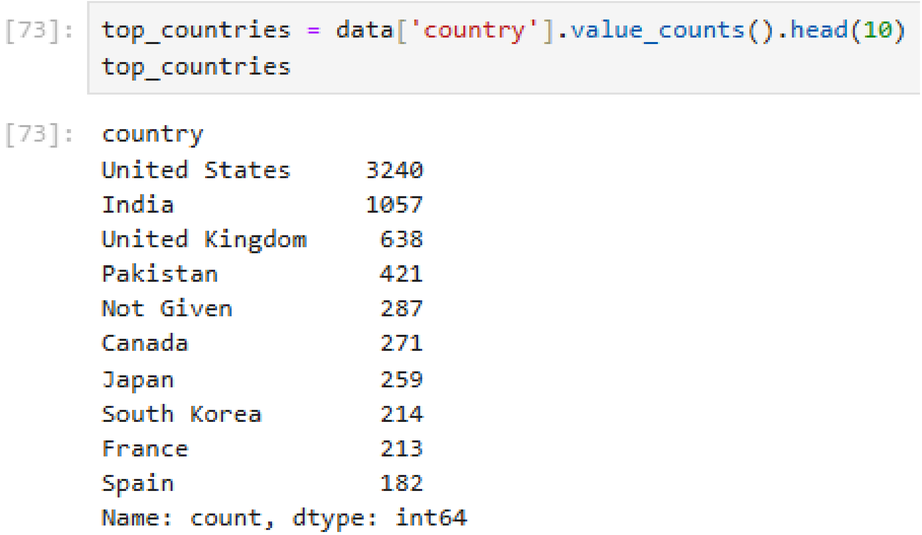


Top 10 countries surfing Netflix



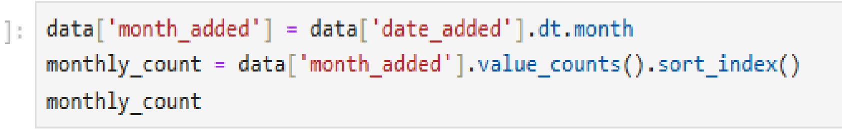
```
plt.figure(figsize=(12, 6))
sns.barplot(x=top_countries.values, y=top_countries.index, hue=top_countries.values, palette='viridis')
plt.title('Top 10 Countries by Content')
plt.xlabel('Number of Titles')
plt.ylabel('Country')
plt.show()
```



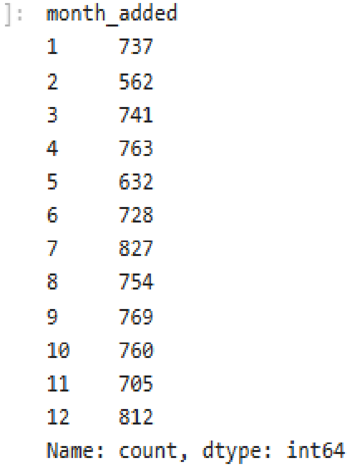


```
[73]: top_countries = data['country'].value_counts().head(10)
      top_countries

[73]: country
      United States     3240
      India             1057
      United Kingdom     638
      Pakistan           421
      Not Given          287
      Canada             271
      Japan              259
      South Korea        214
      France             213
      Spain              182
      Name: count, dtype: int64
```

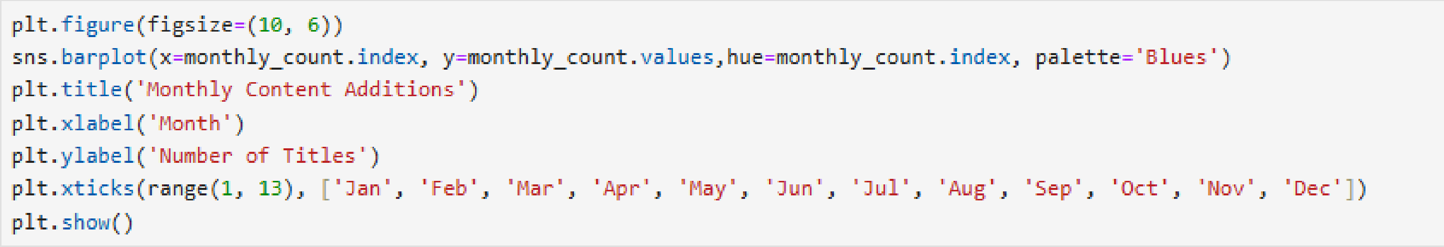
Content addition Month-Wise



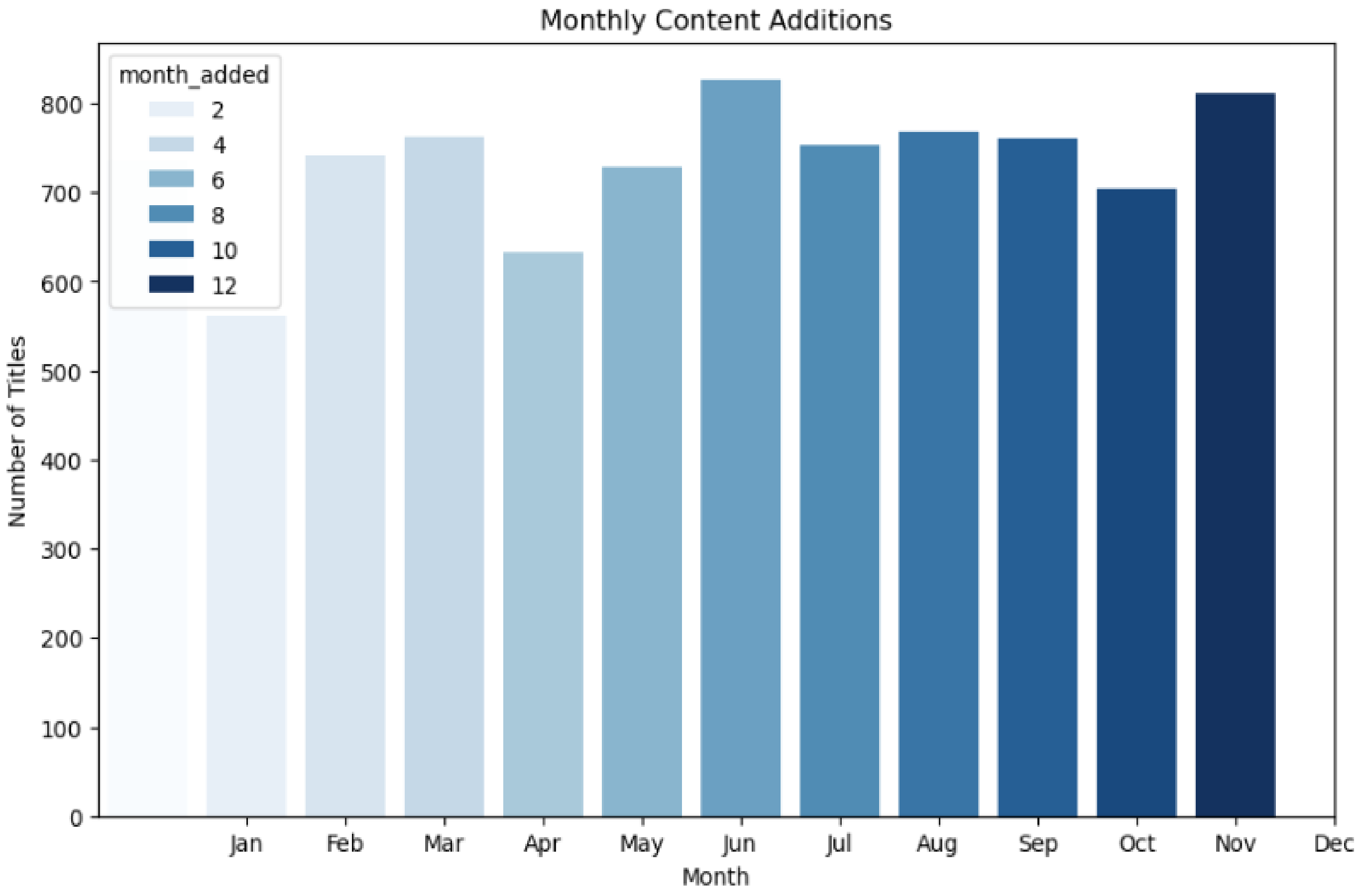
```
data['month_added'] = data['date_added'].dt.month
monthly_count = data['month_added'].value_counts().sort_index()
monthly_count
```



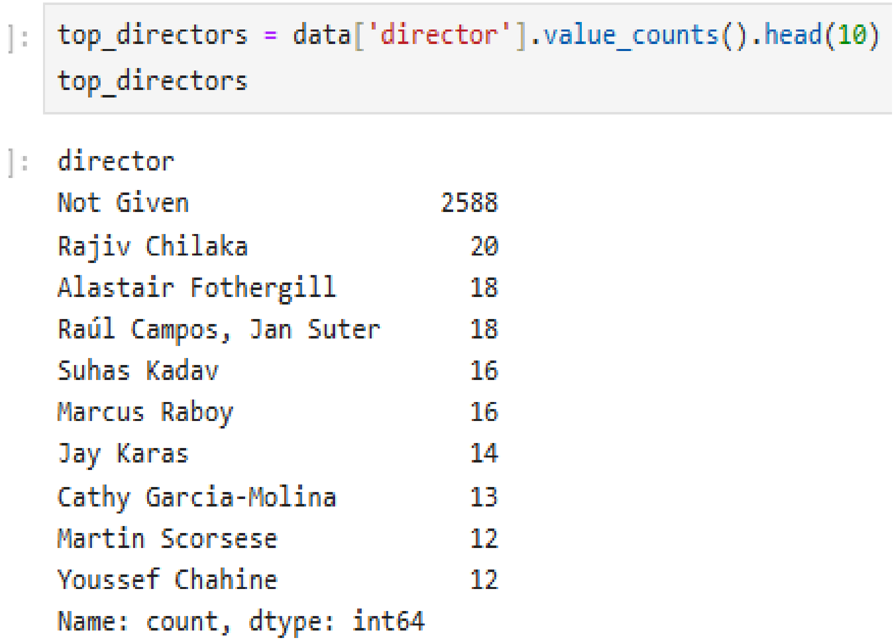
```
month_added
1     737
2     562
3     741
4     763
5     632
6     728
7     827
8     754
9     769
10    760
11    705
12    812
Name: count, dtype: int64
```



```
plt.figure(figsize=(10, 6))
sns.barplot(x=monthly_count.index, y=monthly_count.values,hue=monthly_count.index, palette='Blues')
plt.title('Monthly Content Additions')
plt.xlabel('Month')
plt.ylabel('Number of Titles')
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.show()
```

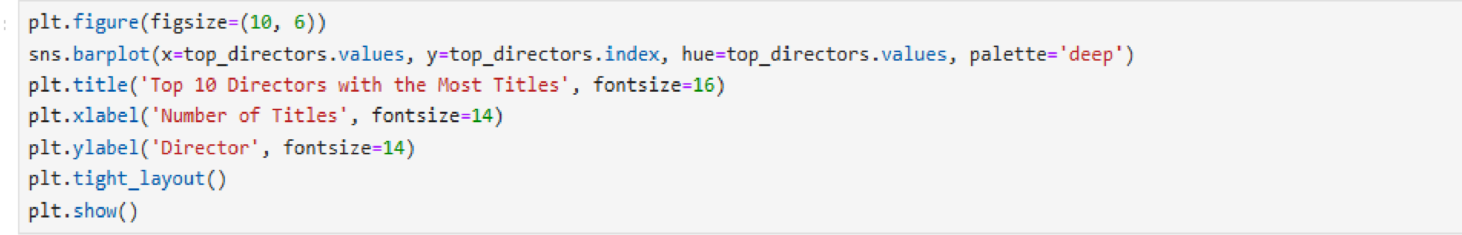


Top 10 most Title hosting Directors

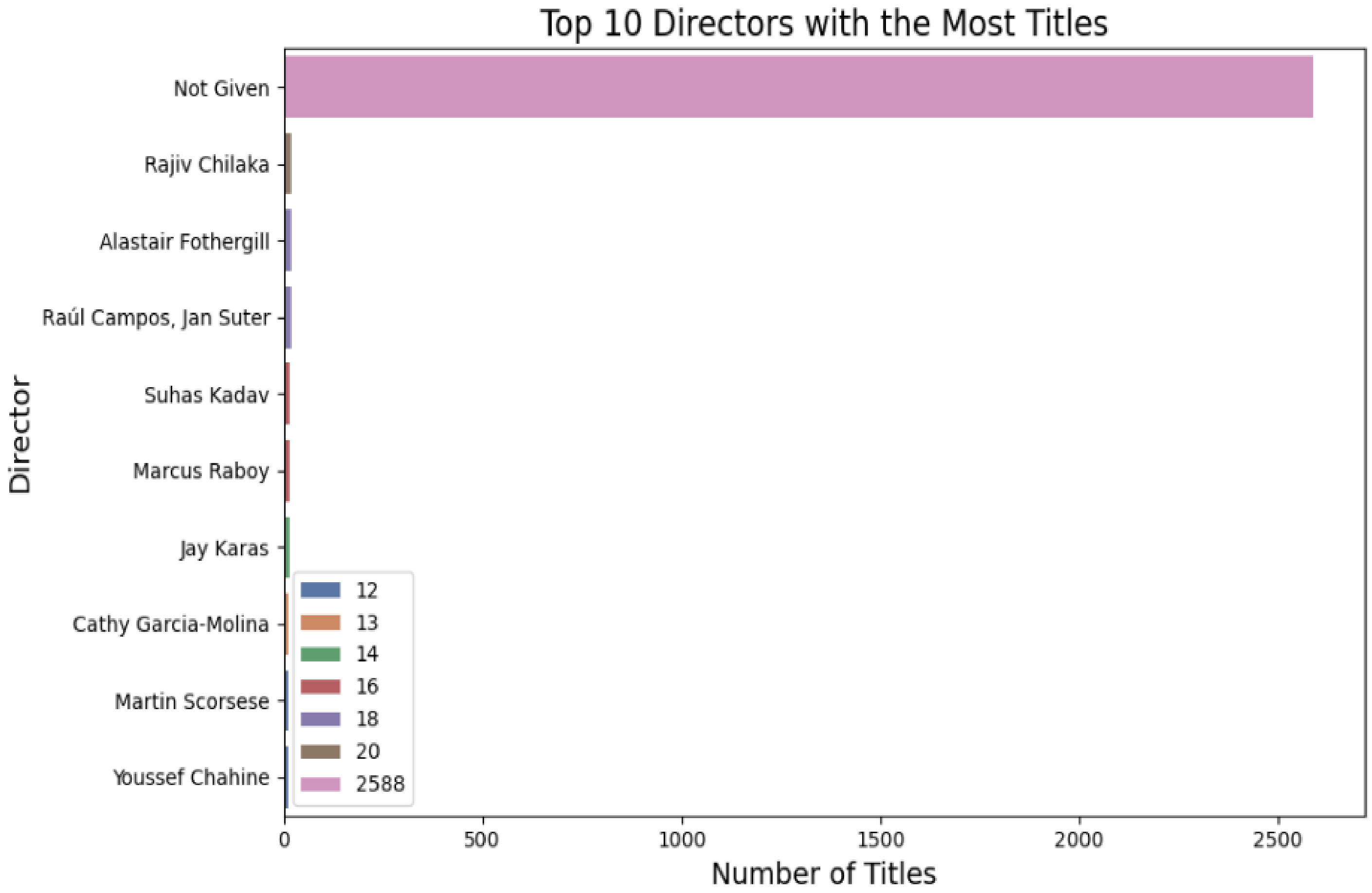


```
]: top_directors = data['director'].value_counts().head(10)
   top_directors
```

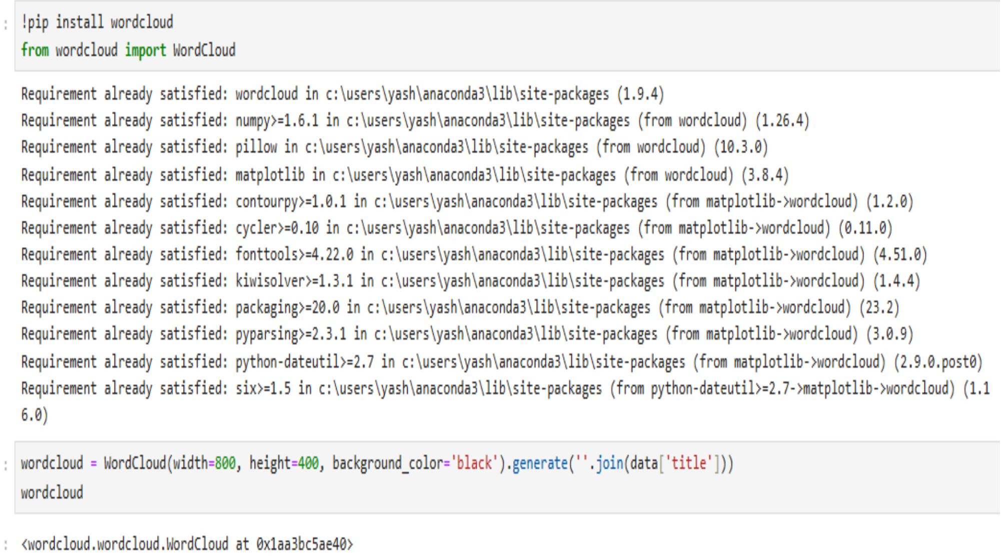
```
]: director
   Not Given                 2588
   Rajiv Chilaka               20
   Alastair Fothergill         18
   Raúl Campos, Jan Suter      18
   Suhas Kadav                 16
   Marcus Raboy                16
   Jay Karas                   14
   Cathy Garcia-Molina         13
   Martin Scorsese             12
   Youssef Chahine             12
   Name: count, dtype: int64
```



```
plt.figure(figsize=(10, 6))
sns.barplot(x=top_directors.values, y=top_directors.index, hue=top_directors.values, palette='deep')
plt.title('Top 10 Directors with the Most Titles', fontsize=16)
plt.xlabel('Number of Titles', fontsize=14)
plt.ylabel('Director', fontsize=14)
plt.tight_layout()
plt.show()
```



Word Cloud Netflix Titles looks like a cool banner

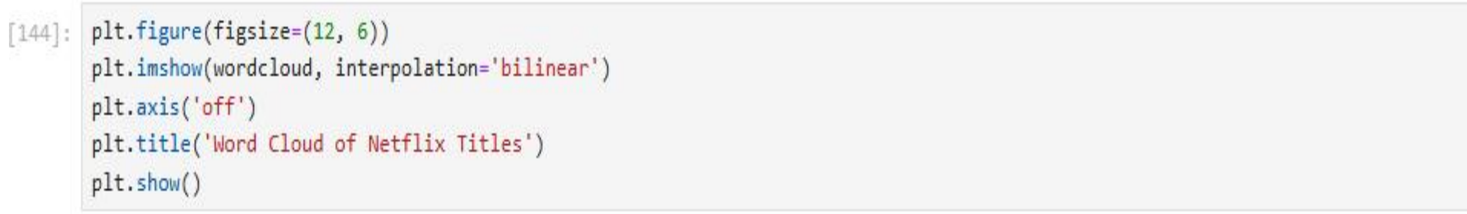


```
!pip install wordcloud
from wordcloud import WordCloud
```

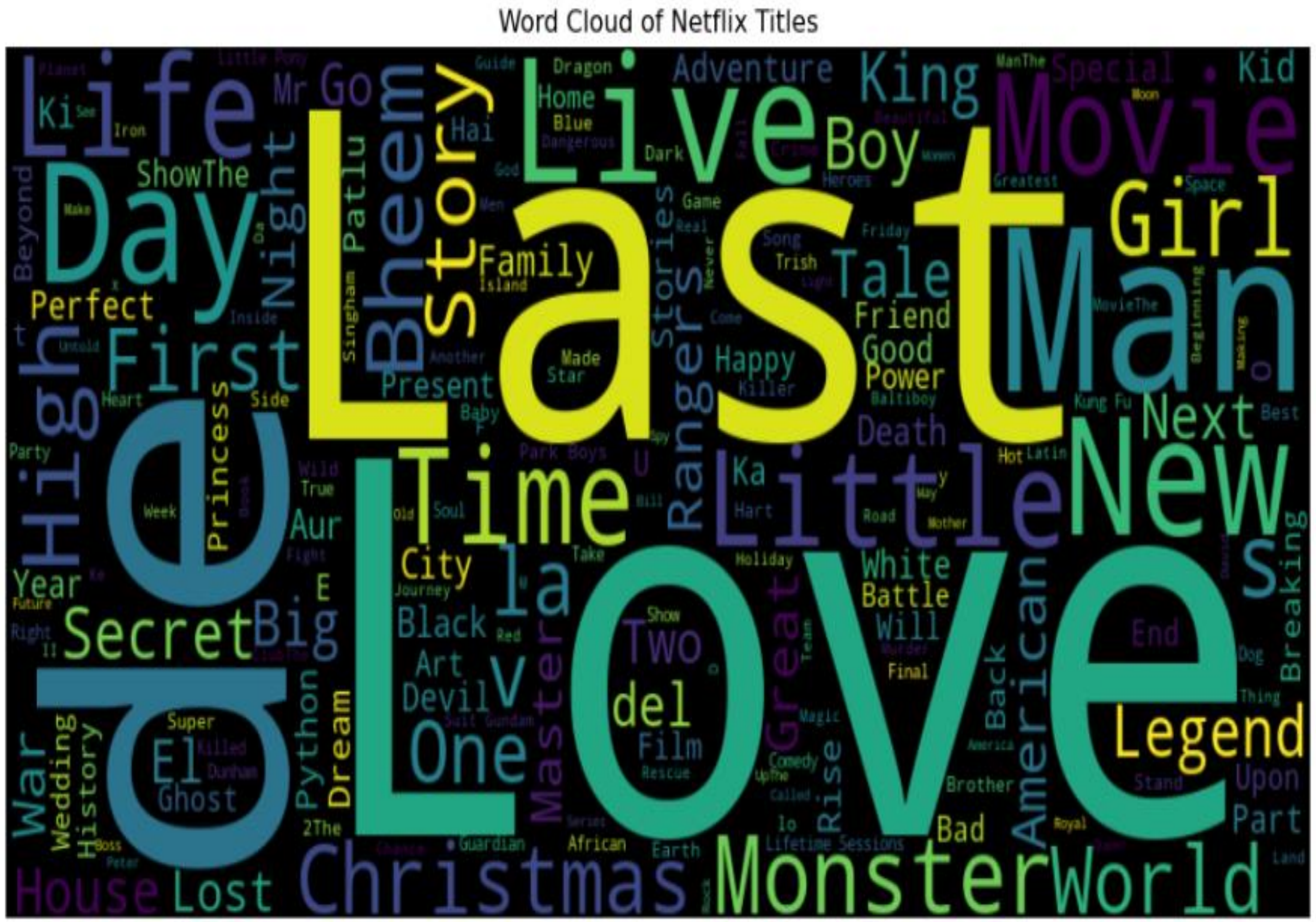
```
Requirement already satisfied: wordcloud in c:\users\yash\anaconda3\lib\site-packages (1.9.4)
Requirement already satisfied: numpy>=1.6.1 in c:\users\yash\anaconda3\lib\site-packages (from wordcloud) (1.26.4)
Requirement already satisfied: pillow in c:\users\yash\anaconda3\lib\site-packages (from wordcloud) (10.3.0)
Requirement already satisfied: matplotlib in c:\users\yash\anaconda3\lib\site-packages (from wordcloud) (3.8.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\yash\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\yash\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\yash\anaconda3\lib\site-packages (from matplotlib->wordcloud) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\yash\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\yash\anaconda3\lib\site-packages (from matplotlib->wordcloud) (23.2)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\yash\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\yash\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\yash\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.1
6.0)
```

```
wordcloud = WordCloud(width=800, height=400, background_color='black').generate(''.join(data['title']))
wordcloud
```

```
<wordcloud.wordcloud.WordCloud at 0x1aa3bc5ae40>
```

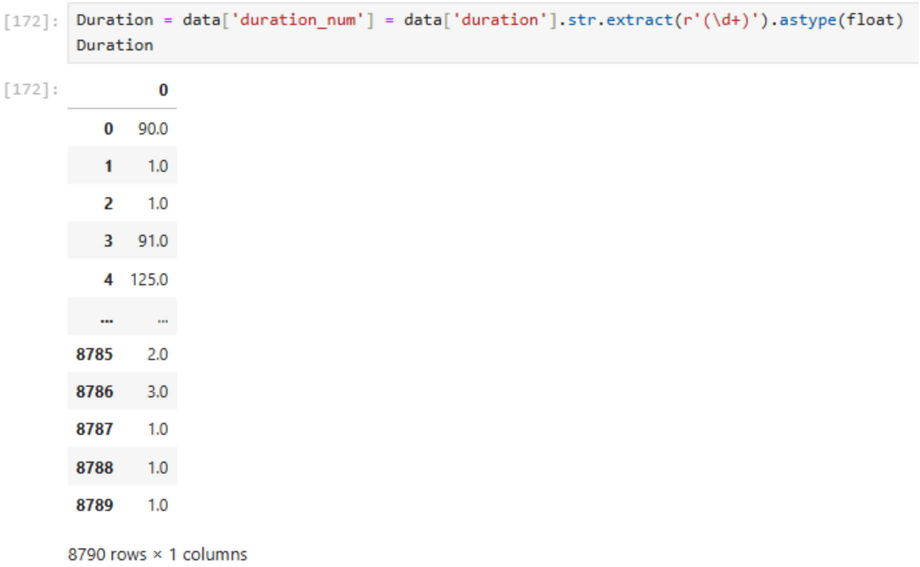


```
[144]: plt.figure(figsize=(12, 6))
       plt.imshow(wordcloud, interpolation='bilinear')
       plt.axis('off')
       plt.title('Word Cloud of Netflix Titles')
       plt.show()
```



Word Cloud of Netflix Titles

Distribution of Movies by their Duration

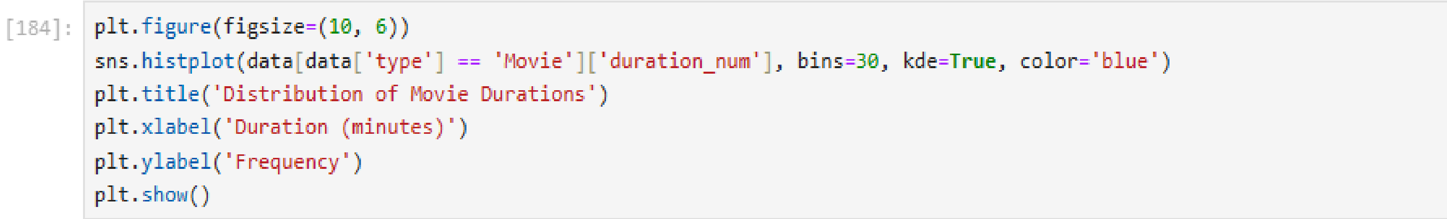


```
[172]: Duration = data['duration_num'] = data['duration'].str.extract(r'(\d+)').astype(float)
       Duration
```

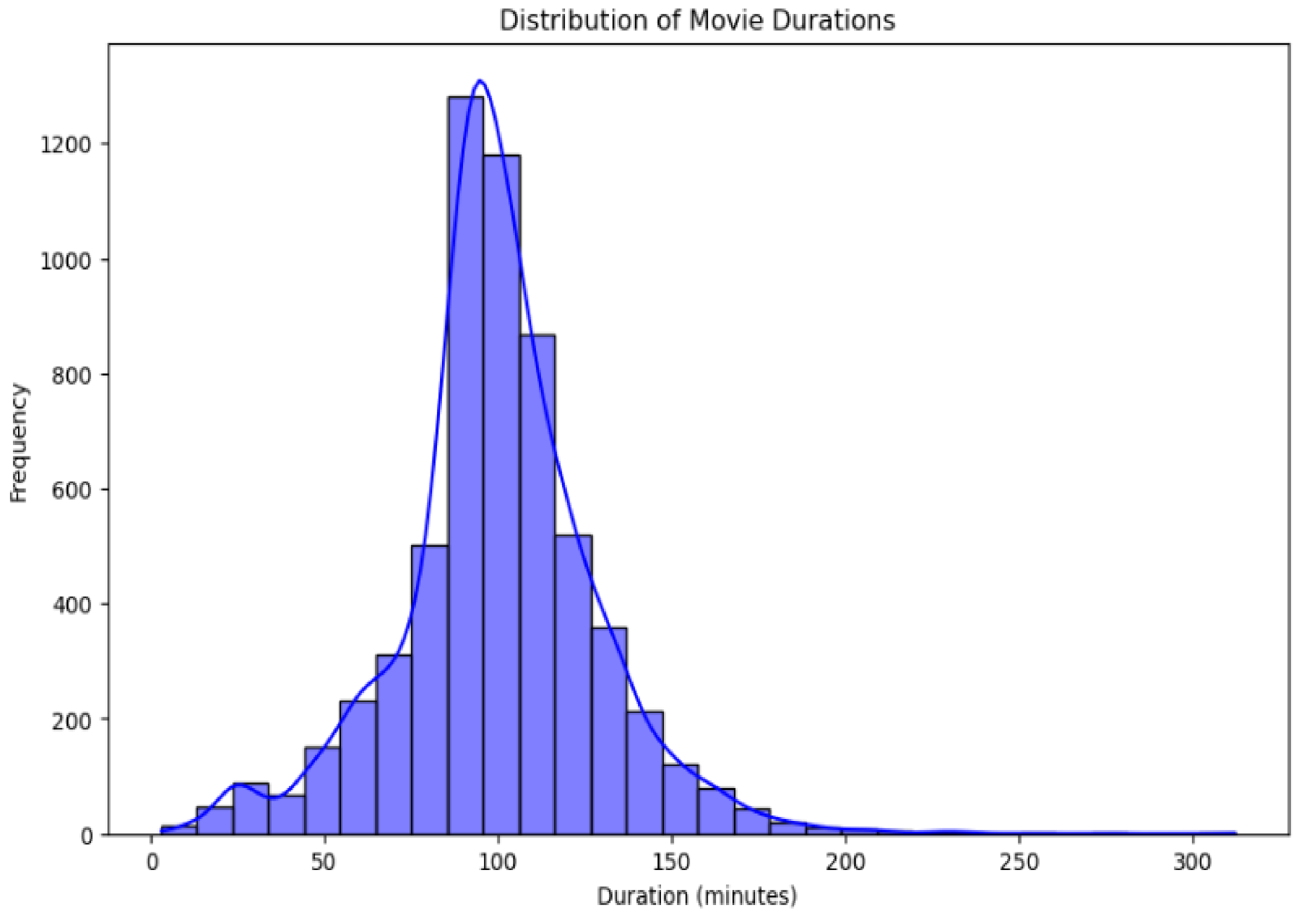
[172]:

	0
0	90.0
1	1.0
2	1.0
3	91.0
4	125.0
...	...
8785	2.0
8786	3.0
8787	1.0
8788	1.0
8789	1.0

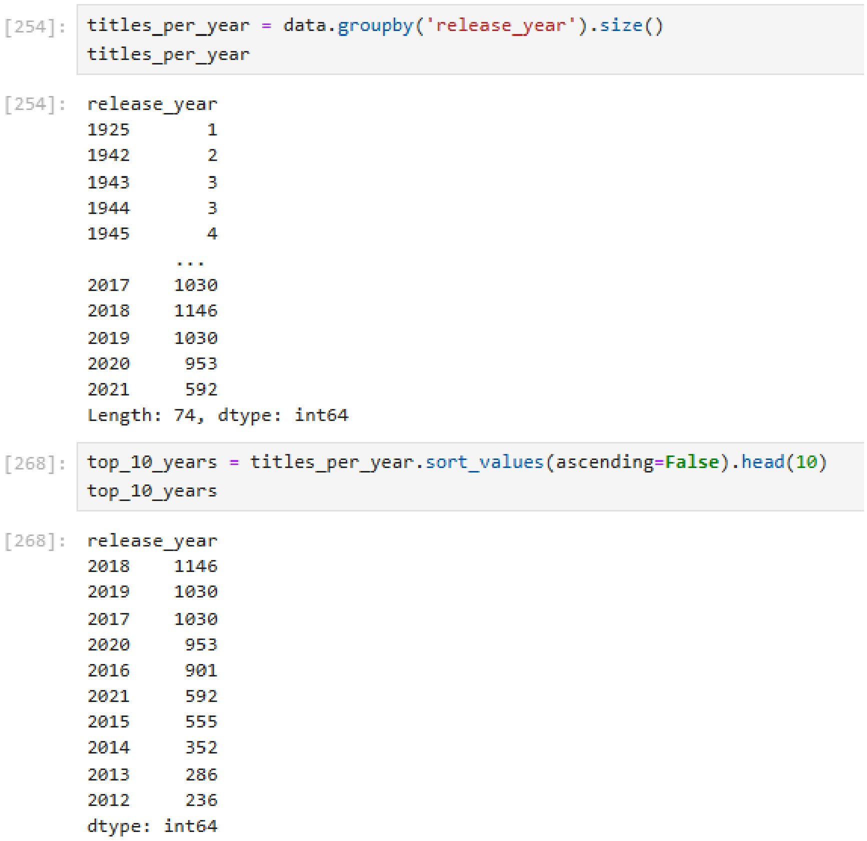
8790 rows × 1 columns



```
[184]: plt.figure(figsize=(10, 6))
       sns.histplot(data[data['type'] == 'Movie']['duration_num'], bins=30, kde=True, color='blue')
       plt.title('Distribution of Movie Durations')
       plt.xlabel('Duration (minutes)')
       plt.ylabel('Frequency')
       plt.show()
```



Top 10 years in which most Movies or Series are released

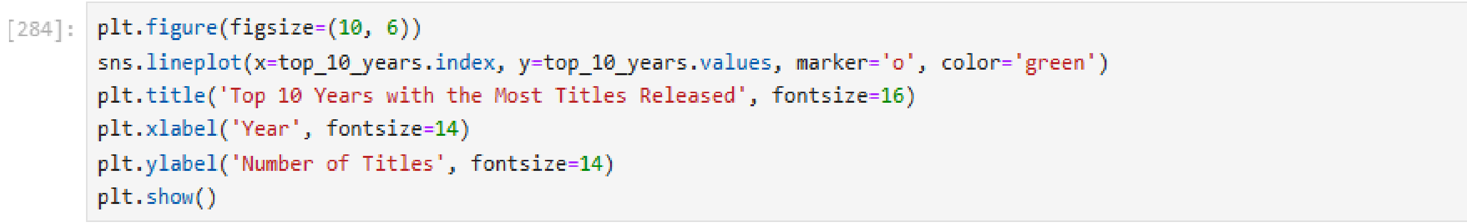


```
[254]: titles_per_year = data.groupby('release_year').size()
       titles_per_year
```

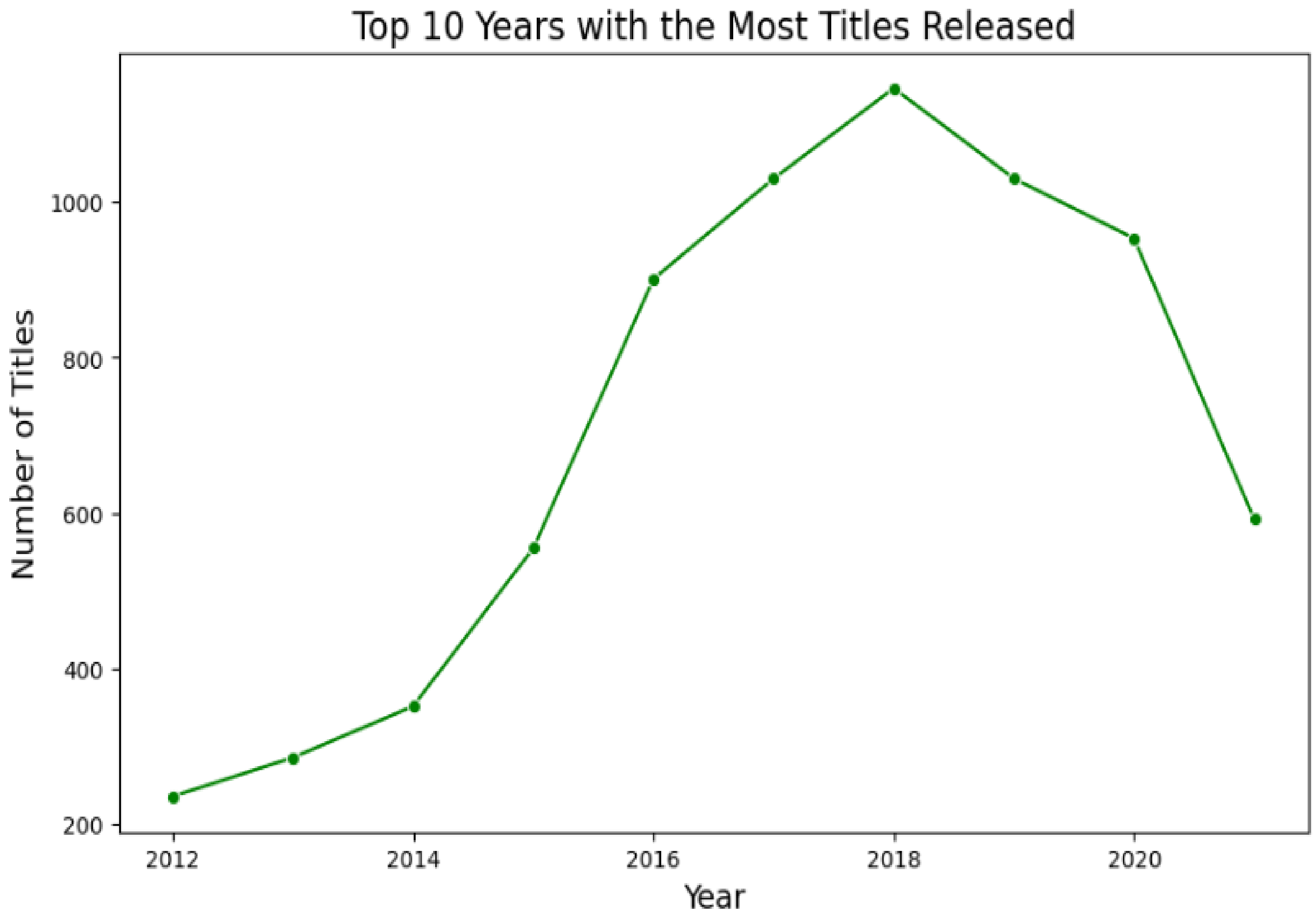
```
[254]: release_year
       1925       1
       1942       2
       1943       3
       1944       3
       1945       4
                ...
       2017    1030
       2018    1146
       2019    1030
       2020     953
       2021     592
       Length: 74, dtype: int64
```

```
[268]: top_10_years = titles_per_year.sort_values(ascending=False).head(10)
       top_10_years
```

```
[268]: release_year
       2018    1146
       2019    1030
       2017    1030
       2020     953
       2016     901
       2021     592
       2015     555
       2014     352
       2013     286
       2012     236
       dtype: int64
```



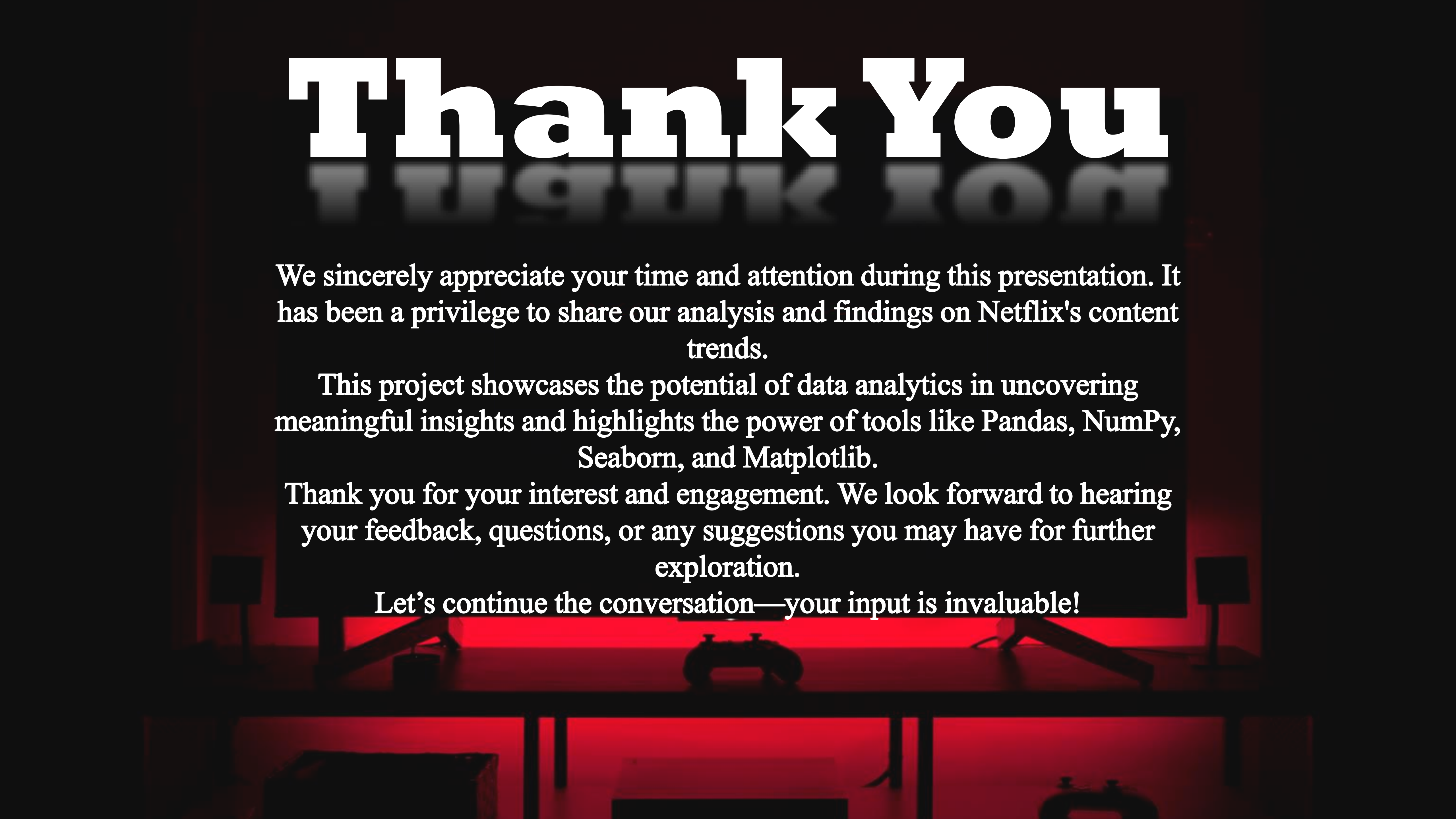
```
[284]: plt.figure(figsize=(10, 6))
       sns.lineplot(x=top_10_years.index, y=top_10_years.values, marker='o', color='green')
       plt.title('Top 10 Years with the Most Titles Released', fontsize=16)
       plt.xlabel('Year', fontsize=14)
       plt.ylabel('Number of Titles', fontsize=14)
       plt.show()
```



NETFLIX

Key Insights

1. Movies make up 70% of Netflix’s content, while TV Shows are 30%.
2. Top genres: Documentaries, Dramas, and Comedies.
3. The U.S., India, and the U.K. are leading content producers.
4. Most content is rated TV-MA, focusing on mature audiences.
5. Recent years show a peak in content additions.



Thank You

We sincerely appreciate your time and attention during this presentation. It has been a privilege to share our analysis and findings on Netflix's content trends.

This project showcases the potential of data analytics in uncovering meaningful insights and highlights the power of tools like Pandas, NumPy, Seaborn, and Matplotlib.

Thank you for your interest and engagement. We look forward to hearing your feedback, questions, or any suggestions you may have for further exploration.

Let’s continue the conversation—your input is invaluable!