

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

```
df = pd.read_csv("netflix_2_project.csv")
df
```

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	25-Sep-21	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	24-Sep-21	2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	24-Sep-21	2021
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	24-Sep-21	2021
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Rai, Alam	India	24-Sep-21	2021

Comments on the Range of Attributes:

In our dataset, we have a diverse range of attributes, each providing different types of information about the movies and TV shows. Below, we describe the key attributes present in our dataset:

- 1. Title: This attribute represents the title or name of each movie or TV show. It serves as a unique identifier for each entry and is crucial for reference.
- 2. Type: This categorical attribute indicates whether the entry is a "Movie" or "TV Show." It helps us distinguish between different types of content.
- 3. Director: This categorical attribute contains the names of the directors responsible for the production of movies or TV shows. It allows us to analyze directorial patterns and identify top directors.
- 4. Cast: Similar to the director attribute, "Cast" is a categorical attribute that lists the names of actors and actresses involved in the production. It helps us understand the popularity and appearances of actors.
- 5. Release Year: This numerical attribute represents the year when the movie or TV show was released. It enables us to analyze temporal trends and patterns over time.
- 6. Country: This categorical attribute specifies the country or countries associated with the production of each entry. It allows us to explore geographic trends and preferences.
- 7. Date Added: This attribute records the date when the content was added to the Netflix platform. It helps us understand when movies or TV shows became available for streaming.
- 8. Rating: This categorical attribute indicates the content rating assigned to each entry, such as "TV-MA," "PG-13," etc. It helps viewers make informed choices based on content appropriateness.
- 9. Duration: This attribute specifies the duration or runtime of each movie or TV show in terms of minutes or seasons/episodes.

10. Listed In: This categorical attribute categorizes entries into genres, themes, or content types. It assists in content recommendation and categorization.

The range of attributes in our dataset provides a rich source of information, allowing us to perform various analyses, such as exploring trends in release years, identifying popular directors and actors, and understanding the distribution of content across countries and genres.

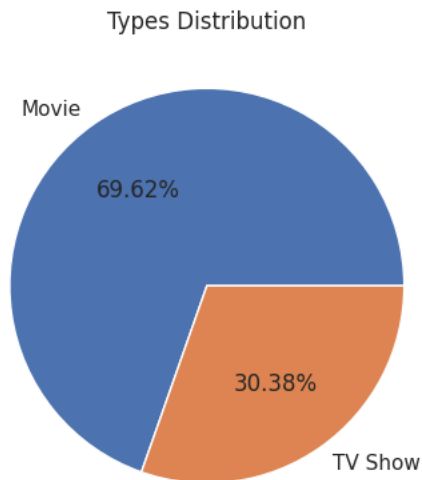
Basic Analysis

```
# Calculate the counts of each type
type_counts = df['type'].value_counts(normalize=True)*100
type_counts

Movie      69.615079
TV Show    30.384921
Name: type, dtype: float64

# Create a pie chart
plt.pie(type_counts, labels=type_counts.index, autopct="%.2f%")
plt.title('Types Distribution')

# Show the pie chart
plt.show()
```



```
# Gives the overall information of all the columns
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  -
0   show_id         8807 non-null   object
1   type            8807 non-null   object
2   title           8807 non-null   object
3   director        6173 non-null   object
4   cast            7982 non-null   object
5   country         7976 non-null   object
6   date_added      8797 non-null   object
7   release_year    8807 non-null   int64
8   rating          8803 non-null   object
9   duration        8804 non-null   object
10  listed_in       8807 non-null   object
11  description      8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
df.shape
```

```
(8807, 12)
```

```
df.isna().sum()

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
```

▼ Statistical Summary

```
# This function provides summary statistics (count, mean, std, min, 25%, 50%, 75%, max)
# for numerical columns in the DataFrame, allowing us to understand the central tendency,
# spread, and distribution of the data.
descriptive_stats = df.describe()
```

descriptive_stats



	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	

```
# This command provides information about categorical data, including count, unique categories,
# the most frequent category, and its frequency, for each categorical column in the DataFrame.

# The .T attribute transposes the output for better readability.

categorical_stats = df.describe(include='object').T

categorical_stats
```

	count	unique	top	freq	
show_id	8807	8807	s1	1	
type	8807	2	Movie	6131	
title	8807	8804	15-Aug	2	
director	8807	4529	Unknown_Director	2634	
cast	8807	7693	Unknown_Actor	825	
country	8807	749	United States	2818	
rating	8807	18	TV-MA	3207	
duration	8807	221	1 Season	1793	
listed_in	8807	514	Dramas, International Movies	362	
description	8807	8775	Paranormal activity at a lush, abandoned prope...	4	

▼ Data Cleaning

Handling null values a. For categorical variables with null values, update those rows as unknown_column_name. Example : Replace missing value with Unknown Actor for missing value in Actors column. b. Replace with 0 for continuous variables having null values.

```
# Replace NaN values in the 'cast' column with 'Unknown_Actor'
df['cast'].fillna('Unknown_Actor', inplace=True)
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Actor	United States	25-Sep-21	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	24-Sep-21	2021
2	s3	TV	Ganglands	Julien	Sami Bouajila, Tracy Gotoas,	NaN	24-Sep-21	2021

```
# Replace NaN values in the 'director' column with 'Unknown_Director'
df['director'].fillna('Unknown_Director', inplace=True)
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Actor	United States	25-Sep-21	
1	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	24-Sep-21	
2	s3	TV	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas,	NaN	24-Sep-21	

```
# Replace NaN values in the 'country' column with 'Unknown_Country'
df['country'].fillna('Unknown_Country', inplace=True)
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Actor	United States	25-Sep-21	2020	PG-13	90 min	Documentaries
1	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	24-Sep-21	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy,	Unknown_Country	24-Sep-21	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV

```
# Remove leading spaces from the 'date_added' column and replace null values with a default value
default_value = 'No Date' # Replace null values with this value
df['date_added'] = df['date_added'].str.strip().fillna(default_value)
```

```
# Convert the 'date_added' column to datetime
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
df
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Actor	United States	2021-09-25	2020	PG-13	90 min	Documentari
1	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Internation TV Shows, ' Dramas, ' Mysteri
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown_Country	2021-09-24	2021	TV-MA	1 Season	Crime ' Show Internation TV Shows, ' Ac
3	s4	TV Show	Jailbirds New Orleans	Unknown_Director	Unknown_Actor	Unknown_Country	2021-09-24	2021	TV-MA	1 Season	Docuserie Reality '
4	s5	TV Show	Kota Factory	Unknown_Director	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	Internation TV Show Romantic ' Shows, TV
...
					Mark Ruffalo, Jake						Cult Movie
# Replace NaN values in the 'rating' column with a default value default_value = 'Not Rated' # Replace NaN values with this value df['rating'] = df['rating'].fillna(default_value) df											

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Actor	United States	2021-09-25	2020	PG-13	90 min	Documentari
1	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Internation TV Shows, ' Dramas, ' Mysteri
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown_Country	2021-09-24	2021	TV-MA	1 Season	Crime ' Show Internation TV Shows, ' Ac
3	s4	TV Show	Jailbirds New Orleans	Unknown_Director	Unknown_Actor	Unknown_Country	2021-09-24	2021	TV-MA	1 Season	Docuserie Reality '
4	s5	TV Show	Kota Factory	Unknown_Director	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	Internation TV Show Romantic ' Shows, TV
...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey	United States	2019-11-20	2007	R	158 min	Cult Movie Drama Thrille

```
# Replace NaN values in the 'duration' column with a default value
default_value = 'Not Available' # Replace NaN values with this value
df['duration'] = df['duration'].fillna(default_value)
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Actor	United States	2021-09-25	2020	PG-13	90 min	Documentaries
1	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries
		TV			Sami Bouajila, Tracy Cato						Crime TV Shows,

▼ Unnest the columns

a. Un-nest the columns those have cells with multiple comma separated values by creating multiple rows

```
# Step 1: Create a DataFrame 'df_new' from the 'cast' column of DataFrame 'df'
constraint = df['cast'].apply(lambda x: str(x).split(',')).tolist() # Split the 'cast' column by ',' and convert to a list
df_new = pd.DataFrame(constraint, index=df['title']) # Create a DataFrame with the split values, using 'title' as the index

# Step 2: Stack the DataFrame to transform it from wide to long format
df_new = df_new.stack() # Stack the DataFrame, turning columns into a multi-index

# Step 3: Convert the stacked DataFrame to a regular DataFrame and reset the index
df_new = pd.DataFrame(df_new) # Convert the stacked DataFrame to a regular DataFrame
df_new.reset_index(inplace=True) # Reset the index to numeric values

# Step 4: Select and rename columns to have 'title' and 'cast' columns
df_new = df_new[['title', 0]] # Select only the 'title' and the stacked 'cast' column
df_new.columns = ['title', 'cast'] # Rename the columns to 'title' and 'cast'

# Display the final DataFrame 'df_new'
df_new
```

	title	cast
0	Dick Johnson Is Dead	Unknown_Actor
1	Blood & Water	Ama Qamata
2	Blood & Water	Khosi Ngema
3	Blood & Water	Gail Mabalane
4	Blood & Water	Thabang Molaba
...
64946	Zubaan	Manish Chaudhary
64947	Zubaan	Meghna Malik
64948	Zubaan	Malkeet Rauni
64949	Zubaan	Anita Shabdish
64950	Zubaan	Chittaranjan Tripathy

64951 rows × 2 columns

```
# Step 1: Create a DataFrame 'df_new_2' from the 'director' column of DataFrame 'df'
constraint_2 = df['director'].apply(lambda x: str(x).split(',')).tolist() # Split the 'director' column by ',' and convert to a list
df_new_2 = pd.DataFrame(constraint_2, index=df['title']) # Create a DataFrame with the split values, using 'title' as the index

# Step 2: Stack the DataFrame to transform it from wide to long format
df_new_2 = df_new_2.stack() # Stack the DataFrame, turning columns into a multi-index

# Step 3: Convert the stacked DataFrame to a regular DataFrame and reset the index
df_new_2 = pd.DataFrame(df_new_2) # Convert the stacked DataFrame to a regular DataFrame
df_new_2.reset_index(inplace=True) # Reset the index to numeric values

# Step 4: Select and rename columns to have 'title' and 'director' columns
df_new_2 = df_new_2[['title', 0]] # Select only the 'title' and the stacked 'director' column
df_new_2.columns = ['title', 'director'] # Rename the columns to 'title' and 'director'
```

```
# Display the final DataFrame 'df_new_2'
df_new_2
```

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	Unknown_Director
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	Unknown_Director
4	Kota Factory	Unknown_Director
...
9607	Zodiac	David Fincher
9608	Zombie Dumb	Unknown_Director
9609	Zombieland	Ruben Fleischer
9610	Zoom	Peter Hewitt
9611	Zubaan	Mozez Singh

9612 rows × 2 columns

```
# Step 1: Create a DataFrame 'df_new_3' from the 'country' column of DataFrame 'df'
constraint_3 = df['country'].apply(lambda x: str(x).split(',')).tolist() # Split the 'country' column by ', ' and convert to a list
df_new_3 = pd.DataFrame(constraint_3, index=df['title']) # Create a DataFrame with the split values, using 'title' as the index
```

```
# Step 2: Stack the DataFrame to transform it from wide to long format
df_new_3 = df_new_3.stack() # Stack the DataFrame, turning columns into a multi-index
```

```
# Step 3: Convert the stacked DataFrame to a regular DataFrame and reset the index
df_new_3 = pd.DataFrame(df_new_3) # Convert the stacked DataFrame to a regular DataFrame
df_new_3.reset_index(inplace=True) # Reset the index to numeric values
```

```
# Step 4: Select and rename columns to have 'title' and 'country' columns
df_new_3 = df_new_3[['title', 0]] # Select only the 'title' and the stacked 'country' column
df_new_3.columns = ['title', 'country'] # Rename the columns to 'title' and 'country'
```

```
# Display the final DataFrame 'df_new_3'
df_new_3
```

	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	Unknown_Country
3	Jailbirds New Orleans	Unknown_Country
4	Kota Factory	India
...
10840	Zodiac	United States
10841	Zombie Dumb	Unknown_Country
10842	Zombieland	United States
10843	Zoom	United States
10844	Zubaan	India

10845 rows × 2 columns

```
# Step 1: Create a DataFrame 'df_new_4' from the 'listed_in' column of DataFrame 'df'
constraint_4 = df['listed_in'].apply(lambda x: str(x).split(',')).tolist() # Split the 'listed_in' column by ', ' and convert to a list
df_new_4 = pd.DataFrame(constraint_4, index=df['title']) # Create a DataFrame with the split values, using 'title' as the index
```

```
# Step 2: Stack the DataFrame to transform it from wide to long format
df_new_4 = df_new_4.stack() # Stack the DataFrame, turning columns into a multi-index
```

```
# Step 3: Convert the stacked DataFrame to a regular DataFrame and reset the index
df_new_4 = pd.DataFrame(df_new_4) # Convert the stacked DataFrame to a regular DataFrame
```

```
df_new_4.reset_index(inplace=True) # Reset the index to numeric values

# Step 4: Select and rename columns to have 'title' and 'listed_in' columns
df_new_4 = df_new_4[['title', 0]] # Select only the 'title' and the stacked 'listed_in' column
df_new_4.columns = ['title', 'listed_in'] # Rename the columns to 'title' and 'listed_in'

# Display the final DataFrame 'df_new_4'
df_new_4
```

	title	listed_in
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows
2	Blood & Water	TV Dramas
3	Blood & Water	TV Mysteries
4	Ganglands	Crime TV Shows
...
19318	Zoom	Children & Family Movies
19319	Zoom	Comedies
19320	Zubaan	Dramas
19321	Zubaan	International Movies
19322	Zubaan	Music & Musicals

19323 rows × 2 columns

▼ Merge nested columns

```
# Step 1: Merge df_new_4 with df_new_3 based on the 'title' column
merge_df_3_4 = df_new_3.merge(df_new_4, on='title')

# Step 2: Merge df_new_2 with the result of step 1 based on the 'title' column
merge_df_2_3_4 = df_new_2.merge(merge_df_3_4, on='title')

# Step 3: Merge df_new with the result of step 2 based on the 'title' column
merge1 = df_new.merge(merge_df_2_3_4, on='title')

# Step 4: Merge the original DataFrame df with the result of step 3 based on the 'title' column
data = df.merge(merge1, on='title')

# Display the final merged DataFrame 'data'
data
```


	show_id	type	title	director_x	cast_x	country_x	date_added	release_year	rating	duration	listed_in_x	des
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown_Actor	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As lil
1	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata, Khosi Ngema, Gail Mabalan... Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	par
2	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata, Khosi Ngema, Gail Mabalan... Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	par
3	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata, Khosi Ngema, Gail Mabalan... Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	par
4	s2	TV Show	Blood & Water	Unknown_Director	Ama Qamata, Khosi Ngema, Gail Mabalan... Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	par
...
204566	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	2019-03-02	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	bu

▼ Dropping the duplicate columns

Chanan

& Musicals

Approach 1

Vicky Kaushal,

Dramas, ...

```
# Drop one or more columns by specifying their names in a list
data.drop(columns=['director_x','cast_x','country_x','listed_in_x'], inplace=True)
```

```
data.head()
```

	show_id	type	title	date_added	release_year	rating	duration	description	cast_y	director_y	country_y	listed_i
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90 min	As her father nears the end of his life, filmm...	Unknown_Actor	Kirsten Johnson	United States	Documenta
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Ama Qamata	Unknown_Director	South Africa	Internatic TV Sh
2	s2	TV	Blood &	2021-09-24	2021	TV-MA	2	After crossing paths at a	Ama Qamata	Unknown_Director	South	TV Dra

▼ Rename the columns

```
rename_column = {
    'cast_y': 'cast',
    'director_y': 'director',
    'country_y': 'country',
    'listed_in_y': 'listed_in'
}
```

```
# Use the rename() method to rename multiple columns
data.rename(columns=rename_column, inplace=True)
```

```
data.head()
```

	show_id	type	title	date_added	release_year	rating	duration	description	cast	director	country	listed_in
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90 min	As her father nears the end of his life, filmm...	Unknown_Actor	Kirsten Johnson	United States	Documentarie
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Ama Qamata	Unknown_Director	South Africa	Internation: TV Show
2	s2	TV	Blood &	2021-09-24	2021	TV-MA	2	After crossing paths at a	Ama Qamata	Unknown_Director	South	TV Drama

```
data['release_year'] = data['release_year'].astype(str)
data.head()
```

	show_id	type	title	date_added	release_year	rating	duration	description	cast	director	country	listed_in
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90 min	As her father nears the end of his life, filmm...	Unknown_Actor	Kirsten Johnson	United States	Documentarie
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...	Ama Qamata	Unknown_Director	South Africa	Internation: TV Show
2	s2	TV	Blood &	2021-09-24	2021	TV-MA	2	After crossing paths at a	Ama Qamata	Unknown_Director	South	TV Drama

```
data['date_added'] = pd.to_datetime(data['date_added'])
data.head(2)
```

	show_id	type	title	date_added	release_year	rating	duration	description	cast	director	country	listed_in
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90 min	As her father nears the end of his life, filmm...	Unknown_Actor	Kirsten Johnson	United States	Documentarie

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 204571 entries, 0 to 204570
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  -
0   show_id         204571 non-null object
1   type            204571 non-null object
2   title           204571 non-null object
3   date_added      204413 non-null datetime64[ns]
4   release_year    204571 non-null object
5   rating          204571 non-null object
6   duration         204571 non-null object
7   description      204571 non-null object
8   cast            204571 non-null object
9   director        204571 non-null object
10  country          204571 non-null object
11  listed_in       204571 non-null object
dtypes: datetime64[ns](1), object(11)
memory usage: 20.3+ MB
```

1 : Find the counts of each categorical variable both using graphical and non-graphical analysis.

Non graphical analysis

```
# Get the count of each value in the 'show_id' column
show_id_counts = data['show_id'].value_counts()
show_id_counts.head(10)
```

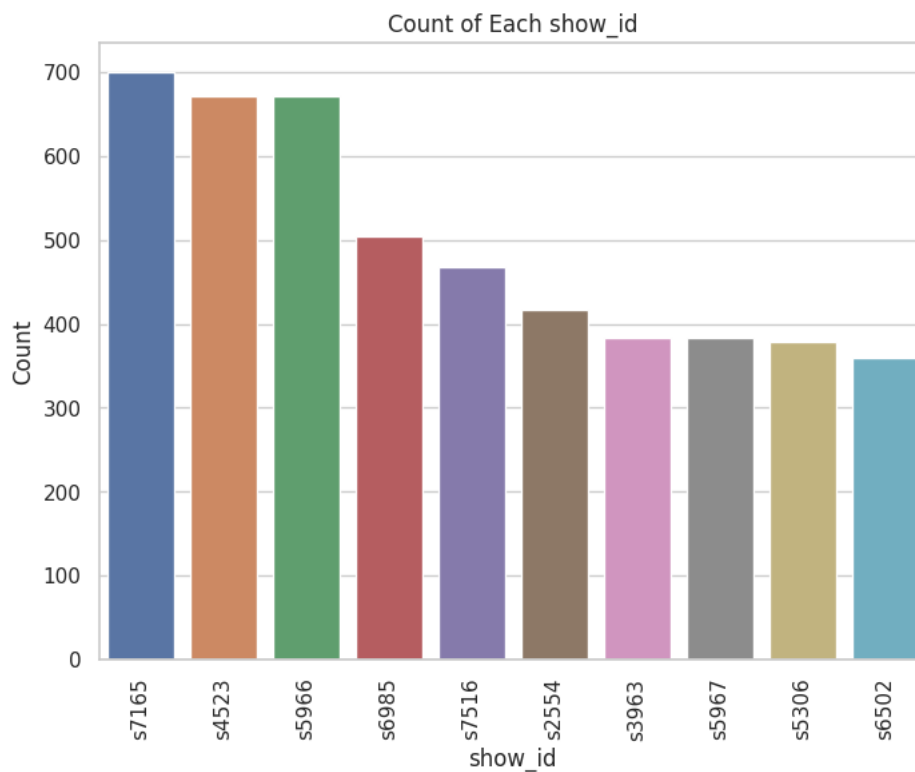
```
s7165    700
s4523    672
s5966    672
s6985    504
s7516    468
s2554    416
s3963    384
s5967    384
s5306    378
s6502    360
Name: show_id, dtype: int64
```

Graphical Analysis

```
# Create a count plot

# Set the style with gridlines
sns.set(style="whitegrid")

plt.figure(figsize=(8, 6))
sns.countplot(data=data, x='show_id', order = show_id_counts.head(10).index)
plt.xticks(rotation=90)
plt.xlabel('show_id')
plt.ylabel('Count')
plt.title('Count of Each show_id')
plt.show()
```



Non graphical analysis

```
# Calculate the counts of each type
type_counts = data['type'].value_counts(normalize=True)*100
type_counts
```

```
Movie      72.259998
TV Show    27.740002
Name: type, dtype: float64
```

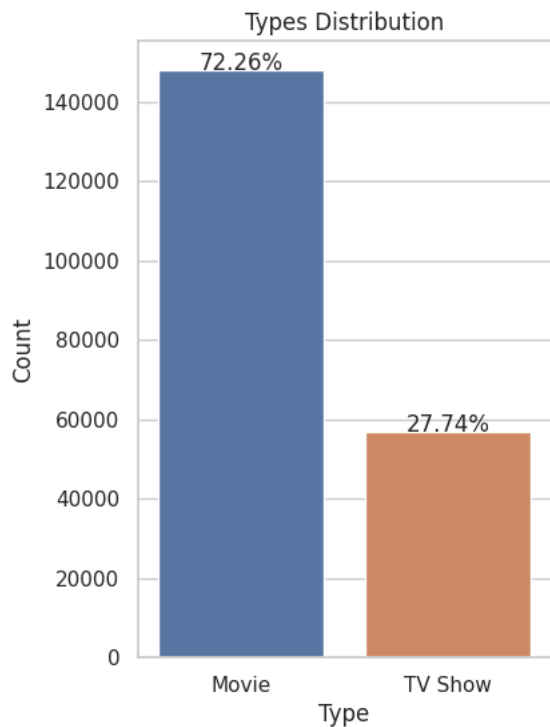
Graphical analysis

Approach 1 countplot

```
plt.figure(figsize=(4, 6)) # Adjust the figure size if needed
ax = sns.countplot(data=data, x='type')
plt.title('Types Distribution')
plt.xlabel('Type')
plt.ylabel('Count')

# Annotate the bars with percentages
total_count = len(data)
for p in ax.patches:
    percentage = '{:.2f}%'.format(100 * p.get_height() / total_count)
    x = p.get_x() + p.get_width() / 2
    y = p.get_height()
    ax.annotate(percentage, (x, y), ha='center')

# Show the count plot
plt.show()
```

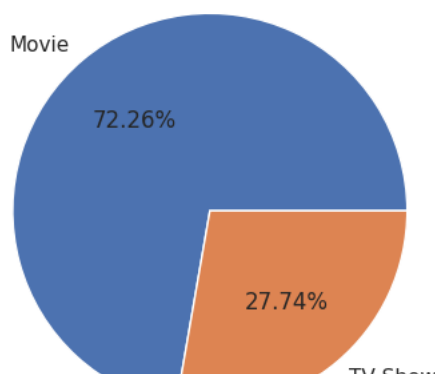


approach 2 pie chart

```
# Create a pie chart
plt.pie(type_counts, labels=type_counts.index, autopct="%.2f%%")
plt.title('Types Distribution')

# Show the pie chart
plt.show()
```

Types Distribution



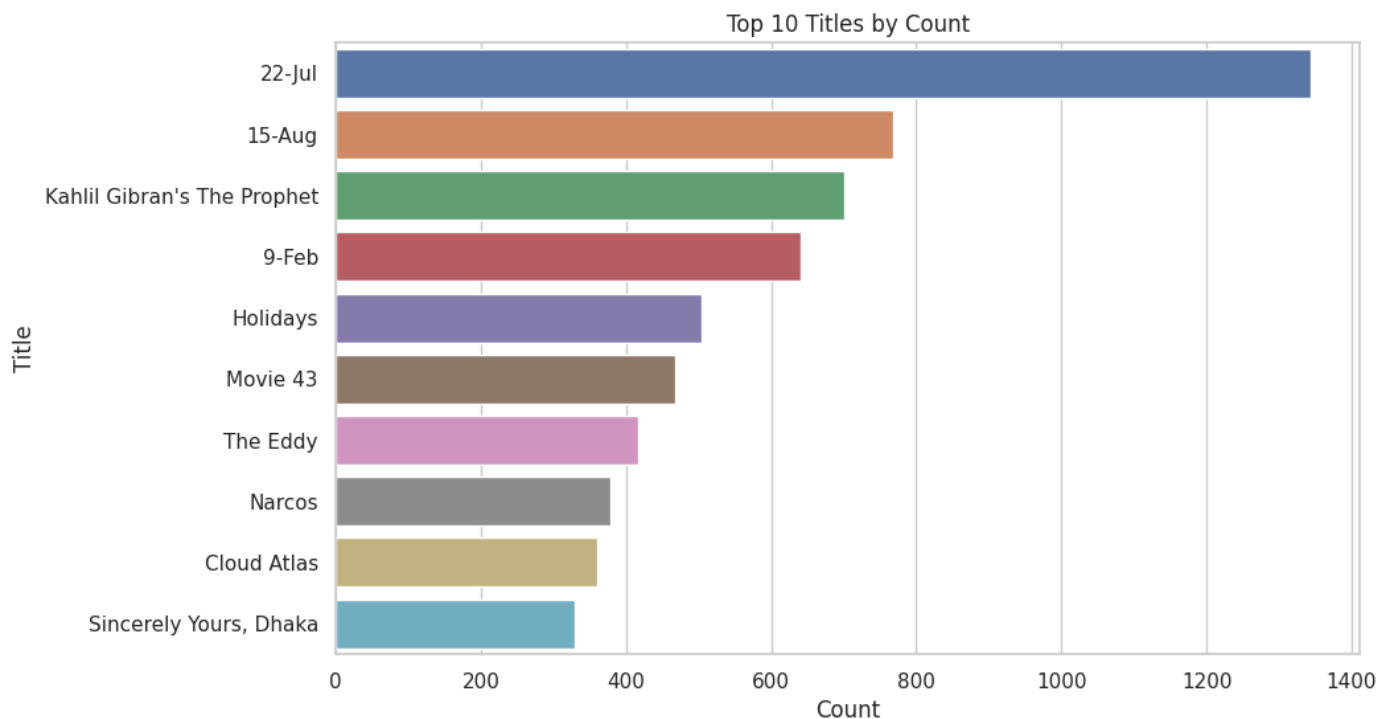
```
# Get the count of each value in the 'title' column
title_counts = data['title'].value_counts()
title_counts.head(10)
```

```
22-Jul      1344
15-Aug      768
Kahlil Gibran's The Prophet  700
9-Feb       640
Holidays    504
Movie 43     468
The Eddy     416
Narcos       378
Cloud Atlas  360
Sincerely Yours, Dhaka  330
Name: title, dtype: int64
```

```
# Create a count plot for the 'title' column
```

```
# Set the style with gridlines
sns.set(style="whitegrid")
```

```
plt.figure(figsize=(10, 6))
sns.countplot(data=data, y='title', order=title_counts.index[:10]) # Adjust the number of displayed items as needed
plt.xlabel('Count')
plt.ylabel('Title')
plt.title('Top 10 Titles by Count')
plt.show()
```



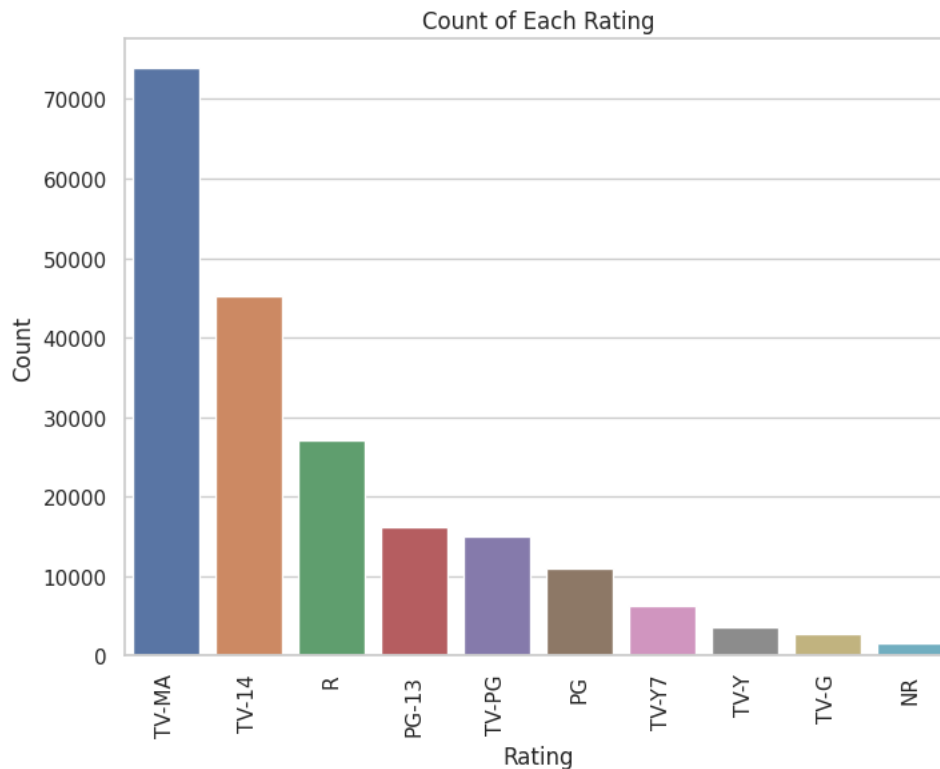
```
# Get the count of each value in the 'rating' column
rating_counts = data['rating'].value_counts()
rating_counts.head(10)
```

```
TV-MA    73867
TV-14    45251
R         27120
PG-13    16246
TV-PG    14926
PG        10919
TV-Y7     6304
TV-Y      3665
TV-G      2779
NR        1573
Name: rating, dtype: int64
```

```
# Create a count plot for the 'rating' column
```

```
# Set the style with gridlines
sns.set(style="whitegrid")
```

```
plt.figure(figsize=(8, 6))
sns.countplot(data=data, x='rating', order=rating_counts.index[:10])
plt.xticks(rotation=90)
plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Count of Each Rating')
plt.show()
```

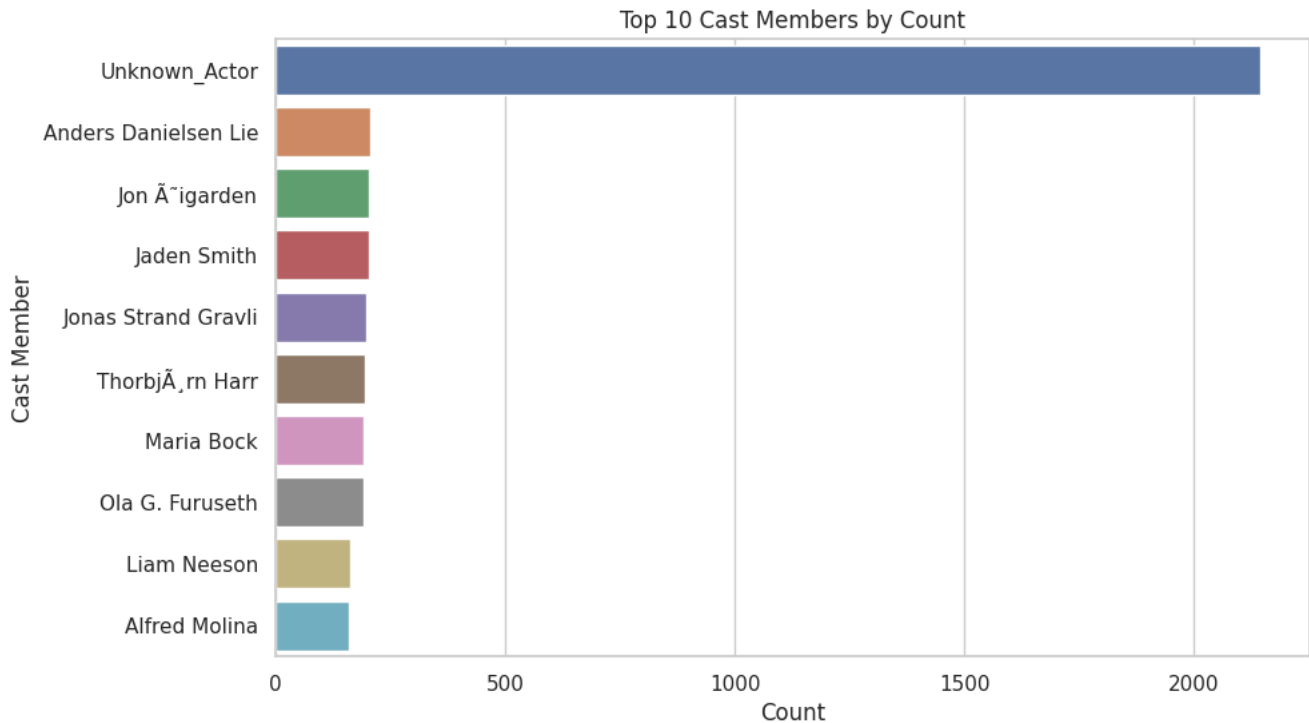


```
# Get the count of each value in the 'cast' column
cast_counts = data['cast'].value_counts()
cast_counts
```

```
Unknown Actor    2146
Anders Danielsen Lie    207
Jon Åigarden    203
Jaden Smith    202
Jonas Strand Gravli    198
...
Dario Yazbek    1
Corinne Foxx    1
Jacob Craner    1
Laila Berzins    1
Richard Ryan    1
Name: cast, Length: 36440, dtype: int64
```

```
sns.set(style="whitegrid")

plt.figure(figsize=(10, 6))
sns.countplot(data=data, y='cast', order=cast_counts.index[:10]) # Adjust the number of displayed items as needed
plt.xlabel('Count')
plt.ylabel('Cast Member')
plt.title('Top 10 Cast Members by Count')
plt.show()
```

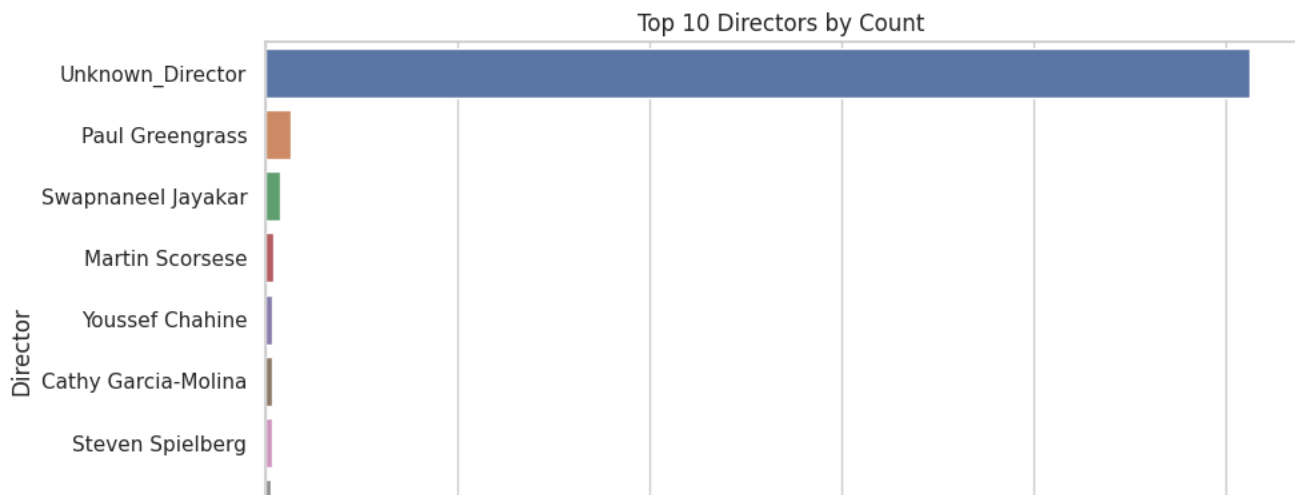


```
# Get the count of each value in the 'director_y' column
director_counts = data['director'].value_counts()
director_counts.head(10)
```

```
Unknown_Director    51243
Paul Greengrass      1384
Swapnaneel Jayakar   768
Martin Scorsese      419
Youssef Chahine      409
Cathy Garcia-Molina  356
Steven Spielberg     355
Lars von Trier       336
Raja Gosnell         308
Tom Hooper           306
Name: director, dtype: int64
```

```
# Set the style with gridlines
sns.set(style="whitegrid")
```

```
# Create a count plot for the 'director' column
plt.figure(figsize=(10, 6))
sns.countplot(data=data, y='director', order=director_counts.index[:10]) # Adjust the number of displayed items as needed
plt.xlabel('Count')
plt.ylabel('Director')
plt.title('Top 10 Directors by Count')
plt.show()
```

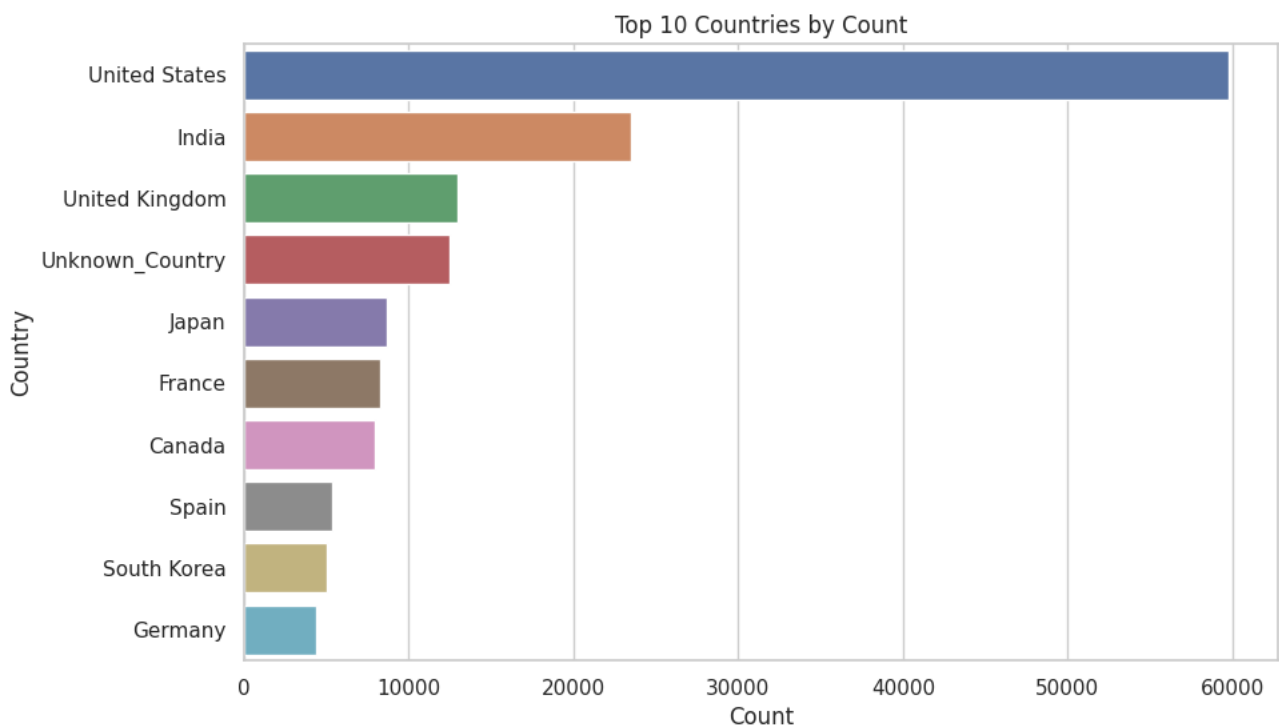


```
# Get the count of each value in the 'country' column
country_counts = data['country'].value_counts()
country_counts.head(10)
```

```
United States    59769
India            23534
United Kingdom   12945
Unknown_Country  12497
Japan            8679
France           8254
Canada           7915
Spain            5315
South Korea      5043
Germany          4383
Name: country, dtype: int64
```

```
# Set the style with gridlines
sns.set(style="whitegrid")
```

```
# Create a count plot for the 'country' column
plt.figure(figsize=(10, 6))
sns.countplot(data=data, y='country', order=country_counts.index[:10]) # Adjust the number of displayed items as needed
plt.xlabel('Count')
plt.ylabel('Country')
plt.title('Top 10 Countries by Count')
plt.show()
```



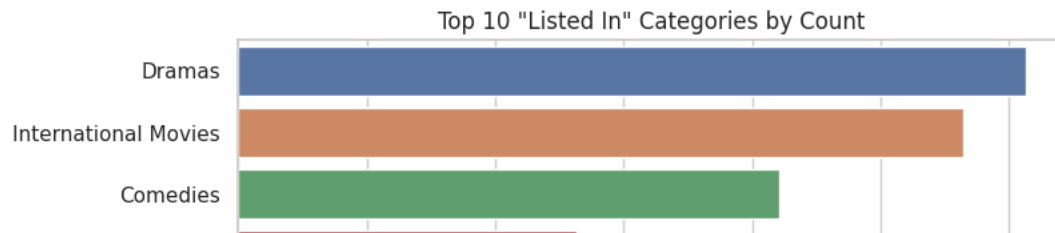

```
# Get the count of each value in the 'listed_in' column
listed_in_counts = data['listed_in'].value_counts()
listed_in_counts
```

Dramas	30645
International Movies	28211
Comedies	21069
International TV Shows	13145
Action & Adventure	12216
Independent Movies	10074
Children & Family Movies	9771
TV Dramas	9242
Thrillers	7737
Romantic Movies	6412
TV Comedies	4963
Crime TV Shows	4733
Horror Movies	4571
Kids' TV	4568
Sci-Fi & Fantasy	4037
Music & Musicals	3077
Romantic TV Shows	3049
Documentaries	2407
Anime Series	2313
TV Action & Adventure	2288
Spanish-Language TV Shows	2126
British TV Shows	1808
Sports Movies	1531
Classic Movies	1434
TV Mysteries	1281
Korean TV Shows	1122
Cult Movies	1077
TV Sci-Fi & Fantasy	1045
Anime Features	1045
TV Horror	941
Docuseries	845
LGBTQ Movies	838
TV Thrillers	768
Teen TV Shows	742
Reality TV	735
Faith & Spirituality	719
Stand-Up Comedy	540
Movies	412
TV Shows	337
Classic & Cult TV	272
Stand-Up Comedy & Talk Shows	268
Science & Nature TV	157

Name: listed_in, dtype: int64

```
# Set the style with gridlines
sns.set(style="whitegrid")
```

```
# Create a count plot for the 'listed_in' column
plt.figure(figsize=(8, 6))
sns.countplot(data=data, y='listed_in', order=listed_in_counts.index[:10]) # Adjust the number of displayed items as needed
plt.xlabel('Count')
plt.ylabel('Listed In')
plt.title('Top 10 "Listed In" Categories by Count')
plt.show()
```



2. Comparison of tv shows vs. movies.

a. Find the number of movies produced in each country and pick the top 10 countries.

```
# Filter the DataFrame to include only movies
movies_data = data[data['type'] == 'Movie']

# Group by country and count the number of unique titles of movies for each country
country_movie_counts = movies_data.groupby('country')['title'].nunique().reset_index()

# Rename the columns for clarity
country_movie_counts.columns = ['Country', 'Number_of_Movies_Produced']

# Sort the DataFrame by the number of movies produced in descending order
country_movie_counts = country_movie_counts.sort_values(by='Number_of_Movies_Produced', ascending=False)

# Select the top 10 countries
top_10_countries = country_movie_counts.head(10)

top_10_countries
```

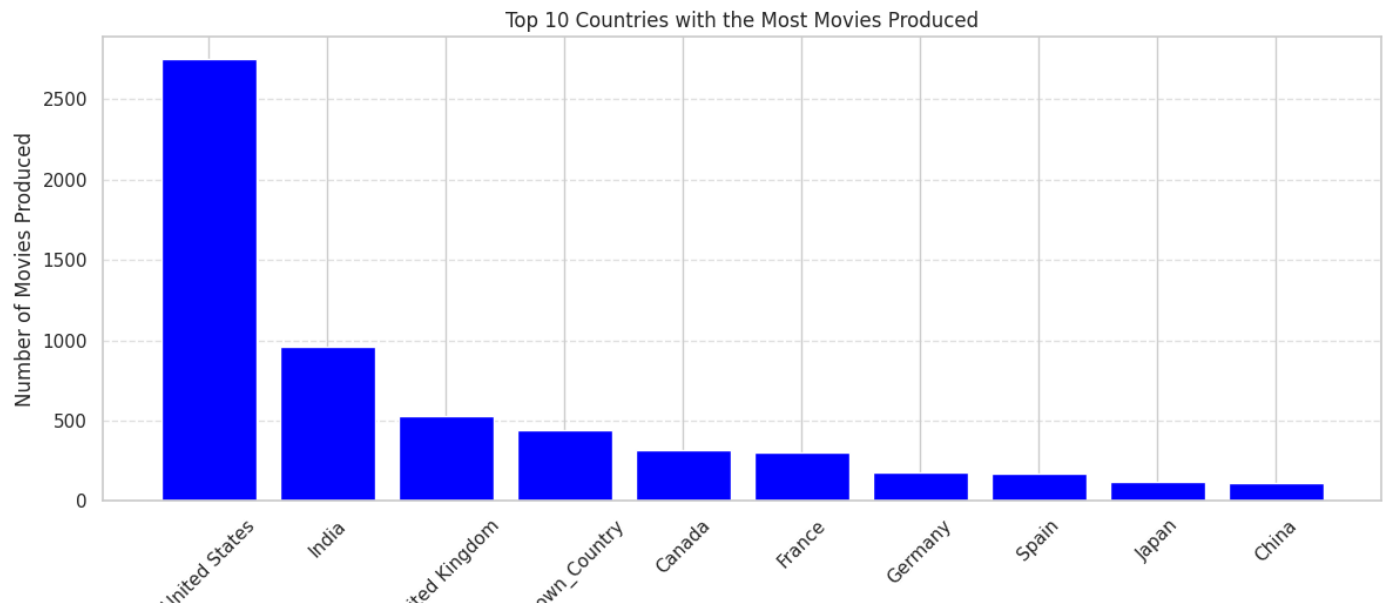
	Country	Number_of_Movies_Produced
114	United States	2750
43	India	961
112	United Kingdom	532
116	Unknown_Country	440
20	Canada	319
34	France	303
36	Germany	182
100	Spain	171
51	Japan	119
23	China	114

```
# Create a bar plot for the top 10 countries
plt.figure(figsize=(12, 6))
plt.bar(top_10_countries['Country'], top_10_countries['Number_of_Movies_Produced'], color='blue')
plt.xlabel('Country')
plt.ylabel('Number of Movies Produced')
plt.title('Top 10 Countries with the Most Movies Produced')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Add gridlines
plt.grid(axis='y', linestyle='--', alpha=0.6)

# Show the plot
plt.tight_layout()
plt.show()
```



- ▼ b. Find the number of Tv-Shows produced in each country and pick the top 10 countries.

```
# Filter the DataFrame to include only TV shows
tv_shows_data = data[data['type'] == 'TV Show']

# Group by country and count the number of unique titles of TV shows for each country
country_tv_show_counts = tv_shows_data.groupby('country')['title'].nunique().reset_index()

# Rename the columns for clarity
country_tv_show_counts.columns = ['Country', 'Number_of_TV_Shows_Produced']

# Sort the DataFrame by the number of TV shows produced in descending order
country_tv_show_counts = country_tv_show_counts.sort_values(by='Number_of_TV_Shows_Produced', ascending=False)

# Select the top 10 countries
top_10_countries = country_tv_show_counts.head(10)

top_10_countries
```

	Country	Number_of_TV_Shows_Produced	
63	United States	938	
64	Unknown_Country	390	
62	United Kingdom	272	
30	Japan	199	
52	South Korea	170	
8	Canada	126	
19	France	90	
25	India	84	
57	Taiwan	70	
2	Australia	66	

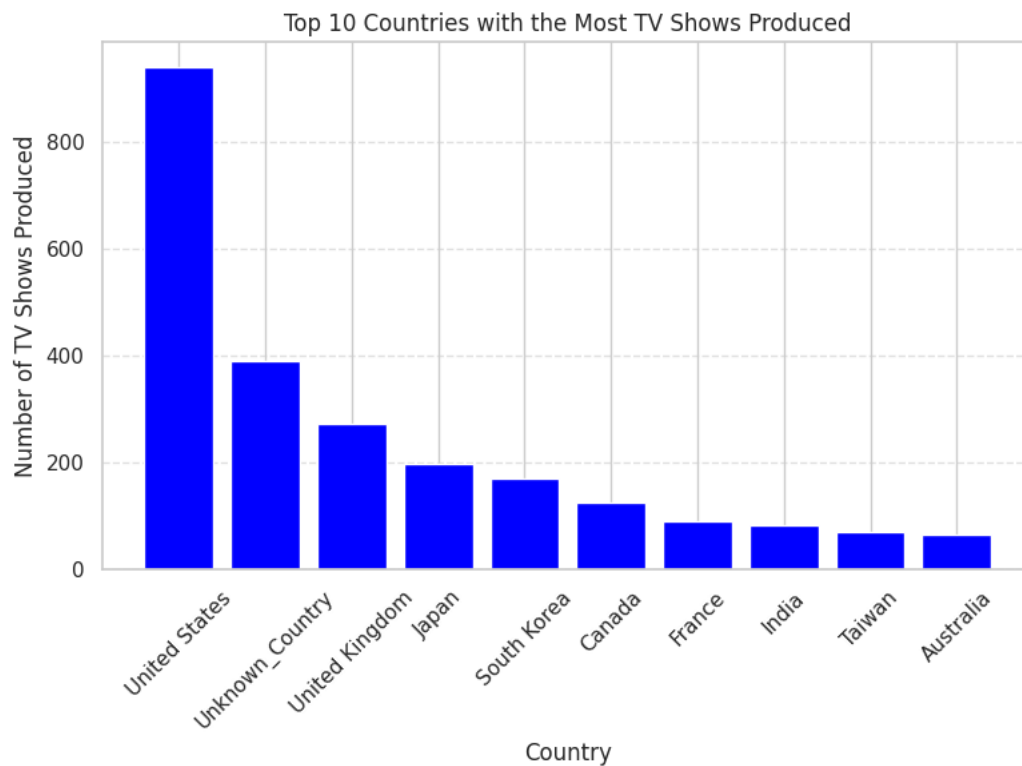
```
# Create a bar plot for the top 10 countries
plt.figure(figsize=(8, 6))
plt.bar(top_10_countries['Country'], top_10_countries['Number_of_TV_Shows_Produced'], color='blue')
plt.xlabel('Country')
plt.ylabel('Number of TV Shows Produced')
plt.title('Top 10 Countries with the Most TV Shows Produced')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Add gridlines
```

```
plt.grid(axis='y', linestyle='--', alpha=0.6)
```

```
# Show the plot
plt.tight_layout()
plt.show()
```



▼ 3. What is the best time to launch a TV show or the Movie ?

a. Find which is the best week to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies

find which is the best week to release the tv-show

```
# Create a new column for week of release
data['week_of_release'] = data['date_added'].dt.strftime('%U-%Y')

# Separate data into TV shows
tv_shows_data = data[data['type'] == 'TV Show']

# Group by week of release and count the number of TV shows and movies
tv_show_release_counts = tv_shows_data.groupby('week_of_release').size().reset_index(name='Count')

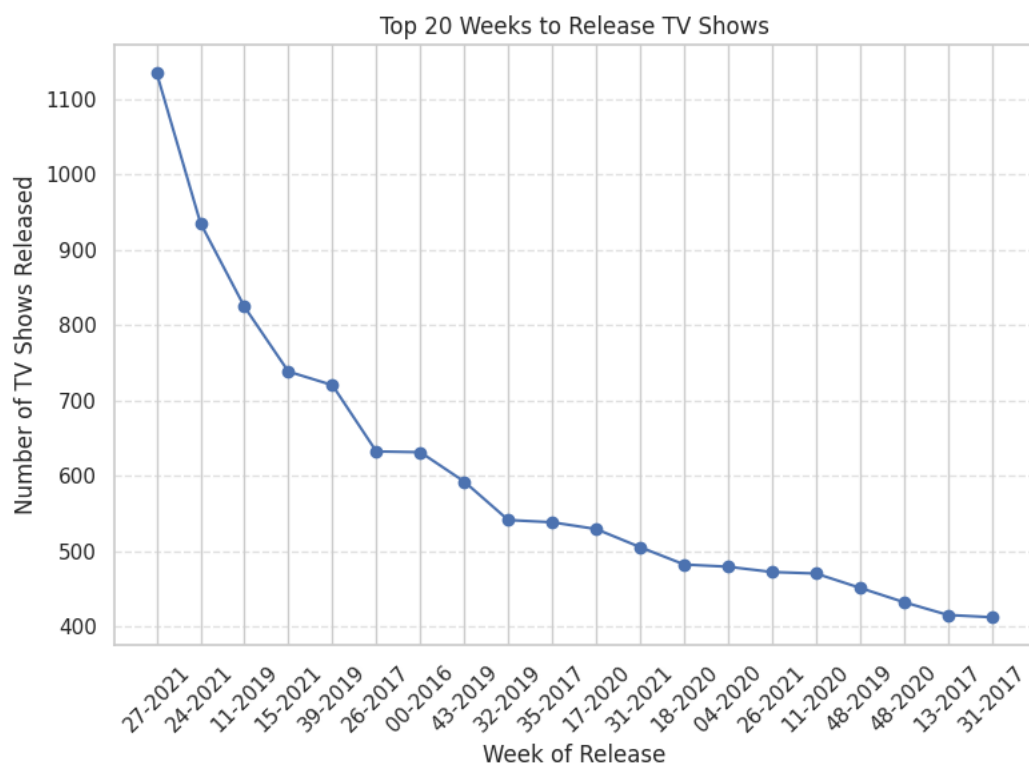
# Sort by count in descending order for TV shows
tv_show_release_counts = tv_show_release_counts.sort_values(by='Count', ascending=False)

tv_show_release_counts.head(20)
```

	week_of_release	Count
169	27-2021	1135
150	24-2021	935
70	11-2019	824
98	15-2021	738
242	39-2019	720
159	26-2017	632
0	00-2016	631
265	43-2019	592
197	32-2019	541
214	35-2017	538
110	17-2020	529

```
# Select the top 20 weeks for TV shows
top_20_tv_show_weeks = tv_show_release_counts.head(20)

# Create a line plot for top 20 TV show release weeks
plt.figure(figsize=(8, 6))
plt.plot(top_20_tv_show_weeks['week_of_release'], top_20_tv_show_weeks['Count'], marker='o', linestyle='--')
plt.title('Top 20 Weeks to Release TV Shows')
plt.xlabel('Week of Release')
plt.ylabel('Number of TV Shows Released')
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



From the data showing the number of TV shows added to Netflix for different weeks and years, here are some insights and comments:

1. Week 27-2021 (1135 TV Shows): This week had the highest number of TV show additions to Netflix. It's possible that Netflix was making a concerted effort to provide fresh content during this period, which falls in the mid-year.



- 2. Week 24-2021 (935 TV Shows): Week 24 of 2021 follows closely with a substantial number of TV show additions. This may have been part of Netflix's strategy to cater to viewers during the early summer months.
- 3. Week 15-2021 (738 TV Shows): Week 15 of 2021 recorded a notable number of TV show additions. It's likely that Netflix was responding to viewer demand as the year progressed.
- 4. Week 39-2019 (720 TV Shows): In Week 39 of 2019, Netflix added a considerable number of TV shows, possibly to enhance its library before the holiday season.
- 5. Week 26-2017 (632 TV Shows): Week 26 of 2017 had a high count of TV show additions, indicating Netflix's ongoing efforts to expand its content offering.
- 6. Week 00-2016 (631 TV Shows): The beginning of 2016 saw a substantial number of TV show additions, potentially aiming to attract new subscribers at the start of the year.
- 7. Week 43-2019 (592 TV Shows): Week 43 of 2019 featured a notable number of TV show additions, suggesting that Netflix continued to invest in content throughout the year.
- 8. Week 32-2019 (541 TV Shows): Week 32 of 2019 recorded a significant number of TV show additions, indicating a consistent approach to content distribution.
- 9. Week 35-2017 (538 TV Shows): Week 35 of 2017 also had a high count of TV show additions, aligning with Netflix's strategy to keep viewers engaged.
- 10. Week 17-2020 (529 TV Shows): In Week 17 of 2020, Netflix added a substantial number of TV shows, potentially targeting viewers during the early spring period.

```
# Separate data into movies
movies_data = data[data['type'] == 'Movie']

# Group by week of release and count the number of movies
movie_release_counts = movies_data.groupby('week_of_release').size().reset_index(name='Count')

# Sort by count in descending order for movies
movie_release_counts = movie_release_counts.sort_values(by='Count', ascending=False)

movie_release_counts.head(20)
```

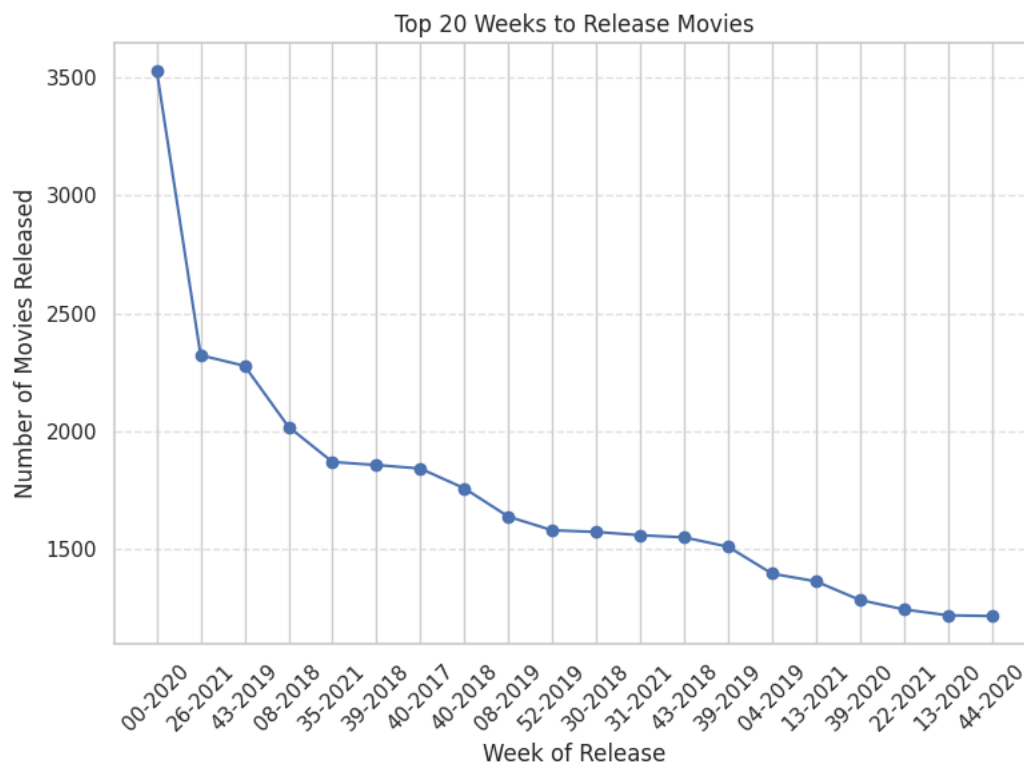
	week_of_release	Count	
4	00-2020	3531	
180	26-2021	2322	
299	43-2019	2277	
57	08-2018	2018	
242	35-2021	1869	
269	39-2018	1856	
275	40-2017	1841	
276	40-2018	1757	
58	08-2019	1637	
362	52-2019	1579	
204	30-2018	1572	
214	31-2021	1558	
298	43-2018	1549	
270	39-2019	1509	
30	04-2019	1395	
92	13-2021	1362	
271	39-2020	1283	
154	22-2021	1243	
91	13-2020	1218	
307	44-2020	1215	

```
# Select the top 20 weeks for movie releases
top_20_movie_weeks = movie_release_counts.head(20)

# Create a line plot for top 20 movie release weeks
plt.figure(figsize=(8, 6))
plt.plot(top_20_movie_weeks['week_of_release'], top_20_movie_weeks['Count'], marker='o', linestyle='--')
plt.title('Top 20 Weeks to Release Movies')
plt.xlabel('Week of Release')
plt.ylabel('Number of Movies Released')
plt.grid(axis='y', linestyle='--', alpha=0.6)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```



From the data showing the number of movies added to Netflix for different weeks and years, here are some insights and comments:

1. Week 00-2020 (3531 Movies): This week, at the beginning of 2020, had the highest number of movie additions to Netflix. It's likely that Netflix aimed to start the year with a substantial collection of movies to attract viewers.
2. Week 26-2021 (2322 Movies): In Week 26 of 2021, there was a significant number of movie additions, possibly aligned with summer releases and the expectation of higher viewership during this season.
3. Week 43-2019 (2277 Movies): Week 43 of 2019 recorded a notable number of movie additions, possibly as part of Netflix's strategy to provide diverse content throughout the year.
4. Week 08-2018 (2018 Movies): In Week 08 of 2018, there were a substantial number of movie additions, indicating a consistent effort by Netflix to expand its movie library.
5. Week 35-2021 (1869 Movies): Week 35 of 2021 featured a significant number of movie additions, likely catering to viewers' preferences during late summer and early fall.
6. Week 39-2018 (1856 Movies): In Week 39 of 2018, there was a high count of movie additions, possibly coinciding with the back-to-school season.
7. Week 40-2017 (1841 Movies): Week 40 of 2017 recorded a substantial number of movie additions, indicating a focus on content diversity and viewer engagement.

8. Week 08-2019 (1637 Movies): In Week 08 of 2019, Netflix added a significant number of movies, potentially targeting viewers during the early spring period.
9. Week 52-2019 (1579 Movies): In Week 52 of 2019, there was a notable number of movie additions, possibly aligned with the holiday season.
10. Week 30-2018 (1572 Movies): Week 30 of 2018 featured a high count of movie additions, indicating Netflix's consistent approach to content distribution.

b. Find which is the best month to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies

TV-show Analysis

```
# Create a new column for month of release
data['month_of_release'] = data['date_added'].dt.strftime('%B-%Y')

# Separate data into TV shows
tv_shows_data = data[data['type'] == 'TV Show']

# Group by month of release and count the number of TV shows
tv_show_release_counts = tv_shows_data.groupby('month_of_release').size().reset_index(name='Count')

# Sort by count in descending order for TV shows
tv_show_release_counts = tv_show_release_counts.sort_values(by='Count', ascending=False)

tv_show_release_counts.head(20)
```

	month_of_release	Count
43	July-2021	2210
50	June-2021	1921
71	November-2019	1708
56	March-2019	1681
21	December-2020	1520
64	May-2020	1416
78	October-2019	1348
14	August-2021	1297
12	August-2019	1292
79	October-2020	1254
29	February-2020	1238
87	September-2021	1161
30	February-2021	1158
35	January-2020	1139
72	November-2020	1125
17	December-2016	1113
57	March-2020	1108
7	April-2021	1080
6	April-2020	1067
48	June-2019	1052

```
# Select the best month for TV show releases
best_month_tv_show = tv_show_release_counts.head(20)

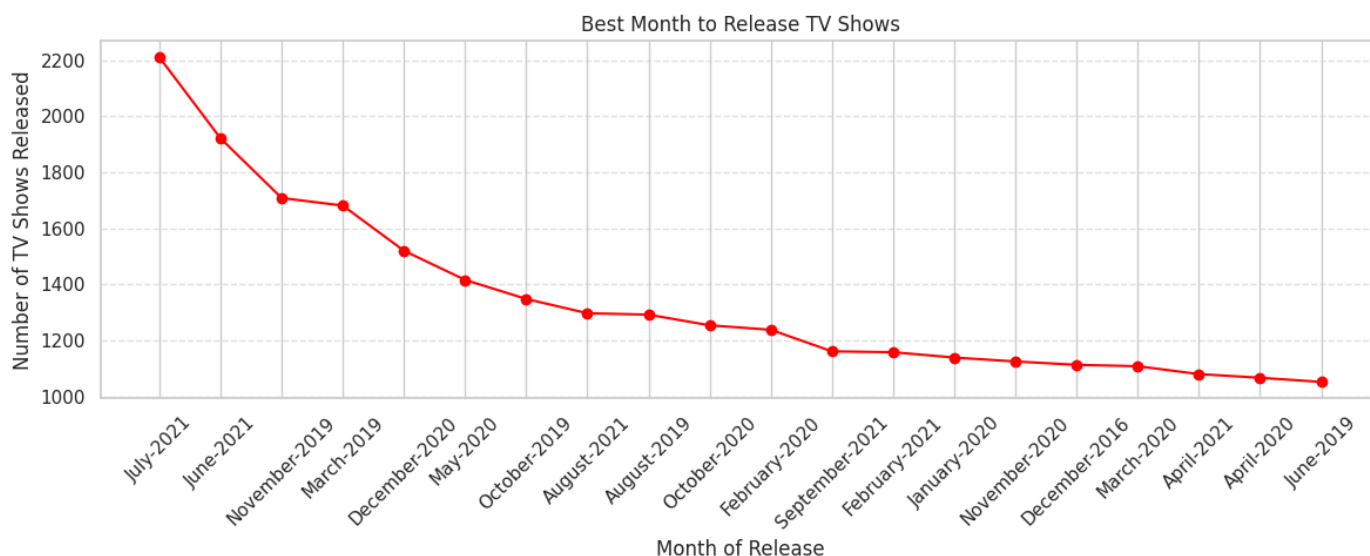
# Create a line plot for the best month to release TV shows
plt.figure(figsize=(12, 5))
```



```
plt.plot(best_month_tv_show['month_of_release'], best_month_tv_show['Count'], color = 'red', marker='o', linestyle='--')
plt.title('Best Month to Release TV Shows')
plt.xlabel('Month of Release')
plt.ylabel('Number of TV Shows Released')
plt.grid(axis='y', linestyle='--', alpha=0.6)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```



The data provided represents the count of TV shows added to Netflix for various months and years. Here are some insights and comments based on this data:

1. July-2021 (2210 TV Shows): July 2021 had the highest number of TV shows added to Netflix. This suggests that Netflix strategically planned a significant TV show addition during the summer season, likely to attract a larger viewership.
2. June-2021 (1921 TV Shows): June 2021 closely follows as one of the months with a substantial number of TV show additions. This aligns with the approach of providing fresh content during the peak of the summer season.
3. November-2019 (1708 TV Shows): November 2019 recorded a notable number of TV show additions. This could be linked to holiday season planning, as many people tend to watch more TV shows during the holiday period.
4. December-2020 (1520 TV Shows): December 2020 shows a significant number of TV show additions, indicating that Netflix recognizes the holiday season as an opportunity to launch new content, catering to viewers' increased leisure time.
5. May-2020 (1416 TV Shows): May 2020 had a substantial number of TV show additions. This might be connected to viewership trends during the spring season when people spend more time indoors.
6. October-2019 (1348 TV Shows): October 2019 reflects a high count of TV show additions, suggesting Netflix's commitment to expanding its content library and keeping viewers engaged.
7. August-2021 (1297 TV Shows): August 2021 also had a notable number of TV show additions, possibly targeting late summer audiences with new content offerings.
8. August-2019 (1292 TV Shows): August 2019 recorded a high count of TV show additions, similar to August 2021. This suggests that August might be strategically important for content launches.
9. October-2020 (1254 TV Shows): October 2020 had a considerable number of TV show additions, likely related to the fall season and content themed around Halloween.
10. February-2020 (1238 TV Shows): February 2020 witnessed a substantial number of TV show additions. This might be associated with Valentine's Day-themed content releases to cater to a romantic audience.

Movie Analysis

```
# Create a new column for month of release
data['month_of_release'] = data['date_added'].dt.strftime('%B-%Y')

# Separate data into movies
movies_data = data[data['type'] == 'Movie']

# Group by month of release and count the number of movies
movie_release_counts = movies_data.groupby('month_of_release').size().reset_index(name='Count')

# Sort by count in descending order for movies
movie_release_counts = movie_release_counts.sort_values(by='Count', ascending=False)

movie_release_counts.head(20)
```

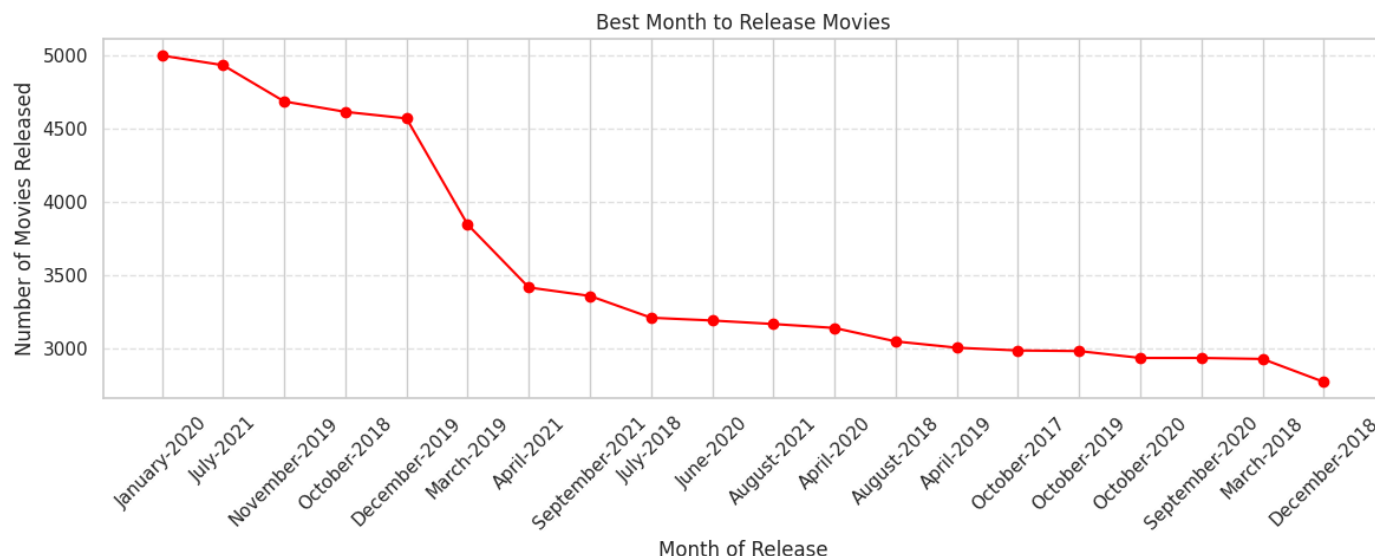
	month_of_release	Count	
41	January-2020	5000	
50	July-2021	4934	
84	November-2019	4686	
92	October-2018	4615	
23	December-2019	4570	
63	March-2019	3844	
7	April-2021	3414	
104	September-2021	3356	
47	July-2018	3207	
57	June-2020	3188	
15	August-2021	3164	
6	April-2020	3137	
12	August-2018	3045	
5	April-2019	3002	
91	October-2017	2983	
93	October-2019	2979	
94	October-2020	2932	
103	September-2020	2932	
62	March-2018	2925	
22	December-2018	2768	

```
# Select the best month for movie releases
best_month_movie = movie_release_counts.head(20)

# Create a line plot for the best month to release movies
plt.figure(figsize=(12, 5))
plt.plot(best_month_movie['month_of_release'], best_month_movie['Count'], color='red', marker='o', linestyle='--')
plt.title('Best Month to Release Movies')
plt.xlabel('Month of Release')
plt.ylabel('Number of Movies Released')
plt.grid(axis='y', linestyle='--', alpha=0.6)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```



Based on the provided data for the number of movies added to Netflix for various months and years, here are some insights and comments:

1. January-2020 (5000 Movies): January 2020 had the highest number of movie additions to Netflix. This suggests that the beginning of the year is a strategic time to release movies, possibly capitalizing on the holiday season and New Year's resolutions.
2. July-2021 (4934 Movies): July 2021 closely follows with a significant number of movie additions. Summer months often witness increased streaming activity, making it an opportune time to launch new movies.
3. November-2019 (4686 Movies): November 2019 recorded a substantial number of movie additions, indicating Netflix's commitment to expanding its movie library, especially in the lead-up to the holiday season.
4. December-2019 (4570 Movies): December 2019 shows a high count of movie additions, likely in preparation for the holiday season when viewers have more leisure time.
5. April-2021 (3414 Movies): April 2021 witnessed a notable number of movie additions. This could be associated with the spring season when viewers may be looking for fresh content.
6. September-2021 (3356 Movies): September 2021 also had a substantial number of movie additions, possibly targeting viewers returning from summer vacations.
7. October-2018 (3355 Movies): October 2018 reflects a high count of movie additions, suggesting Netflix's consistent effort to provide diverse content throughout the year.
8. July-2018 (3207 Movies): July 2018 shows a considerable number of movie additions, possibly aligning with summer blockbuster releases.
9. June-2020 (3188 Movies): June 2020 had a notable number of movie additions. This might be related to viewership trends during the early summer months.
10. August-2021 (3164 Movies): August 2021 also recorded a significant number of movie additions, likely aiming to engage viewers during the late summer period.

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

a. Identify the top 10 directors who have appeared in most movies or TV shows.

```
# Filter out rows where 'cast' is not null (i.e., movies/TV shows with known directors)
actors_data = data[data['cast'].notnull()]

# Group by cast and count the number of unique titles they have worked
actors_counts = actors_data.groupby('cast')['title'].nunique().reset_index()

# Rename the columns for clarity
actors_counts.columns = ['Actors', 'Appearances']

# Sort the cast by the number of appearances in descending order
```

```
top_10_actors = actors_counts.sort_values(by='Appearances', ascending=False).head(10)
```

```
top_10_actors
```

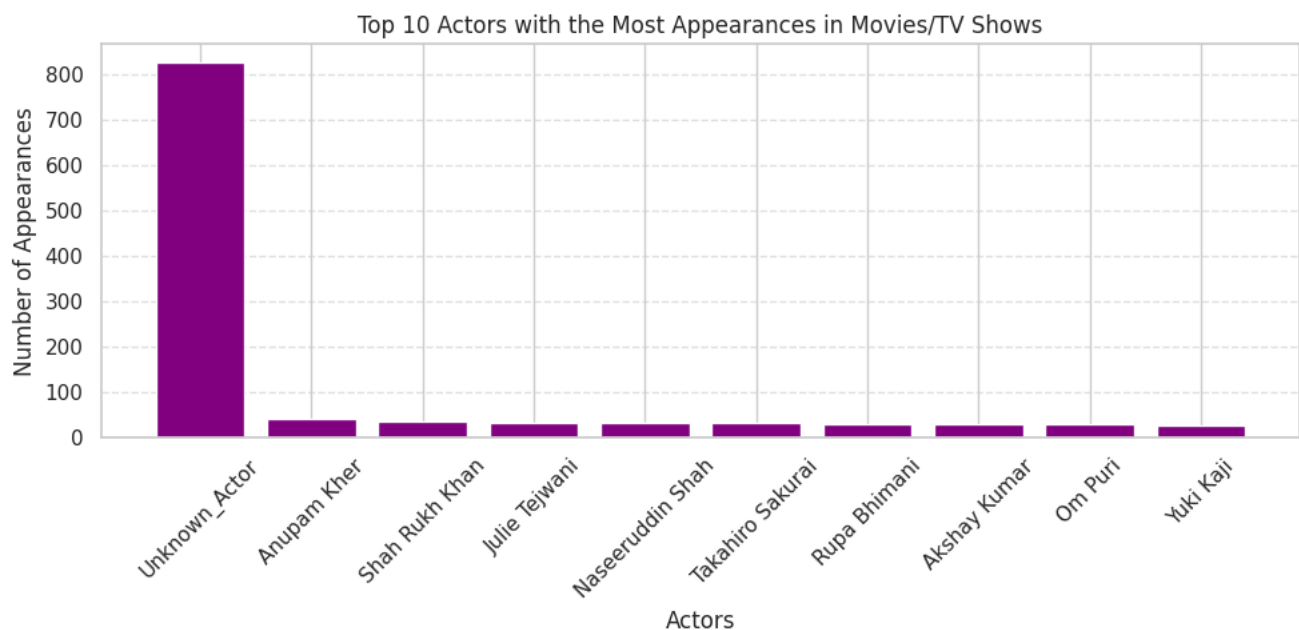
	Actors	Appearances
34214	Unknown_Actor	825
2833	Anupam Kher	43
30489	Shah Rukh Khan	35
16697	Julie Teiwani	33
24215	Naseeruddin Shah	32
32591	Takahiro Sakurai	32
28974	Rupa Bhimani	31
846	Akshay Kumar	30
25424	Om Puri	30
35881	Yuki Kaji	29

```
# Create a bar plot for the top 10 actors
plt.figure(figsize=(10, 5))
plt.bar(top_10_actors['Actors'], top_10_actors['Appearances'], color='purple')
plt.xlabel('Actors')
plt.ylabel('Number of Appearances')
plt.title('Top 10 Actors with the Most Appearances in Movies/TV Shows')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Add gridlines
plt.grid(axis='y', linestyle='--', alpha=0.6)

# Show the plot
plt.tight_layout()
plt.show()
```



The data provided contains information about actors and the number of appearances they have made in Netflix content.

Here are some observations and comments based on the data:

- 1. Unknown Artist (825 Appearances): The category "Unknown_Artist" stands out with a remarkably high number of appearances (825). This category likely includes instances where specific actor information is not available or is not credited.
- 2. Anupam Kher (43 Appearances): Anupam Kher is one of the top actors on Netflix with 43 appearances. This suggests a prolific career in Netflix content.
- 3. Shah Rukh Khan (35 Appearances): Shah Rukh Khan, a prominent Bollywood actor, has made 35 appearances on Netflix. His presence indicates the popularity of Indian content on the platform.
- 4. Julie Tejwani (33 Appearances): Julie Tejwani has appeared in 33 titles on Netflix, indicating a significant contribution to the platform.
- 5. Naseeruddin Shah (32 Appearances): Naseeruddin Shah, a respected Indian actor, has 32 appearances on Netflix. His association adds to the diversity of content.
- 6. Takahiro Sakurai (32 Appearances): Takahiro Sakurai is known for voice acting in anime and has appeared in 32 titles, suggesting a presence in anime content.
- 7. Rupa Bhimani (31 Appearances): Rupa Bhimani has made 31 appearances on Netflix, showcasing a substantial body of work.
- 8. Om Puri (30 Appearances): The late Om Puri, a celebrated Indian actor, has 30 appearances on the platform, contributing to Netflix's library.
- 9. Akshay Kumar (30 Appearances): Akshay Kumar, a popular Bollywood actor known for his versatile roles, has 30 appearances on Netflix.
- 10. Yuki Kaji (29 Appearances): Yuki Kaji's 29 appearances indicate a strong presence in anime and animated content on Netflix.

b. Identify the top 10 directors who have appeared in most movies or TV shows.



```
# Filter out rows where 'director' is not null (i.e., movies/TV shows with known directors)
directors_data = data[data['director'].notnull()]

# Group by director and count the number of unique titles they have directed
director_counts = directors_data.groupby('director')['title'].nunique().reset_index()

# Rename the columns for clarity
director_counts.columns = ['Director', 'Unique_Titles']

# Sort the directors by the number of unique titles in descending order
top_10_directors = director_counts.sort_values(by='Unique_Titles', ascending=False).head(10)

top_10_directors
```

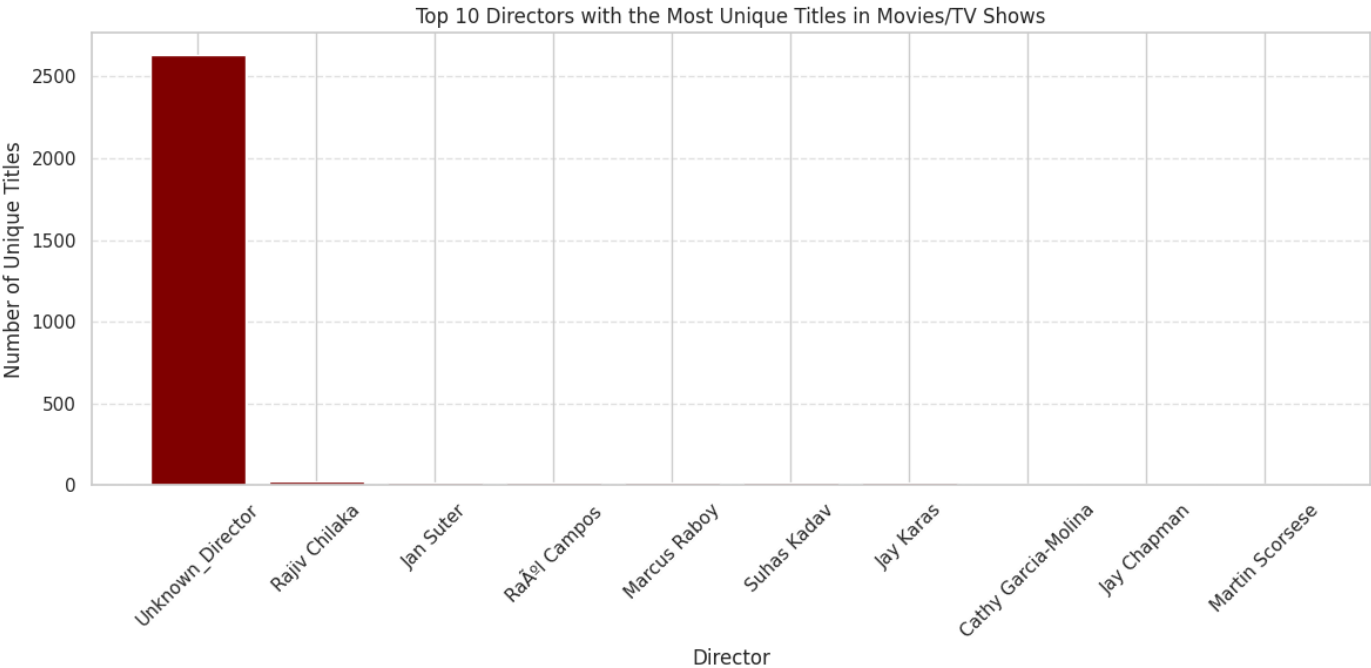
	Director	Unique_Titles	
4744	Unknown_Director	2633	
3749	Rajiv Chilaka	22	
1906	Jan Suter	21	
3800	Raúl Campos	19	
2866	Marcus Raboy	16	
4457	Suhas Kadav	16	
1954	Jay Karas	15	
755	Cathy Garcia-Molina	13	
1951	Jay Chapman	12	
2945	Martin Scorsese	12	

```
# Create a bar plot for the top 10 directors
plt.figure(figsize=(12, 6))
plt.bar(top_10_directors['Director'], top_10_directors['Unique_Titles'], color='maroon')
plt.xlabel('Director')
plt.ylabel('Number of Unique Titles')
plt.title('Top 10 Directors with the Most Unique Titles in Movies/TV Shows')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Add gridlines
plt.grid(axis='y', linestyle='--', alpha=0.6)
```

```
# Show the plot
plt.tight_layout()
plt.show()
```



The data provided represents a list of directors along with the number of unique titles they have worked on. Here are some observations and comments:

- 1. Unknown Director (4744 Unique Titles): The "Unknown_Director" category stands out with a significantly high number of unique titles (2634). This category likely includes content where the director's information is missing or not available.
- 2. Rajiv Chilaka (22 Unique Titles): Rajiv Chilaka has worked on 22 unique titles. This could indicate a prolific director with a notable contribution to the platform.
- 3. Jan Suter (21 Unique Titles): Jan Suter is another director with a substantial number of unique titles (21). Their work appears to be well-represented on Netflix.
- 4. Raúl Campos (19 Unique Titles): Raúl Campos has directed 19 unique titles, suggesting a diverse range of projects.
- 5. Marcus Raboy (16 Unique Titles): Marcus Raboy has worked on 16 unique titles, indicating a consistent presence on the platform.
- 6. Suhas Kadav (16 Unique Titles): Suhas Kadav also has 16 unique titles to their name, which may include animated content.
- 7. Jay Karas (15 Unique Titles): Jay Karas has directed 15 unique titles, showcasing their contribution to Netflix's content library.
- 8. Cathy Garcia-Molina (13 Unique Titles): Cathy Garcia-Molina is associated with 13 unique titles, suggesting a significant body of work.
- 9. Jay Chapman (12 Unique Titles): Jay Chapman has directed 12 unique titles, indicating a noteworthy presence on the platform.
- 10. Martin Scorsese (12 Unique Titles): Even a renowned director like Martin Scorsese has 12 unique titles on Netflix, which demonstrates the diversity of content available.

5 . Which genre movies are more popular or produced more

```
# Import the WordCloud class from the wordcloud library
from wordcloud import WordCloud

# Concatenate all genre values into a single string
genre_text = ' '.join(data['listed_in'].dropna())
```

```
# Create a WordCloud object
wordcloud = WordCloud(width=800, height=400, background_color='white', colormap='viridis', max_words=50).generate(genre_text)

wordcloud

<wordcloud.wordcloud.WordCloud at 0x7b982886ffd0>

# Create a figure for the word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.title('Word Cloud of Movie Genres')
plt.axis('off') # Turn off axis labels

# Display the word cloud
plt.show()
```



6. Find After how many days the movie will be added to Netflix after the release of the movie

```
# Calculate the time difference in days between date added and release year
data['days_to_add'] = (data['date_added'] - pd.to_datetime(data['release_year'], format='%Y')).dt.days

# Find the mode (most common) value of days to add
mode_duration = data['days to add'].mode()[0]
```

mode_duration

282.0

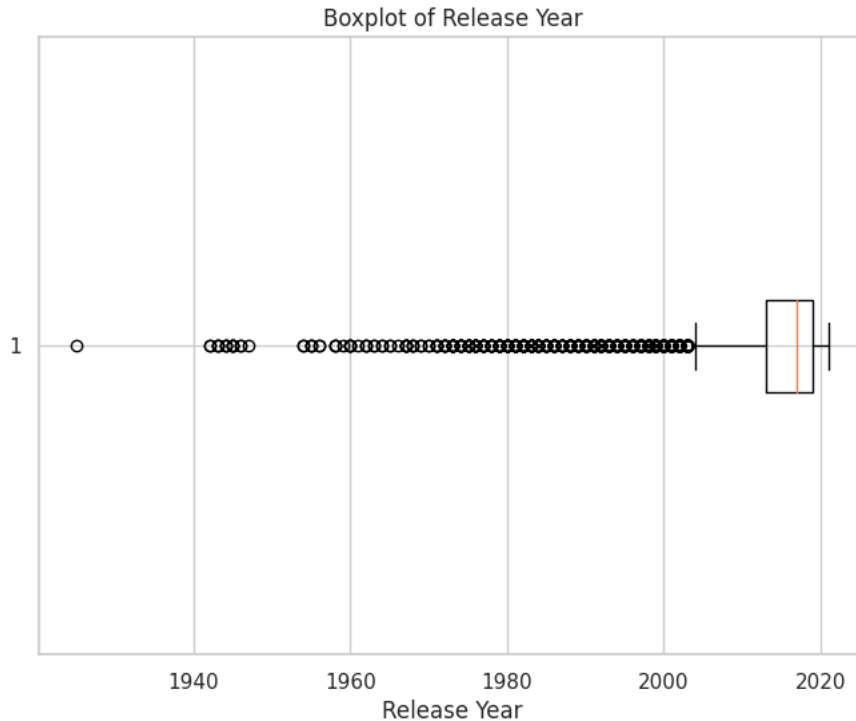
```
data.head(2)
```

show_id	type	title	date_added	release_year	rating	duration	description		cast	director	country	listed_in
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90 min	As her father nears the end of his life, filmm...	Unknown_Actor	Kirsten Johnson	United States	Documentarie
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape	Ama Qamata	Unknown_Director	South Africa	Internationa TV Show

▼ Boxplot

there is a lot of historical content which is not preferred by modern audience

```
plt.figure(figsize=(8, 6))
plt.boxplot(df['release_year'], vert=False)
plt.xlabel('Release Year')
plt.title('Boxplot of Release Year')
plt.show()
```



▼ Heatmap

It shows that TV shows and movies are highly rated as TV-MA, and from this, we can conclude that Netflix should focus more on content that is based on this particular rating

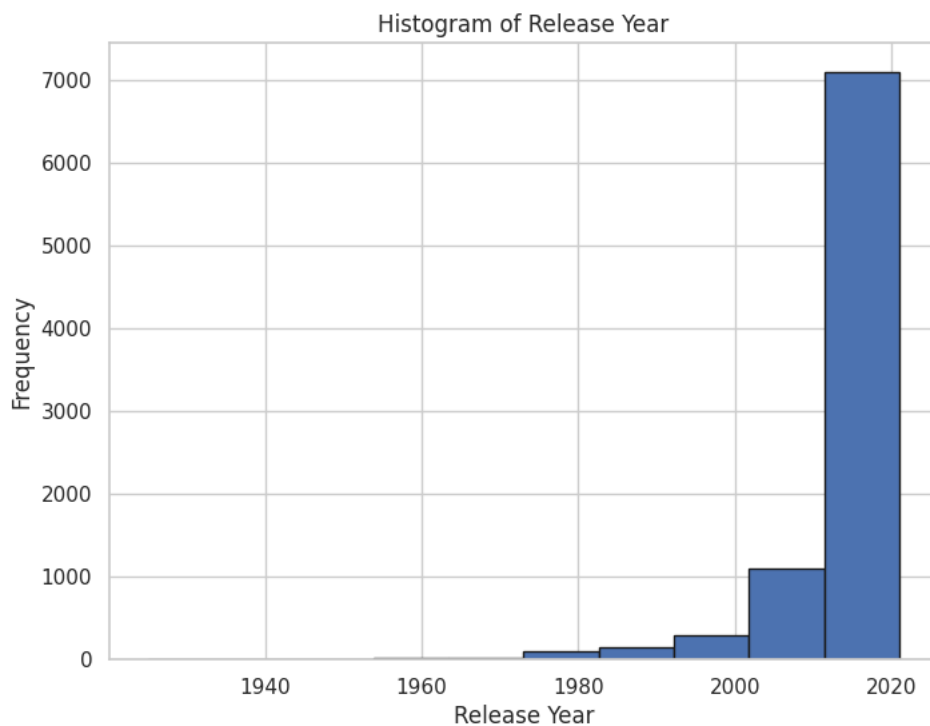
```
colormap = plt.cm.plasma
sns.heatmap(pd.crosstab(data["rating"], data["type"]), cmap = colormap)
```


<Axes: xlabel='type', ylabel='rating'>

▼ Histogram

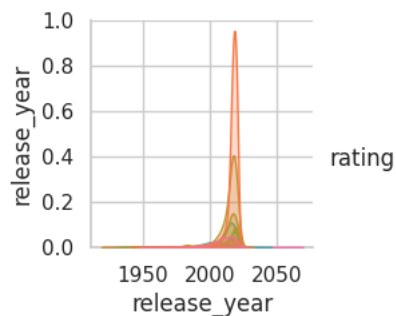
G 

```
plt.figure(figsize=(8, 6))
plt.hist(df['release_year'], bins=10, edgecolor='k')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.title('Histogram of Release Year')
plt.grid(True)
plt.show()
```



▼ Pairplot

```
sns.pairplot(df, hue='rating', markers=['o', 's', 'D'])
plt.show()
```

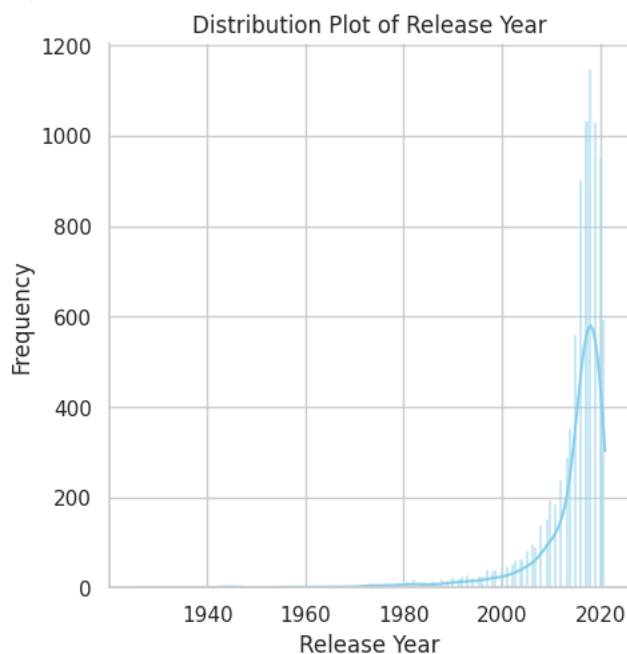


▼ Distplot

```
plt.figure(figsize=(8, 6))
sns.displot(data=df, x='release_year', kde=True, color='skyblue')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.title('Distribution Plot of Release Year')
```

```
plt.grid(True)
plt.show()
```

<Figure size 800x600 with 0 Axes>



▼ Countplot

```
import seaborn as sns
import matplotlib.pyplot as plt

# Create a countplot with grid lines
plt.figure(figsize=(12, 6))
sns.countplot(x='release_year', data=data, palette='Set2', order=data['release_year'].value_counts().index)
plt.xticks(rotation=90) # Rotate x-axis labels for better readability
plt.title('Count of Titles by Release Year')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.grid(axis='y', linestyle='--', alpha=0.7) # Add grid lines on the y-axis
plt.show()
```

Count of Titles by Release Year



""" Business insights :

- 1 . The analysis shows us that the there is high amountt of movies produced per year than tv shows
- 2 . Corona virus has impacted the content quantity
- 3 . The USA and India are the top 2 countries content wise
- 4 . The content targeted in india is teens while the content being targeted at usa is adult audience
- 5 . Lack of child content produced in india
- 6 . India and South Korea have similar taste and usa and uk audience have similar taste
- 7 . Lack of diverse content for indian audience

"""



Recommendations:

- 1. Produce more tv shows in high markets like india with diverse quantity
- 2. More movies targeting untapped young adult audience
- 3. More children quanity should be created

