Netflix INTRO

Netflix is an American subscription video on-demand over-the-top streaming service. The service primarily distributes original and acquired films and television shows from various genres, and it is available internationally in multiple languages. Netflix was founded by Marc Randolph and Reed Hastings on August 29, 1997, in Scotts Valley, California.

Problem Statement:

Analyze the Netflix dataset to gain insights into its content catalog and user engagement patterns. Identify trends, preferences, and potential areas for improvement in content offerings and platform performance

---

Basic Metrics Analysis:

1.Content Popularity Analysis: • Identify the most popular genres, themes, and formats of content among subscribers in various countries. Analyze viewing patterns, ratings, and feedback to understand audience preferences. • Explore metrics such as viewership duration, and user engagement to gauge the success of different types of content.

1. Market Segmentation: • Segment the audience based on different type of modes of viewership like TV Shows or Movies. • Identify what can be the best periods to deliver a particular content and what kind of content is available in different countries.
2. Content Performance Metrics: • Evaluate the performance of existing content in terms of viewer retention, audience satisfaction, and critical acclaim. • Identify trends and patterns in viewer behavior to inform content production decisions and optimize the content library for maximum engagement.
3. Localization and Cultural Relevance:• Analyze the success of localized content and its impact on subscriber growth and reten- tion in different markets. •Determine the level of cultural relevance and authenticity required to resonate with audiences in specific regions and time periods.
4. User Feedback and Recommendations: • Gather insights from user reviews, ratings, and recommendations to understand audience preferences and improve content selection algorithms. • Understand what kind of genres are more popular than others and visual analysis as much as possible to understand key points regarding preferences.

In [1]:
```
import warnings
warnings.filterwarnings("ignore")
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv("netflix.csv")
df.head(5)
```

Out[2]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

2.Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required),*

missing value detection, statistical summary

shape of data

In [4]:
```
df.shape
```

Out[4]:
```
(8807, 12)
```

Insights:There are 8807 rows and 12 columns in this dataframe.

data types of all the attributes

In [5]:
```
df.dtypes
```

Out[5]:
```
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
```

Insight: The above mentioned cell gives us the information about shape and datatype all the columns.

conversion of categorical attributes to 'category' (If required)

Converting categorical attributes to the 'category' data type in Python (using pandas) is not compulsory, but it can offer several advantages like ; Memory Efficiency,Performance Improvements,Implicit Ordering,Compatibility However,these advantages arent crucial for my specific use case, as i am working with small datasets where memory efficiency isnt a concern, so i will not convert categorical attributes to 'category'

statistical summary before data cleaning:

In [5]:
```
df.describe()
```

Out[5]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

missing value detection

In [6]:
```
df.isnull().any()
```

Out[6]:
```
show_id         False
type            False
title           False
director         True
cast             True
country          True
date_added       True
release_year    False
rating           True
duration         True
listed_in       False
description     False
dtype: bool
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
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
```

In [8]:
```
df.isnull().sum().sum()
```

Out[8]:
```
4307
```

Insight:The above mentioned cell gives us the information about all the columns, their data types and Non-Null Count.Director column ,cast column , country column have the most null values.The total null values is 4307.

In [9]:
```
df.nunique()
```

Out[9]:
```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1767
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
```

In [10]:
```
df.nunique().sum()
```

Out[10]:
```
41951
```

Insight: The above cell gives us the number of unique values present in the dataset.there is total 41951 unique values.

filling missing values

In [11]:
```
df.director.fillna('Unknown Director',inplace=True)
df.cast.fillna('Unknown Cast',inplace=True)
df.country.fillna('Unknown country',inplace=True)
df.dropna(subset=['date_added','rating','duration'],inplace=True)
df
```

Out[11]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown Cast	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown country	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	Unknown Director	Unknown Cast	Unknown country	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	Unknown Director	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...
...	...	...	...	...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8803	s8804	TV Show	Zombie Dumb	Unknown Director	Unknown Cast	Unknown country	July 1, 2019	2018	TV-Y7	2 Seasons	Kids' TV, Korean TV Shows, TV Comedies	While living alone in a spooky town, a young g...
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	Comedies, Horror Movies	Looking to survive in a world taken over by zo...
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

8790 rows × 12 columns

missing value detection

In [12]:
```
df.isnull().any()
```

Out[12]:
```
show_id         False
type            False
title           False
director        False
cast            False
country         False
date_added      False
release_year    False
rating          False
duration        False
listed_in       False
description     False
dtype: bool
```

Insights: It is clear from above data that there are no remaining NaN values in any column.

Statistical Summary After Data Cleaning

In [13]:
```
df.describe()
```

Out[13]:

	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

Unnesting of data

Unnesting data typically refers to the process of flattening nested data structures, such as lists of lists or dictionaries of dictionaries, into a single-level structure for easier manipulation and analysis

Unnesting of Cast Column

In [14]:
```
cast_df = df[["title", "cast"]]
cast_df["list_of_cast"] = cast_df.cast.apply(lambda x: str(x).split(", "))
cast_df = cast_df.explode("list_of_cast")
cast_df
```

Out[14]:

	title	cast	list_of_cast
0	Dick Johnson Is Dead	Unknown Cast	Unknown Cast
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Ama Qamata
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Khosi Ngema
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Gail Mabalane
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Thabang Molaba
...	...	...	...
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Manish Chaudhary
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Meghna Malik
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Malkeet Rauni
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Anita Shabdish
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Chittaranjan Tripathy

64841 rows × 3 columns

Unnesting of Director Column

In [15]:
```
director_df = df[["title","type", "director"]]
director_df['list_of_director']=director_df.director.apply(lambda x:str(x).split(","))
director_df=director_df.explode('list_of_director')
director_df
```

Out[15]:

	title	type	director	list_of_director
0	Dick Johnson Is Dead	Movie	Kirsten Johnson	Kirsten Johnson
1	Blood & Water	TV Show	Unknown Director	Unknown Director
2	Ganglands	TV Show	Julien Leclercq	Julien Leclercq
3	Jailbirds New Orleans	TV Show	Unknown Director	Unknown Director
4	Kota Factory	TV Show	Unknown Director	Unknown Director
...	...	...	...	...
8802	Zodiac	Movie	David Fincher	David Fincher
8803	Zombie Dumb	TV Show	Unknown Director	Unknown Director
8804	Zombieland	Movie	Ruben Fleischer	Ruben Fleischer
8805	Zoom	Movie	Peter Hewitt	Peter Hewitt
8806	Zubaan	Movie	Mozez Singh	Mozez Singh

9595 rows × 4 columns

Unnesting of Country column

In [16]:

```
country_df = df[["title","type","country"]]
def split_a_str(s):
    return str(s).split(', ')
country_df["list_of_country"] = country_df.country.apply(split_a_str)
country_df = country_df.explode("list_of_country")
country_df
```

Out[16]:

	title	type	country	list_of_country
0	Dick Johnson Is Dead	Movie	United States	United States
1	Blood & Water	TV Show	South Africa	South Africa
2	Ganglands	TV Show	Unknown country	Unknown country
3	Jailbirds New Orleans	TV Show	Unknown country	Unknown country
4	Kota Factory	TV Show	India	India
...	...	...	...	...
8802	Zodiac	Movie	United States	United States
8803	Zombie Dumb	TV Show	Unknown country	Unknown country
8804	Zombieland	Movie	United States	United States
8805	Zoom	Movie	United States	United States
8806	Zubaan	Movie	India	India

10828 rows × 4 columns

Merging of unnested data

Merging DataFrames in Python, particularly using the pandas library, is a common operation when working with tabular data. There are several ways to merge DataFrames, including concatenation, joining, and merging based on common columns

In [17]:
```
merge_df = pd.merge(left=cast_df,right = country_df,on="title")
merge_df
```

Out[17]:

	title	cast	list_of_cast	type	country	list_of_country
0	Dick Johnson Is Dead	Unknown Cast	Unknown Cast	Movie	United States	United States
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Ama Qamata	TV Show	South Africa	South Africa
2	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Khosi Ngema	TV Show	South Africa	South Africa
3	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Gail Mabalane	TV Show	South Africa	South Africa
4	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Thabang Molaba	TV Show	South Africa	South Africa
...	...	...	...	...	...	...
81593	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Manish Chaudhary	Movie	India	India
81594	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Meghna Malik	Movie	India	India
81595	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Malkeet Rauni	Movie	India	India
81596	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Anita Shabdish	Movie	India	India
81597	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	Chittaranjan Tripathy	Movie	India	India

81598 rows × 6 columns

Visual Analysis

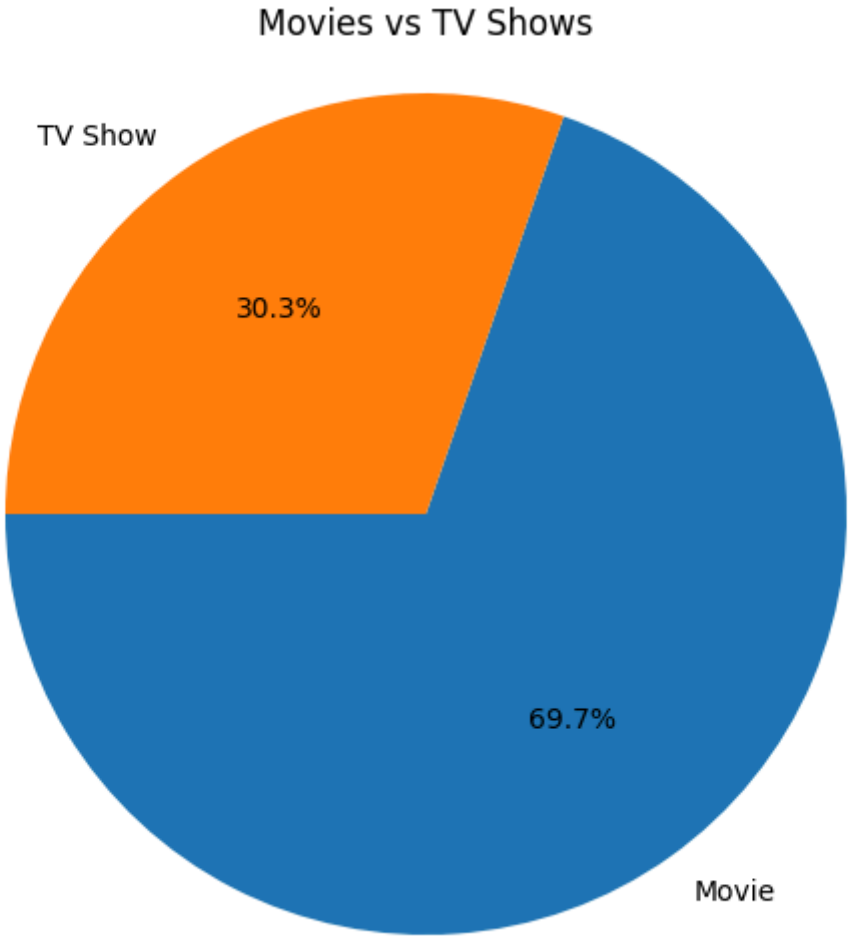
Visual analysis of data is a crucial step in exploring and understanding your datasets. Python provides several powerful libraries for creating visualizations, with matplotlib and seaborn being two of the most popular ones.Here I have used various plots like - Line plot - Box plot - Heatmap - Pie Chart - Bar Chart

1.Comparison of TV Show and Movies

In [18]:
```
content_type_counts = df['type'].value_counts()
content_type_counts
```

Out[18]:
```
Movie      6126
TV Show    2664
Name: type, dtype: int64
```

In [19]:
```
plt.figure(figsize=(8, 6))
plt.title("Movies vs TV Shows")
plt.pie(content_type_counts, labels=content_type_counts.index, autopct='%1.1f%%', startangle=180)
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle
plt.show()
```



Insights: There are far more movie titles (69.7%) than that of TV shows titles (30.3%) in terms of title.The above piechart shows us that the number of movies being made is vastly more than number of TV Shows being made. From this we can infer that it is more profitable to release movies on netflix than TV Shows.

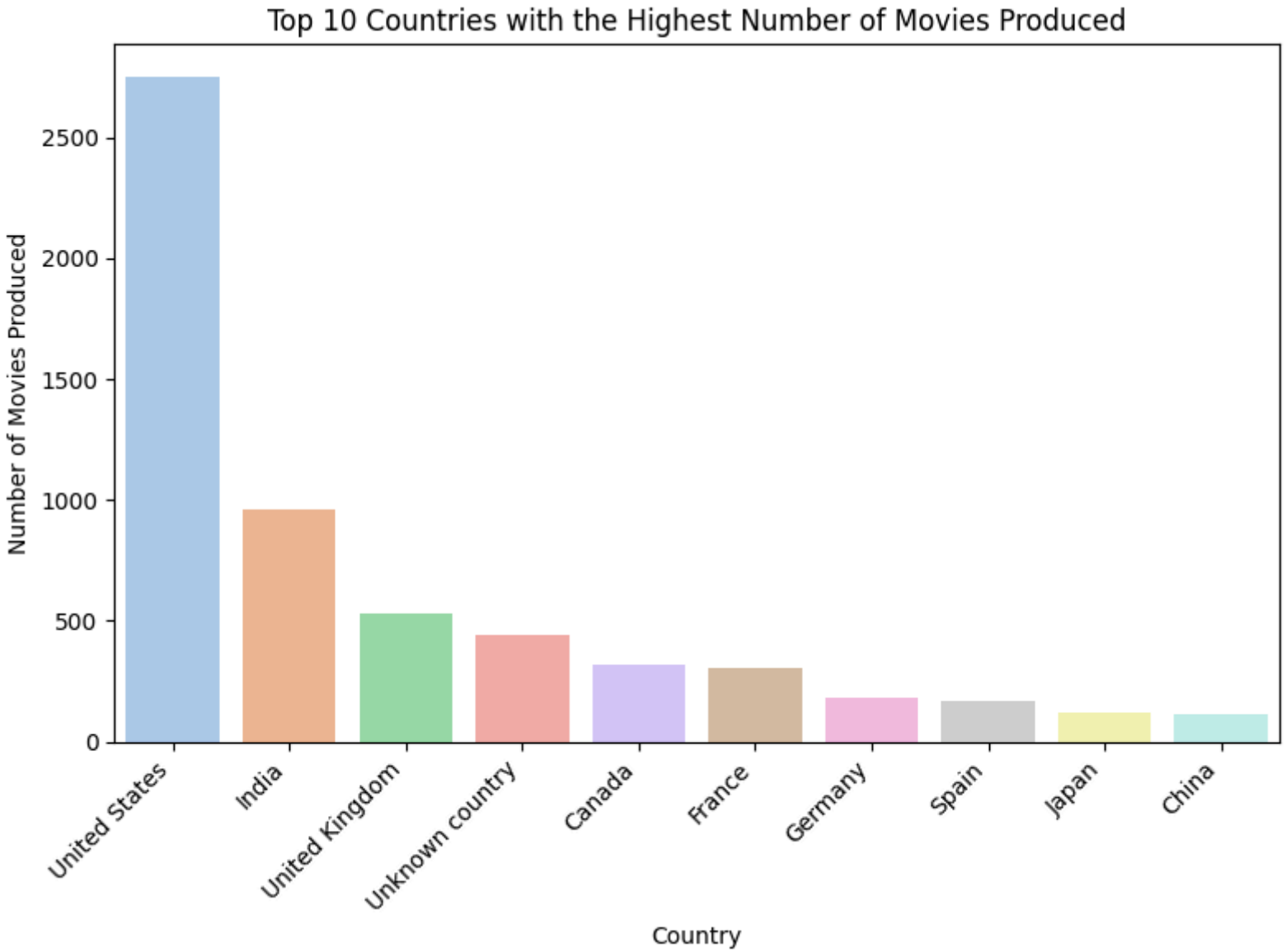
2.The number of movies produced in each country and pick the top 10 countries.

In [24]:
```
movies_df=merge_df[merge_df.type=="Movie"]
movies_by_country = movies_df.groupby('list_of_country')['title'].nunique().reset_index()
top_10_countries = movies_by_country.sort_values(by='title', ascending=False).head(10).reset_index()
top_10_countries
```

Out[24]:

	index	list_of_country	title
0	114	United States	2748
1	43	India	962
2	112	United Kingdom	532
3	116	Unknown country	439
4	20	Canada	319
5	34	France	303
6	36	Germany	182
7	100	Spain	171
8	51	Japan	119
9	23	China	114

In [27]:
```
plt.figure(figsize=(8, 6))
sns.barplot(x='list_of_country', y='title', data=top_10_countries, palette='pastel')
plt.xlabel('Country')
plt.ylabel('Number of Movies Produced')
plt.title('Top 10 Countries with the Highest Number of Movies Produced')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Insights:The United States stands out as the leading producer of movies, with a significantly higher count compared to other countries. India ranks second in terms of the number of movies produced, indicating the growing significance of the Indian film industry.

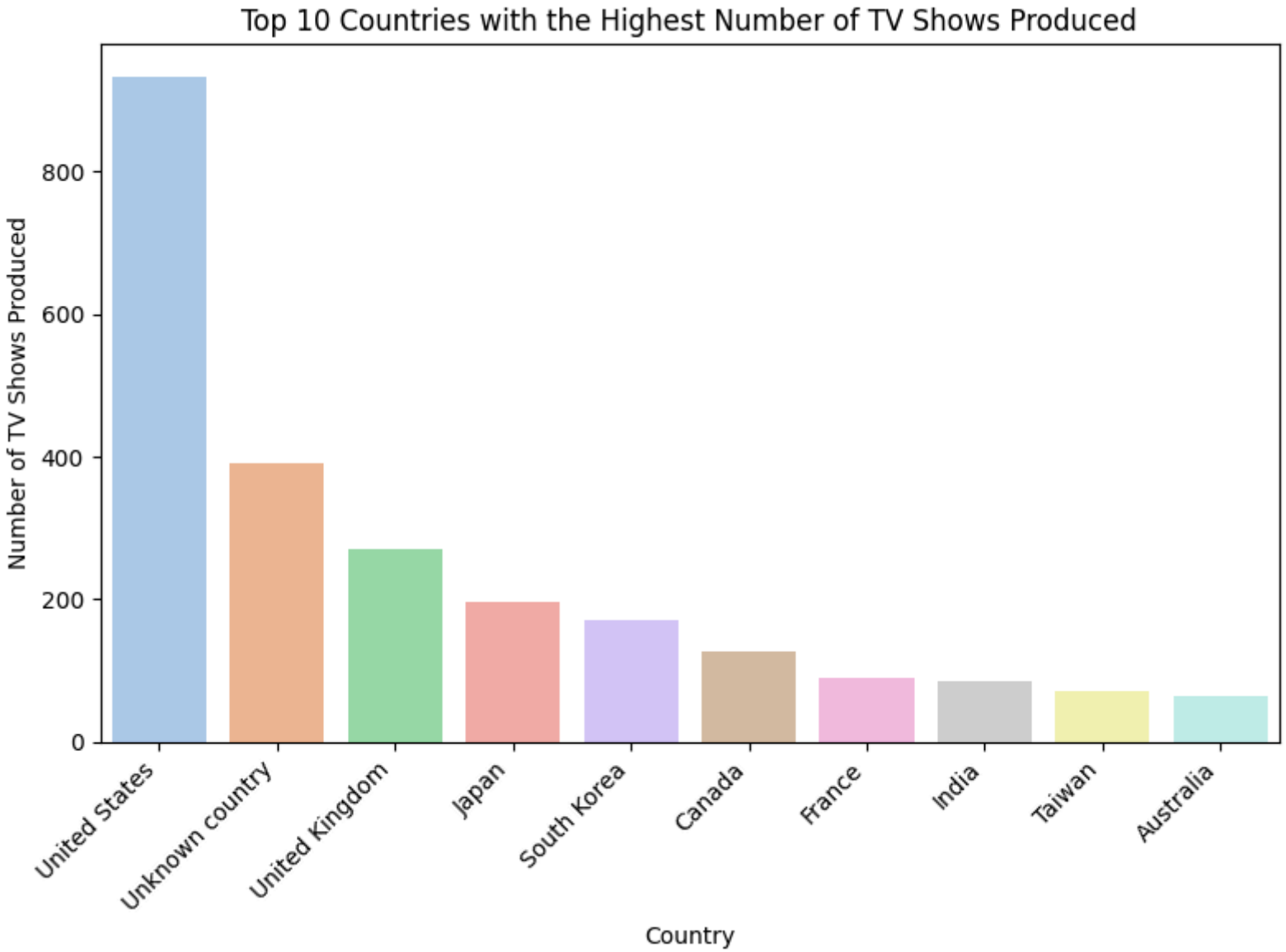
3.The number of Tv-Shows produced in each country and pick the top 10 countries.

In [32]:
```
TV_Show_df=merge_df[merge_df['type']=='TV Show']
TV_Show_by_country=TV_Show_df.groupby('list_of_country')['title'].nunique().reset_index()
top_10_countries=TV_Show_by_country.sort_values(by='title',ascending=False).head(10).reset_index()
top_10_countries
```

Out[32]:

	index	list_of_country	title
0	63	United States	932
1	64	Unknown country	390
2	62	United Kingdom	271
3	30	Japan	197
4	52	South Korea	170
5	8	Canada	126
6	19	France	90
7	25	India	84
8	57	Taiwan	70
9	2	Australia	64

In [34]:
```
plt.figure(figsize=(8, 6))
sns.barplot(x='list_of_country', y='title', data=top_10_countries, palette='pastel')
plt.xlabel('Country')
plt.ylabel('Number of TV Shows Produced')
plt.title('Top 10 Countries with the Highest Number of TV Shows Produced')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Insights:Similar to the movie production trend, the United States emerges as the leading producer of TV shows, with a significantly higher count compared to other countries. The United Kingdom, Canada, Australia, and South Korea are among the top producers of TV shows, reflecting the influence of English-language and Korean-language television content globally.

4.Analysis of actors/directors of different types of shows/movies.

4.1the top 10 directors who have appeared in most movies or TV shows

In [36]:
```
merge2_df = pd.merge(left=director_df,right = country_df,on="title")
merge2_df
```

Out[36]:

	title	type_x	director	list_of_director	type_y	country	list_of_country
0	Dick Johnson Is Dead	Movie	Kirsten Johnson	Kirsten Johnson	Movie	United States	United States
1	Blood & Water	TV Show	Unknown Director	Unknown Director	TV Show	South Africa	South Africa
2	Ganglands	TV Show	Julien Leclercq	Julien Leclercq	TV Show	Unknown country	Unknown country
3	Jailbirds New Orleans	TV Show	Unknown Director	Unknown Director	TV Show	Unknown country	Unknown country
4	Kota Factory	TV Show	Unknown Director	Unknown Director	TV Show	India	India
...	...	...	...	...	...	...	...
11890	Zodiac	Movie	David Fincher	David Fincher	Movie	United States	United States
11891	Zombie Dumb	TV Show	Unknown Director	Unknown Director	TV Show	Unknown country	Unknown country
11892	Zombieland	Movie	Ruben Fleischer	Ruben Fleischer	Movie	United States	United States
11893	Zoom	Movie	Peter Hewitt	Peter Hewitt	Movie	United States	United States
11894	Zubaan	Movie	Mozez Singh	Mozez Singh	Movie	India	India

11895 rows × 7 columns

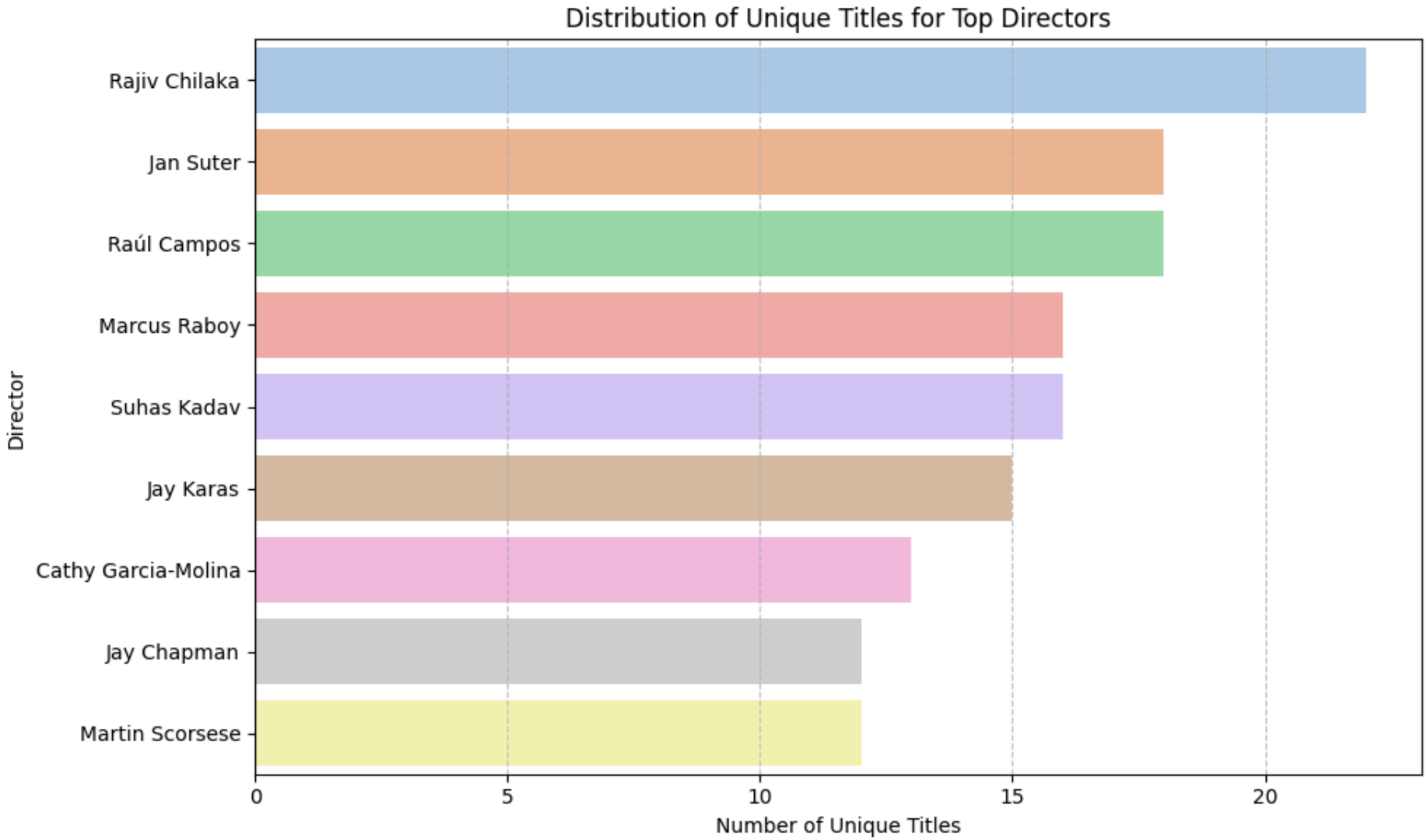
In [61]:
```
top_directors=merge2_df.groupby(["list_of_director"])['title'].nunique().sort_values(ascending=False).head(10).reset_
top_directors=top_directors[1:10]
top_directors
```

Out[61]:

	list_of_director	title
1	Rajiv Chilaka	22
2	Jan Suter	18
3	Raúl Campos	18
4	Marcus Raboy	16
5	Suhas Kadav	16
6	Jay Karas	15
7	Cathy Garcia-Molina	13
8	Jay Chapman	12
9	Martin Scorsese	12

In [63]:

```
plt.figure(figsize=(10, 6))
sns.barplot(x='title', y='list_of_director', data=top_directors, palette='pastel')
plt.xlabel('Number of Unique Titles')
plt.ylabel('Director')
plt.title('Distribution of Unique Titles for Top Directors')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



Insights:1.The "Unknown Director" has the highest number of unique titles (2621). This suggests that a significant portion of the dataset either lacks director information. 2.Rajiv Chilaka and Suhas Kadav are prominent directors with 22 and 16 unique titles.

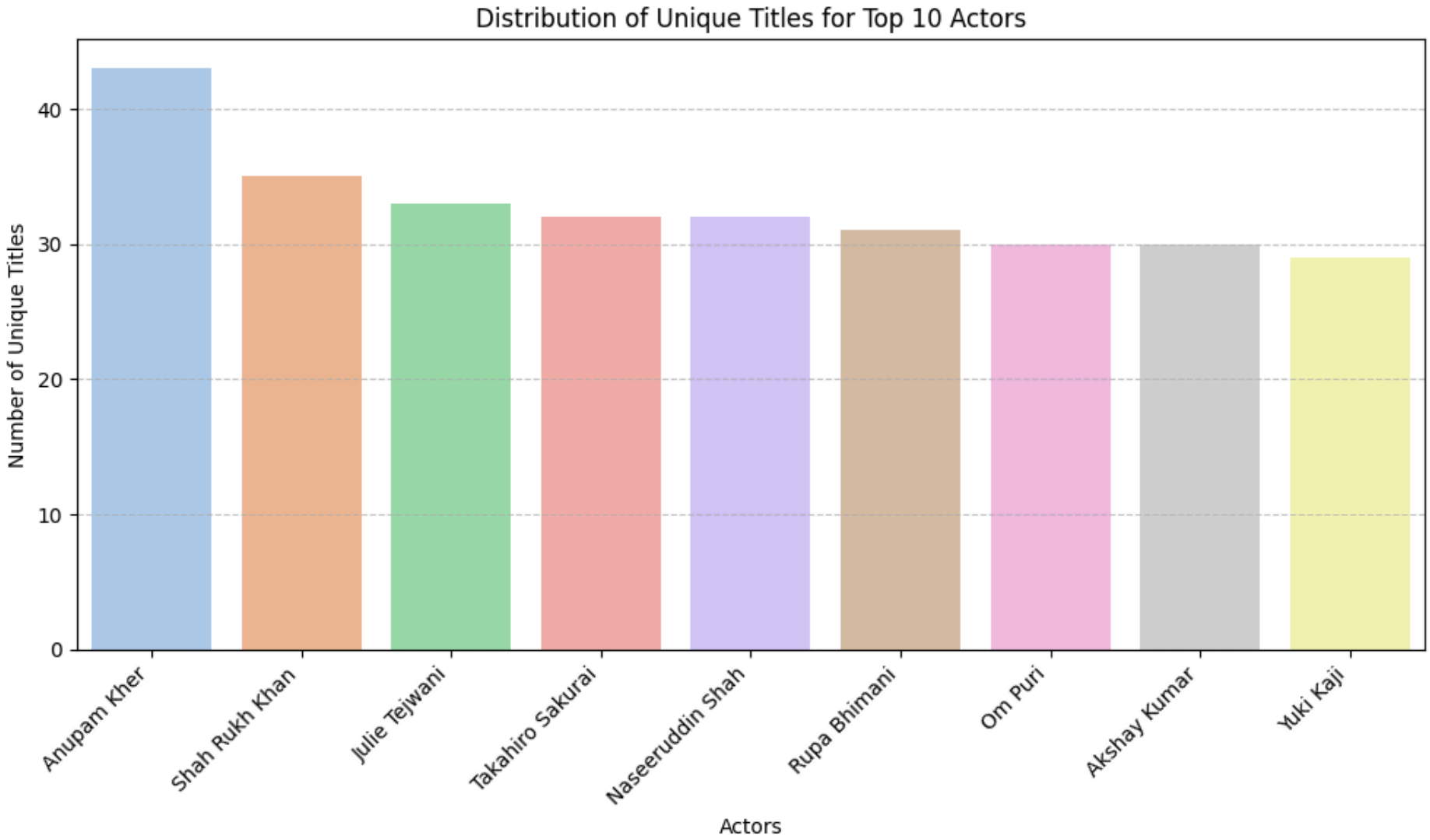
4.2 The top 10 Actor who have appeared in most movies or TV shows

In [58]:
```
Actors_counts=merge_df.groupby(["list_of_cast"])['title'].nunique().sort_values(ascending=False).head(10).reset_index(
Actors_counts=Actors_counts[1:10]
Actors_counts
```

Out[58]:

	list_of_cast	title
1	Anupam Kher	43
2	Shah Rukh Khan	35
3	Julie Tejwani	33
4	Takahiro Sakurai	32
5	Naseeruddin Shah	32
6	Rupa Bhimani	31
7	Om Puri	30
8	Akshay Kumar	30
9	Yuki Kaji	29

In [60]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x='list_of_cast', y='title', data=Actors_counts, palette='pastel')
plt.xlabel('Actors')
plt.ylabel('Number of Unique Titles')
plt.title('Distribution of Unique Titles for Top 10 Actors')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



Insights;The category labeled as "Unknown Cast" has the highest number of unique titles (825). This suggests a substantial portion of the dataset either lacks specific actor information.Anupam Kher and Shah Rukh Khan are among the top actors in the dataset, with 43 and 35 unique titles, respectively.

5.Genre movies are more popular or produced more.

In [64]:
```
filtered_genre = df[['title', 'listed_in']]
filtered_genre['list_of_genre'] = filtered_genre['listed_in'].apply(lambda x: str(x).split(', '))
filtered_genre = filtered_genre.explode('list_of_genre')

# Count unique genres and select the top 20
filtered_genre = filtered_genre['list_of_genre'].value_counts().reset_index()
print(filtered_genre.head(20))
```

```
                       index  list_of_genre
0       International Movies           2752
1                     Dramas           2427
2                   Comedies           1674
3     International TV Shows           1351
4              Documentaries            869
5         Action & Adventure            859
6                  TV Dramas            763
7         Independent Movies            756
8   Children & Family Movies            641
9            Romantic Movies            616
10               TV Comedies            581
11                 Thrillers            577
12            Crime TV Shows            470
13                 Kids' TV            451
14                Docuseries            395
15          Music & Musicals            375
16         Romantic TV Shows            370
17             Horror Movies            357
18           Stand-Up Comedy            343
19                Reality TV            255
```

In [65]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
genre_counts = dict(zip(filtered_genre['index'], filtered_genre['list_of_genre']))
# Generate word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(genre_counts)
# Plot the word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.title('Word Cloud of Movie Genres')
plt.axis('off')
plt.show()
```

Insights:In order to display which type of movies are more popular, we have used wordcloud. This helps us in seeing all the major and important and major genres in a bigger font and different colour than others.

6.1 The best month to release the Tv-show .

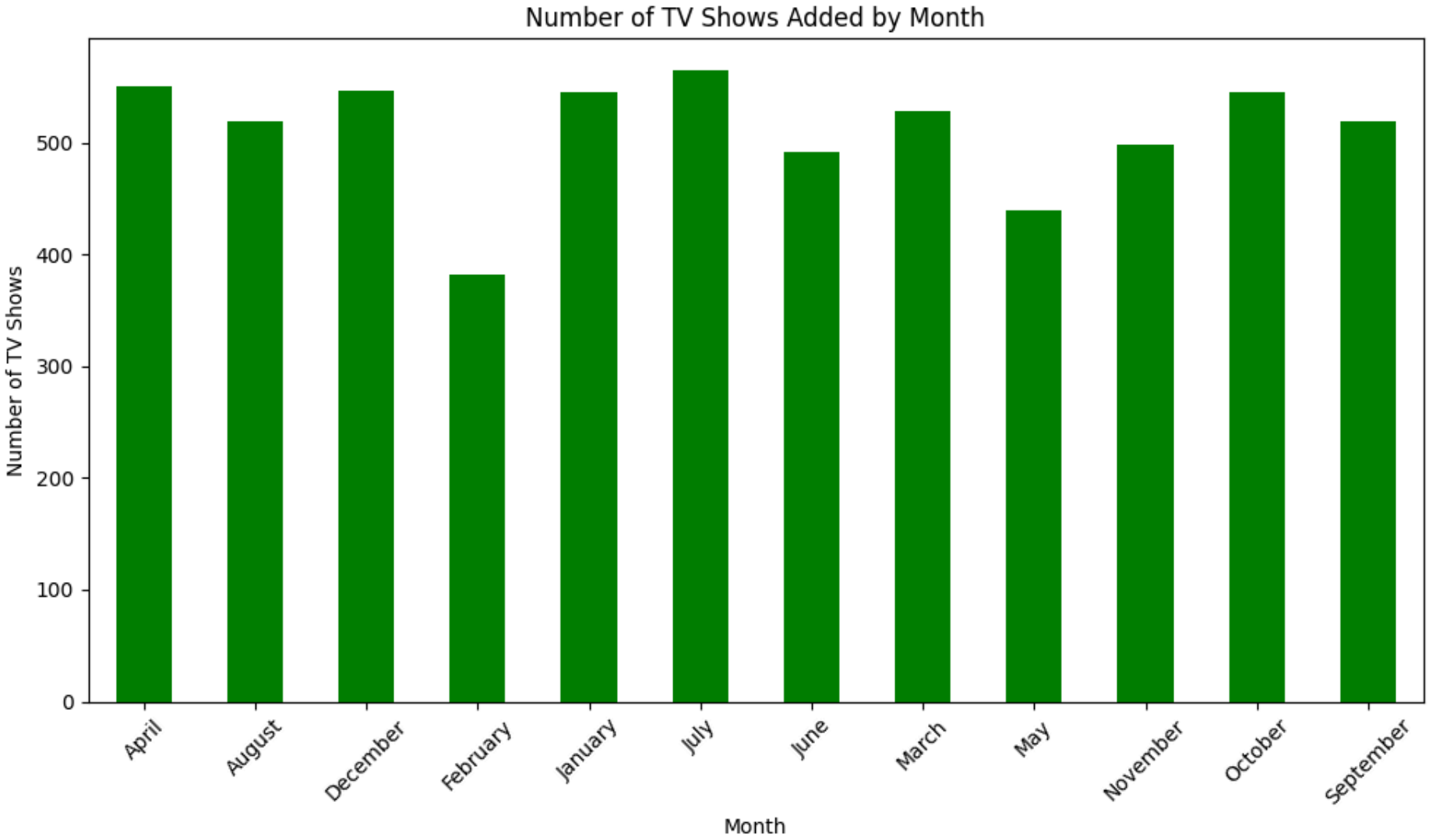
In [68]:
```
tv_show=df[df['type']=='TV Show']
tv_show['date_added']=pd.to_datetime(tv_show['date_added'])
tv_show['Month']=tv_show['date_added'].dt.month_name()

Month_counts=tv_show['Month'].value_counts()
Month_counts
```

Out[68]:
```
December     266
July         262
September    251
August       236
June         236
October      215
April        214
March        213
November     207
May          193
January      192
February     181
Name: Month, dtype: int64
```

In [69]:
```
best_Month_counts=Month_counts.idxmax()
best_Month_counts
```

Out[69]:
```
'December'
```

In [104…
```
plt.figure(figsize=(10, 6))
Month_counts.plot(kind='bar', color='green')
plt.title('Number of TV Shows Added by Month')
plt.xlabel('Month')
plt.ylabel('Number of TV Shows')
plt.xticks(rotation=45)  # Rotating x-axis labels for better readability
plt.tight_layout()
plt.show()
```



Insights:December and July have the highest numbers of TV shows added, with 266 and 262 respectively. This could be due to various factors such as holiday seasons, school breaks. Months like June, July, and August (typically summer months in many regions) show relatively high numbers of TV shows added.

6.2 The best month to release the Movie .

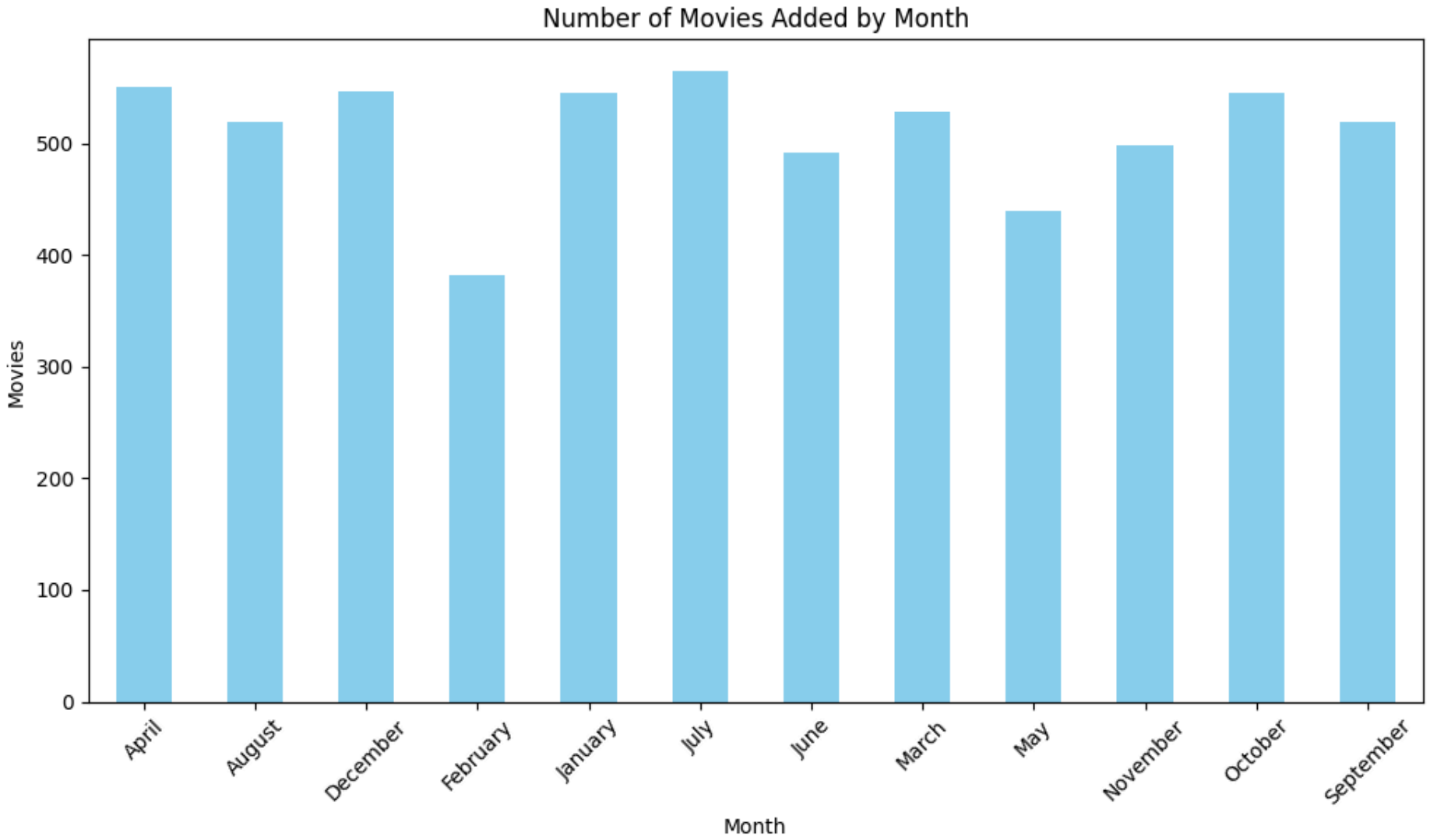
In [77]:
```
Movie_df=df[df['type']=='Movie']
Movie_df['date_added']=pd.to_datetime(Movie_df['date_added'])
Movie_df['Month']=Movie_df['date_added'].dt.month_name()

Month_counts=Movie_df['Month'].value_counts()
Month_counts
```

Out[77]:
```
July         565
April        550
December     547
January      546
October      545
March        529
September    519
August       519
November     498
June         492
May          439
February     382
Name: Month, dtype: int64
```

In [78]:
```
best_month=Month_counts.idxmax()
best_month
```

Out[78]:
```
'July'
```

In [80]:
```
Month_counts = Month_counts.sort_index()
plt.figure(figsize=(10, 6))
Month_counts.plot(kind='bar', color='skyblue')
plt.title('Number of Movies Added by Month')
plt.xlabel('Month')
plt.ylabel('Movies')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Insights:December and July have the highest numbers of Movies added, with 565 and 550 respectively. This could be due to various factors such as holiday seasons, school breaks. Months like June, July, and August (typically summer months in many regions) show relatively high numbers of TV shows added.

6.3 The best week to release the MOVIE

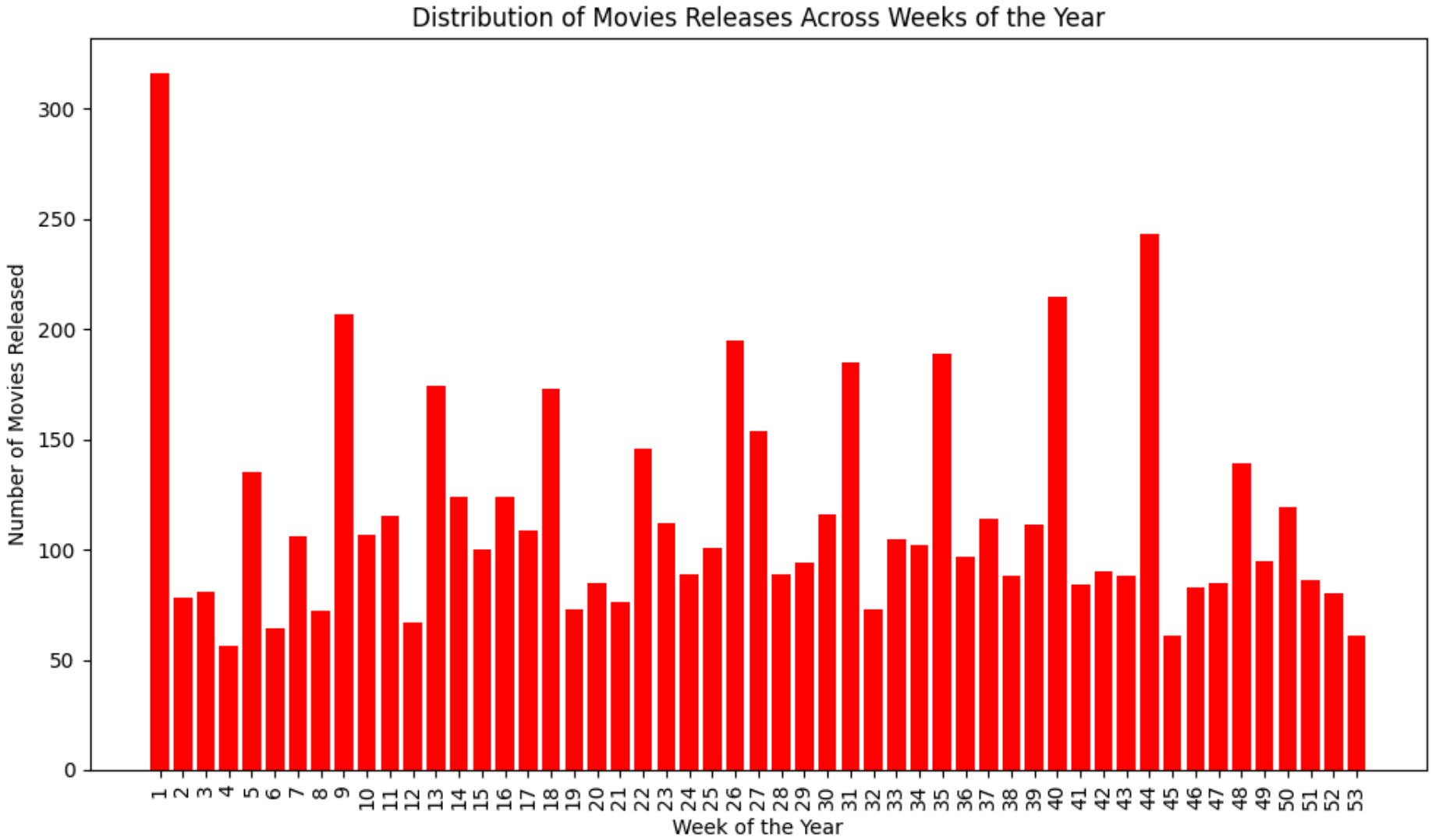
In [106…
```
movies_df=df[df['type']=='Movie']
movies_df['date_added']=pd.to_datetime(movies_df['date_added'])
movies_df['week']=movies_df['date_added'].dt.isocalendar().week
movie_week_counts=movies_df['week'].value_counts()
movie_week_counts.head(10)
```

Out[106]:
```
1     316
44    243
40    215
9     207
26    195
35    189
31    185
13    174
18    173
27    154
Name: week, dtype: Int64
```

In [107…
```
best_movie_week = movie_week_counts.idxmax()
best_movie_week
```

Out[107]:
```
1
```

In [109…
```
movie_week_counts = movies_df['week'].value_counts().sort_values()
plt.figure(figsize=(10, 6))
plt.bar(movie_week_counts.index, movie_week_counts.values, color='red')
plt.xlabel('Week of the Year')
plt.ylabel('Number of Movies Released')
plt.title('Distribution of Movies Releases Across Weeks of the Year')
plt.xticks(range(1, 54), rotation=90)
plt.tight_layout()
plt.show()
```



Insights:Week 1 stands out with the highest number of Movies added (316). This suggests a trend of increased activity in content releases at the beginning of the year.

6.4 The best week to release the TV SHOW

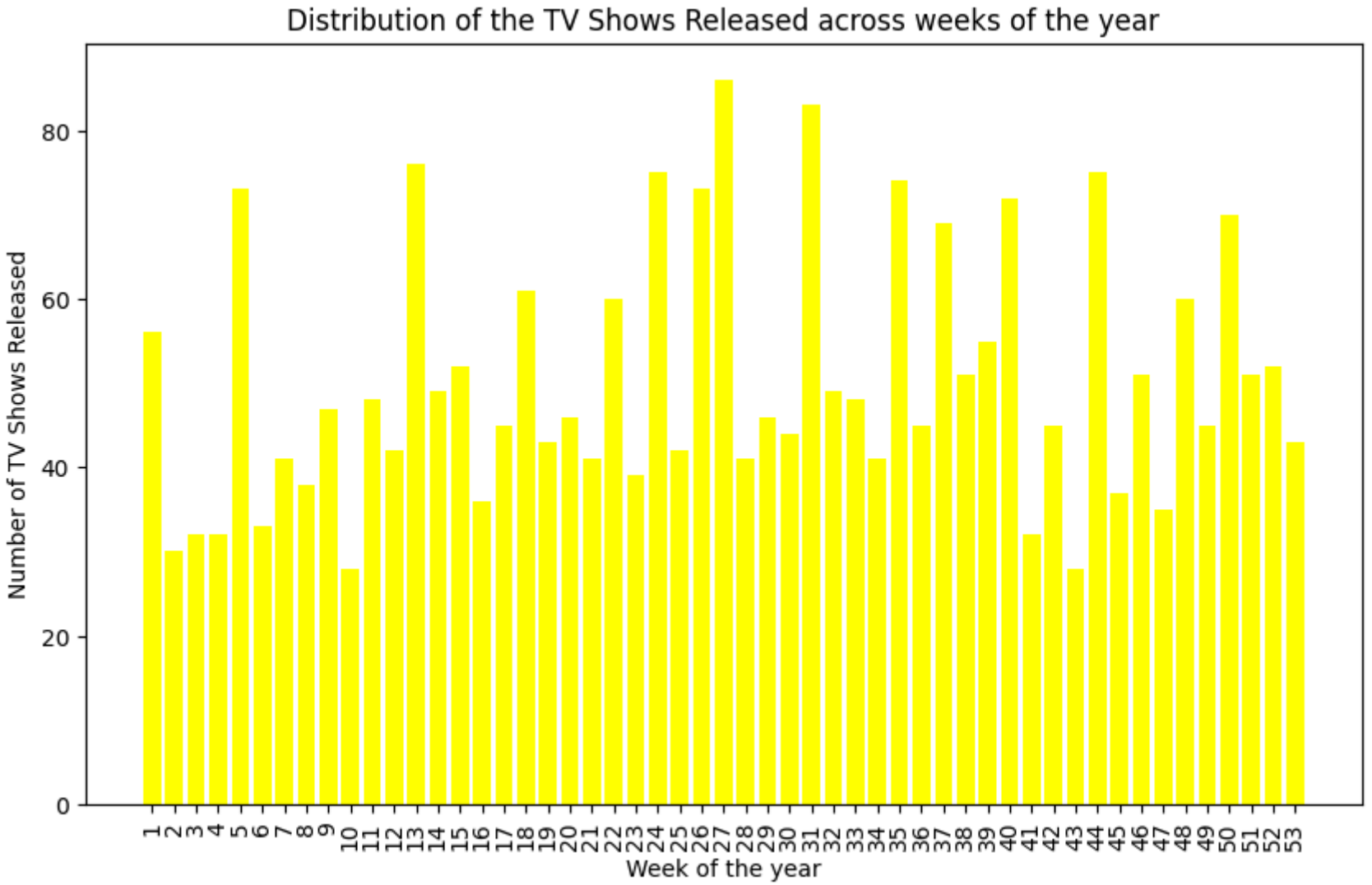
In [88]:
```
tv_show=df[df['type']=='TV Show']
tv_show['date_added']=pd.to_datetime(tv_show['date_added'])
tv_show['week']=tv_show['date_added'].dt.isocalendar().week
tv_show_week_counts=tv_show['week'].value_counts()
tv_show_week_counts.head(10)
```

Out[88]:
```
27    86
31    83
13    76
44    75
24    75
35    74
26    73
5     73
40    72
50    70
Name: week, dtype: Int64
```

In [89]:
```
best_tv_show_week=tv_show_week_counts.idxmax()
best_tv_show_week
```

Out[89]:
```
27
```

In [111…
```
plt.figure(figsize=(10,6))
plt.bar(tv_show_week_counts.index,tv_show_week_counts.values,color='yellow')
plt.xlabel('Week of the year')
plt.ylabel('Number of TV Shows Released')
plt.title('Distribution of the TV Shows Released across weeks of the year')
plt.xticks(range(1,54),rotation = 90)
plt.show()
```



Insights:Week 27 stands out with the highest number of TV shows added (86)

7 After how many days the movie will be added to Netflix after the release of the movie (you can consider the recent past data)

In [113…
```
movies_df=df[df['type']=='Movie']
movies_df['date_added'] = pd.to_datetime(movies_df['date_added'])
movies_df['release_year'] = pd.to_datetime(movies_df['release_year'], format='%Y')
movies_df['Days_to_addition'] = (movies_df['date_added'] - movies_df['release_year']).dt.days

movies_df[['show_id','type','title','Days_to_addition']]
```

Out[113]:

	show_id	type	title	Days_to_addition
0	s1	Movie	Dick Johnson Is Dead	633
6	s7	Movie	My Little Pony: A New Generation	266
7	s8	Movie	Sankofa	10493
9	s10	Movie	The Starling	266
12	s13	Movie	Je Suis Karl	265
...	...	...	...	...
8801	s8802	Movie	Zinzana	433
8802	s8803	Movie	Zodiac	4706
8804	s8805	Movie	Zombieland	3956
8805	s8806	Movie	Zoom	5123
8806	s8807	Movie	Zubaan	1521

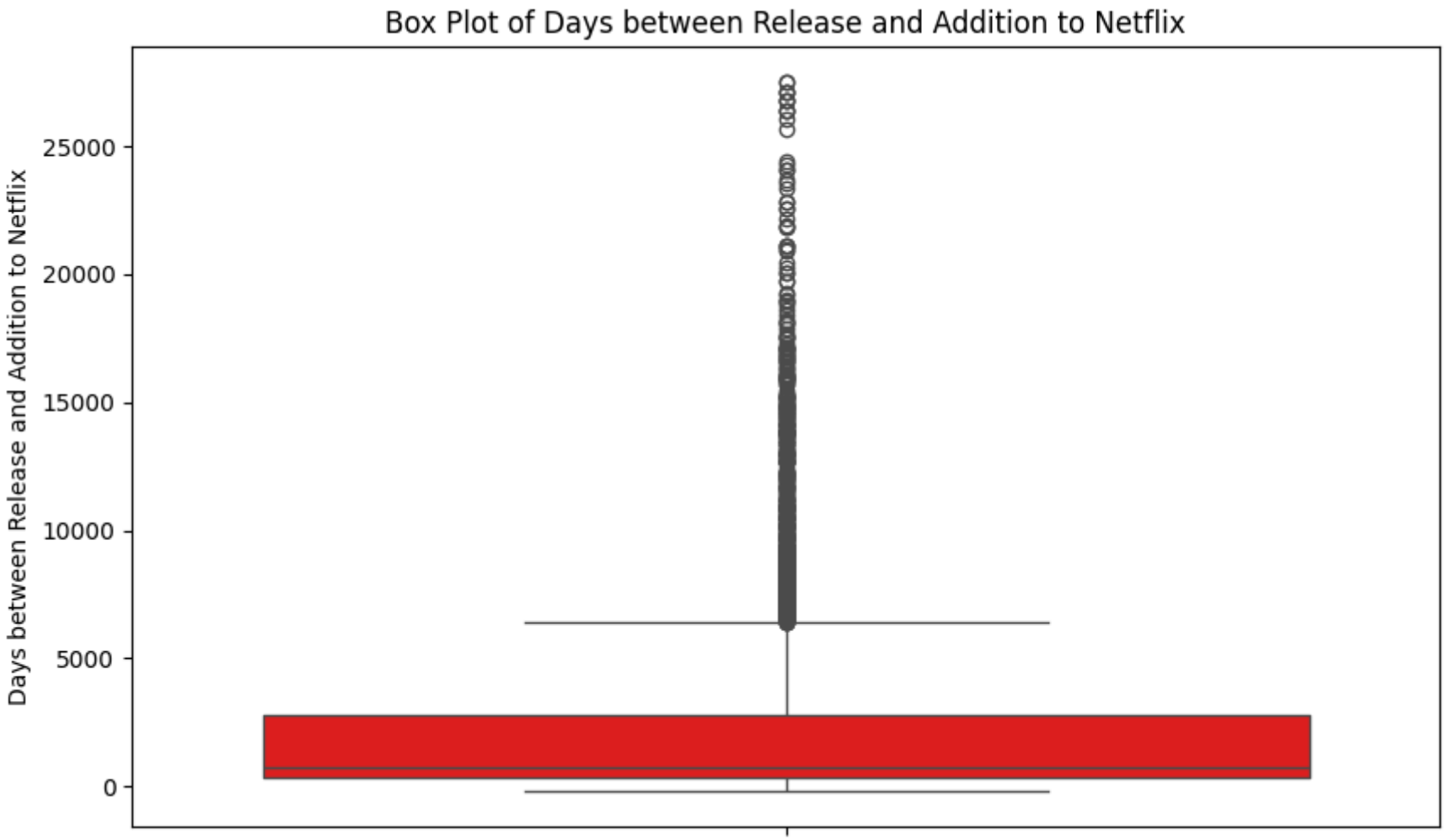
6131 rows × 4 columns

In [114…
```
average_days_to_addition = movies_df['Days_to_addition'].median()  # or median()
average_days_to_addition
```

Out[114]:
```
741.0
```

In [115…
```
plt.figure(figsize=(10, 6))

sns.boxplot(data=movies_df, y='Days_to_addition', color='red')

plt.ylabel('Days between Release and Addition to Netflix')
```

```
plt.title('Box Plot of Days between Release and Addition to Netflix')

plt.show()
```



7.1 After how many days the movie will be added to Netflix after the release of the movie (you can consider the recent past data)

In [116…

```
TV_SHOW_df=df[df['type']=='TV Show']
TV_SHOW_df['date_added'] = pd.to_datetime(TV_SHOW_df['date_added'])
```

```
TV_SHOW_df['release_year'] = pd.to_datetime(TV_SHOW_df['release_year'], format='%Y')
TV_SHOW_df['Days_to_addition'] = (TV_SHOW_df['date_added'] - TV_SHOW_df['release_year']).dt.days
TV_SHOW_df[['show_id','type','title','Days_to_addition']]
```

Out[116]:

	show_id	type	title	Days_to_addition
1	s2	TV Show	Blood & Water	266.0
2	s3	TV Show	Ganglands	266.0
3	s4	TV Show	Jailbirds New Orleans	266.0
4	s5	TV Show	Kota Factory	266.0
5	s6	TV Show	Midnight Mass	266.0
...	...	...	...	...
8795	s8796	TV Show	Yu-Gi-Oh! Arc-V	1216.0
8796	s8797	TV Show	Yunus Emre	382.0
8797	s8798	TV Show	Zak Storm	986.0
8800	s8801	TV Show	Zindagi Gulzar Hai	1810.0
8803	s8804	TV Show	Zombie Dumb	546.0

2676 rows × 4 columns

In [117…

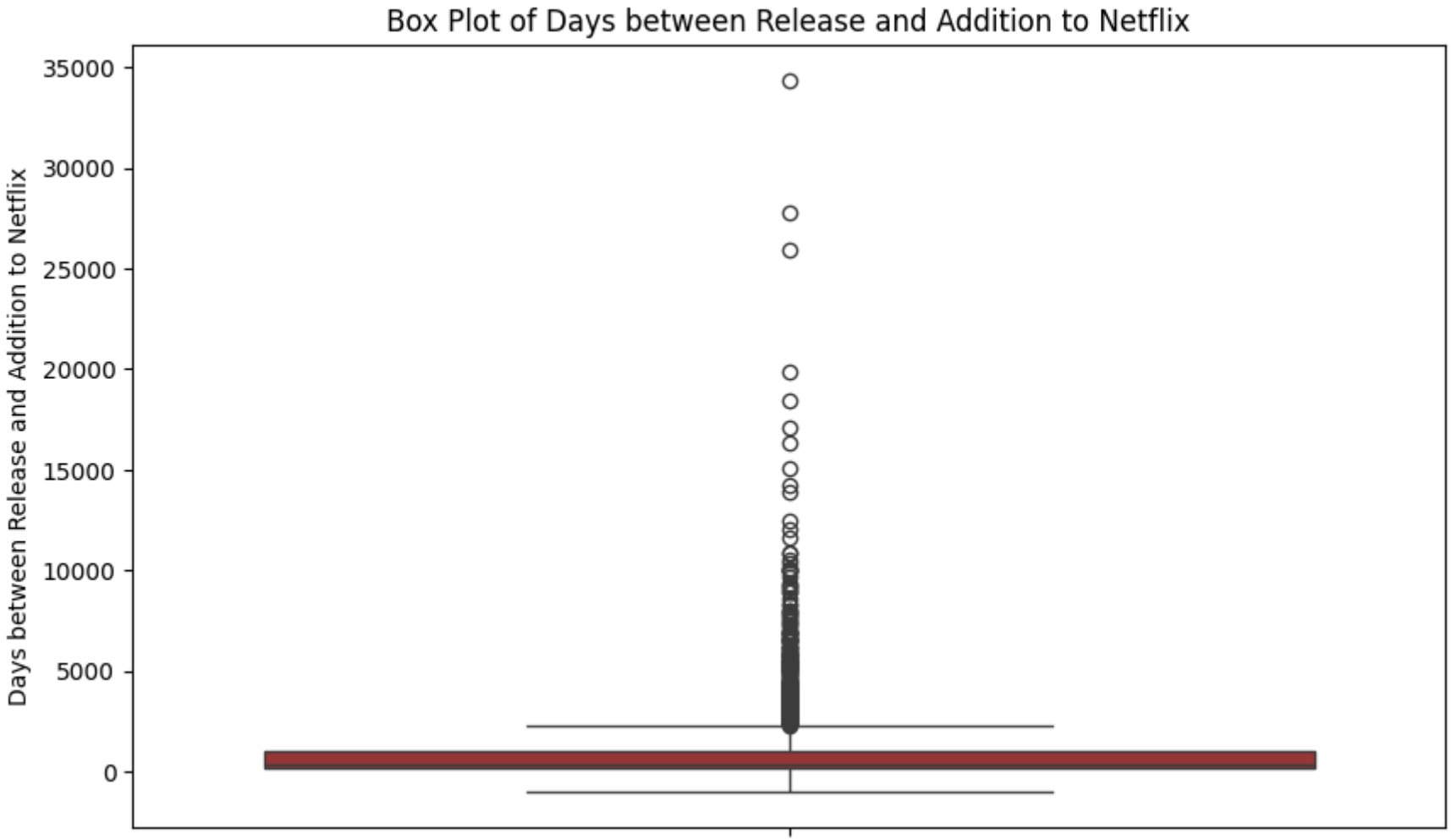
```
average_days_to_addition = TV_SHOW_df['Days_to_addition'].median()  # or median()
average_days_to_addition
```

Out[117]:

```
351.0
```

In [118…

```
plt.figure(figsize=(10, 6))

sns.boxplot(data=TV_SHOW_df, y='Days_to_addition', color='brown')

plt.ylabel('Days between Release and Addition to Netflix')

plt.title('Box Plot of Days between Release and Addition to Netflix')

plt.show()
```



Insights: We have used the box plot to display the numbers for Movie and TV Show because they give us the idea of a median value, low values, high values and outliers. From the plot we can see that in both the cases, Movies and TV Shows, the difference in days is very large and maximum of values lie there only.

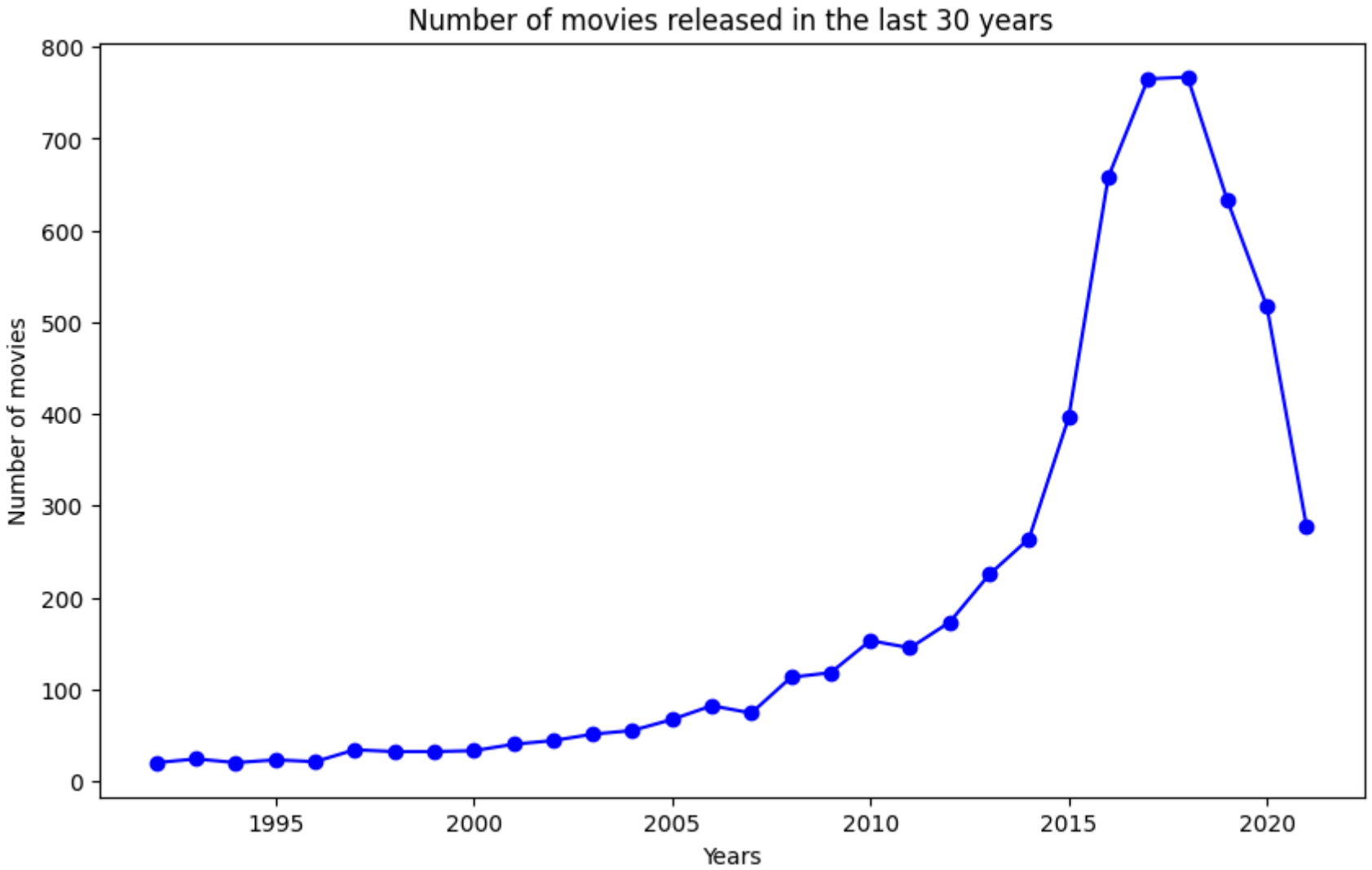
8.How has the number of movies released per year changed over the last 20-30 years?

In [20]:
```
movies_df = df[df['type'] == 'Movie']
movies = movies_df.groupby('release_year')['title'].count()
movies.sort_index(ascending = False)
```

Out[20]:
```
release_year
2021    277
2020    517
2019    633
2018    767
2017    765
       ...
1946      1
1945      3
1944      3
1943      3
1942      2
Name: title, Length: 73, dtype: int64
```

In [21]:
```
movies_last_30_years = movies.tail(30)
```

In [23]:
```
plt.figure(figsize=(10, 6))
plt.plot(movies_last_30_years.index, movies_last_30_years.values, marker='o',color='b', linestyle='-')
plt.title('Number of movies released in the last 30 years')
plt.xlabel('Years')
plt.ylabel('Number of movies')
plt.show()
```



Insights:The above graph shows us that the number of movies released in the last 30 years has been increasing at an extraordinary rate.However, we notice a sharp downward trend from 2015,This is the result of covid pandemic.

Insights and Conclusion:

We have drawn many interesting inferences from the dataset Netflix titles; here's a summary of the few of them: 1.The most viewed content type on Netflix is Movies. 2.The United States stands out as the leading producer of movies and TV Show. 3.The most popular director on Netflix , with the most titles are Rajiv Chilaka and Suhas Kadav, with 22 and 16 unique titles. 4.Anupam Kher and Shah Rukh Khan are among the top actors in the dataset, with 43 and 35 unique titles, respectively. 5.International Movies is a genre that is mostly in Netflix. 6.December and July have the highest numbers of TV shows added, with 266 and 262 respectively. This could be due to various factors such as holiday seasons, school breaks 7.December and July have the highest numbers of Movies added, with 565 and 550 respectively. This could be due to various factors such as holiday seasons, school breaks

It's clear that Netflix has grown over the years. We can see it from the data that the company took certain approaches in their marketing strategy to break into new markets around the world.

Recommendations:

1. Netflix should try to increase the number of TV shows by reaching out to customers and understanding what is popular in order to increase it's TV Shows viewership.
2. The number of movies released drastically went down in 2020. Netflix should partner with Production houses and ramp up the production and making of movies as soon as possible.
3. Netflix should focus on genres Action & Adventure, Comedy, Documentaries and Dramas because they are the ones with most viewership and will likely bring more profits for the company. 4.The most popular markets for the company are United States and India however United States is vastly ahead of India in terms of number of users so, the company should try to penetrate deeper into Indian market by providing local language content in a very personalized way to each state of India. In the US, it should look for genres where there has not been much success and try to promote them as well.
4. Also, according to the data, the best time to release movies is between week 1 and week 15. As for the TV shows it is the month of December and July.

All the above recommendations are some of the possible ways in which Netflix can improve its business and increase its viewership. By implementing them, Netflix can strengthen its position in the streaming industry and sustain long term growth.