In [1]:

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

df=pd.read_csv("netflix.csv")

# Let us check the nu,ber of columns and type of columns in the dataset.

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
```

In [2]:

```
df = df.convert_dtypes()
df["date_added"] = pd.to_datetime(df["date_added"])
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   string        
 1   type          8807 non-null   string        
 2   title         8807 non-null   string        
 3   director      6173 non-null   string        
 4   cast          7982 non-null   string        
 5   country       7976 non-null   string        
 6   date_added    8797 non-null   datetime64[ns]
 7   release_year  8807 non-null   Int64         
 8   rating        8803 non-null   string        
 9   duration      8804 non-null   string        
 10  listed_in     8807 non-null   string        
 11  description   8807 non-null   string        
dtypes: Int64(1), datetime64[ns](1), string(10)
memory usage: 834.4 KB
```

In [3]:

```
# Checking Sample glimpse of data set
df.head(10)
```

Out[3]:

	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	<NA>	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	<NA>	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	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...	<NA>	2021-09-24	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	<NA>	<NA>	<NA>	2021-09-24	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	<NA>	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	<NA>	2021-09-24	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	<NA>	2021-09-24	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	2021-09-24	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8	s9	TV Show	The Great British Baking Show	Andy Devonshire	Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho...	United Kingdom	2021-09-24	2021	TV-14	9 Seasons	British TV Shows, Reality TV	A talented batch of amateur bakers face off in...
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	2021-09-24	2021	PG-13	104 min	Comedies, Dramas	A woman adjusting to life after a loss contend...

In [4]:

```
# Checking the total number of rows in the data set
df.tail()
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	2019-11-20	2007	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...
8803	s8804	TV Show	Zombie Dumb	<NA>	<NA>	<NA>	2019-07-01	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	2019-11-01	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	2020-01-11	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	2019-03-02	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

In [ ]:

```

```

In [ ]:

```

```

In [5]:

```
# as the sample showed up some of the NA values, getting the count of NA values in the dataset for each column.
df.isna().sum()
```

Out[5]:

```
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 [6]:

```
# Dropping null values rows in the specific columns


movie_rating = df.loc[df['type'] == 'Movie', 'rating'].mode()[0]
tv_rating = df.loc[df['type'] == 'TV Show', 'rating'].mode()[0]

# Filling missing rating values based on the type of content
df['rating'] = df.apply(lambda x: movie_rating if x['type'] == 'Movie' and pd.isna(x['rating'])
                        else tv_rating if x['type'] == 'TV Show' and pd.isna(x['rating'])
                        else x['rating'], axis=1)
df.isna().sum()
```

Out[6]:

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             0
duration           3
listed_in          0
description        0
dtype: int64
```

In [7]:

```
# Filling dummy values in some columns where null values are significant.
df[['director', 'cast']] = df[['director', 'cast']].fillna('Unknown')
df['country'] = df['country'].fillna(df['country'].mode()[0])

df.dropna(inplace=True)
df.isna().sum()
```

Out[7]:

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

In [8]:

```
df['month_added'] = df['date_added'].dt.month
df['month_name_added'] = df['date_added'].dt.month_name()
df['year_added'] = df['date_added'].dt.year
```

In [9]:

```
# Splitting and expanding the columns
df_cast = df['cast'].str.split(',', expand=True).stack()
df_cast = df_cast.reset_index(level=1, drop=True).to_frame('cast')
df_cast['show_id'] = df['show_id']
```

```
df_country = df['country'].str.split(',', expand=True).stack()
df_country = df_country.reset_index(level=1, drop=True).to_frame('country')
df_country['show_id'] = df['show_id']
df_country['type'] = df['type']


df_listed_in = df['listed_in'].str.split(',', expand=True).stack()
df_listed_in = df_listed_in.reset_index(level=1, drop=True).to_frame('listed_in')
df_listed_in['show_id'] = df['show_id']

df_director = df['director'].str.split(',', expand=True).stack()
df_director = df_director.reset_index(level=1, drop=True).to_frame('director')
df_director['show_id'] = df['show_id']
```

In [10]:

```
# Which contries are the leading contributors to the contents on Netflix as per the data.
df['country'].value_counts()
```

Out[10]:

```
United States                                           3639
India                                                    972
United Kingdom                                           418
Japan                                                    244
South Korea                                              199
                                                        ...
Ireland, United Kingdom, Greece, France, Nethe...          1
France, Canada, Italy, United States, China                1
United States, Venezuela                                   1
United Kingdom, Canada, Japan                              1
United Arab Emirates, Jordan                               1
Name: country, Length: 748, dtype: Int64
```

In [11]:

```
print(df['release_year'].min())
print(df['release_year'].max())
```

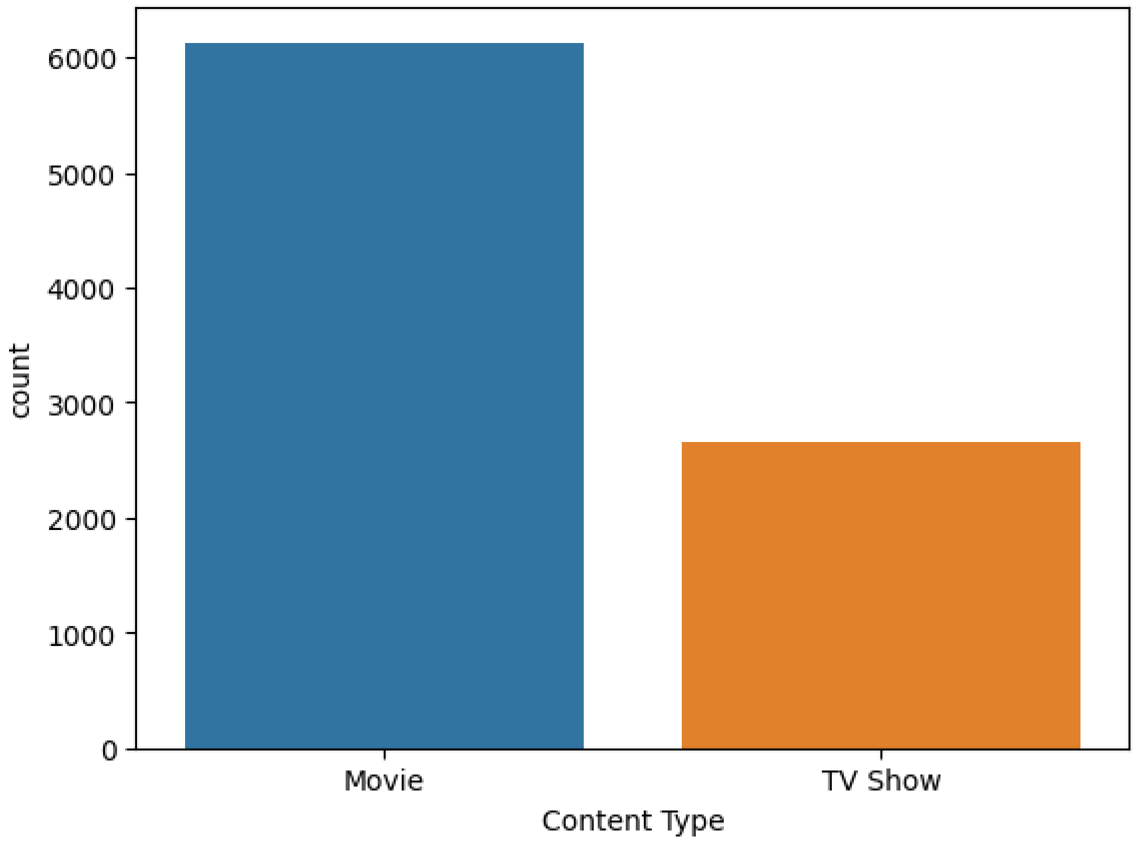
```
1925
2021
```

In [12]:

```
# Q2. Comparison of tv shows vs. movies.
'''Checking the type of content in the dataset and we found almost 30% of the shows in data are TVshows and 70% are Movies'''
sns.countplot(data=df,x=df['type'])
plt.xlabel('Content Type')
```

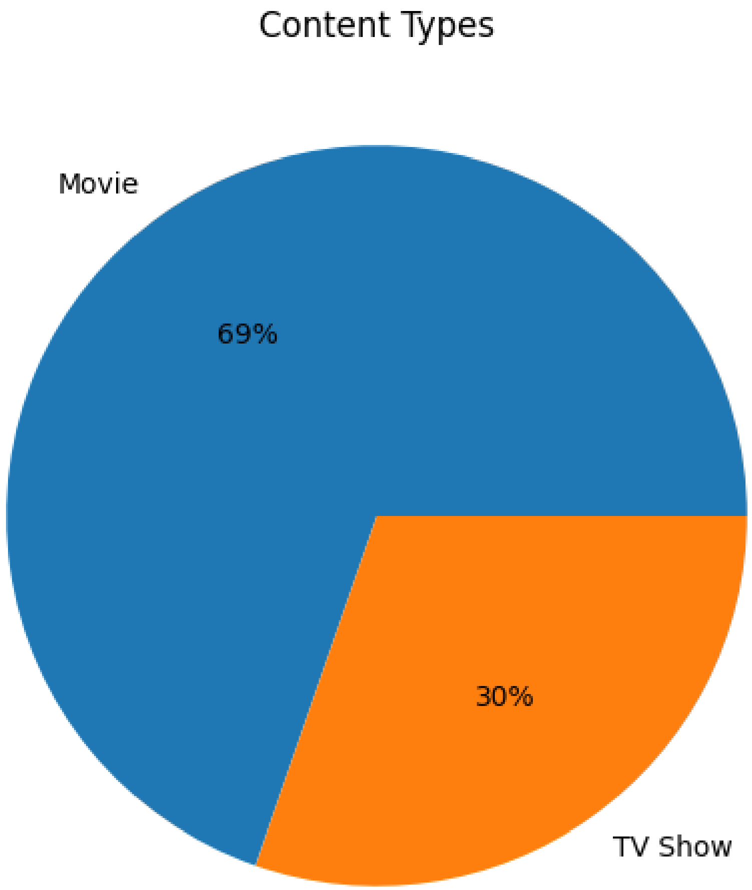
Out[12]:

```
Text(0.5, 0, 'Content Type')
```



In [13]:

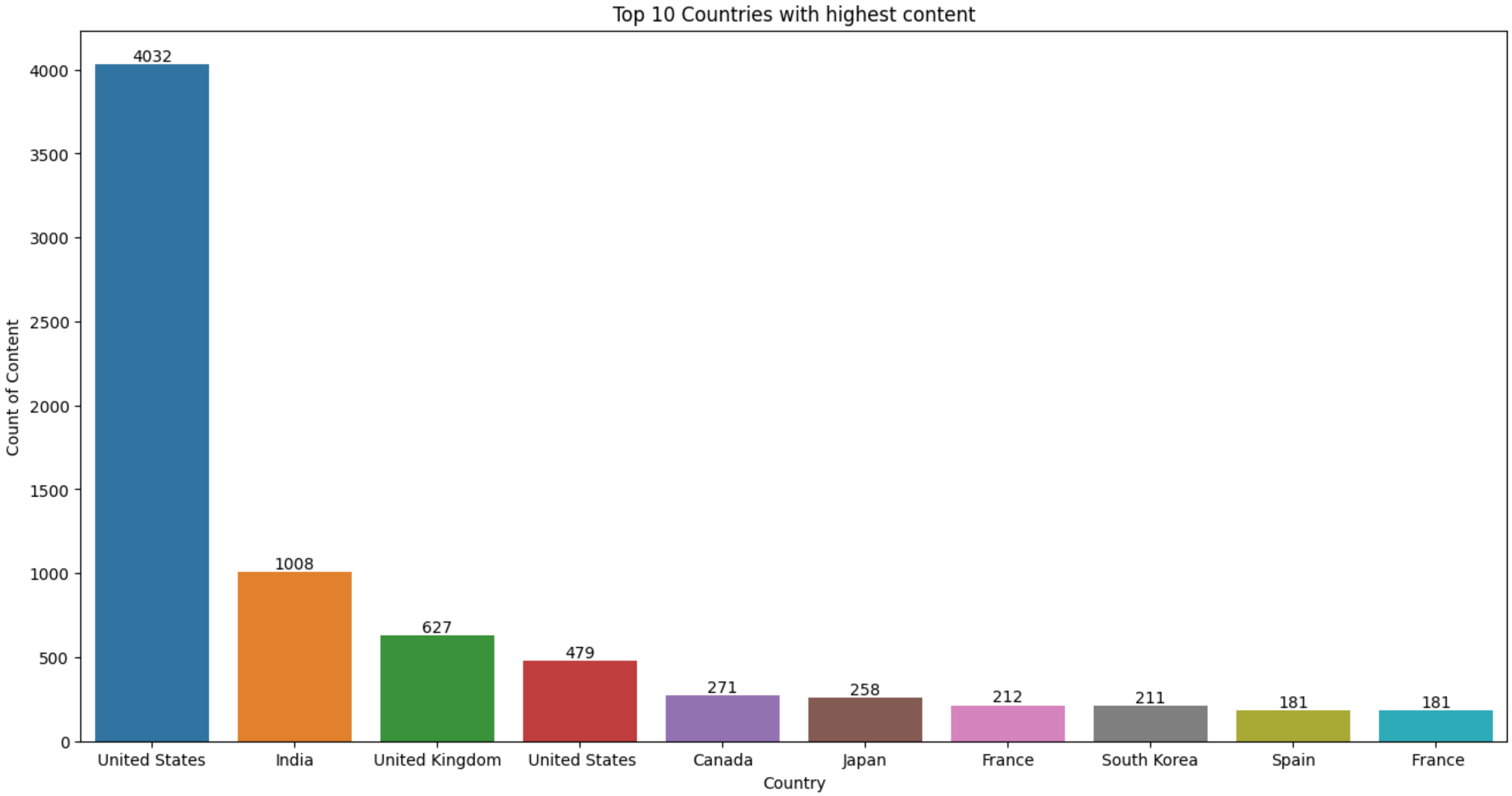
```
# Comparison of content type
x = df.groupby(['type'])['type'].count()
y = len(df)
r = ((x/y) * 100).round(2)

mf_ratio = pd.DataFrame(r)
mf_ratio.rename({'type': '%'}, axis=1, inplace=True)

plt.figure(figsize=(12, 6))
plt.pie(mf_ratio['%'], labels=mf_ratio.index,autopct='%i%%')

plt.title('Content Types')
plt.show()
```



In [14]:

```
# Top 10 countries with highes content contribution

df_country['country'] = df_country['country'].str.rstrip()
country_counts = df_country['country'].value_counts()
top_10_countries = country_counts.head(10)

plt.figure(figsize=(16, 8))

plt.xlabel('Country')
plt.ylabel('Count of Content')
plt.title('Top 10 Countries with highest content')
```

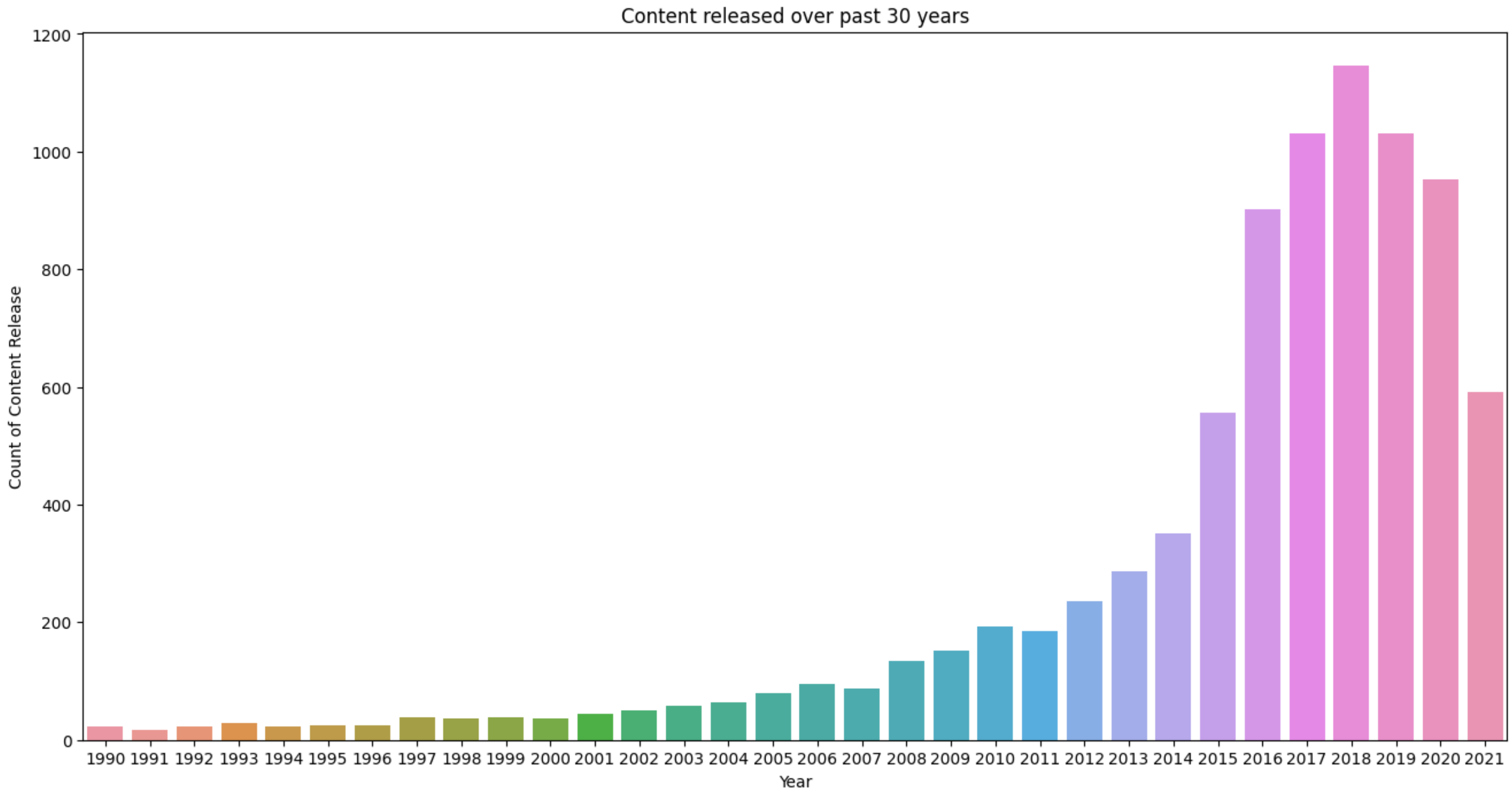
```
bar_plot = sns.barplot(x=top_10_countries.index, y=top_10_countries.values)

for index, value in enumerate(top_10_countries.values):
    bar_plot.text(index, value, str(value), ha='center', va='bottom')


plt.show()
```



In [15]:

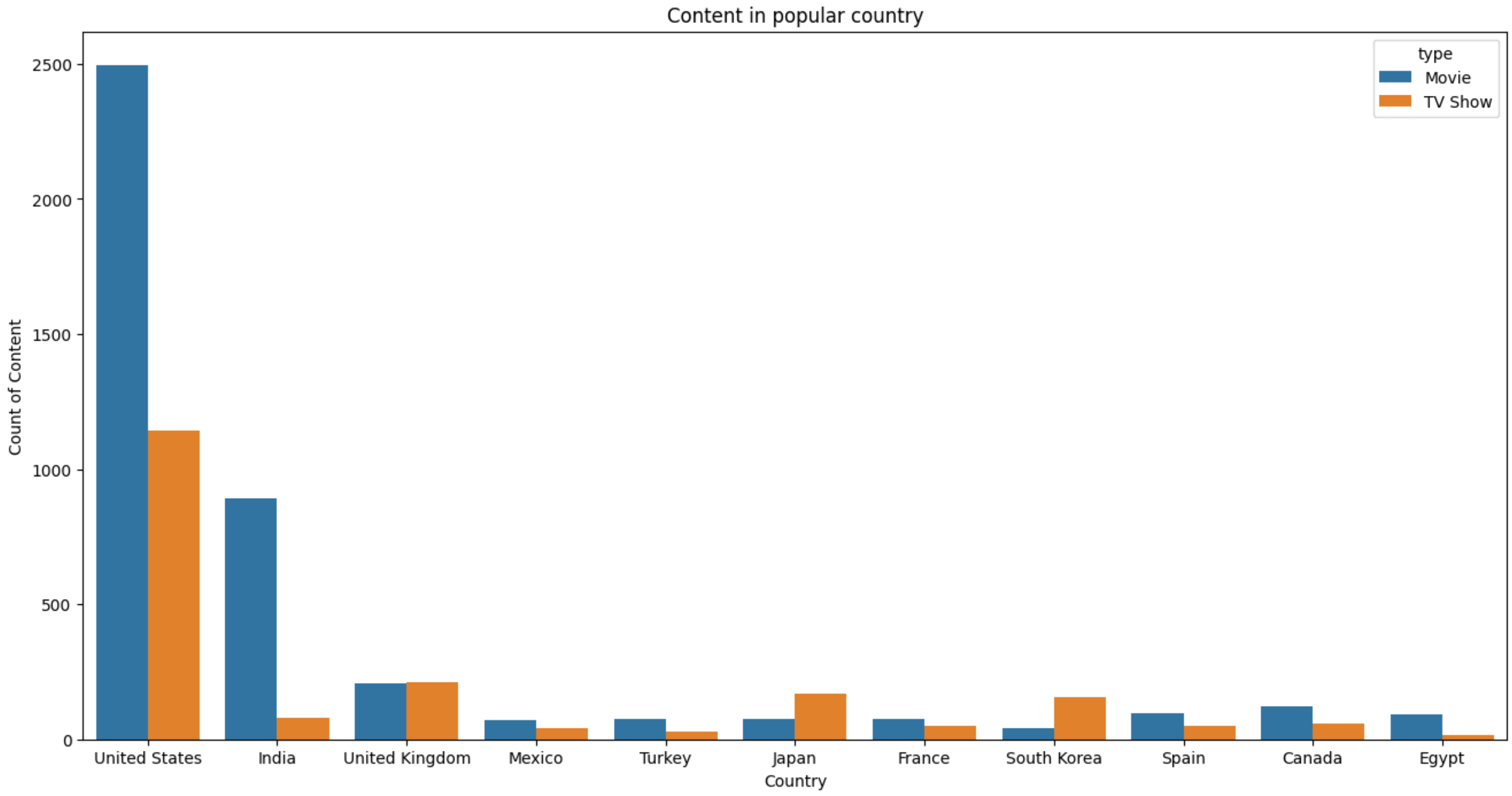
```
df_temp=df[df['release_year']>=1990]
plt.figure(figsize=(16, 8))
sns.countplot(data=df_temp,x=df_temp['release_year'])
```

```
plt.xlabel('Year')
plt.ylabel('Count of Content Release')
plt.title('Content released over past 30 years')
plt.show()
```



In [16]:

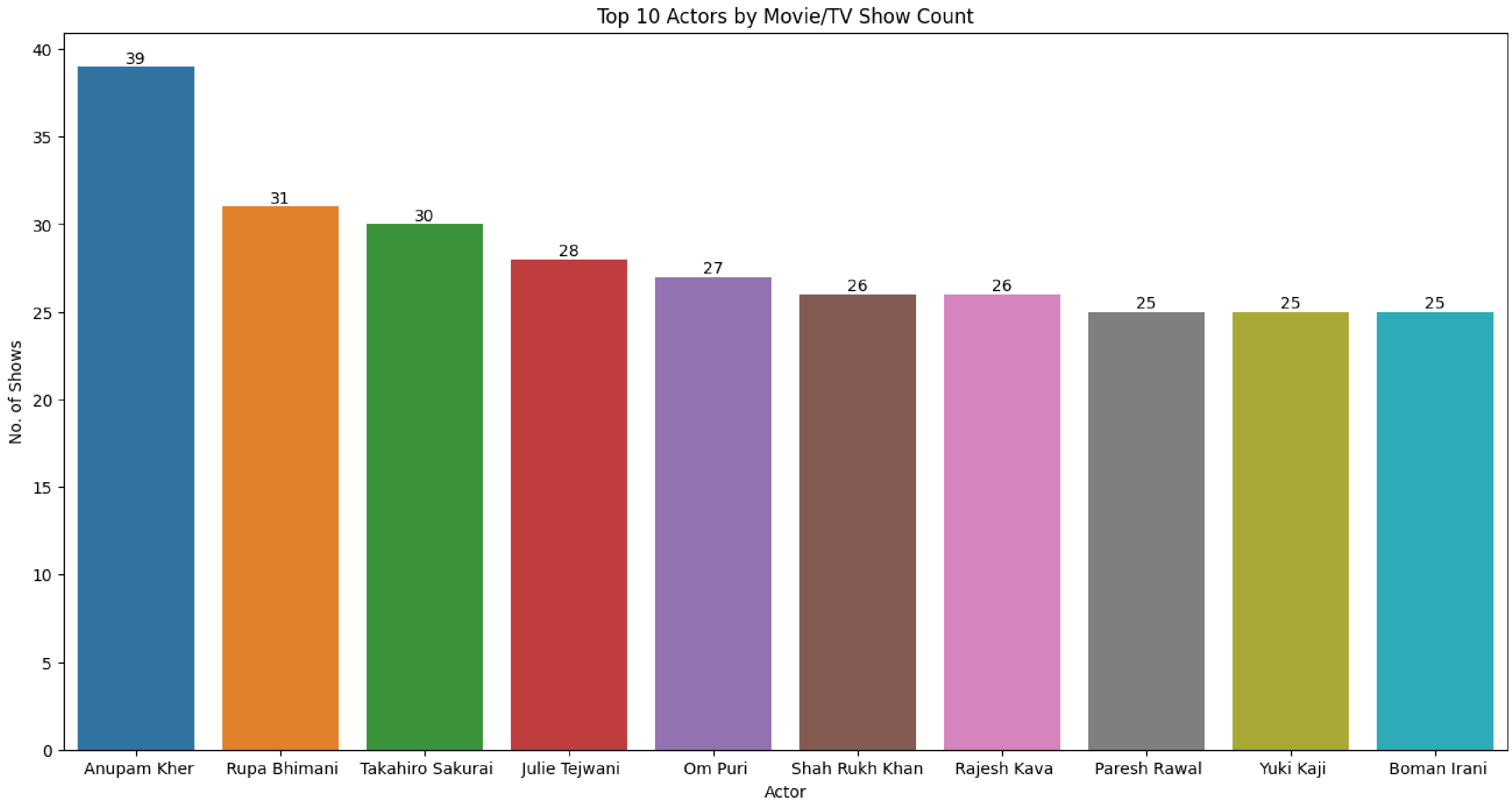
```
df_temp=df[df['country'].map(df['country'].value_counts()) >= 100]
plt.figure(figsize=(16, 8))
sns.countplot(data=df_temp,x=df_temp['country'],hue=df_temp['type'])
plt.xlabel('Country')
plt.ylabel('Count of Content')
plt.title('Content in popular country')
plt.show()
```



In [17]:

```
# Top 10 actors across all content
cast_counts = df_cast['cast'].value_counts()[1:]

top_10_cast = cast_counts.head(10)

plt.figure(figsize=(16, 8))
bar_plot = sns.barplot(x=top_10_cast.index, y=top_10_cast.values)

plt.xlabel('Actor')
plt.ylabel('No. of Shows')
plt.title('Top 10 Actors by Movie/TV Show Count')
```

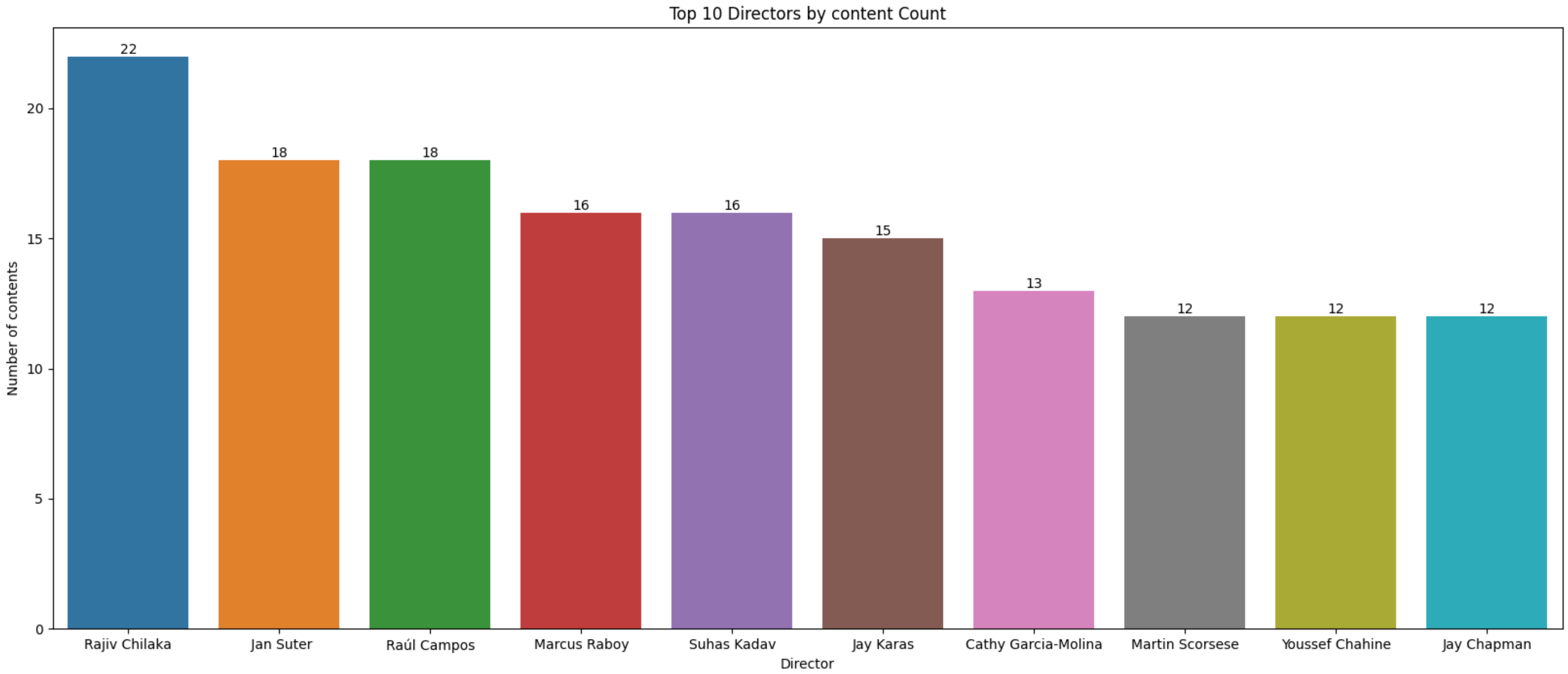
```
for index, value in enumerate(top_10_cast.values):
    bar_plot.text(index, value, str(value), ha='center', va='bottom')

plt.show()
```



In [18]:

```
# Top 10 directors
director_counts = df_director['director'].value_counts()[1:]

top_10_directors = director_counts.head(10)

plt.figure(figsize=(20, 8))
bar_plot = sns.barplot(x=top_10_directors.index, y=top_10_directors.values)
```

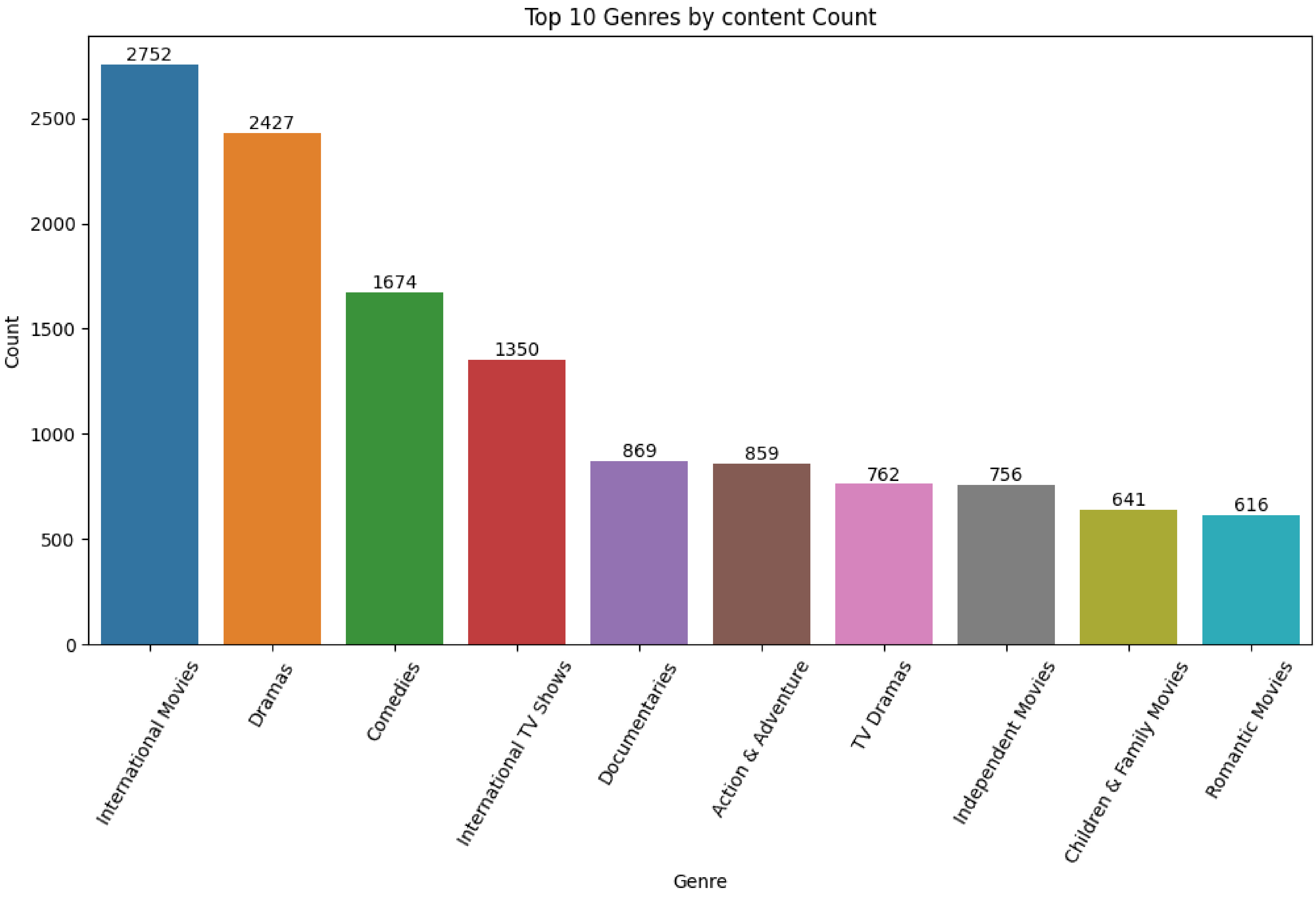
```python
plt.xlabel('Director')
plt.ylabel('Number of contents')
plt.title('Top 10 Directors by content Count')

for index, value in enumerate(top_10_directors.values):
    bar_plot.text(index, value, str(value), ha='center', va='bottom')

plt.show()
```



In [19]:

```python
#Top 10 genres in the content

df_listed_in['listed_in'] = df_listed_in['listed_in'].str.strip()

listed_in_counts = df_listed_in['listed_in'].value_counts()

top_10_listed_in = listed_in_counts.head(10)
```

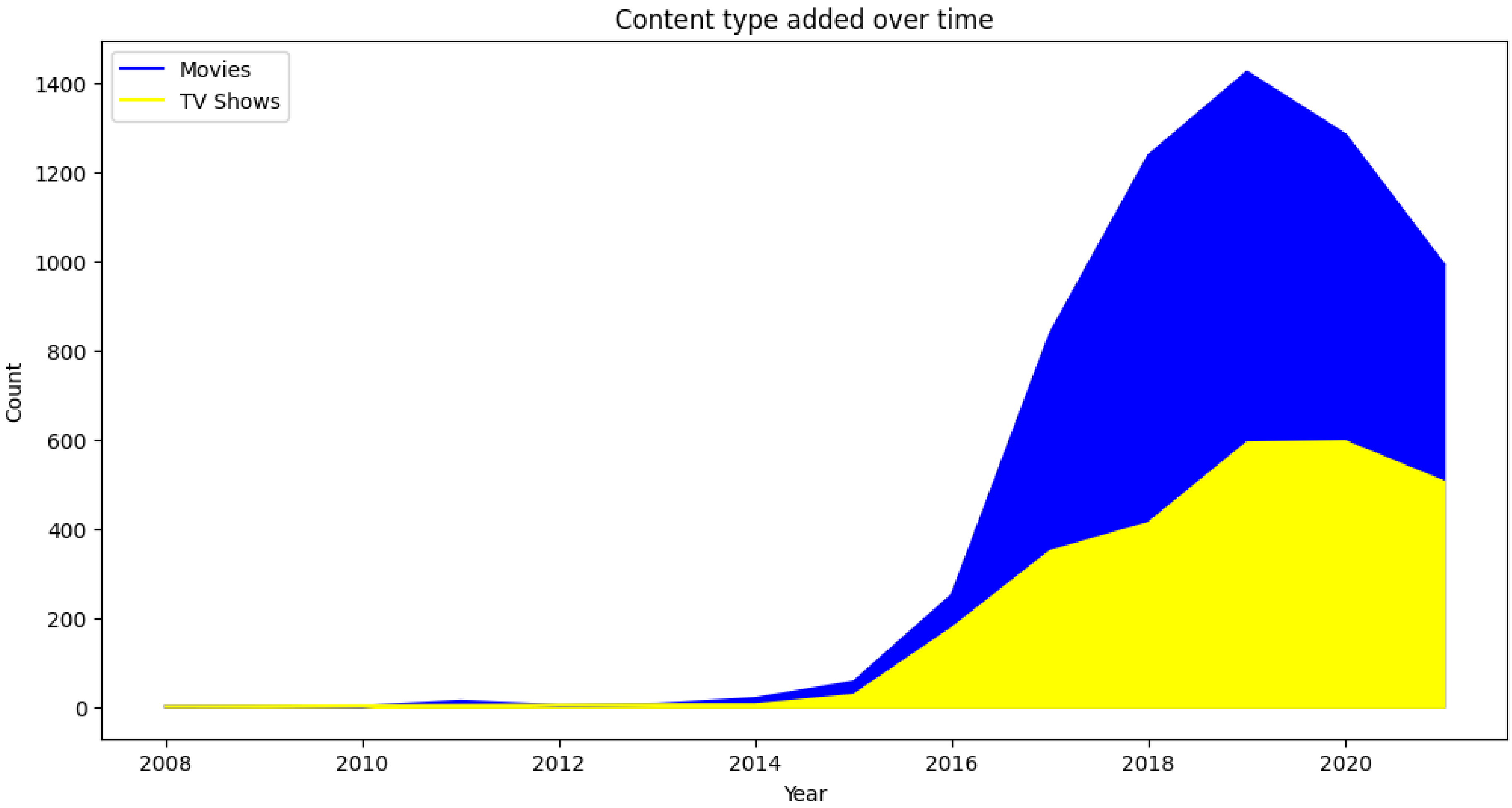
```python
plt.figure(figsize=(12, 6))
bar_plot = sns.barplot(x=top_10_listed_in.index, y=top_10_listed_in.values)

plt.xlabel("Genre")
plt.ylabel("Count")
plt.title('Top 10 Genres by content Count')
plt.xticks(rotation=60)

for index, value in enumerate(top_10_listed_in.values):
    bar_plot.text(index, value, str(value), ha='center', va='bottom')

plt.show()
```



In [20]:

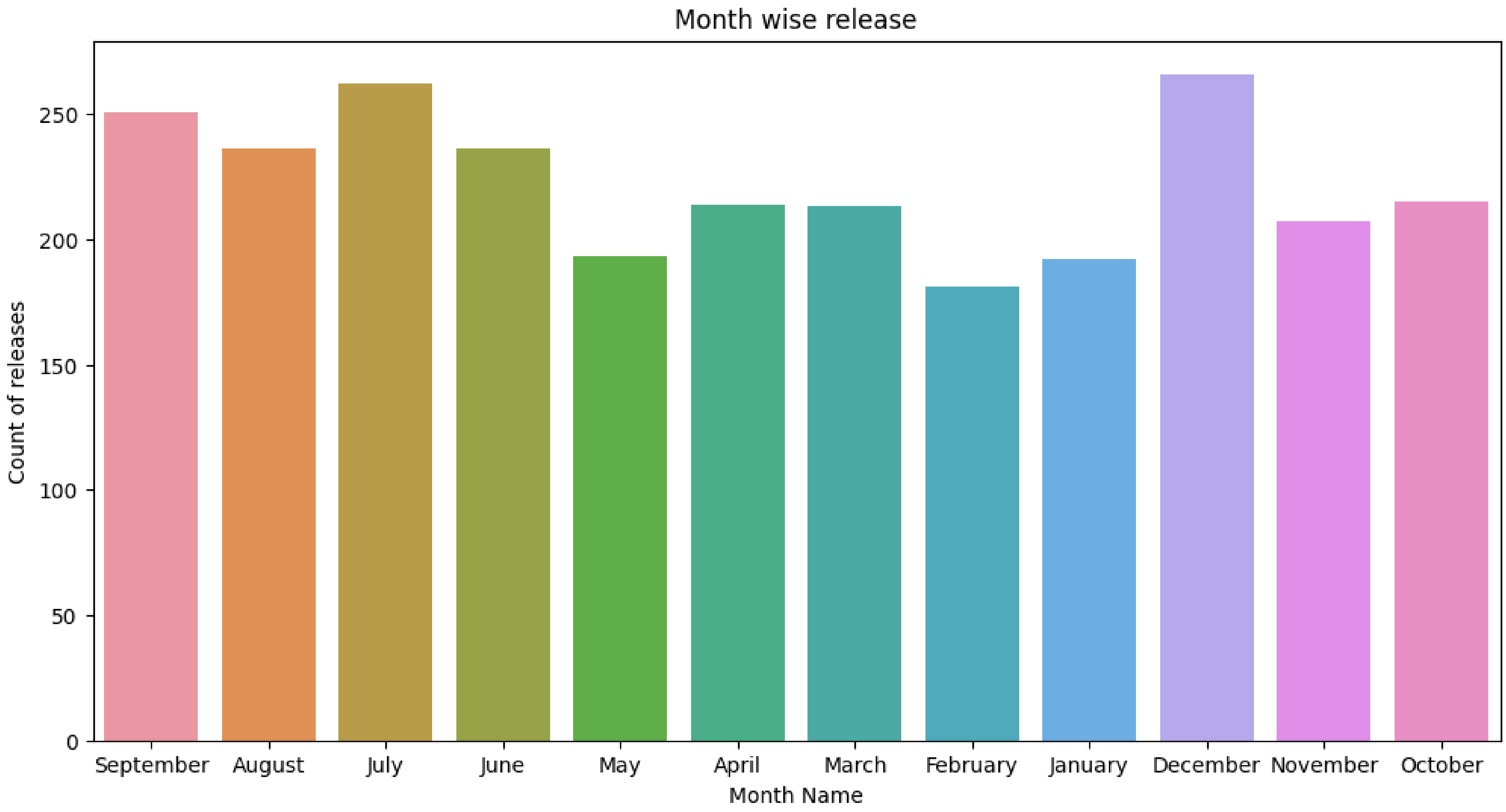
```
df_movies = df[df['type'] == 'Movie']
df_tv_shows = df[df['type'] == 'TV Show']

movies_count = df_movies['year_added'].value_counts().sort_index()
```

```
tv_shows_count = df_tv_shows['year_added'].value_counts().sort_index()

plt.figure(figsize=(12, 6))
plt.plot(movies_count.index, movies_count.values, color='blue', label='Movies')
plt.plot(tv_shows_count.index, tv_shows_count.values, color='yellow', label='TV Shows')

plt.fill_between(movies_count.index, movies_count.values, color='blue')
plt.fill_between(tv_shows_count.index, tv_shows_count.values, color='yellow')

plt.xlabel('Year')
plt.ylabel('Count')
plt.title('Content type added over time')
plt.legend()

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



In [21]:

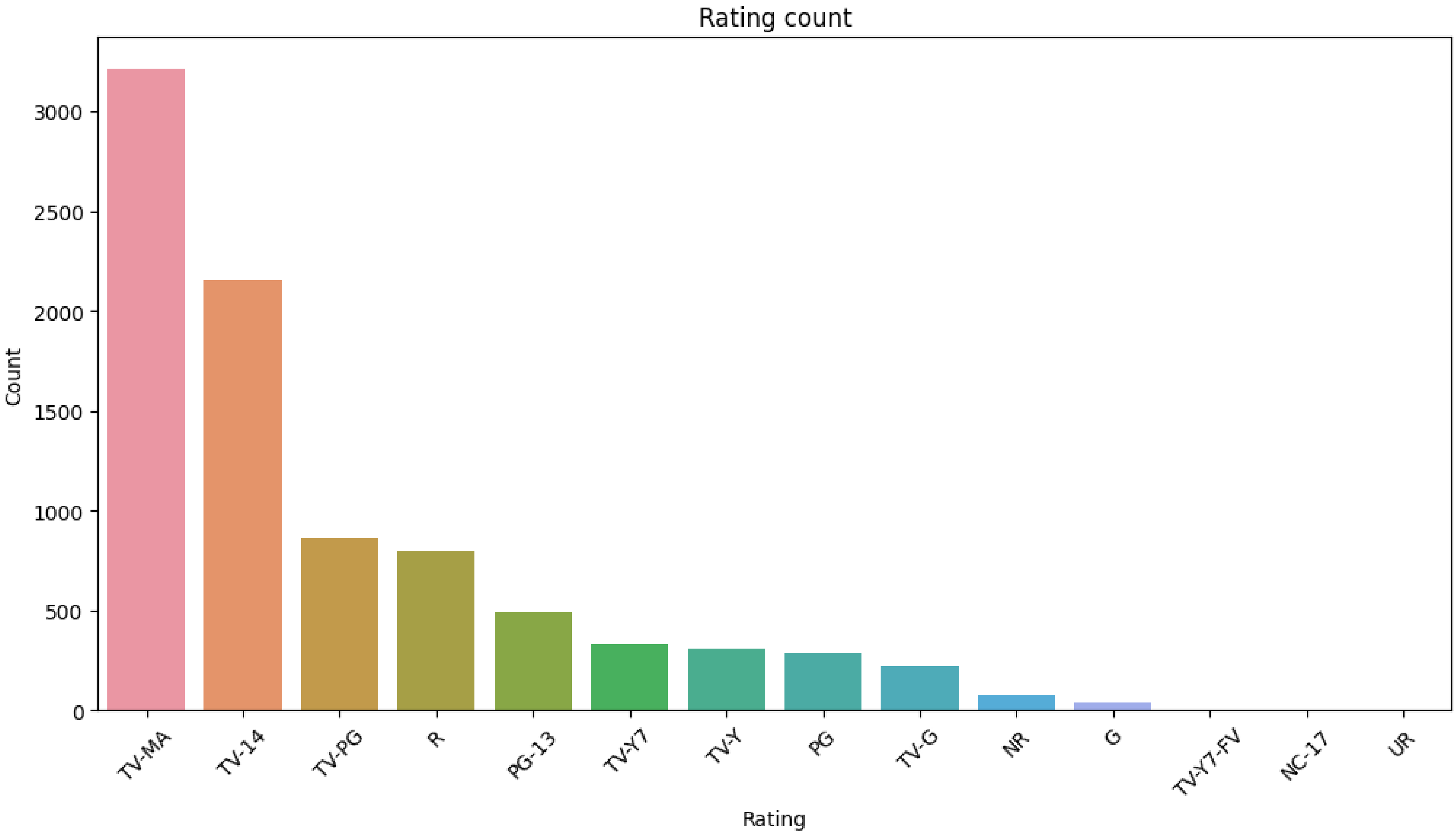
```
# Q3. What is the best time to launch a TV show?
'''
It seems December & July are the best time to launch TV shows.
'''
fig2 = plt.figure(figsize=(12, 6))
df_movie_mon=df[(df['type']=='TV Show')]
sns.countplot(data=df_movie_mon,x=df_movie_mon['date_added'].dt.month_name())
plt.xlabel('Month Name')
plt.ylabel('Count of releases')
plt.title('Month wise release')
plt.show()
```



In [ ]:

In [22]:

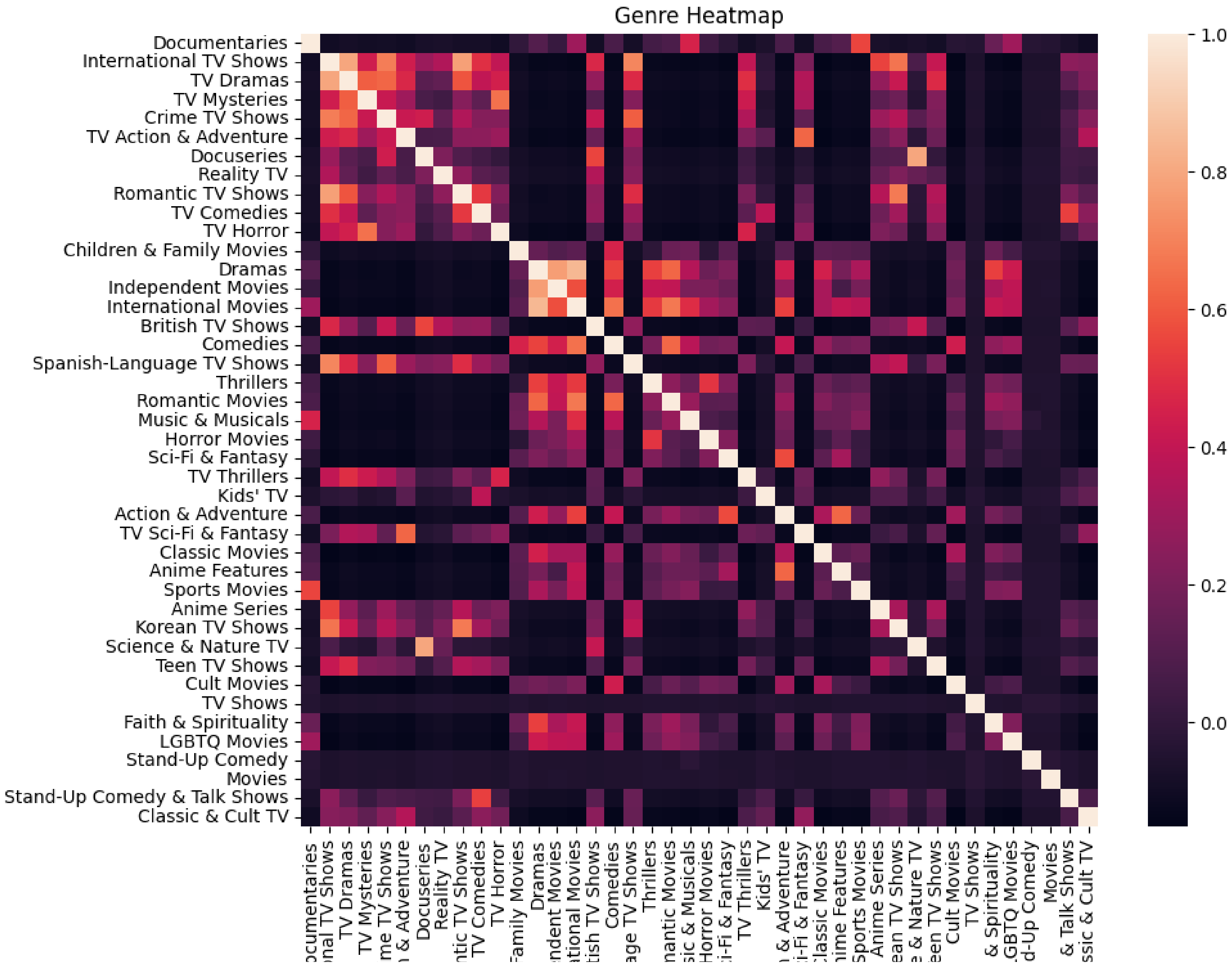
```
# Highest rating count
rating_counts = df['rating'].value_counts()

plt.figure(figsize=(12,6))
sns.barplot(x=rating_counts.index, y=rating_counts.values)

plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Rating count')
plt.xticks(rotation=45)

plt.show()
```

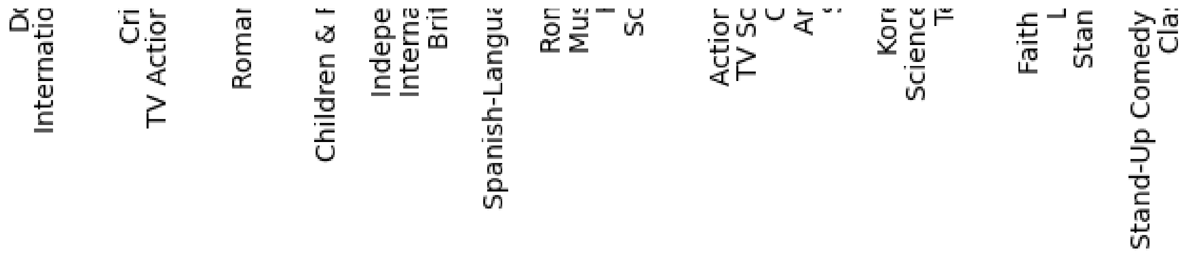


In [23]:

```
genres = df['listed_in'].str.split(', ', expand=True).stack().unique()

genre_data = pd.DataFrame(index=genres, columns=genres, dtype=float)

genre_data.fillna(0, inplace=True)
```

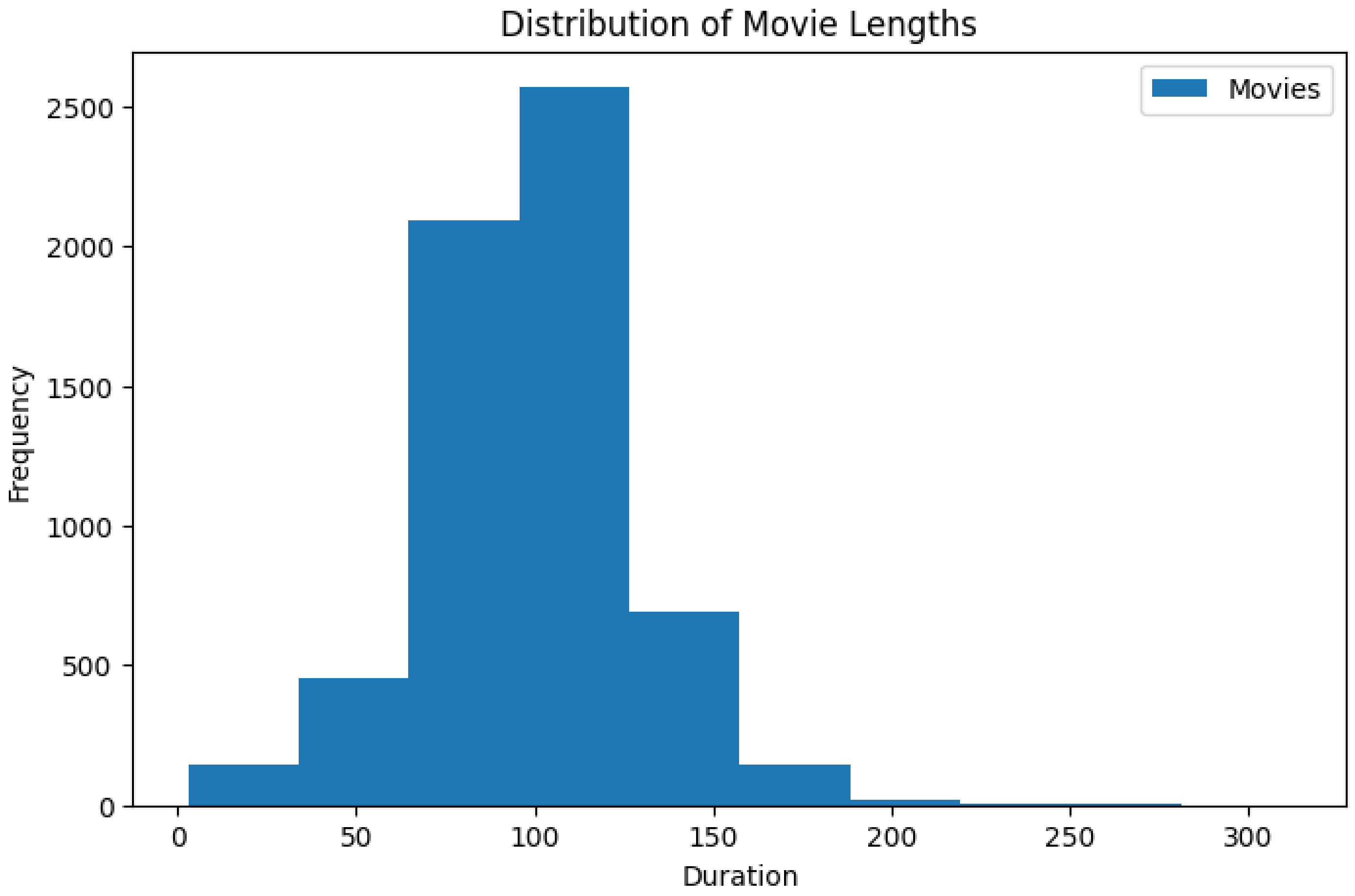
```
for _, row in df.iterrows():
    listed_in = row['listed_in'].split(', ')
    for genre1 in listed_in:
        for genre2 in listed_in:
            genre_data.at[genre1, genre2] += 1

correlation_matrix = genre_data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix)

plt.title('Genre Heatmap')
plt.xticks=rotation=90
plt.yticks=rotation=0

plt.show()
```


Genre Heatmap
Documentaries
International TV Shows
TV Dramas
TV Mysteries
Crime TV Shows
TV Action & Adventure
Docuseries
Reality TV
Romantic TV Shows
TV Comedies
TV Horror
Children & Family Movies
Dramas
Independent Movies
International Movies
British TV Shows
Comedies
Spanish-Language TV Shows
Thrillers
Romantic Movies
Music & Musicals
Horror Movies
Sci-Fi & Fantasy
TV Thrillers
Kids' TV
Action & Adventure
TV Sci-Fi & Fantasy
Classic Movies
Anime Features
Sports Movies
Anime Series
Korean TV Shows
Science & Nature TV
Teen TV Shows
Cult Movies
TV Shows
Faith & Spirituality
LGBTQ Movies
Stand-Up Comedy
Movies
Stand-Up Comedy & Talk Shows
Classic & Cult TV
ocumentaries
onal TV Shows
TV Dramas
TV Mysteries
me TV Shows
n & Adventure
Docuseries
Reality TV
ntic TV Shows
TV Comedies
TV Horror
Family Movies
Dramas
ndent Movies
ational Movies
tish TV Shows
Comedies
age TV Shows
Thrillers
nantic Movies
sic & Musicals
Horror Movies
i-Fi & Fantasy
TV Thrillers
Kids' TV
n & Adventure
i-Fi & Fantasy
lassic Movies
nime Features
Sports Movies
Anime Series
ean TV Shows
e & Nature TV
een TV Shows
Cult Movies
TV Shows
& Spirituality
GBTQ Movies
d-Up Comedy
Movies
& Talk Shows
ssic & Cult TV
1.0
0.8
0.6
0.4
0.2
0.0



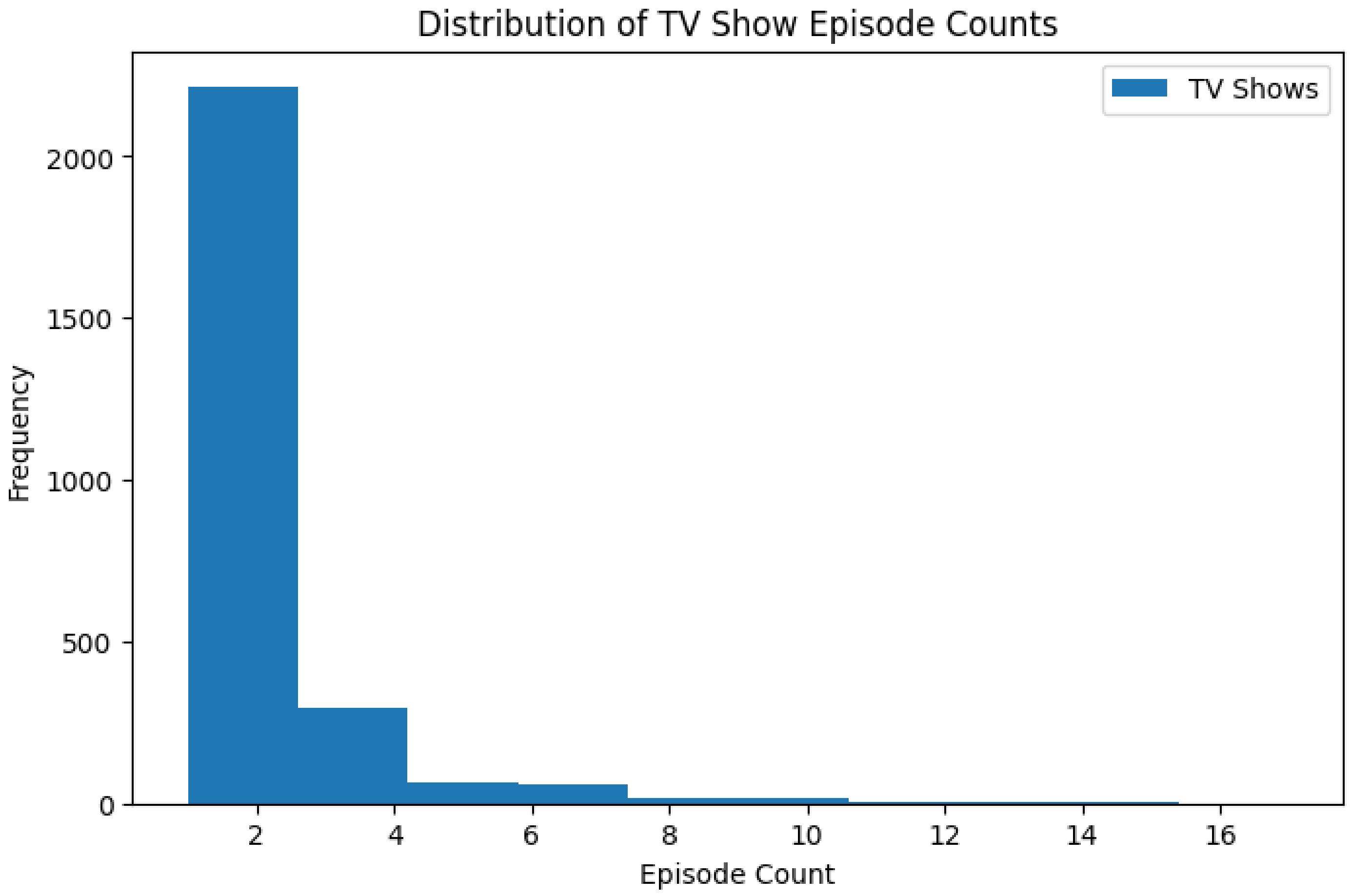
In [24]:

```
# Movie lenght analysis
movie_lengths = df_movies['duration'].str.extract('(\d+)', expand=False).astype(int)

plt.figure(figsize=(8, 5))
plt.hist(movie_lengths, bins=10, label='Movies')

plt.xlabel('Duration')
plt.ylabel('Frequency')
plt.title('Distribution of Movie Lengths')
plt.legend()

plt.show()
```



In [25]:

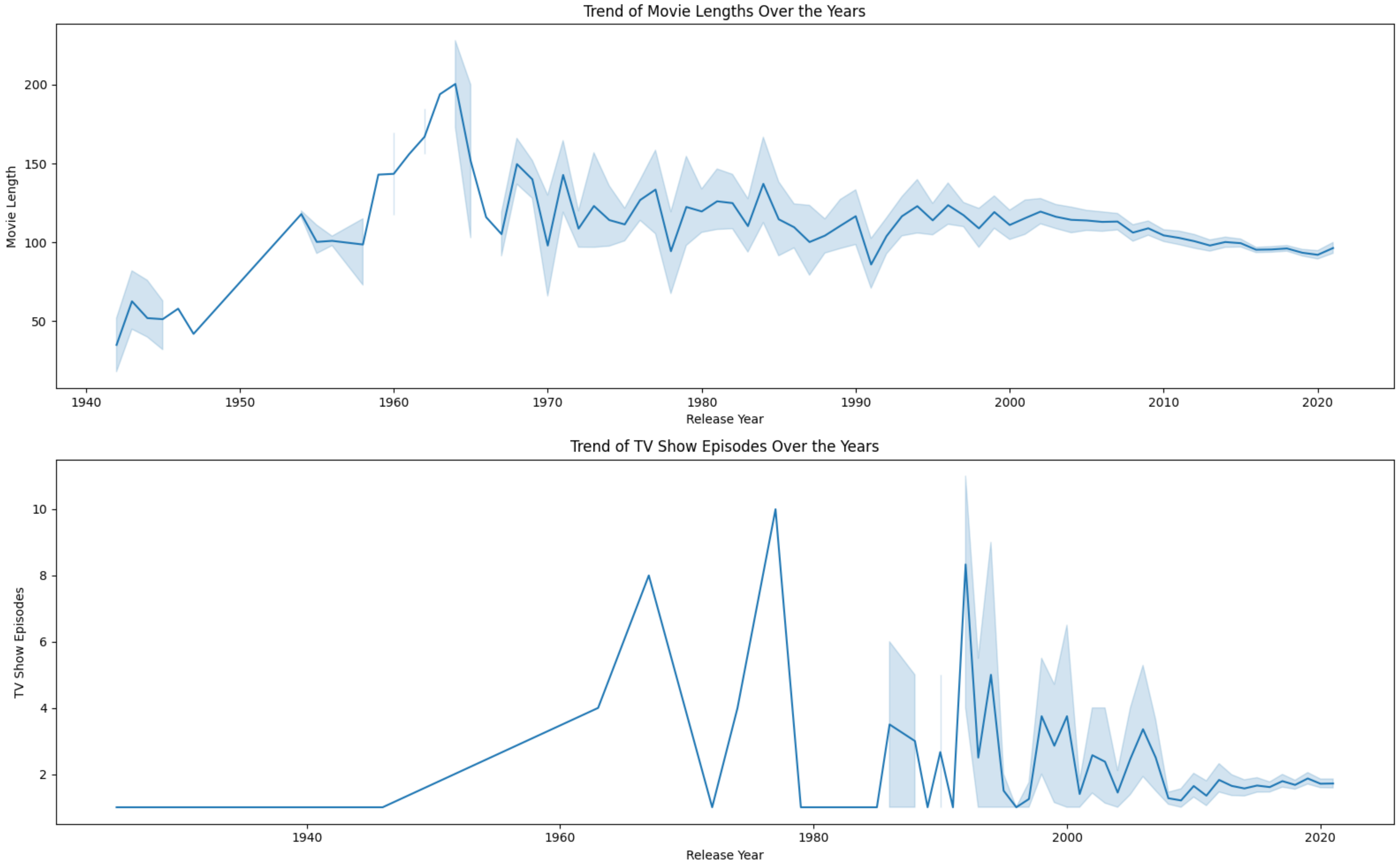
```
# TV Shows analysis
tv_show_episodes = df_tv_shows['duration'].str.extract('(\d+)', expand=False).astype(int)

plt.figure(figsize=(8, 5))
plt.hist(tv_show_episodes, bins=10, label='TV Shows')

plt.xlabel('Episode Count')
plt.ylabel('Frequency')
plt.title('Distribution of TV Show Episode Counts')
plt.legend()

plt.show()
```



In [26]:

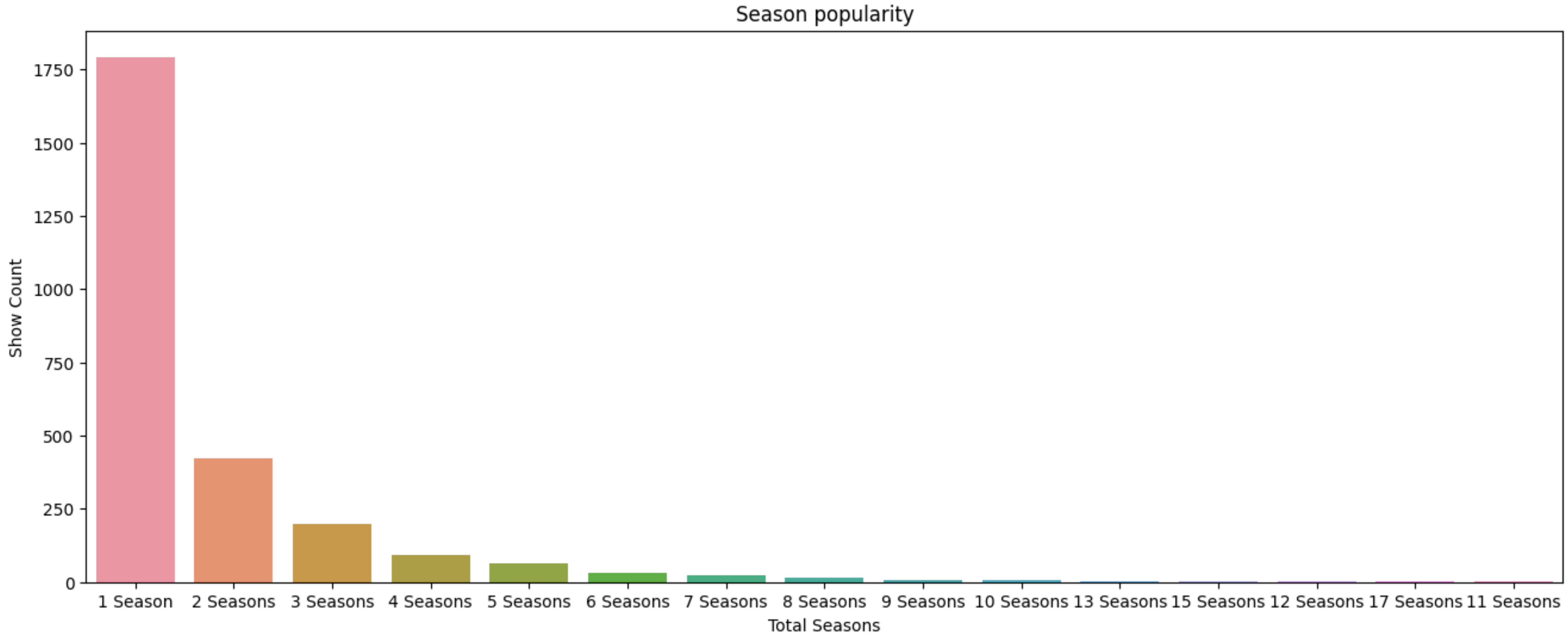
```
# Movie and TV show duration over years
movie_lengths = df_movies['duration'].str.extract('(\d+)', expand=False).astype(int)
tv_show_episodes = df_tv_shows['duration'].str.extract('(\d+)', expand=False).astype(int)

plt.figure(figsize=(16, 10))

plt.subplot(2, 1, 1)
sns.lineplot(data=df_movies, x='release_year', y=movie_lengths)
plt.xlabel('Release Year')
plt.ylabel('Movie Length')
plt.title('Trend of Movie Lengths Over the Years')

plt.subplot(2, 1, 2)
```

```python
sns.lineplot(data=df_tv_shows, x='release_year', y=tv_show_episodes)
plt.xlabel('Release Year')
plt.ylabel('TV Show Episodes')
plt.title('Trend of TV Show Episodes Over the Years')

# Adjust the layout and spacing
plt.tight_layout()

# Show the plots
plt.show()
```



In [27]:

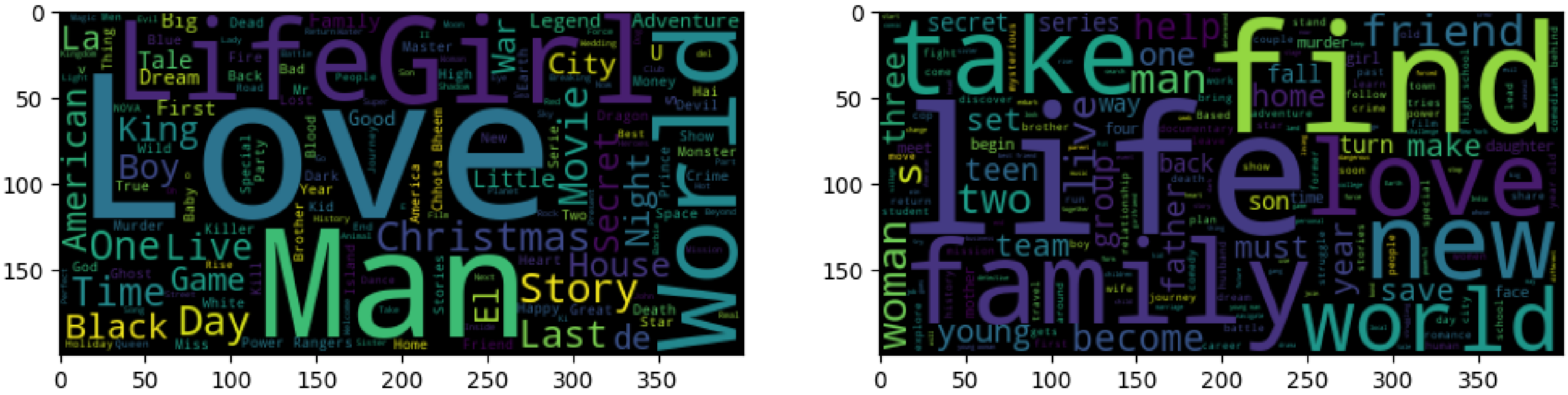
```
# No of seasons popularity

tv_show_episodes = df_tv_shows['duration'].value_counts()

plt.figure(figsize=(16,6))
sns.barplot(x=tv_show_episodes.index,y=tv_show_episodes.values)
plt.xlabel('Total Seasons')
plt.ylabel('Show Count')
plt.title('Season popularity')
plt.show()
```



In [28]:

```
# most frequent word analysis

from wordcloud import WordCloud

text = ' '.join(df['title'])

wordcloud = WordCloud().generate(text)

# plot the WordCloud image
plt.figure(figsize = (12, 12), facecolor = None)
plt.subplot(1,2,1)
plt.imshow(wordcloud)


text = ' '.join(df['description'])
```

```
wordcloud = WordCloud().generate(text)

plt.subplot(1,2,2)
plt.imshow(wordcloud)

plt.show()
```



```
In [ ]:
```

```
In [ ]:
```