Netflix Data Analysis

Netflix is an American subscription video on-demand over-the-top streaming service owned and operated by Netflix, Inc. The service primarily distributes films and television series produced by the media company of the same name from various genres, and it is available internationally in multiple languages

Importing libraries pandas, numpy and matplotlib

In [112…
```
#importing libraries pandas, numpy and matplotlib
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

Loading CSV Data

In [84]:
```
#reading netflix data and assigning it to variable called df
df = pd.read_csv("Netflix Data.csv")
```

Reading Data

In [85]:
```
#getting top 10 rows from the database
df.head(3)
```

Out[85]:

	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	25-Sep-21	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	24-Sep-21	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	24-Sep-21	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...

In [86]:
```
df.tail(3)
```

Out[86]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8806	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	1-Nov-19	2009	R	88 min	Comedies, Horror Movies	Looking to survive in a world taken over by zo...
8807	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	11-Jan-20	2006	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...
8808	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	2-Mar-19	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

In [87]:
```
#Data type of all attributes
print(df.dtypes)
```

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

In [106…]:
```
df.release_year = df.release_year.apply(pd.to_numeric)
print(df.dtypes)
```

```
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
```

In [90]:
```
df.describe()
```

Out[90]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
count	8809	8808	8807	6173	7983	7976	8797	8807	8803	8804	8806	8806
unique	8809	3	8804	4528	7693	749	1768	75	18	221	514	8774
top	s1	Movie	15-Aug	Rajiv Chilaka	David Attenborough	United States	1-Jan-20	2018	TV-MA	1 Season	Dramas, International Movies	Paranormal activity at a lush, abandoned prope...
freq	1	6131	2	19	19	2817	109	1147	3207	1793	362	4

In [91]:
```
df.nunique()
```

Out[91]:
```
show_id         8809
type               3
title           8804
director        4528
cast            7693
country          749
date_added      1768
release_year      75
rating            18
duration         221
listed_in        514
description     8774
dtype: int64
```

In [92]:
```
#getting shape of the data meand number of rows and columns
df.shape
```

Out[92]:
```
(8809, 12)
```

In [93]:
```
#getting all the column names
df.columns
```

Out[93]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

Handling Null Values

In [94]:
```
#getting all the null value count in all columns
df.isnull().sum()
```

Out[94]:
```
show_id            0
type               1
title              2
director        2636
cast             826
country          833
date_added        12
release_year       2
rating             6
duration           5
listed_in          3
description        3
dtype: int64
```

In [95]:
```
df['director'].fillna('No Director', inplace=True)
df['cast'].fillna('No Cast', inplace=True)
df['country'].fillna('Country Unavailable', inplace=True)
df.head(5)
```

Out[95]:

	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	No Cast	United States	25-Sep-21	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	No 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	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Country Unavailable	24-Sep-21	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	No Director	No Cast	Country Unavailable	24-Sep-21	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	No Director	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	24-Sep-21	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

In [96]:
```
#getting all the null value count in all columns
df.isnull().sum()
```

Out[96]:
```
show_id          0
type             1
title            2
director         0
cast             0
country          0
date_added      12
release_year     2
rating           6
duration         5
listed_in        3
description      3
dtype: int64
```

In [97]:
```
df.dropna(subset=['date_added'],inplace=True)
df.dropna(subset=['rating'],inplace=True)
df.dropna(subset=['duration'],inplace=True)
```

In [98]:
```
df.isnull().sum()
```

Out[98]:
```
show_id         0
type            0
title           1
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       1
description     1
dtype: int64
```

Data Cleansing

In [99]:
```
#checking value count in Type column
df["type"].value_counts()
```

Out[99]:
```
Movie            6125
TV Show          2664
William Wyler       1
Name: type, dtype: int64
```

In [100…
```
#checking index number of incorrect data in column
df[df["type"]== "William Wyler"]
```

Out[100]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8421	Flying Fortress"	William Wyler	NaN	United States	31-Mar-17	1944	TV-PG	40 min	Classic Movies, Documentaries	This documentary centers on the crew of the B-...	NaN	NaN

In [101…
```
#dropping row where incorrect data existed
df.drop(8421,axis = 0, inplace = True)
df.head()
```

Out[101]:

	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	No Cast	United States	25-Sep-21	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	No 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	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Country Unavailable	24-Sep-21	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	No Director	No Cast	Country Unavailable	24-Sep-21	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	No Director	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	24-Sep-21	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

In [102…

```
#checking shape of data after dropping a row where incorrect data existed
df.shape
```

Out[102]:

```
(8789, 12)
```

In [103…

```
#checking updated data in column type after dropping incorrect value
cat_counts = df["type"].value_counts()
cat_counts
```

Out[103]:

```
Movie      6125
TV Show    2664
Name: type, dtype: int64
```

In [104…

```
#checking rating column with number of value counts
df["rating"].value_counts()
```

Out[104]:
```
TV-MA       3205
TV-14       2157
TV-PG        860
R            799
PG-13        490
TV-Y7        333
TV-Y         306
PG           287
TV-G         220
NR            79
G             41
TV-Y7-FV       6
NC-17          3
UR             3
Name: rating, dtype: int64
```

In [105…
```
df.head(2)
```

Out[105]:

	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	No Cast	United States	25-Sep-21	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	No 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	After crossing paths at a party, a Cape Town t...

In [24]:
```
df["cast"] = df["cast"].str.split(",")
df["country"] = df["country"].str.split(",")
df["listed_in"] = df["listed_in"].str.split(",")
```

In [25]:
```
df.head(2)
```

Out[25]:

	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	[No Cast]	[United States]	25-Sep-21	2020	PG-13	90 min	[Documentaries]	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	No Director	[Ama Qamata, Khosi Ngema, Gail Mabalane, Th...	[South Africa]	24-Sep-21	2021	TV-MA	2 Seasons	[International TV Shows, TV Dramas, TV Myste...	After crossing paths at a party, a Cape Town t...

In [26]:
```
df_cast = df.explode("cast",ignore_index = True)
```

In [27]:
```
df_cast.head()
```

Out[27]:

	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	No Cast	[United States]	25-Sep-21	2020	PG-13	90 min	[Documentaries]	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	No Director	Ama Qamata	[South Africa]	24-Sep-21	2021	TV-MA	2 Seasons	[International TV Shows, TV Dramas, TV Myste...	After crossing paths at a party, a Cape Town t...
2	s2	TV Show	Blood & Water	No Director	Khosi Ngema	[South Africa]	24-Sep-21	2021	TV-MA	2 Seasons	[International TV Shows, TV Dramas, TV Myste...	After crossing paths at a party, a Cape Town t...
3	s2	TV Show	Blood & Water	No Director	Gail Mabalane	[South Africa]	24-Sep-21	2021	TV-MA	2 Seasons	[International TV Shows, TV Dramas, TV Myste...	After crossing paths at a party, a Cape Town t...
4	s2	TV Show	Blood & Water	No Director	Thabang Molaba	[South Africa]	24-Sep-21	2021	TV-MA	2 Seasons	[International TV Shows, TV Dramas, TV Myste...	After crossing paths at a party, a Cape Town t...

In [28]:
```
df_country = df.explode("country",ignore_index = True)
```

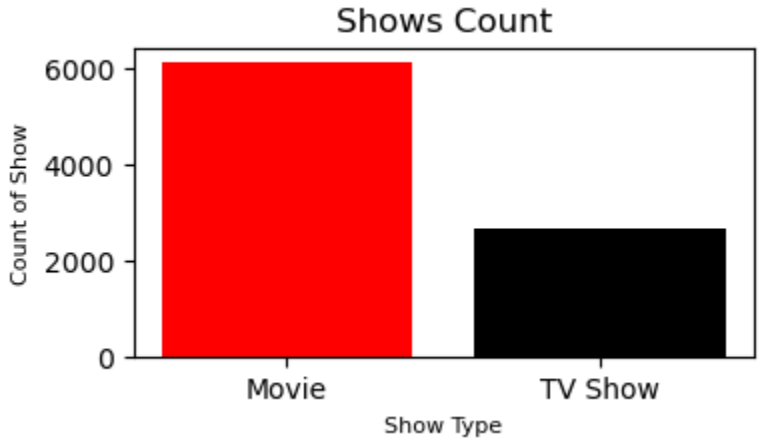
In [29]:
```
df_country.head(3)
```

Out[29]:

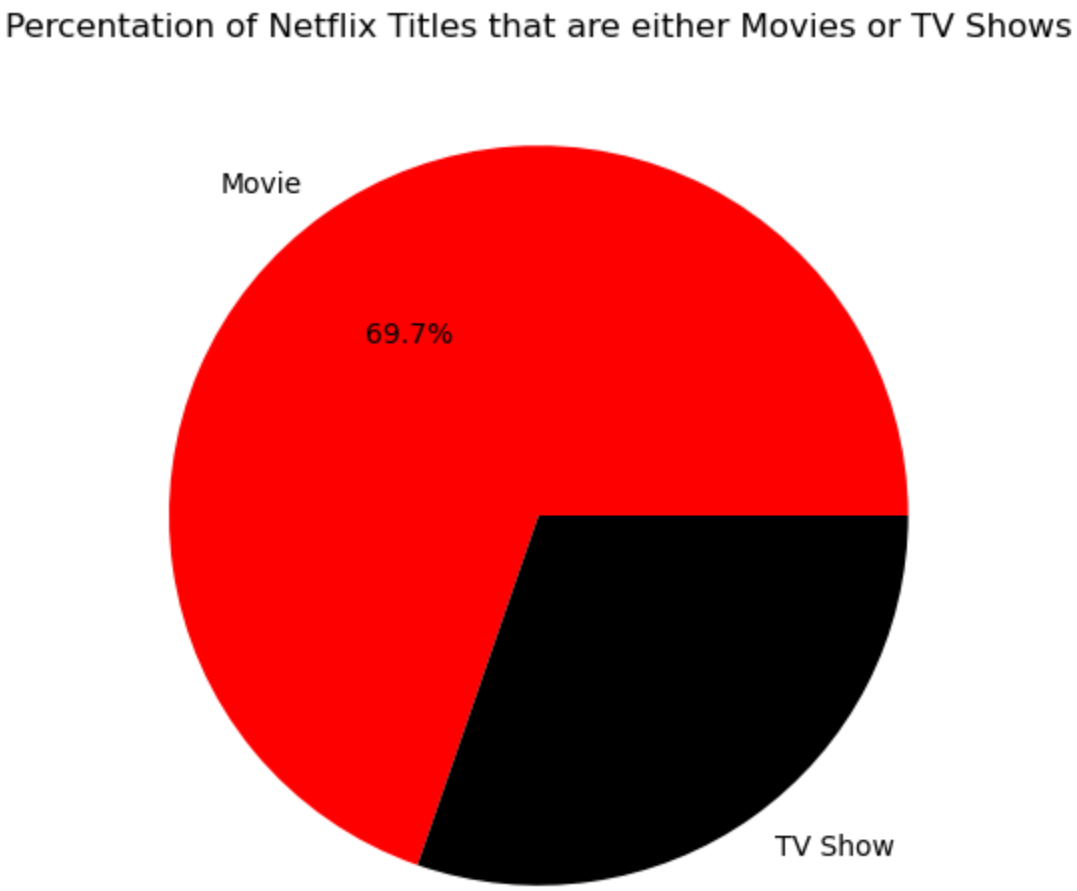
	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	[No Cast]	United States	25-Sep-21	2020	PG-13	90 min	[Documentaries]	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	No Director	[Ama Qamata, Khosi Ngema, Gail Mabalane, Th...	South Africa	24-Sep-21	2021	TV-MA	2 Seasons	[International TV Shows, TV Dramas, TV Myste...	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	[Sami Bouajila, Tracy Gotoas, Samuel Jouy, ...	Country Unavailable	24-Sep-21	2021	TV-MA	1 Season	[Crime TV Shows, International TV Shows, TV ...	To protect his family from a powerful drug lor...

Data Visualization

In [30]:
```
index = cat_counts.index
value = cat_counts.values
index
value
plt.figure(figsize = (4,2))
plt.title("Shows Count")
plt.bar(index,value,color = ["Red", "Black"])
plt.xlabel("Show Type",fontsize = 8)
plt.ylabel("Count of Show",fontsize = 8)
plt.show()
```



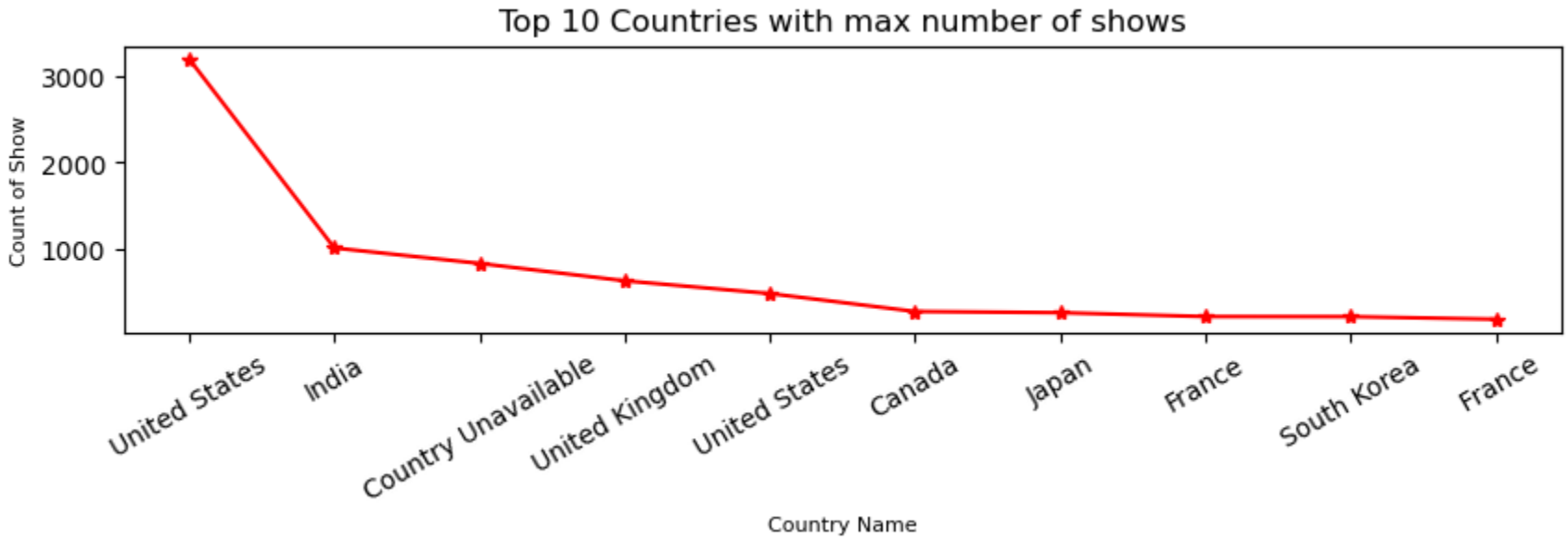
In [63]:
```
labels = df.type.value_counts().index
plt.figure(figsize=(12,6))
plt.title("Percentation of Netflix Titles that are either Movies or TV Shows")
plt.pie(df.type.value_counts(),labels = labels, colors=["red","black"],autopct="%1.1f%%")
plt.show()
```



In [64]:
```
country_count = df_country.country.value_counts()[:10]
country_count
```

Out[64]:
```
United States          3201
India                  1008
Country Unavailable     829
United Kingdom          627
 United States          479
Canada                  271
Japan                   257
France                  212
South Korea             211
 France                 181
Name: country, dtype: int64
```

In [65]:
```
index = country_count.index
value = country_count.values
plt.figure(figsize=(10,2))
plt.xticks(rotation = 30)
plt.title("Top 10 Countries with max number of shows")
plt.plot(index,value,marker = "*",color = "Red")
plt.xlabel("Country Name",fontsize = 8)
plt.ylabel("Count of Show",fontsize = 8)
plt.show()
```



In [34]:
```
df_listed_in = df.explode("listed_in",ignore_index = True)
```

In [35]:
```
df_listed_in.head()
```

Out[35]:

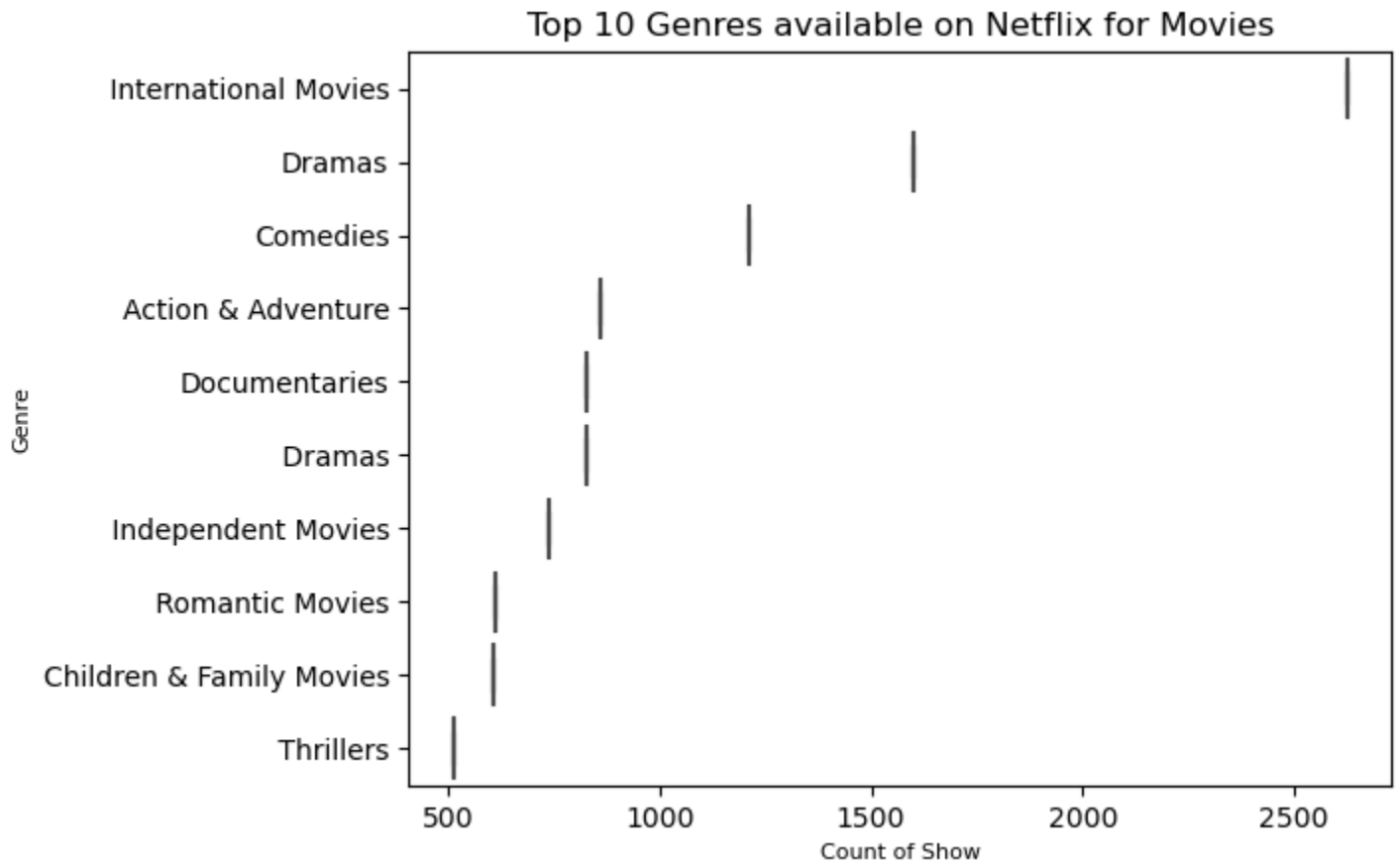
	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	[No Cast]	[United States]	25-Sep-21	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	No Director	[Ama Qamata, Khosi Ngema, Gail Mabalane, Th...	[South Africa]	24-Sep-21	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...
2	s2	TV Show	Blood & Water	No Director	[Ama Qamata, Khosi Ngema, Gail Mabalane, Th...	[South Africa]	24-Sep-21	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...
3	s2	TV Show	Blood & Water	No Director	[Ama Qamata, Khosi Ngema, Gail Mabalane, Th...	[South Africa]	24-Sep-21	2021	TV-MA	2 Seasons	TV Mysteries	After crossing paths at a party, a Cape Town t...
4	s3	TV Show	Ganglands	Julien Leclercq	[Sami Bouajila, Tracy Gotoas, Samuel Jouy, ...	[Country Unavailable]	24-Sep-21	2021	TV-MA	1 Season	Crime TV Shows	To protect his family from a powerful drug lor...

In [36]:
```
listed_in_count = df_listed_in.listed_in.value_counts()
listed_in_count
```

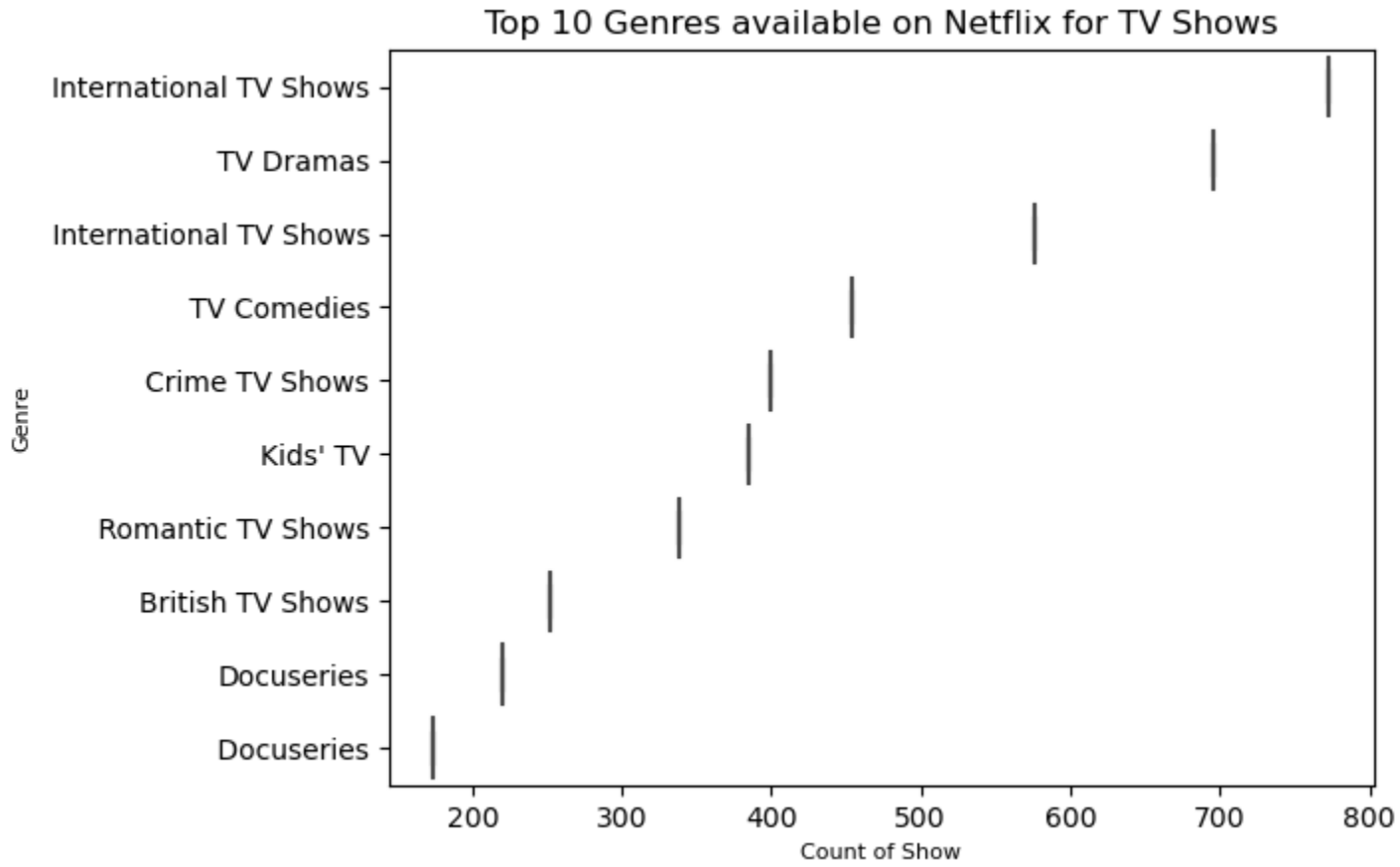
Out[36]:
```
 International Movies        2624
Dramas                       1599
Comedies                     1210
Action & Adventure            859
Documentaries                 829
                             ...
Romantic Movies                 3
Spanish-Language TV Shows       2
LGBTQ Movies                    1
TV Sci-Fi & Fantasy             1
Sports Movies                   1
Name: listed_in, Length: 73, dtype: int64
```

In [66]:
```
plt.title("Top 10 Genres available on Netflix for Movies")
plt.ylabel("Genre",fontsize = 8)
plt.xlabel("Count of Show",fontsize = 8)
sns.boxplot(data = df_listed_in,
            x = df_listed_in[df_listed_in["type"] == "Movie"]["listed_in"].value_counts().values[:10],
            y = df_listed_in[df_listed_in["type"] == "Movie"]["listed_in"].value_counts().index[:10],
```

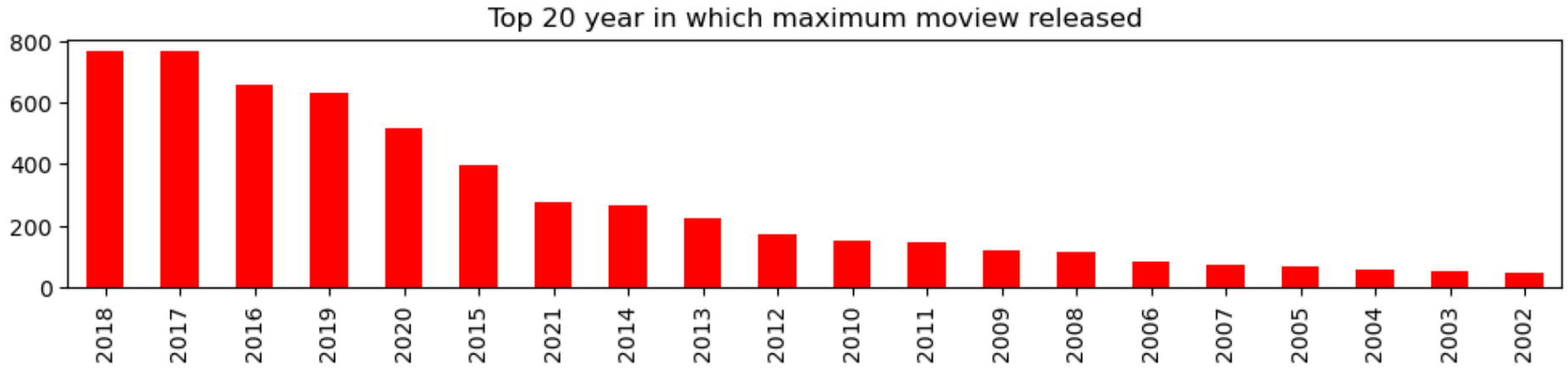
```
            color = "Red")
plt.show()
```



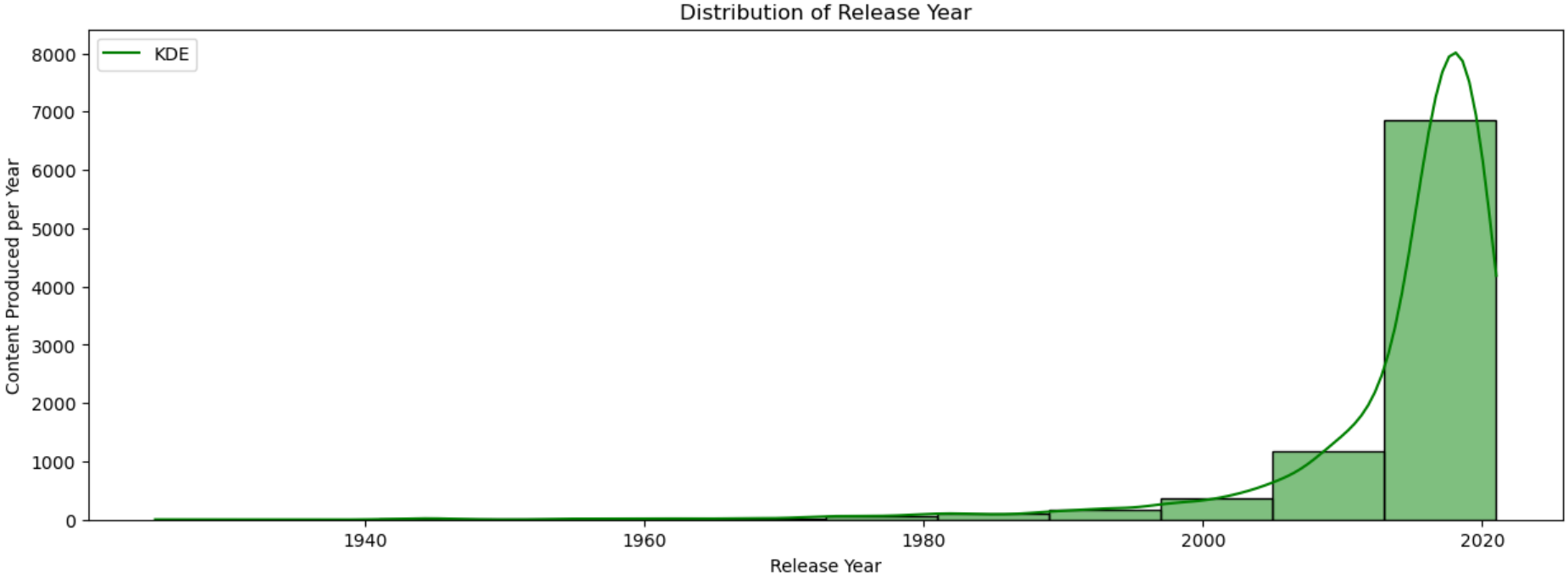
In [38]:
```
plt.title("Top 10 Genres available on Netflix for TV Shows")
plt.ylabel("Genre",fontsize = 8)
plt.xlabel("Count of Show",fontsize = 8)
sns.boxplot(data = df_listed_in,
            x = df_listed_in[df_listed_in["type"] == "TV Show"]["listed_in"].value_counts().values[:10],
            y = df_listed_in[df_listed_in["type"] == "TV Show"]["listed_in"].value_counts().index[:10],
            color = "Red")
plt.show()
```

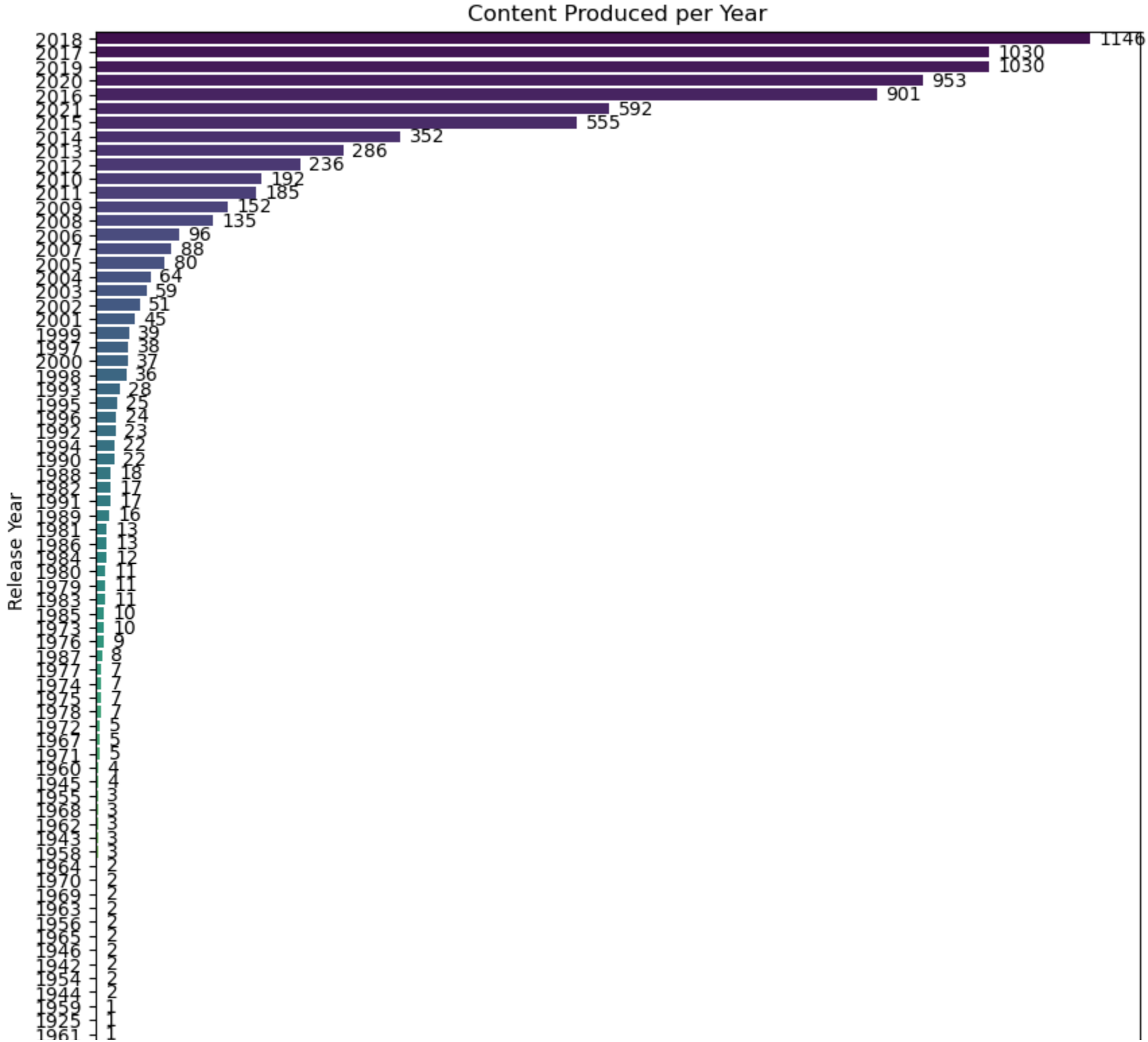


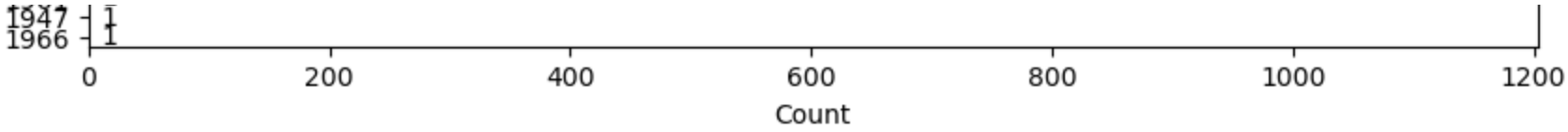
In [67]:
```
plt.figure(figsize = (12,2))
df[df["type"] == "Movie"]["release_year"].value_counts()[:20].plot(kind = "bar",color = "Red")
plt.title("Top 20 year in which maximum moview released")
plt.show()
```



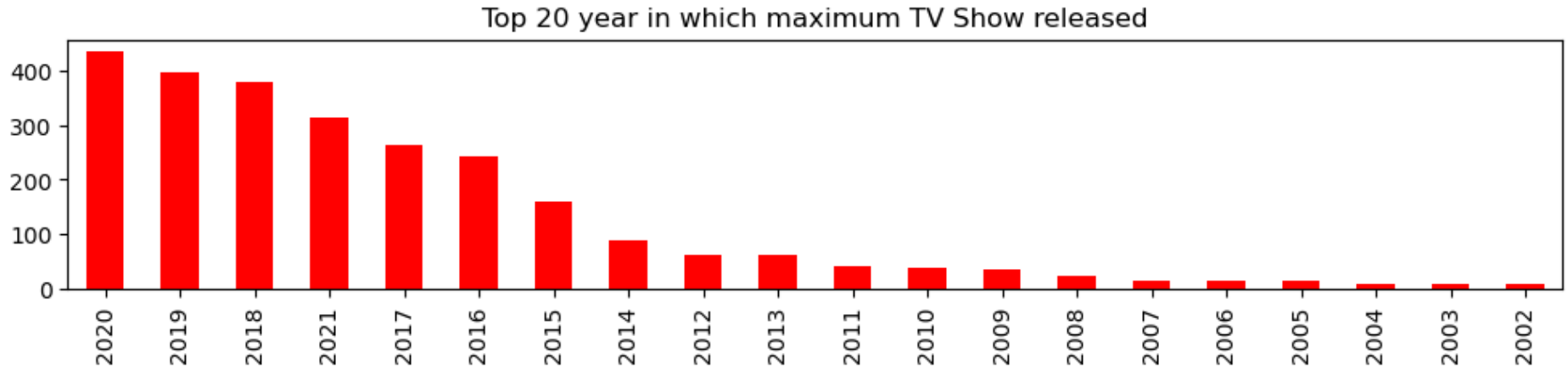
In [70]:
```python
# Distplot for 'release_year'
plt.figure(figsize=(15, 5))
ax = sns.histplot(data =df, x = "release_year", bins = 12, kde=True, color="g")
plt.title('Distribution of Release Year')
plt.xlabel('Release Year')
plt.ylabel('Content Produced per Year')
plt.legend(labels=['KDE'])
plt.show()
# Countplot for 'release_year'
plt.figure(figsize=(10, 10))
ax = sns.countplot(y='release_year', data=df, order=df['release_year'].value_counts().index, palette="viridis")
plt.title('Content Produced per Year')
plt.xlabel('Count')
plt.ylabel('Release Year')
# Adding labels to the bars
for p in ax.patches:
 width = p.get_width()
 plt.text(width + 10, p.get_y() + p.get_height()/2, int(width), va='center')
plt.show()
```


Distribution of Release Year
KDE
Content Produced per Year
Release Year
0
1000
2000
3000
4000
5000
6000
7000
8000
1940
1960
1980
2000
2020

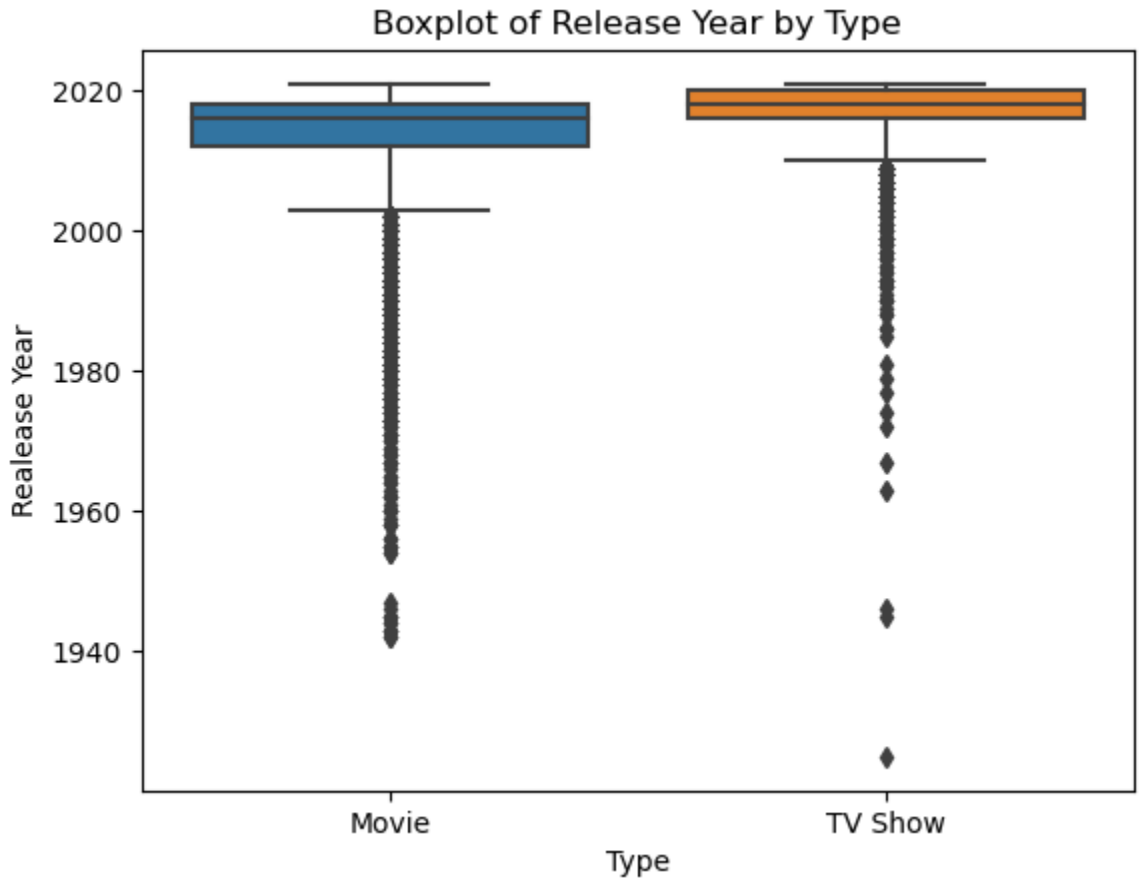

Content Produced per Year
Release Year
2018 1146
2017 1030
2019 1030
2020 953
2016 901
2021 592
2015 555
2014 352
2013 286
2012 236
2010 192
2011 185
2009 152
2008 135
2006 96
2007 88
2005 80
2004 64
2003 59
2002 51
2001 45
1999 39
1997 38
2000 37
1998 36
1993 28
1995 25
1996 24
1992 23
1994 22
1990 22
1988 18
1982 17
1991 17
1989 16
1981 13
1986 13
1984 12
1980 11
1979 11
1983 11
1985 10
1973 10
1976 9
1987 8
1977 7
1974 7
1975 7
1978 7
1972 5
1967 5
1971 5
1960 4
1945 4
1955 3
1968 3
1962 3
1943 3
1958 3
1964 2
1970 2
1969 2
1963 2
1956 2
1965 2
1946 2
1942 2
1954 2
1944 2
1959 1
1925 1
1961 1



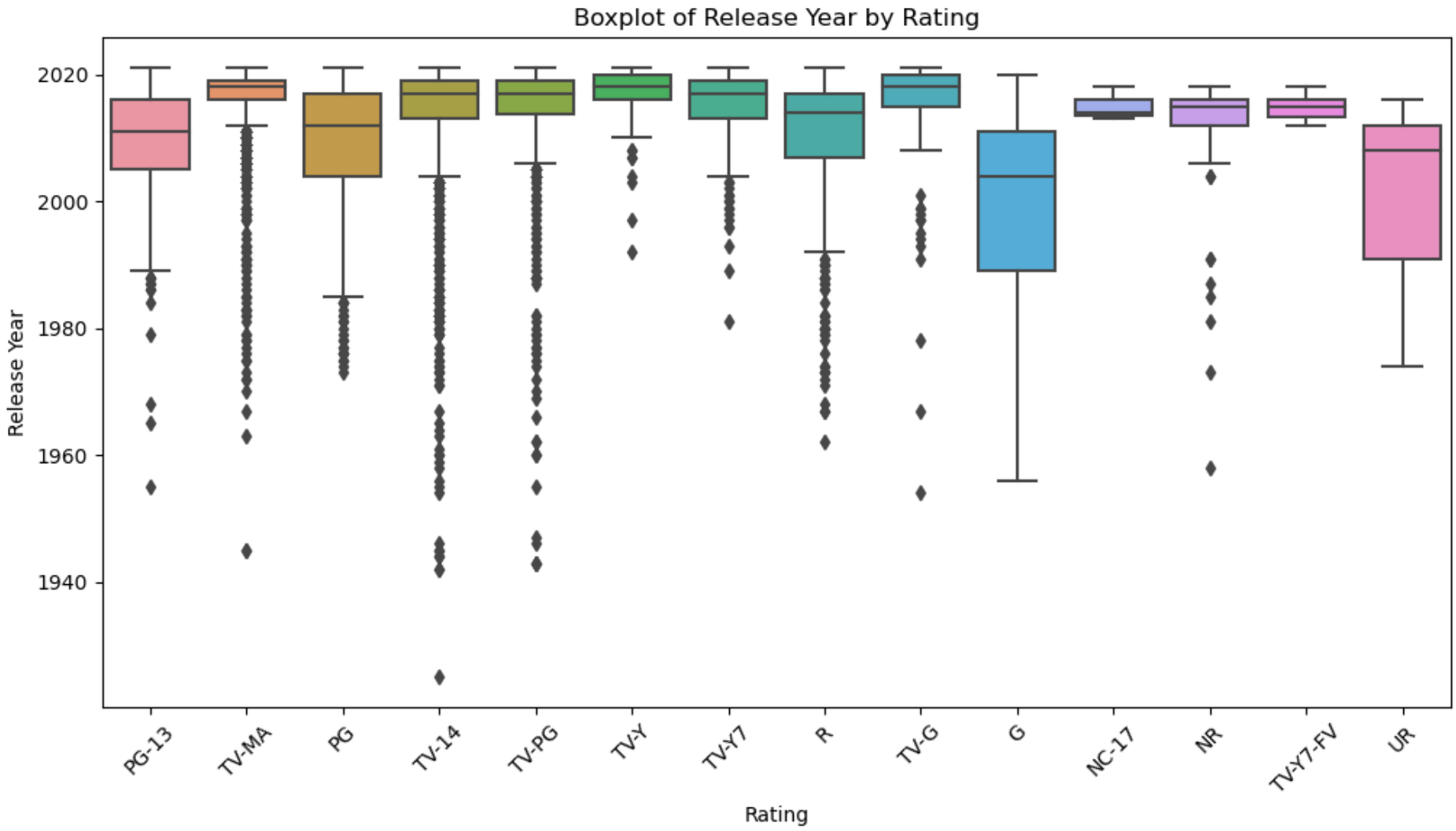
In [71]:
```
plt.figure(figsize = (12,2))
df[df["type"] == "TV Show"]["release_year"].value_counts()[:20].plot(kind = "bar",color = "Red")
plt.title("Top 20 year in which maximum TV Show released")
plt.show()
```



In [73]:
```
sns.boxplot(x='type', y='release_year', data=df)
plt.title('Boxplot of Release Year by Type')
plt.xlabel('Type')
plt.ylabel('Realease Year')
plt.show()
```



In [72]:
```
plt.figure(figsize=(12, 6))
sns.boxplot(x='rating', y='release_year', data=df)
plt.title('Boxplot of Release Year by Rating')
plt.xlabel('Rating')
plt.ylabel('Release Year')
plt.xticks(rotation=45)
plt.show()
```

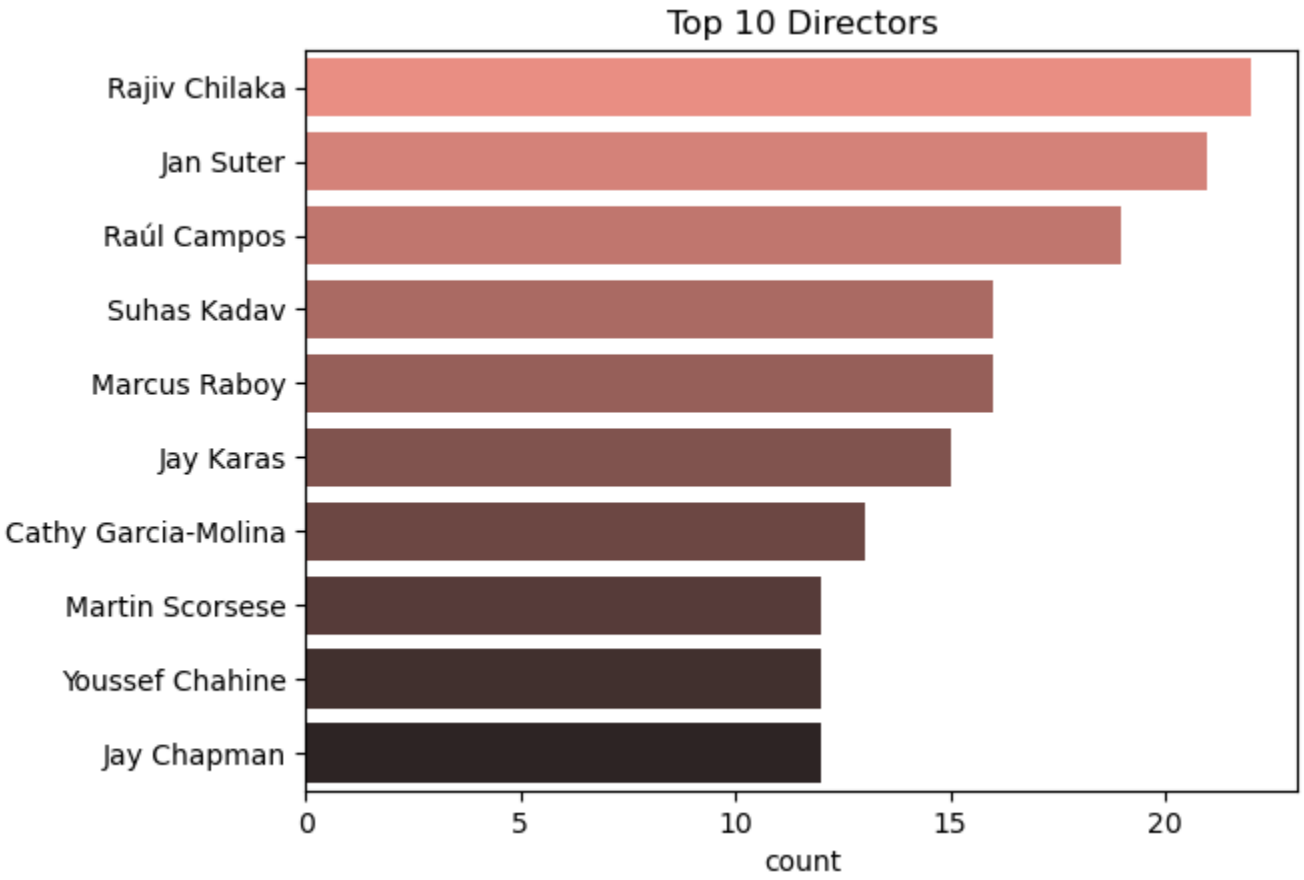


In [403…

```
df.rating.value_counts()
```

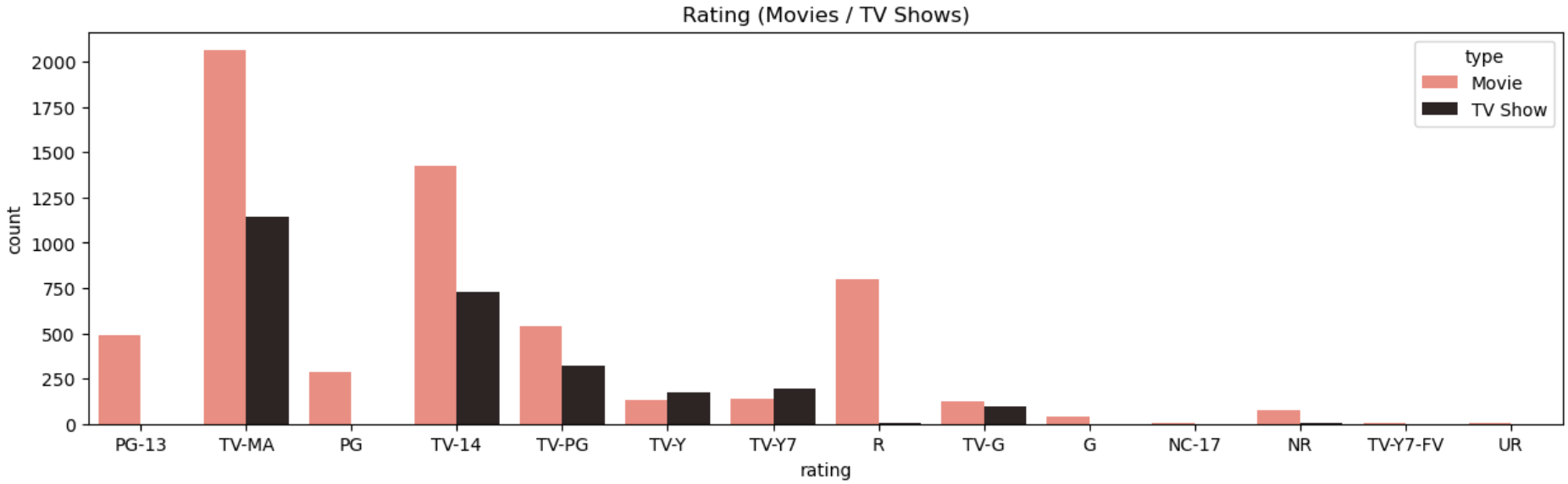
Out[403]:
```
TV-MA       3205
TV-14       2157
TV-PG        860
R            799
PG-13        490
TV-Y7        333
TV-Y         306
PG           287
TV-G         220
NR            79
G             41
TV-Y7-FV       6
NC-17          3
UR             3
Name: rating, dtype: int64
```

In [404…
```
filtered_directors = df[df.director != 'No Director'].director.str.split(', ', expand=True).stack()
sns.countplot(y = filtered_directors, order=filtered_directors.value_counts().index[:10], palette='dark:salmon_r')
plt.title("Top 10 Directors")
plt.show()
```



In [405…

```
plt.figure(figsize=(15,4))
sns.countplot(x='rating',hue='type',data=df, palette='dark:salmon_r' )
plt.title('Rating (Movies / TV Shows)')
plt.show()
```



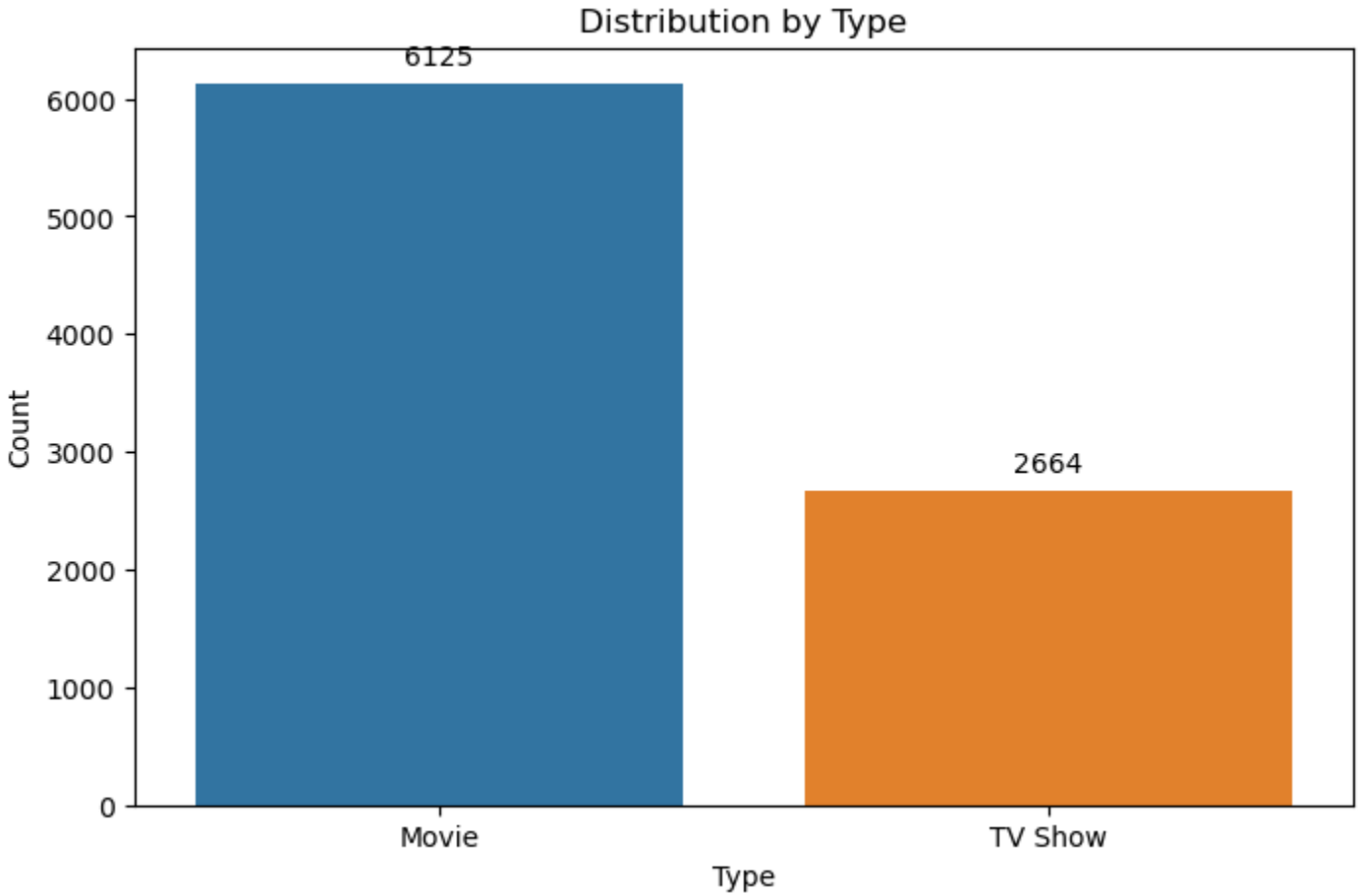
In [406…
```
#Which individual country has the highest no: of TV Shows?
df[df['type']=='TV Show']['country'].value_counts().head(1)
```

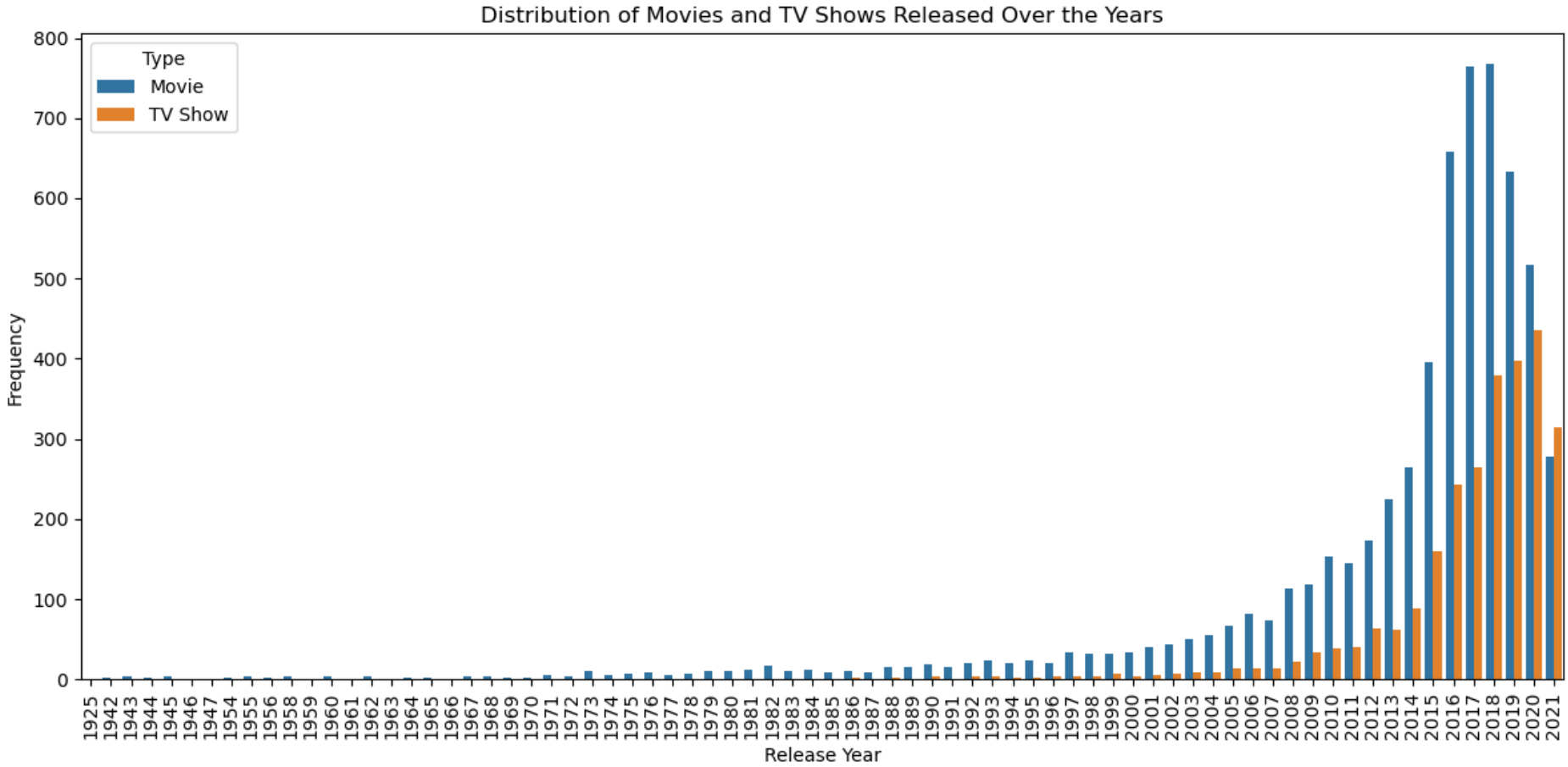
Out[406]:
```
[United States]    754
Name: country, dtype: int64
```

In [77]:
```
plt.figure(figsize=(8, 5))
ax = sns.countplot(data=df, x='type')
plt.title('Distribution by Type')
plt.xlabel('Type')
plt.ylabel('Count')
for p in ax.patches:
 ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', xytext=(0, 10), textcoords='offset points')
plt.show()

#Distribution of TV Show and Movie releases over the Years

plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='release_year', hue='type')
plt.title('Distribution of Movies and TV Shows Released Over the Years')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.legend(title='Type')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
plt.figure(figsize=(12, 6))
```

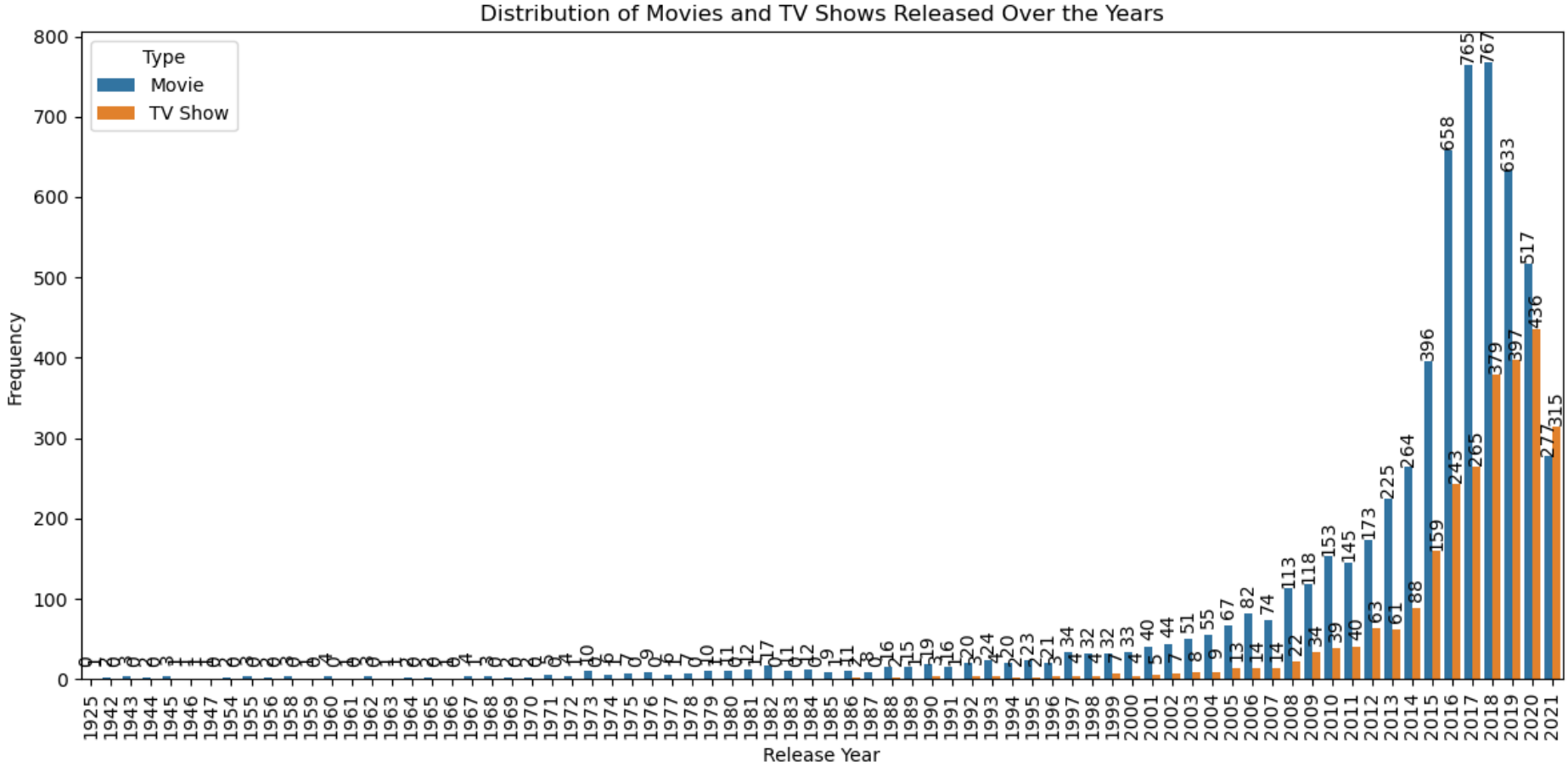
```
#Distribution of TV Show and Movie releases over the Years with labels

plt.figure(figsize=(12, 6))
ax = sns.countplot(data=df, x='release_year', hue='type')
plt.title('Distribution of Movies and TV Shows Released Over the Years')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.legend(title='Type')
plt.xticks(rotation=90)
for p in ax.patches:
 ax.annotate(f'{int(p.get_height())}',
 (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center',
 va='center',
 xytext=(0, 10),
 textcoords='offset points',
 rotation=90)
plt.tight_layout()
plt.show()

# Distribution of content ratings

ax = sns.countplot(y='rating', data=df, order=df['rating'].value_counts().index)
plt.xlabel('Count')
plt.ylabel('Rating')
for p in ax.patches:
 width = p.get_width()
 plt.text(width + 10, p.get_y() + p.get_height() / 2, int(width), ha="center")
plt.show()


# Trend of content production over the years

yearly_production = df['release_year'].value_counts().sort_index()
plt.plot(yearly_production.index, yearly_production.values)
plt.xlabel('Year')
plt.ylabel('Number of Shows')
plt.show()

# Distribution of content across different genres

plt.figure(figsize=(10, 12))
plt.title("Top 10 Genres available on Netflix for Movies")
plt.ylabel("Genre",fontsize = 8)
plt.xlabel("Count of Show",fontsize = 8)
sns.boxplot(data = df_listed_in,
```

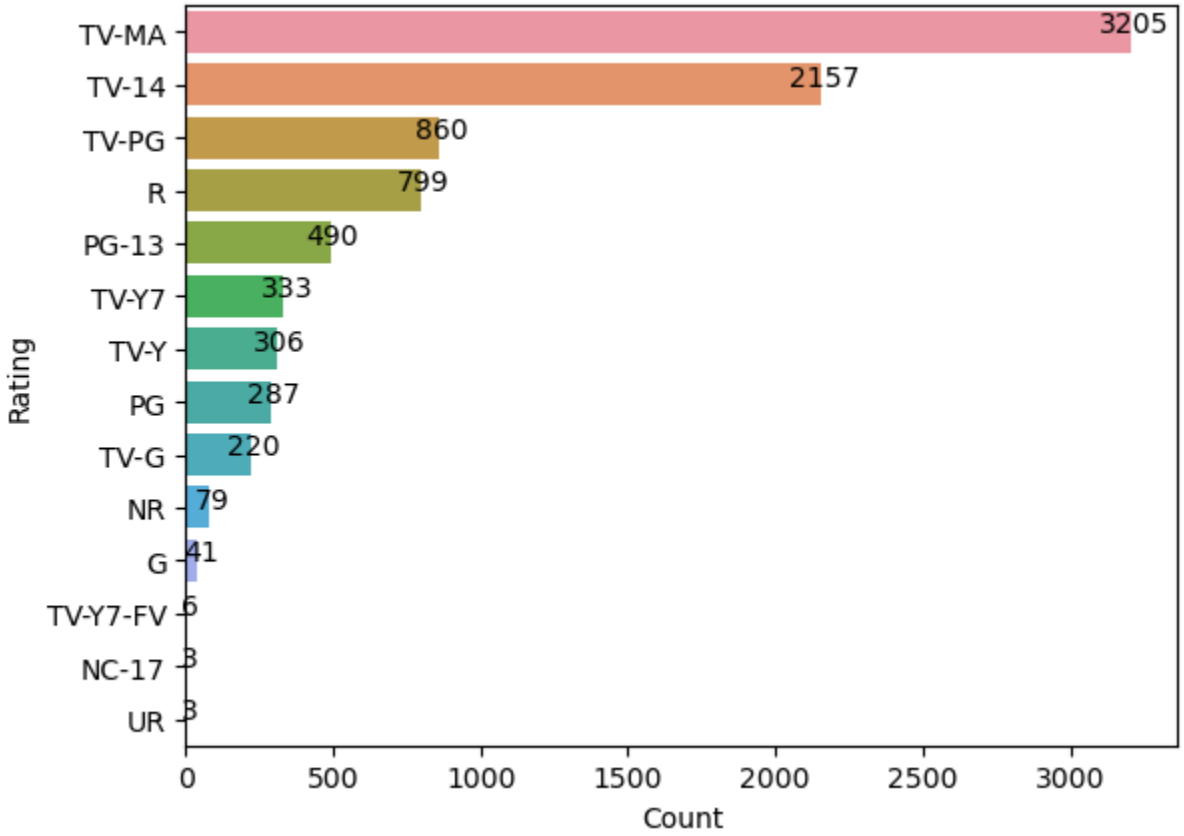
```
            x = df_listed_in[df_listed_in["type"] == "Movie"]["listed_in"].value_counts().values[:10],
            y = df_listed_in[df_listed_in["type"] == "Movie"]["listed_in"].value_counts().index[:10],
            color = "Red")
plt.show()

# DIstribution of content type

df_count = df.groupby(['release_year', 'type']).size().reset_index(name='count')
plt.figure(figsize=(10, 6))
ax = sns.lineplot(x='release_year', y='count', hue='type', data=df_count)
plt.title('Content Produced per Year')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.xticks(rotation=45)
ax.legend(title='Type')
plt.show()

# Heatmap for 'type' and 'rating'

cross_tab = pd.crosstab(df['type'], df['rating'])
plt.figure(figsize=(10, 5))
sns.heatmap(cross_tab, annot=True, cmap='viridis', fmt='d')
plt.title('Heatmap of Type vs Rating')
plt.xlabel('Rating')
plt.ylabel('Type')
plt.show()
```

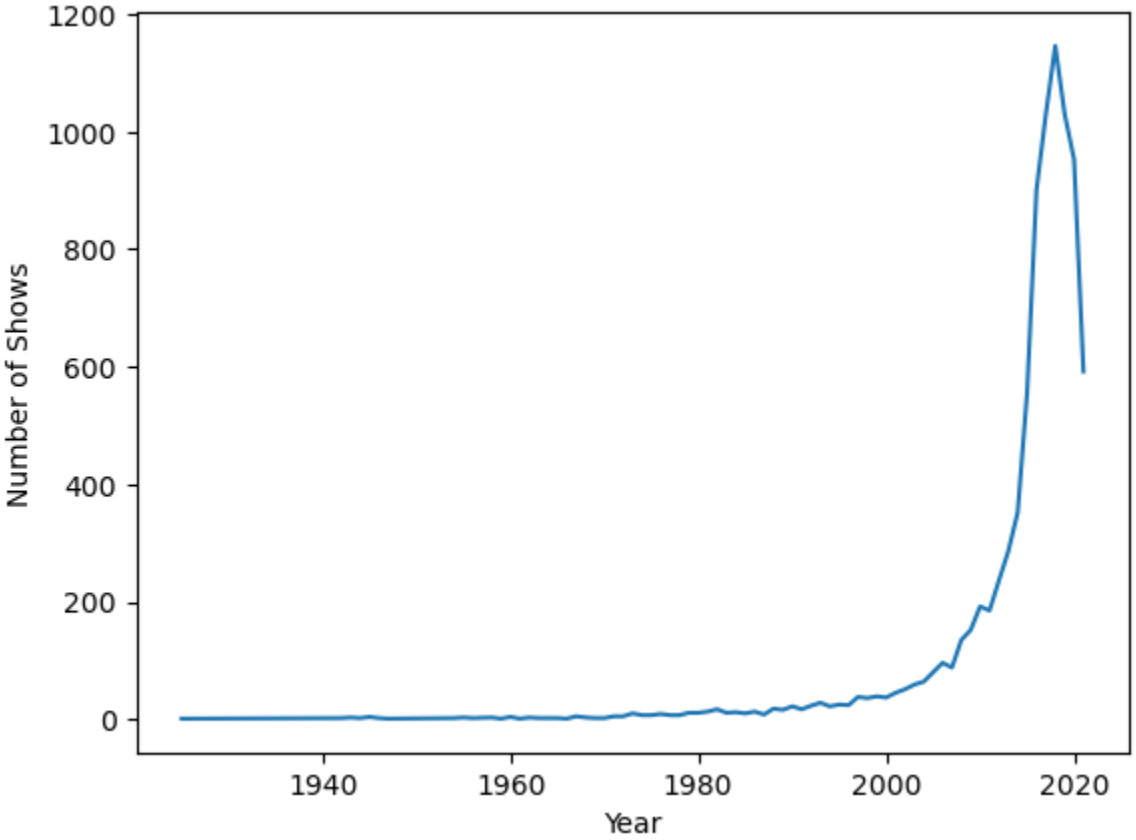

Distribution by Type
6125
2664
6000
5000
4000
3000
2000
1000
0
Count
Movie
TV Show
Type

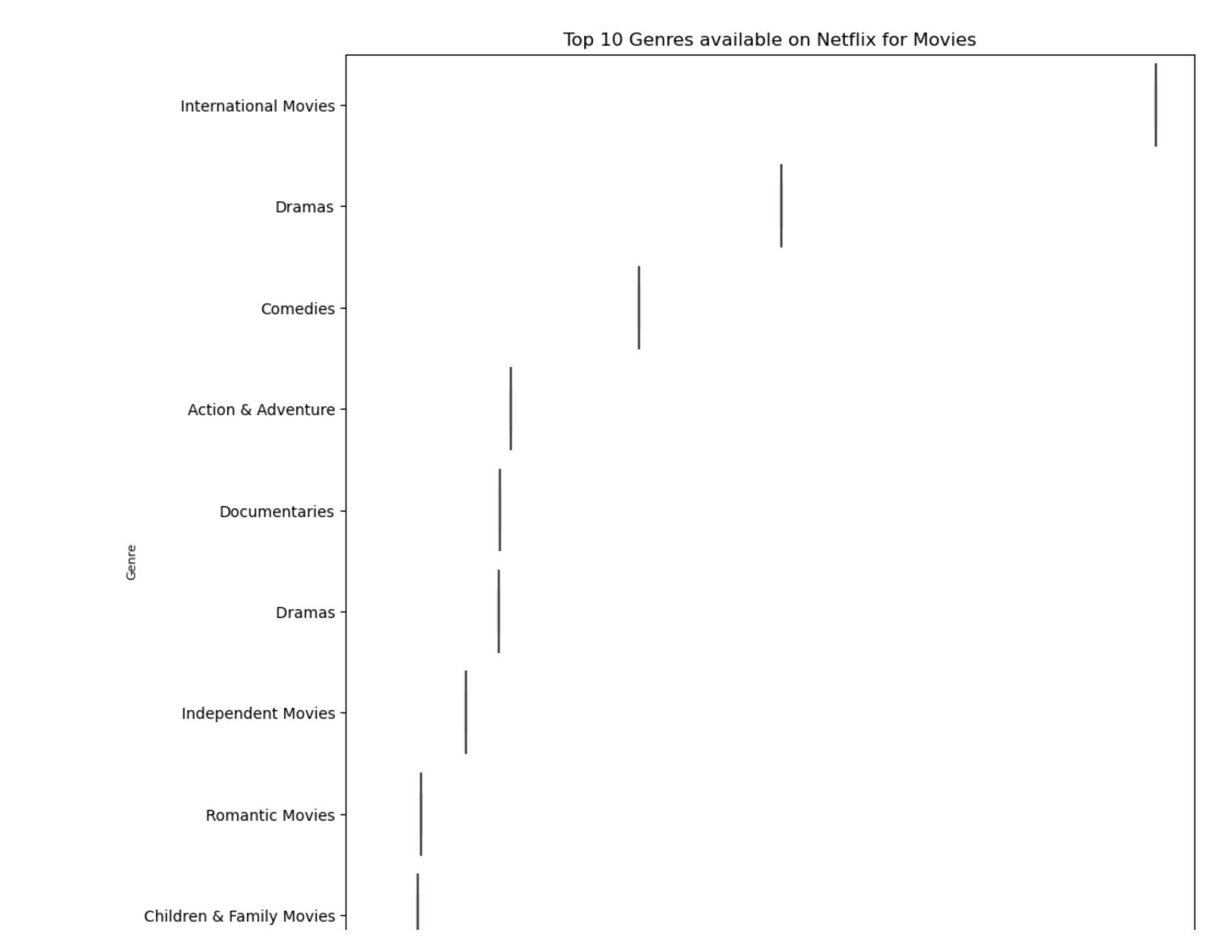


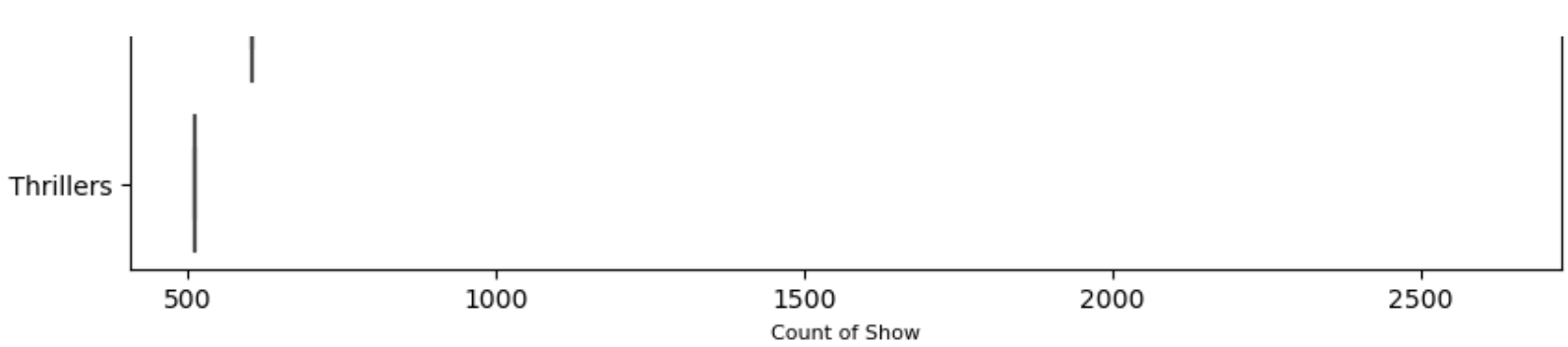
<Figure size 1200x600 with 0 Axes>

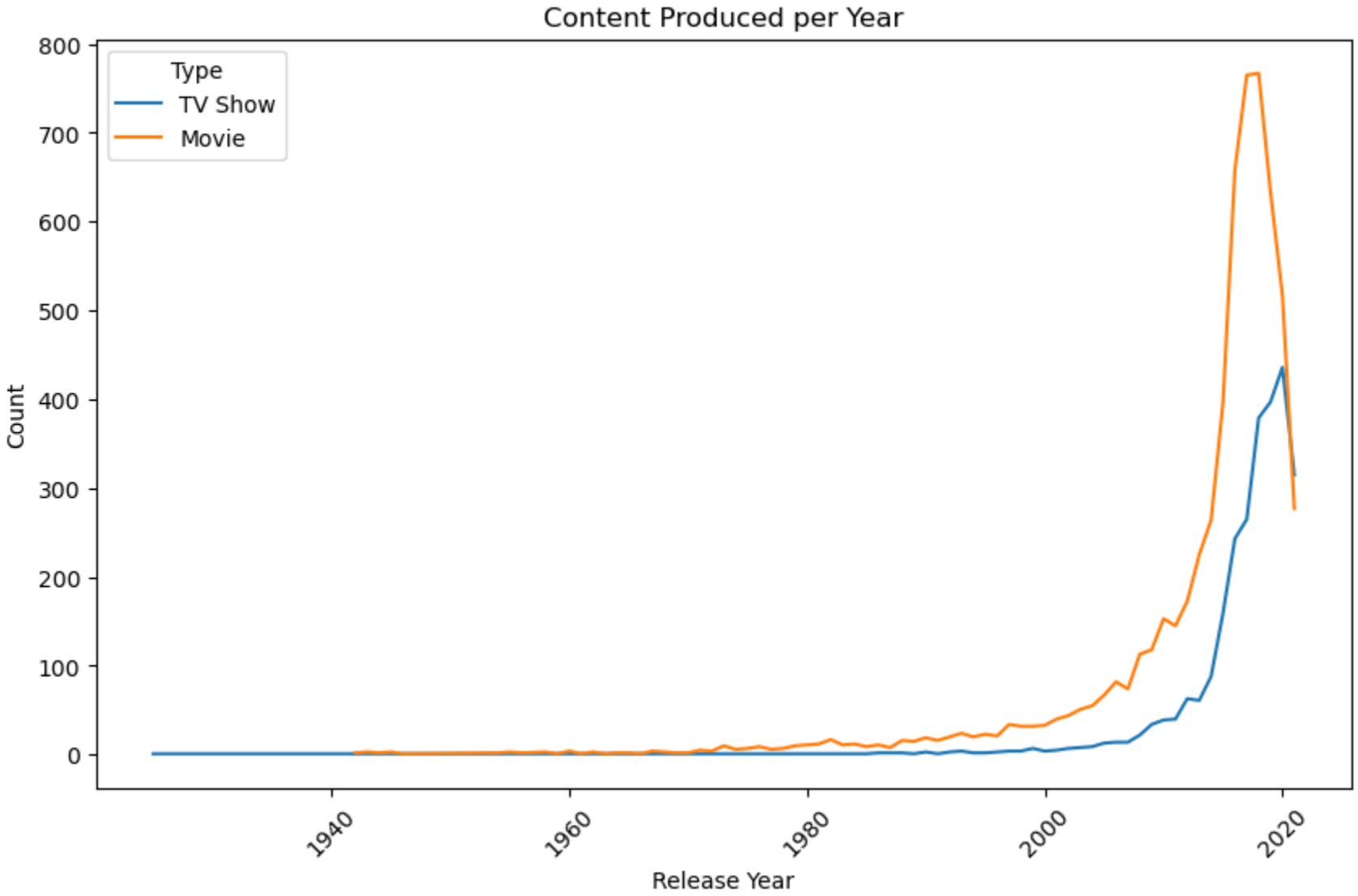

Distribution of Movies and TV Shows Released Over the Years
Type
Movie
TV Show
Frequency
Release Year
0
100
200
300
400
500
600
700
800
1925
1942
1943
1944
1945
1946
1947
1954
1955
1956
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

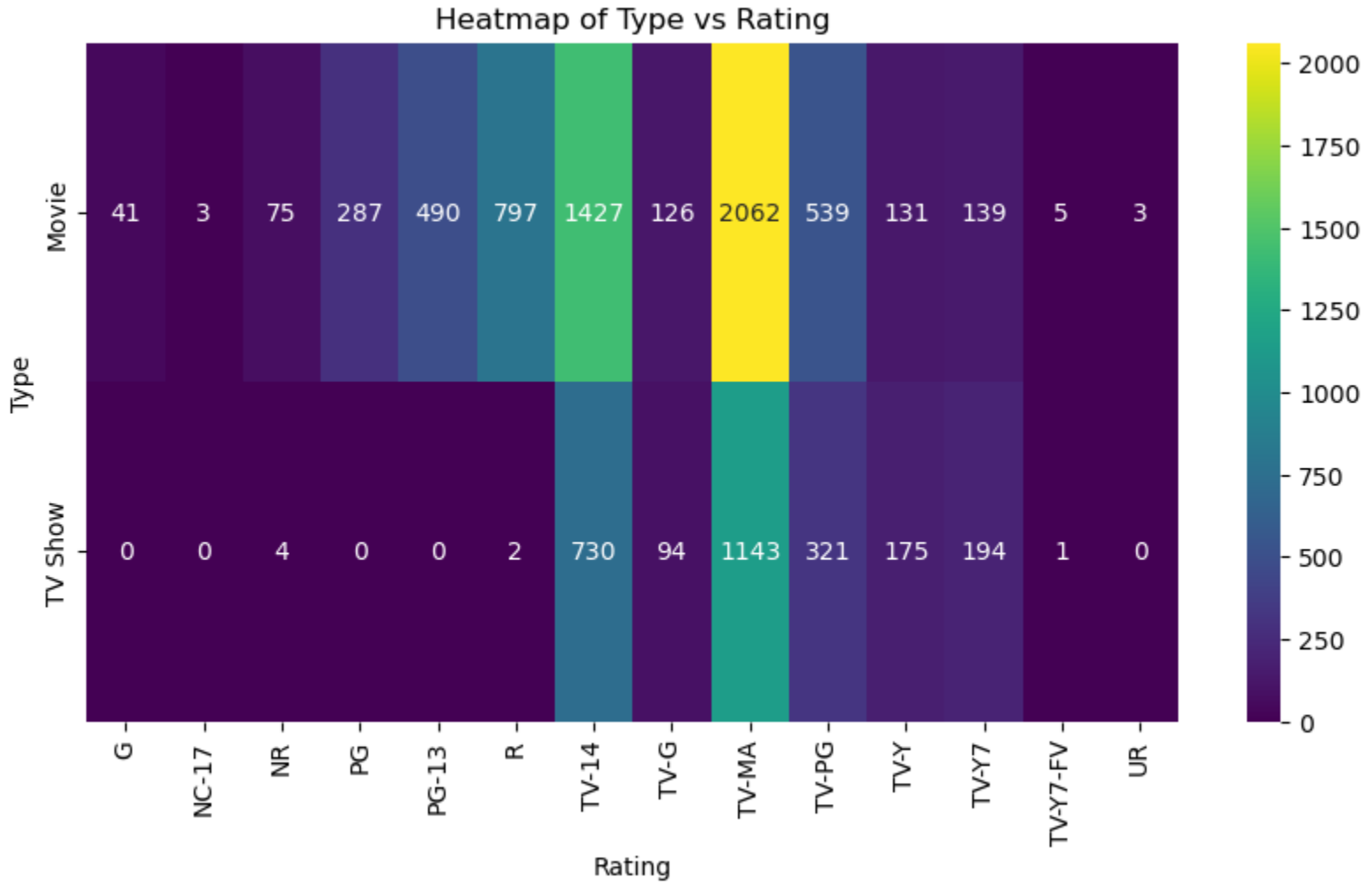

TV-MA
TV-14
TV-PG
R
PG-13
TV-Y7
TV-Y
PG
TV-G
NR
G
TV-Y7-FV
NC-17
UR
3205
2157
860
799
490
333
306
287
220
79
41
6
3
3
Rating
0
500
1000
1500
2000
2500
3000
Count


1200
1000
800
600
400
200
0
Number of Shows
1940
1960
1980
2000
2020
Year

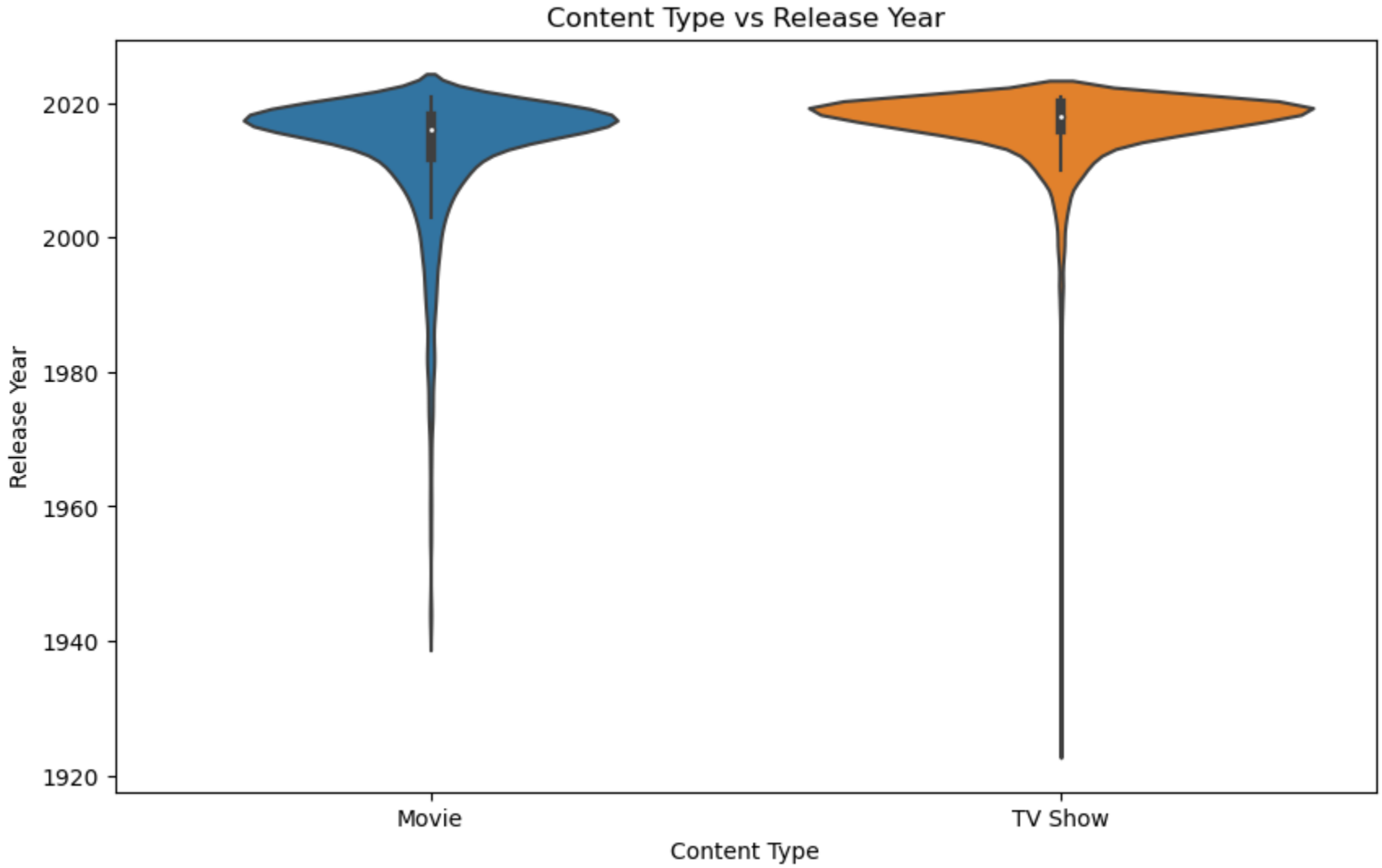

Top 10 Genres available on Netflix for Movies
International Movies
Dramas
Comedies
Action & Adventure
Documentaries
Dramas
Independent Movies
Romantic Movies
Children & Family Movies
Genre


Thrillers
500
1000
1500
2000
2500
Count of Show


Content Produced per Year
Type
TV Show
Movie
800
700
600
500
400
300
200
100
0
Count
1940
1960
1980
2000
2020
Release Year



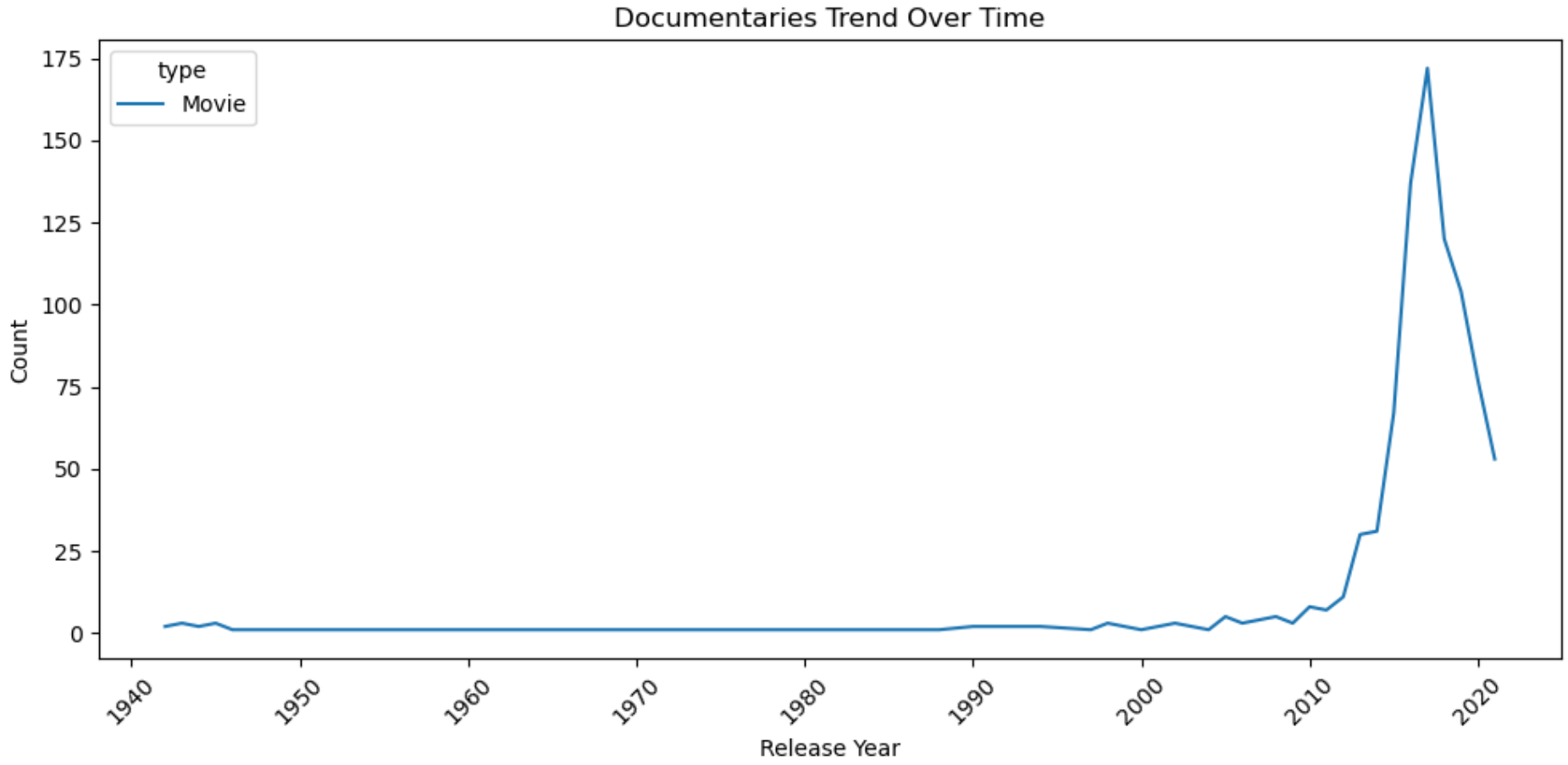
In [78]:
```
plt.figure(figsize=(10, 6))
sns.violinplot(x='type', y='release_year', data=df)
plt.title('Content Type vs Release Year')
plt.xlabel('Content Type')
plt.ylabel('Release Year')
plt.show()
```

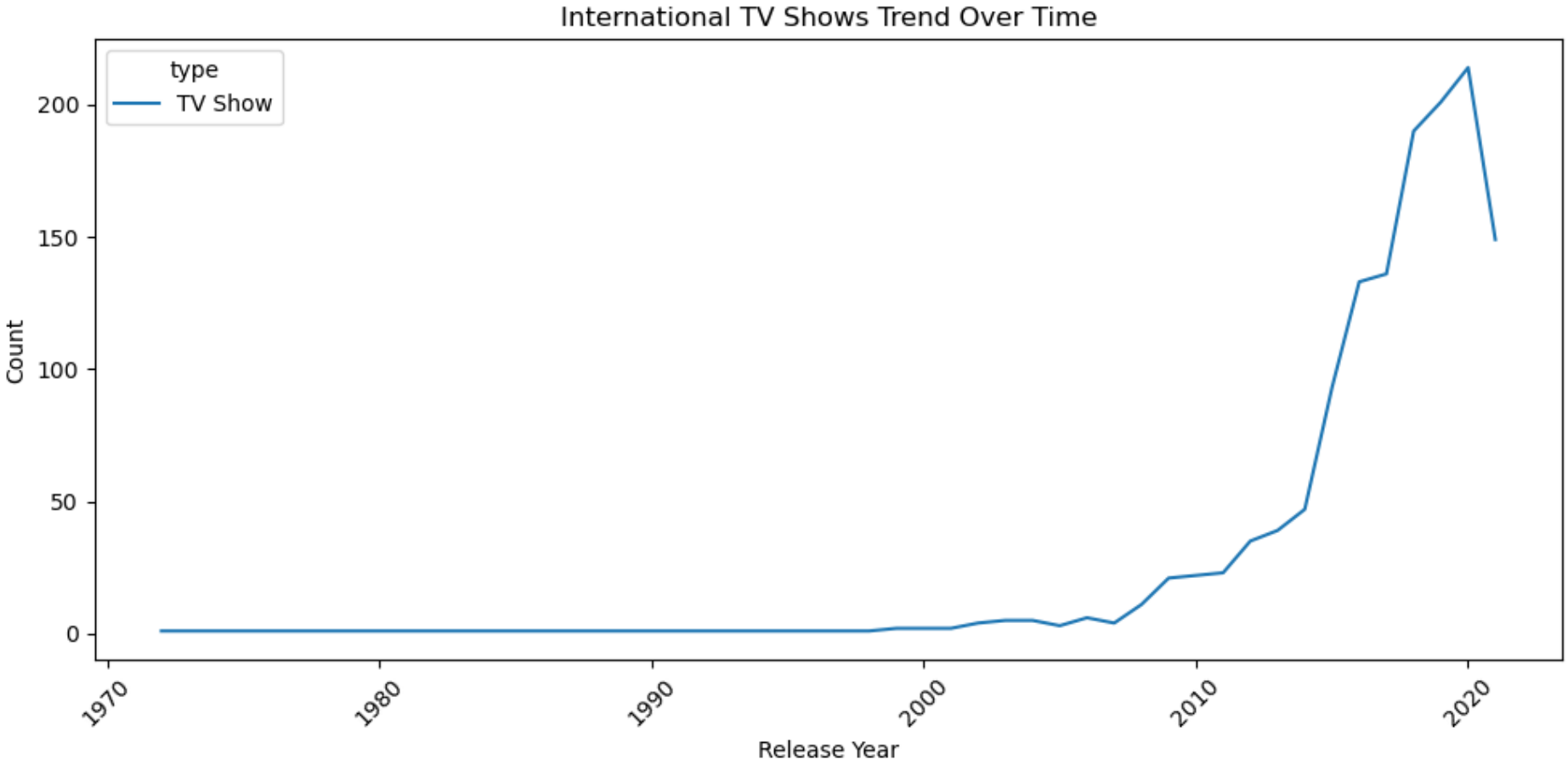


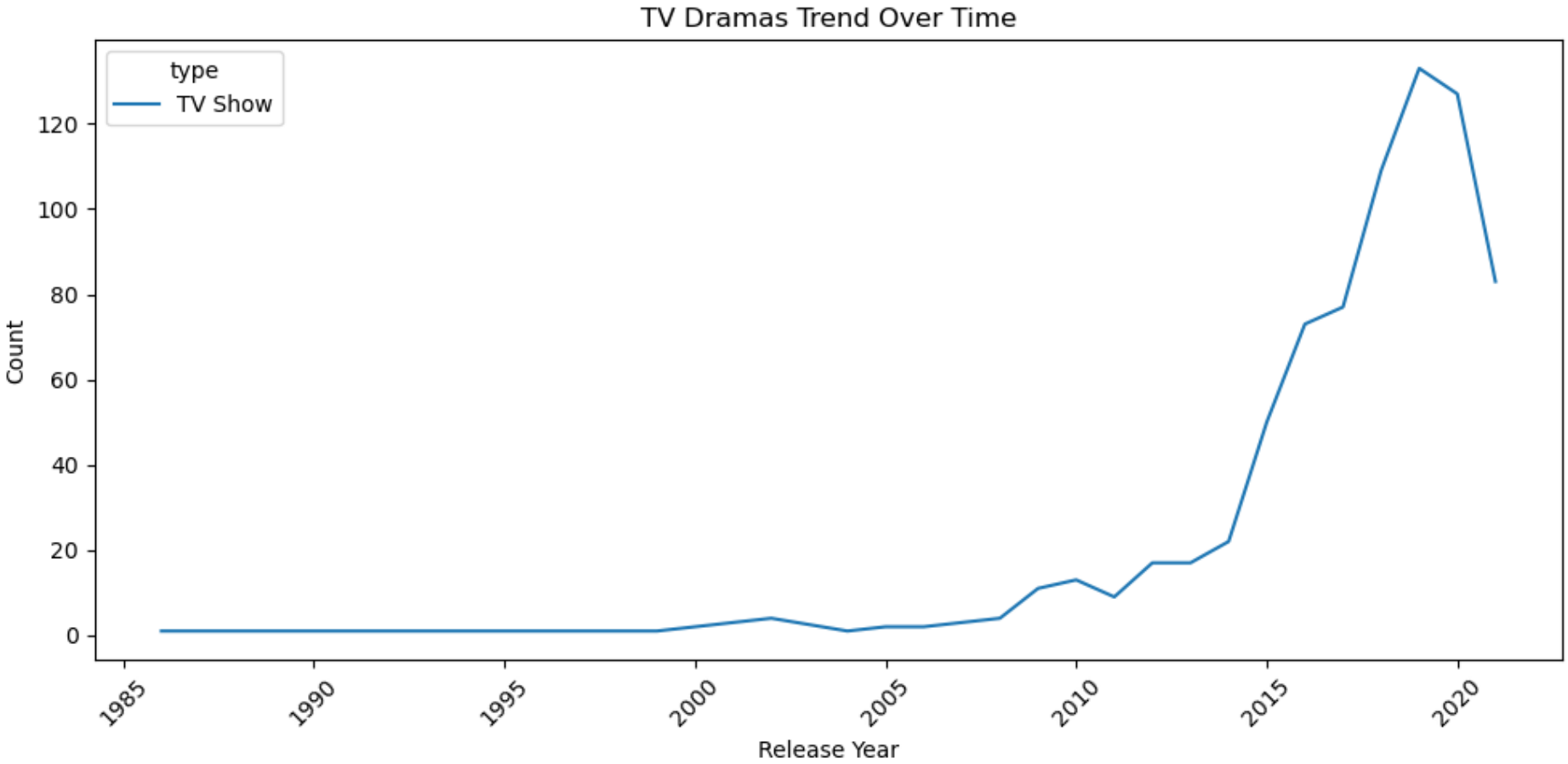
In [107…

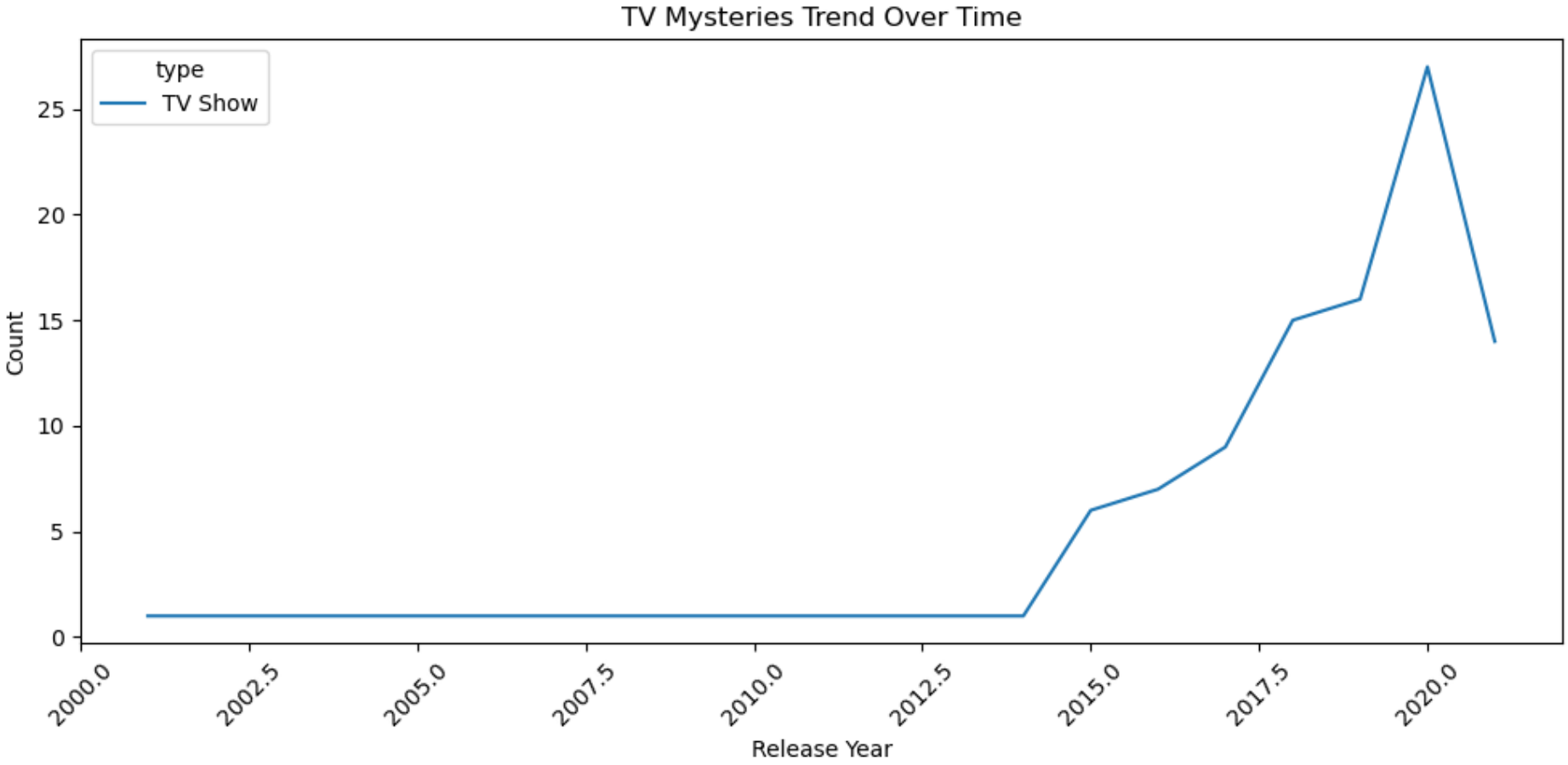
```
# Trend of Individual Genre over Time
df_exploded = df.assign(listed_in=df['listed_in'].str.split(', ')).explode('listed_in')
df_grouped = df_exploded.groupby(['release_year', 'listed_in', 'type']).size().reset_index(name='count')
genres = df_exploded['listed_in'].unique()
for genre in genres:
 plt.figure(figsize=(10, 5))

 data = df_grouped[df_grouped['listed_in'] == genre]
 sns.lineplot(x='release_year', y='count', hue='type', data=data)

 plt.title(f'{genre} Trend Over Time')
 plt.xlabel('Release Year')
 plt.ylabel('Count')
```

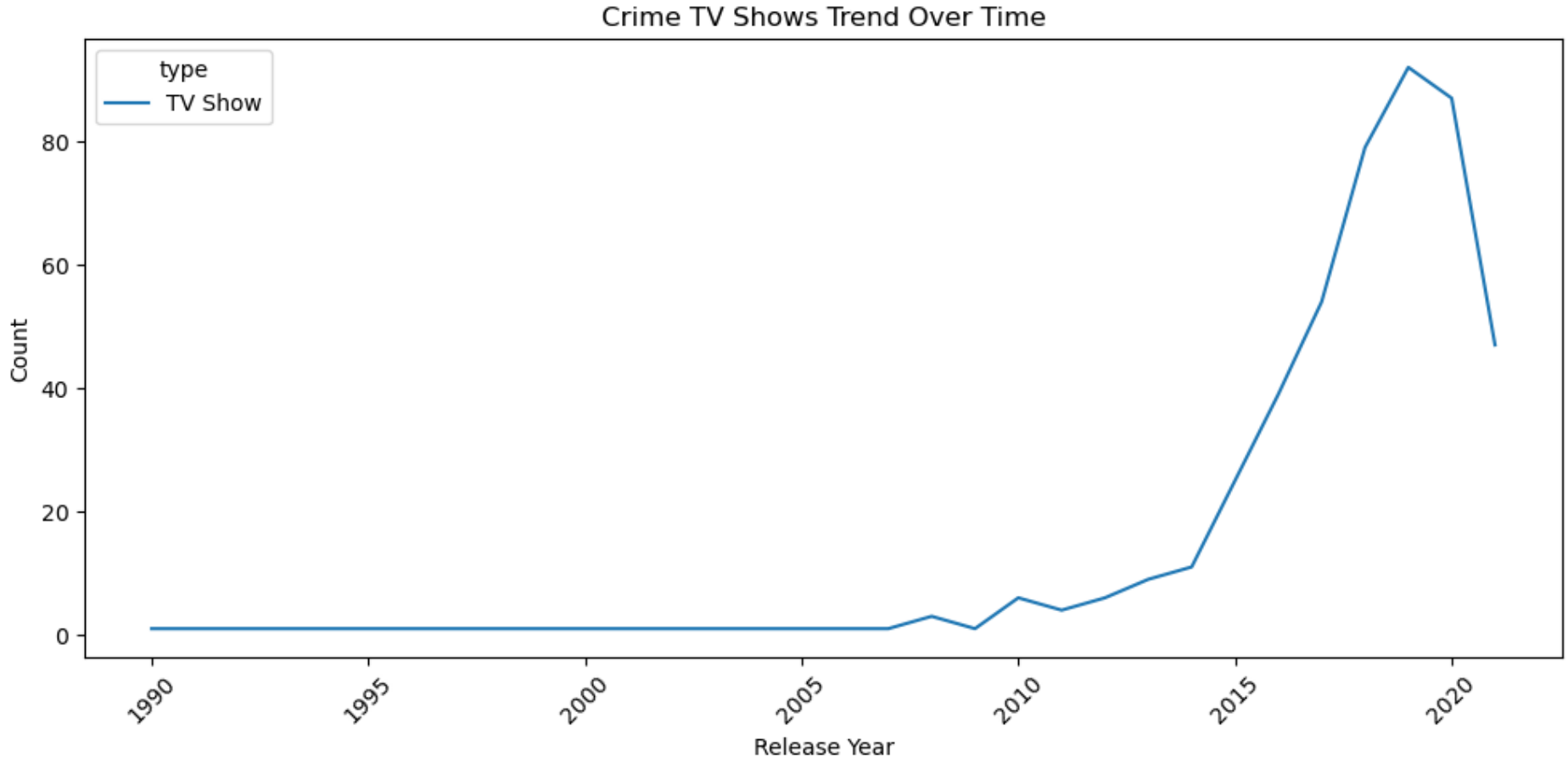
```
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

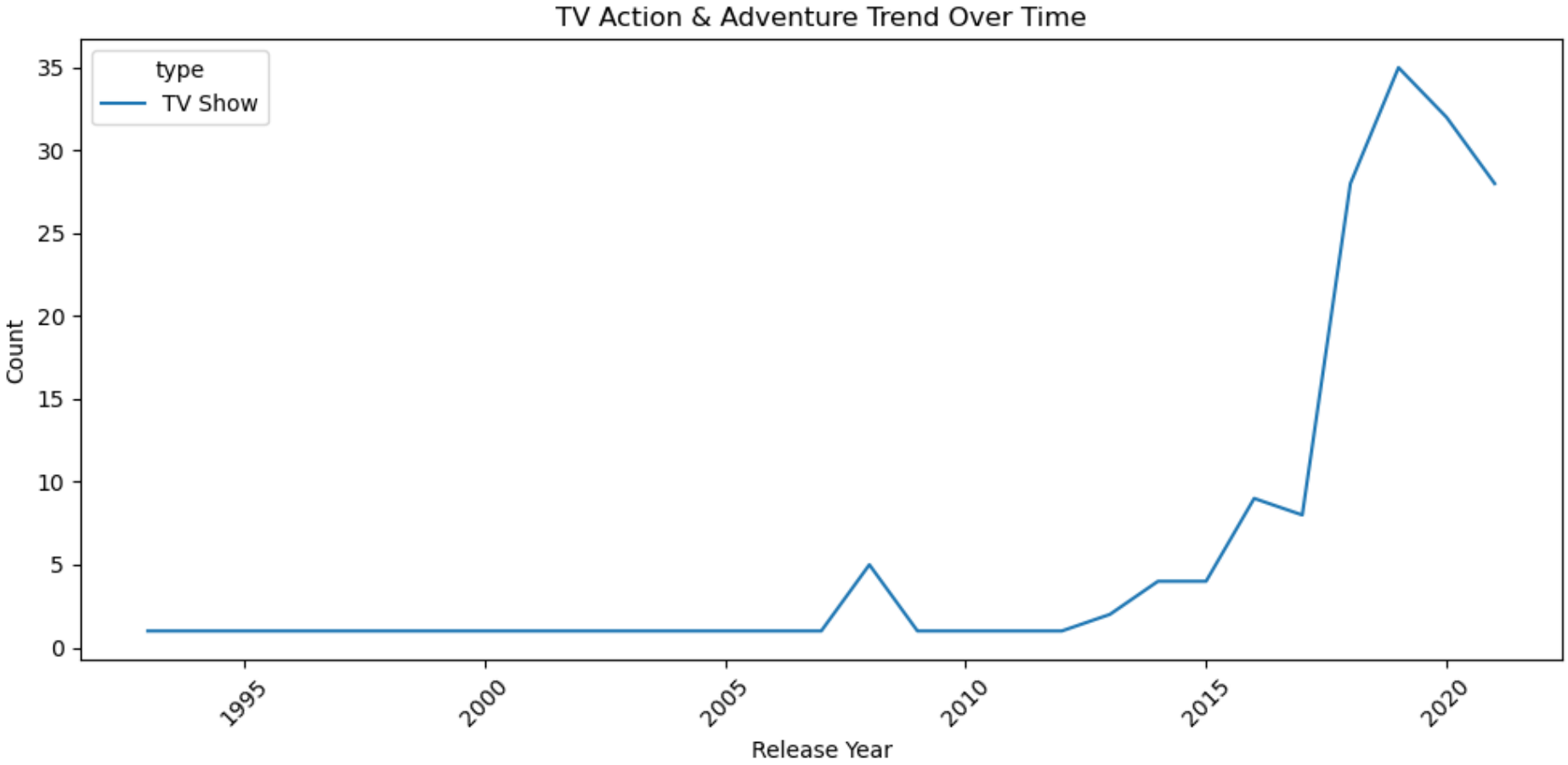


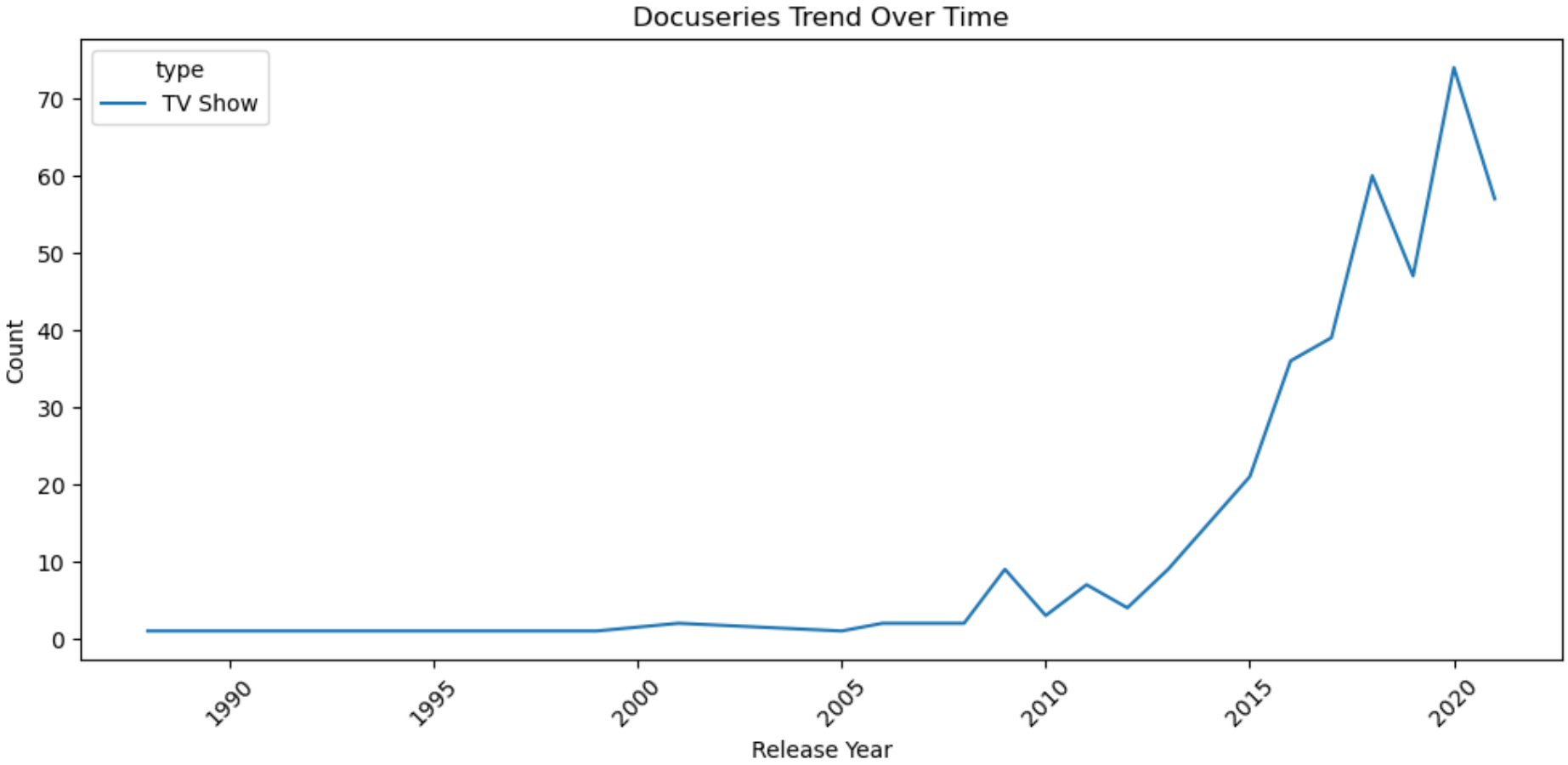

International TV Shows Trend Over Time
type
TV Show
200
150
100
50
0
Count
1970
1980
1990
2000
2010
2020
Release Year

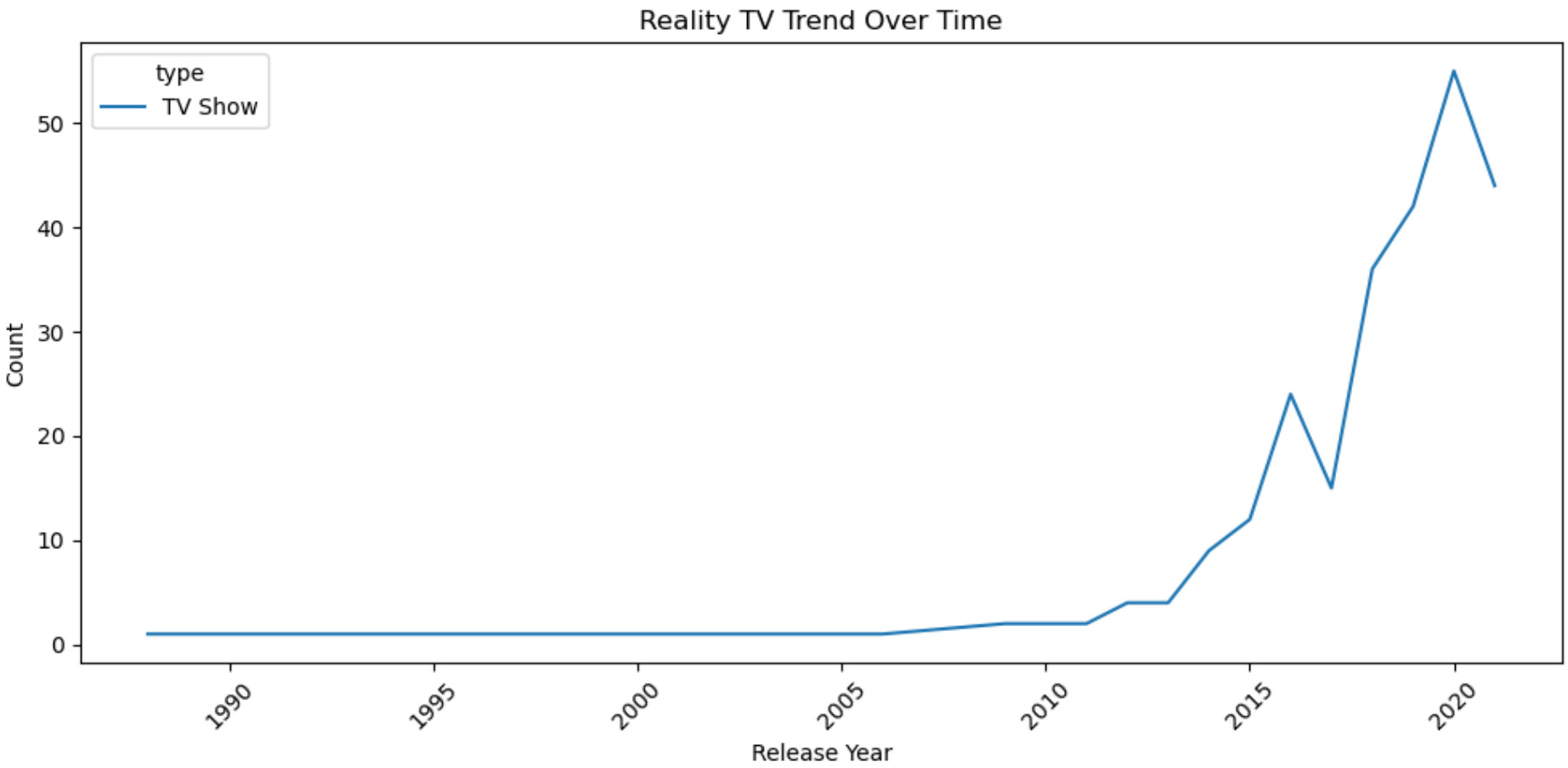

TV Dramas Trend Over Time
type
TV Show
Count
0
20
40
60
80
100
120
1985
1990
1995
2000
2005
2010
2015
2020
Release Year

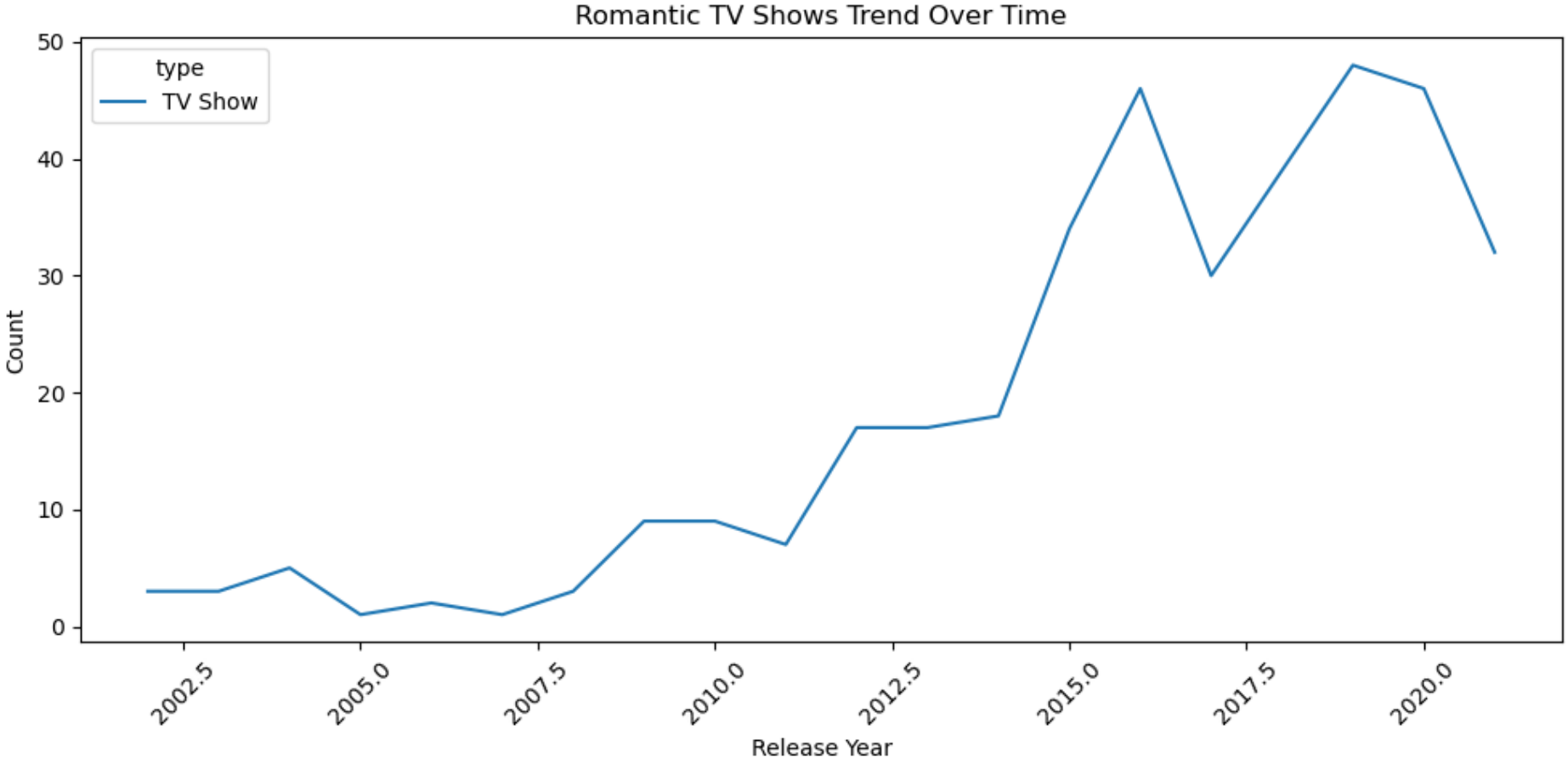

TV Mysteries Trend Over Time
type
TV Show
Count
0
5
10
15
20
25
2000.0
2002.5
2005.0
2007.5
2010.0
2012.5
2015.0
2017.5
2020.0
Release Year

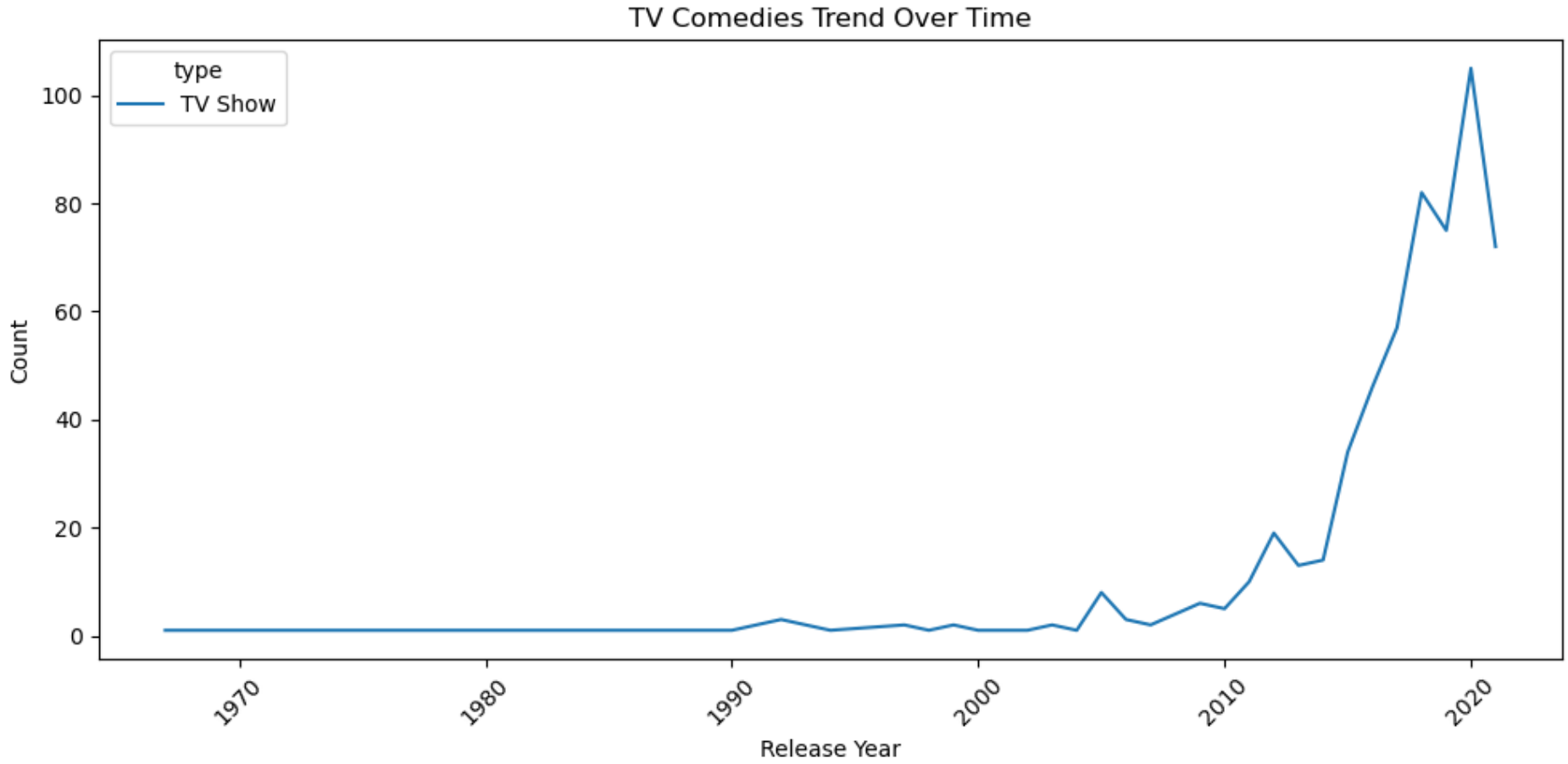

Crime TV Shows Trend Over Time
type
TV Show
Count
0
20
40
60
80
1990
1995
2000
2005
2010
2015
2020
Release Year

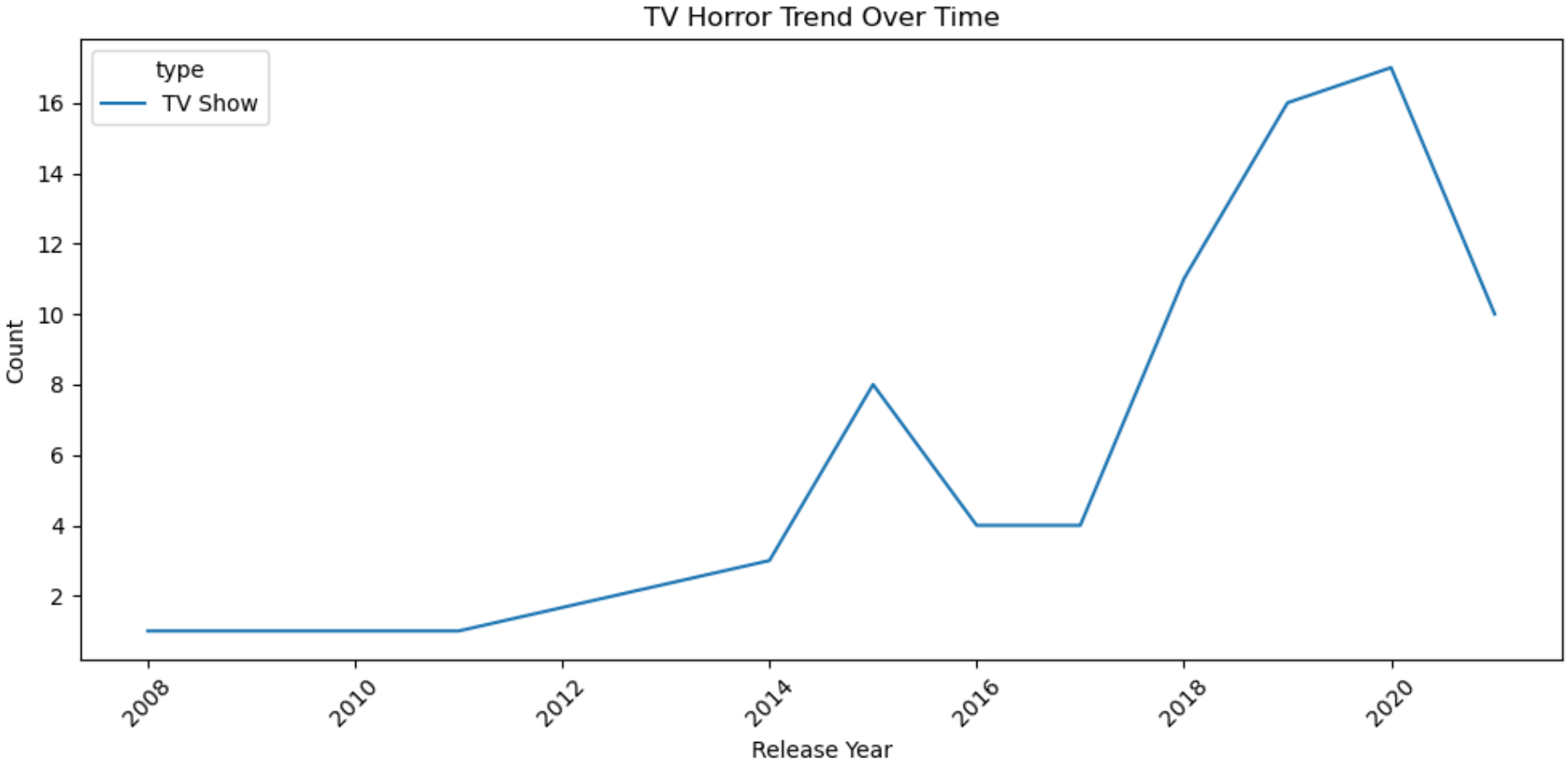

TV Action & Adventure Trend Over Time
type
TV Show
Count
0
5
10
15
20
25
30
35
1995
2000
2005
2010
2015
2020
Release Year

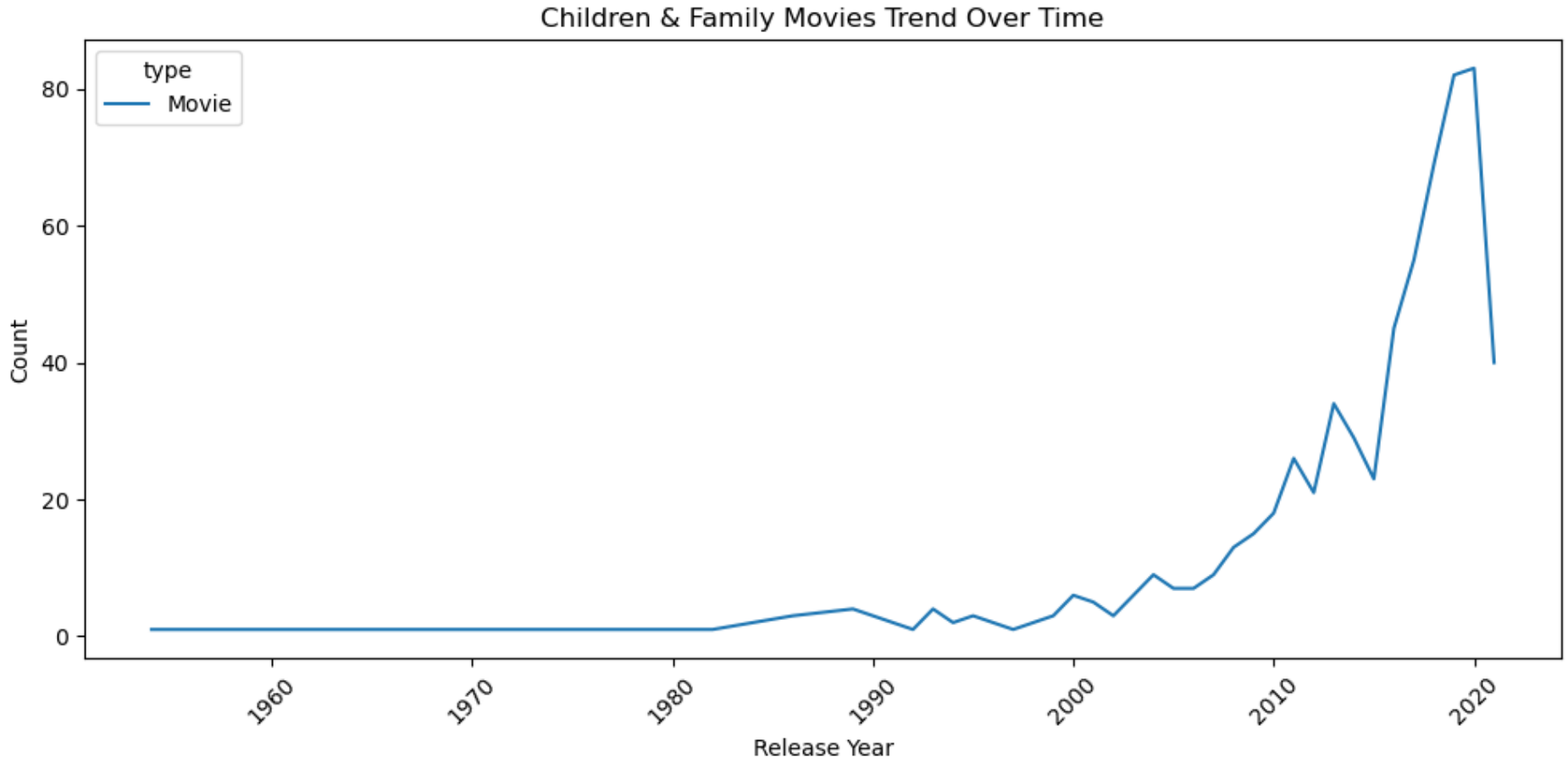

Docuseries Trend Over Time
type
TV Show
Count
0
10
20
30
40
50
60
70
1990
1995
2000
2005
2010
2015
2020
Release Year

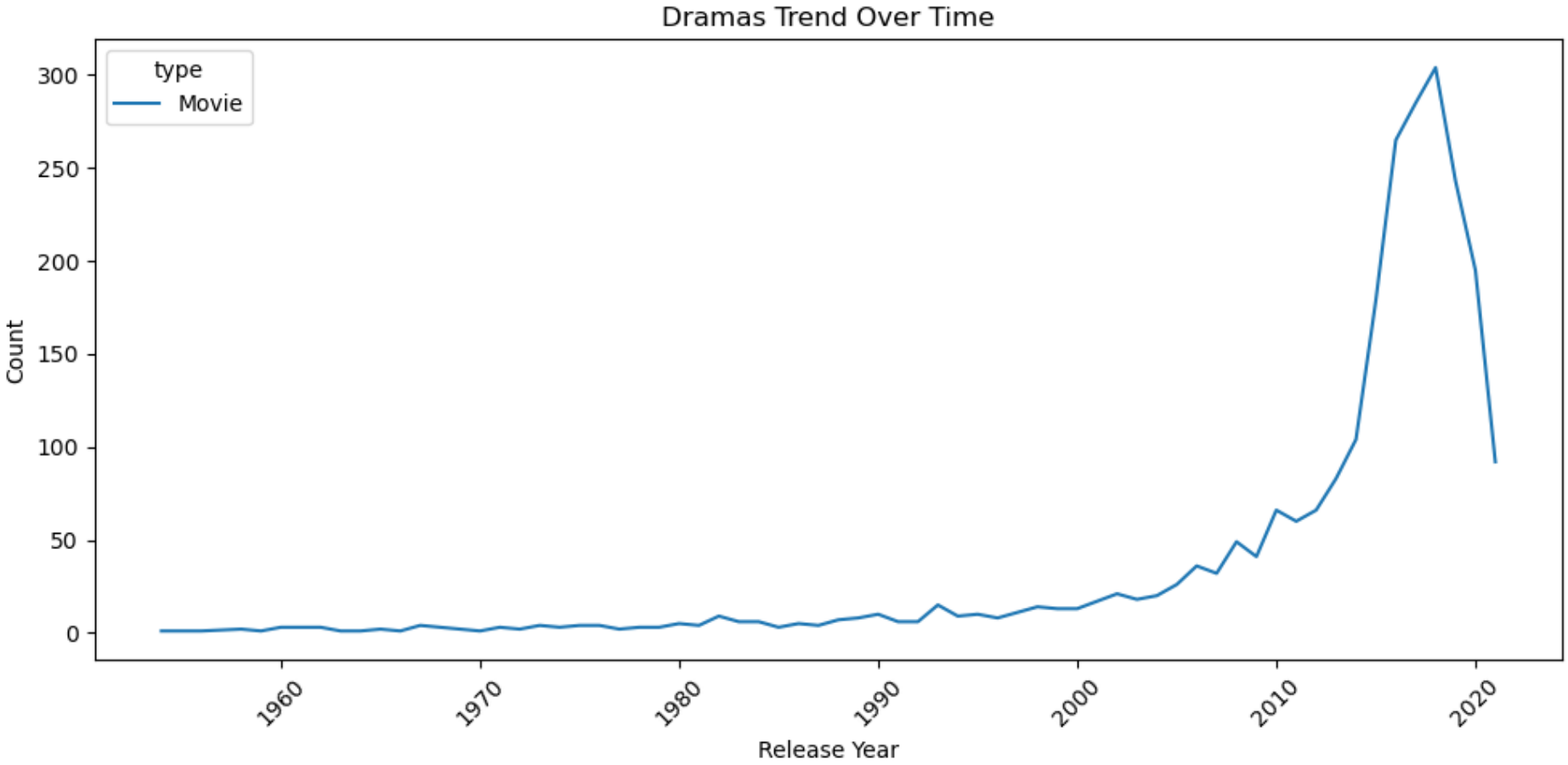

Reality TV Trend Over Time
type
TV Show
Count
0
10
20
30
40
50
1990
1995
2000
2005
2010
2015
2020
Release Year

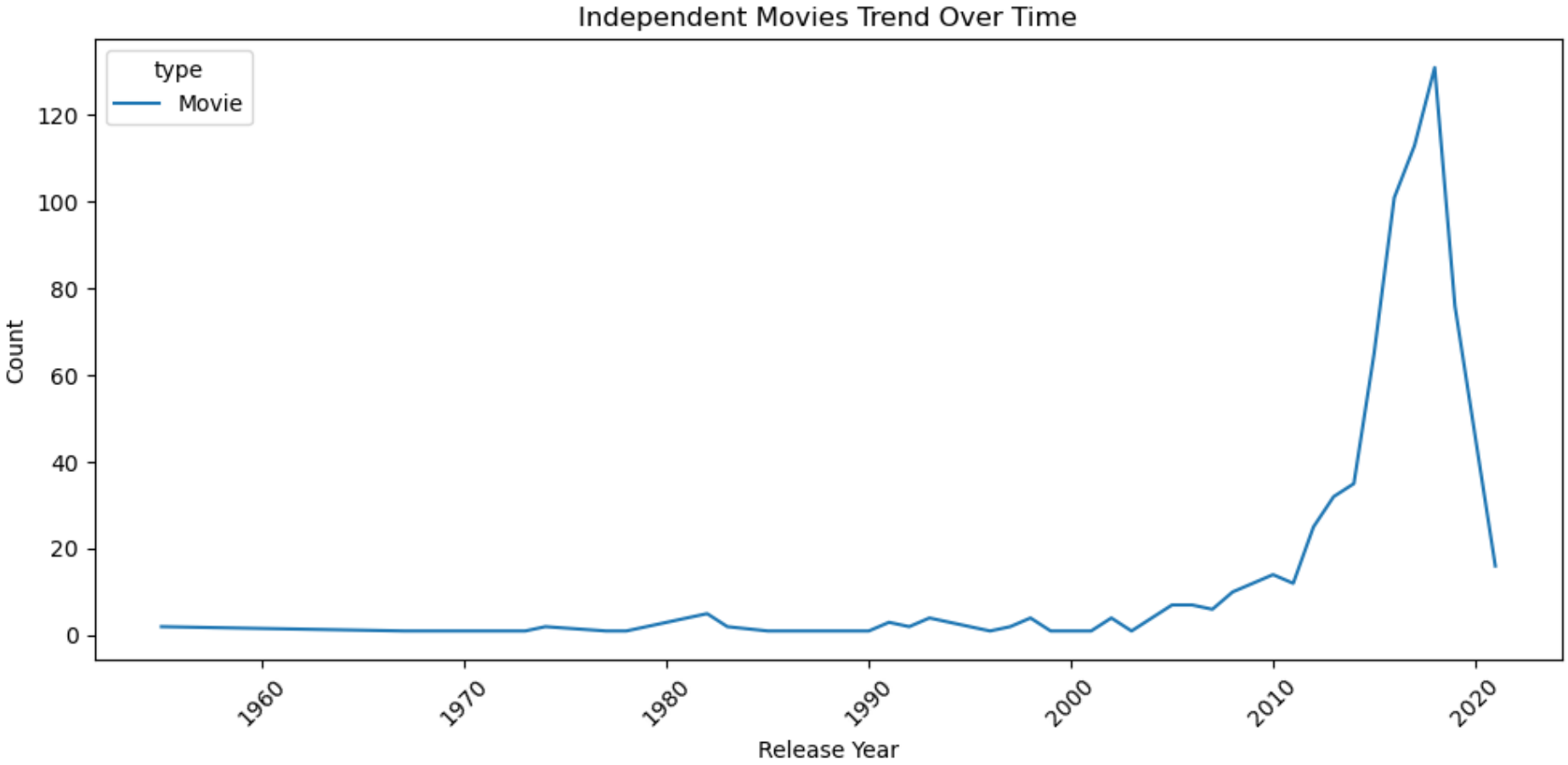

Romantic TV Shows Trend Over Time
type
TV Show
Count
Release Year
0
10
20
30
40
50
2002.5
2005.0
2007.5
2010.0
2012.5
2015.0
2017.5
2020.0

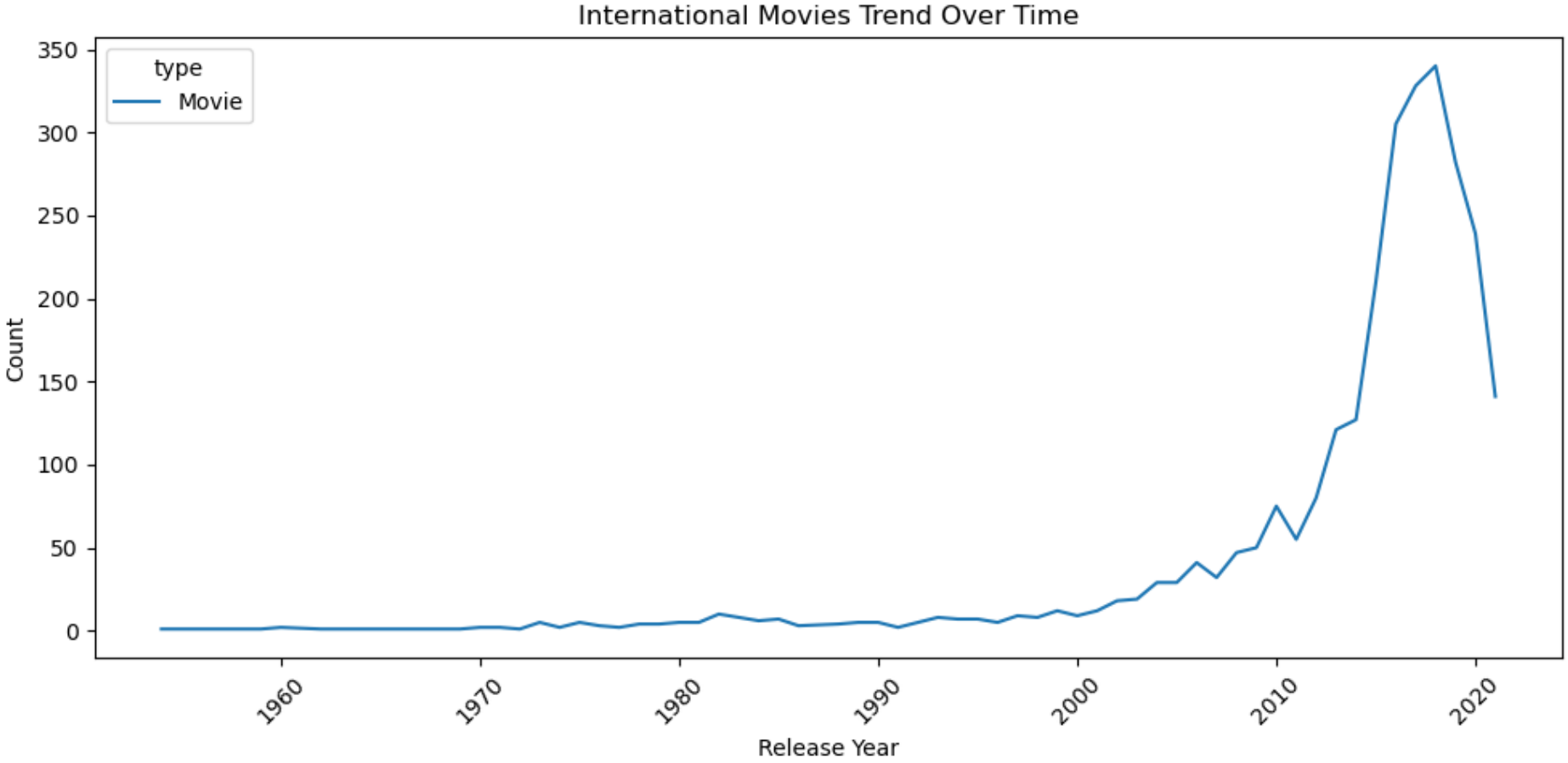

TV Comedies Trend Over Time
type
TV Show
Count
0
20
40
60
80
100
1970
1980
1990
2000
2010
2020
Release Year

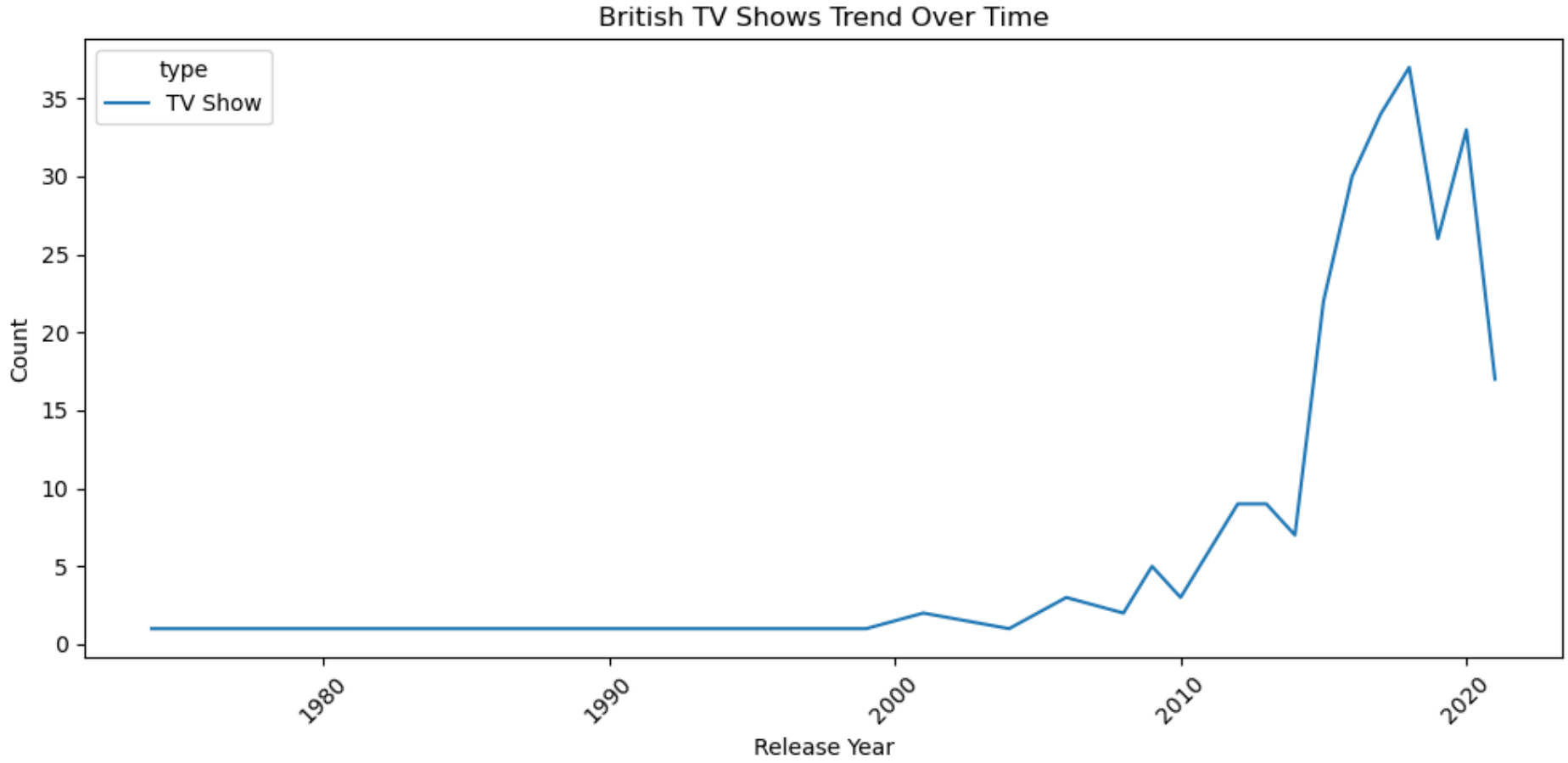

TV Horror Trend Over Time
type
TV Show
Count
16
14
12
10
8
6
4
2
2008
2010
2012
2014
2016
2018
2020
Release Year

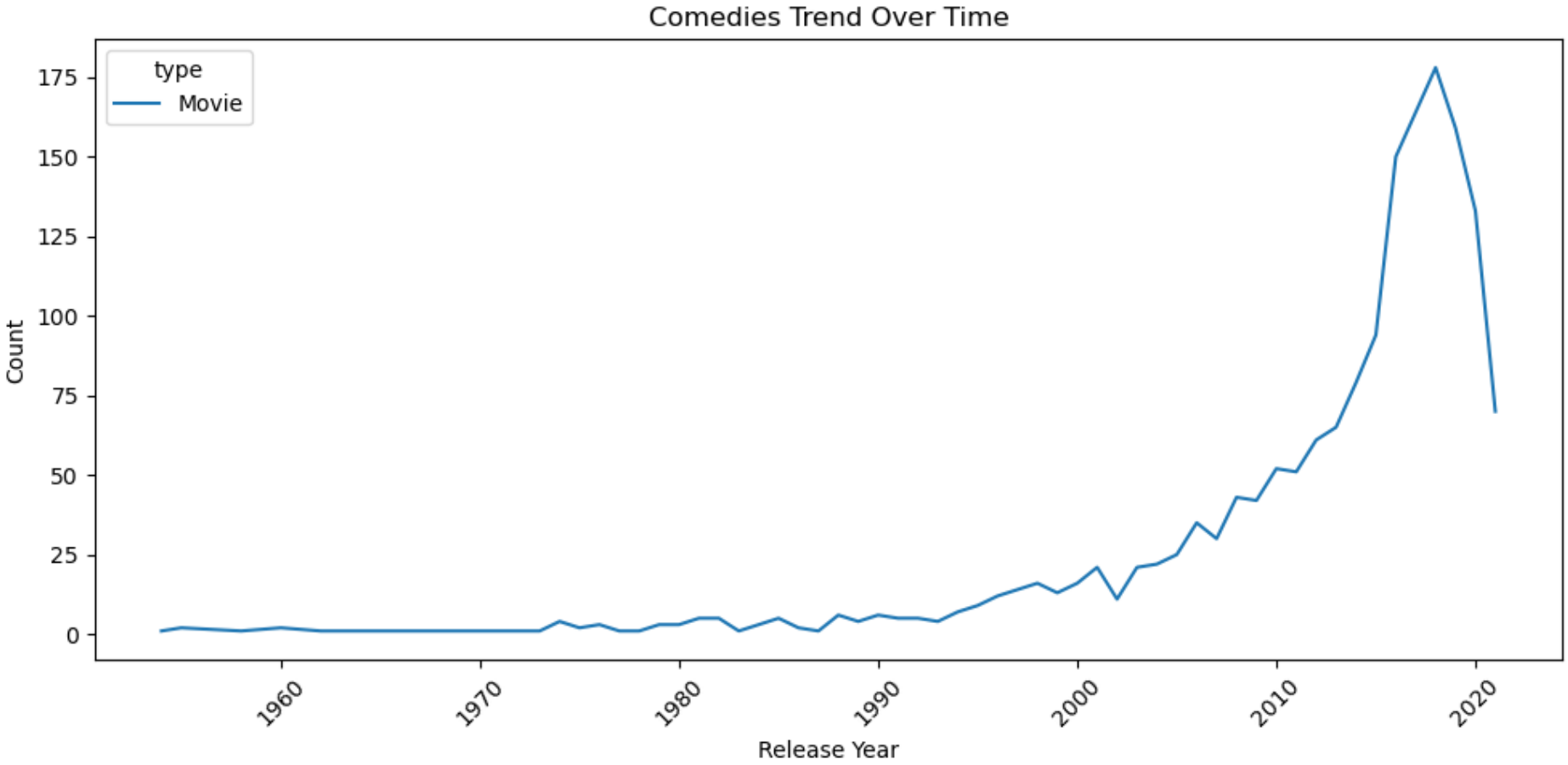

Children & Family Movies Trend Over Time
type
Movie
Count
0
20
40
60
80
1960
1970
1980
1990
2000
2010
2020
Release Year

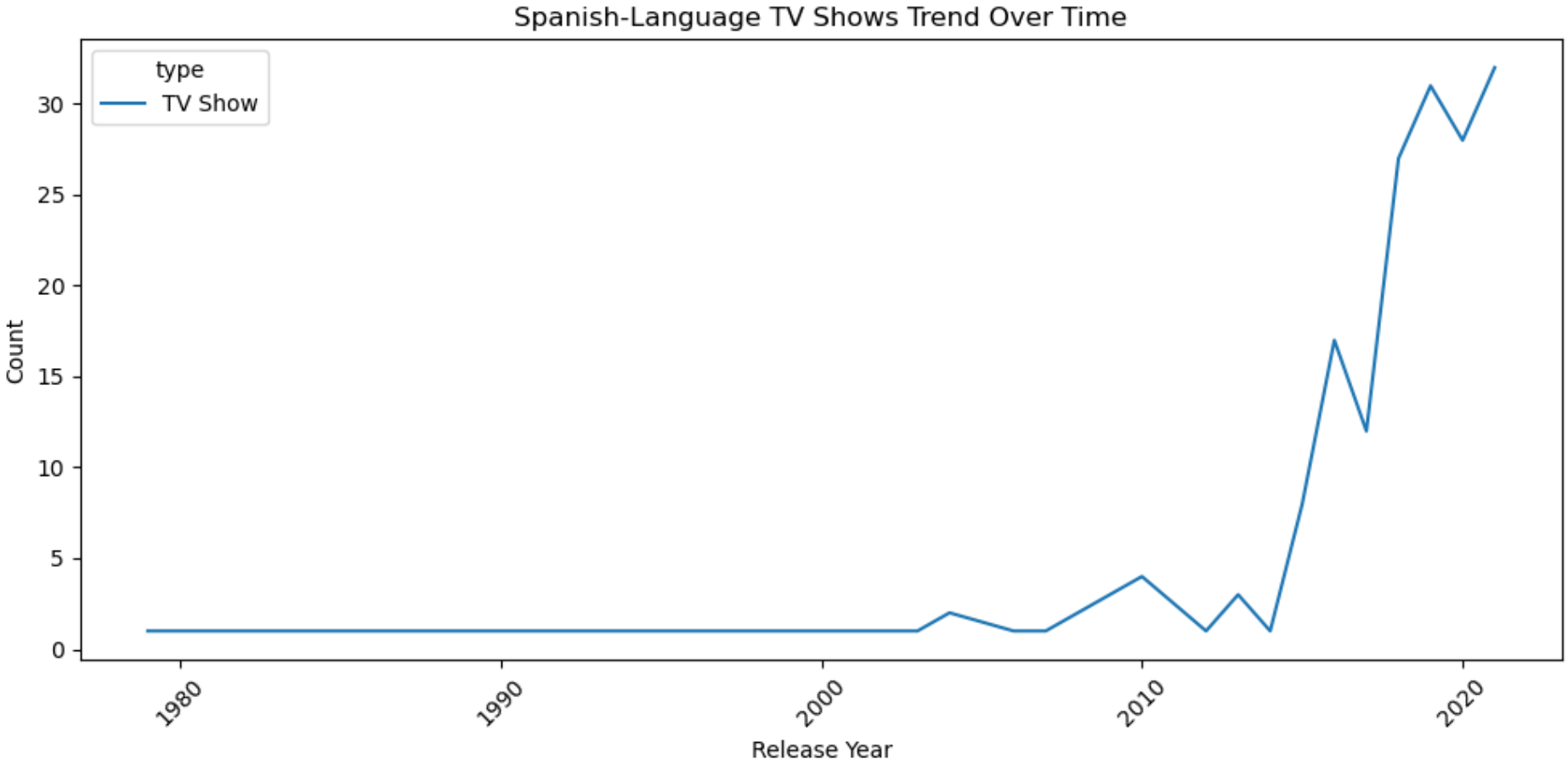

Dramas Trend Over Time
type
Movie
300
250
200
150
100
50
0
Count
1960
1970
1980
1990
2000
2010
2020
Release Year

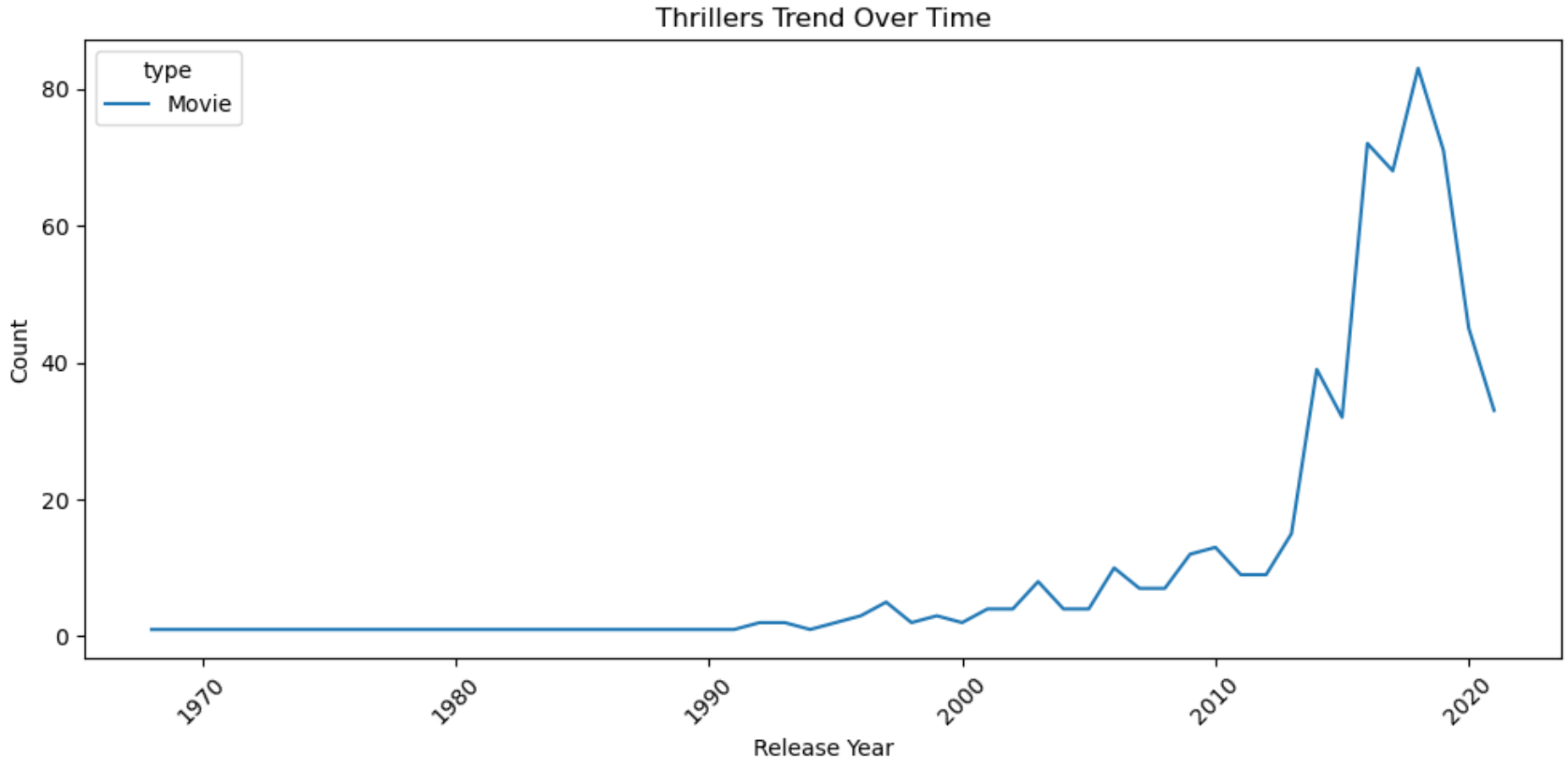

Independent Movies Trend Over Time
type
Movie
120
100
80
60
40
20
0
Count
1960
1970
1980
1990
2000
2010
2020
Release Year

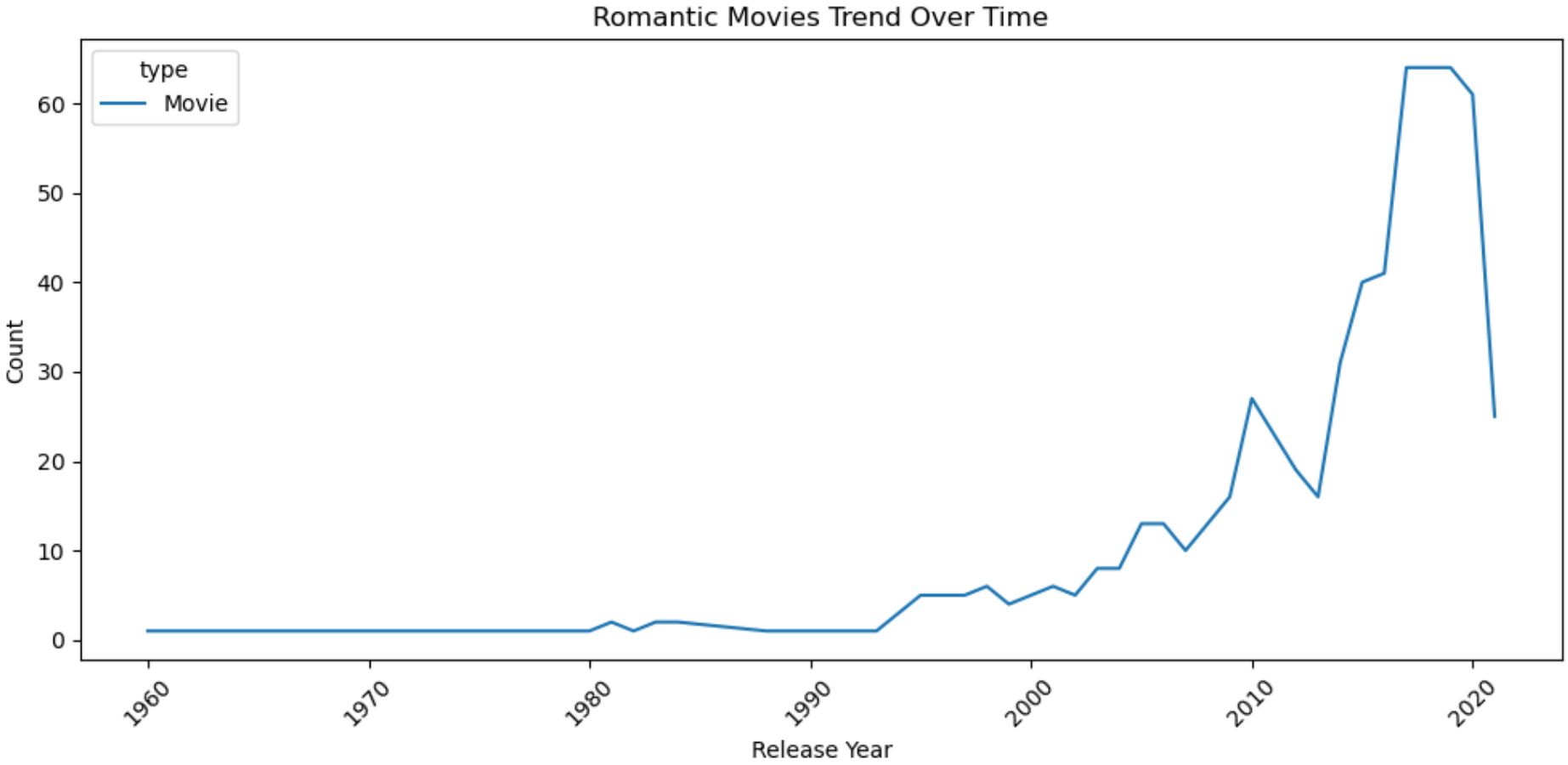

International Movies Trend Over Time
type
Movie
Count
Release Year
0
50
100
150
200
250
300
350
1960
1970
1980
1990
2000
2010
2020

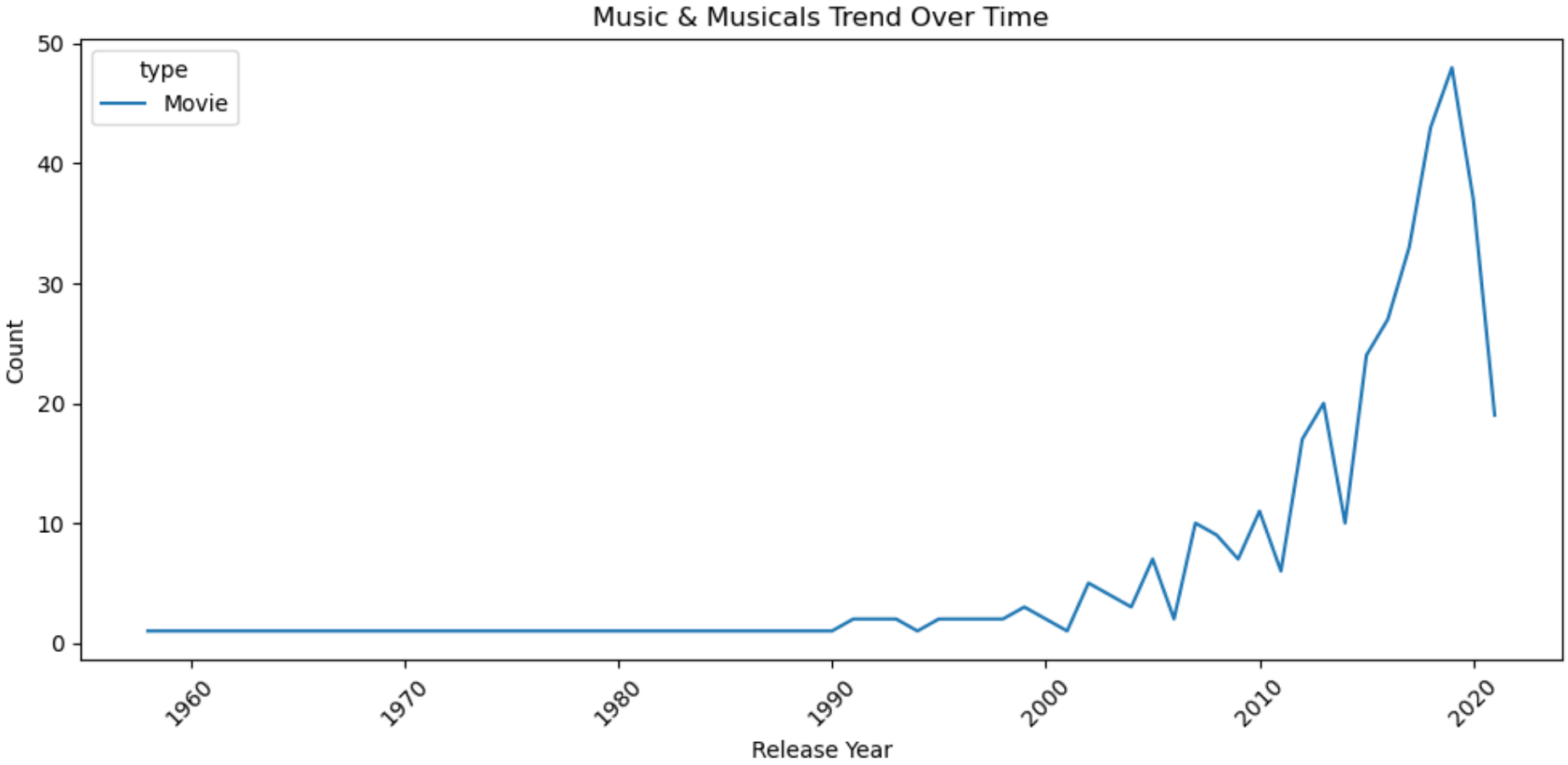

British TV Shows Trend Over Time
type
TV Show
Count
0
5
10
15
20
25
30
35
1980
1990
2000
2010
2020
Release Year

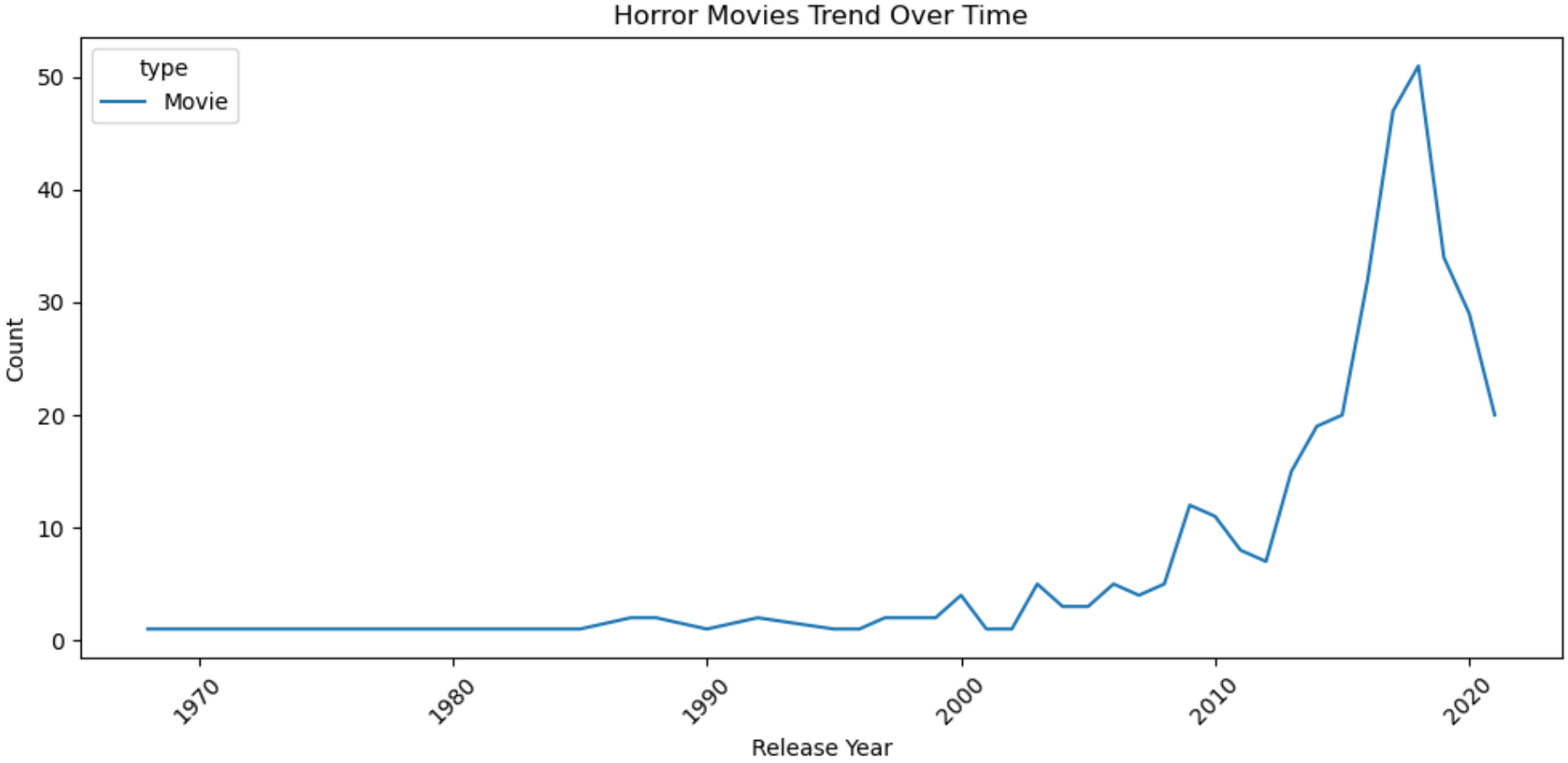

Comedies Trend Over Time
type
Movie
175
150
125
100
75
50
25
0
Count
1960
1970
1980
1990
2000
2010
2020
Release Year

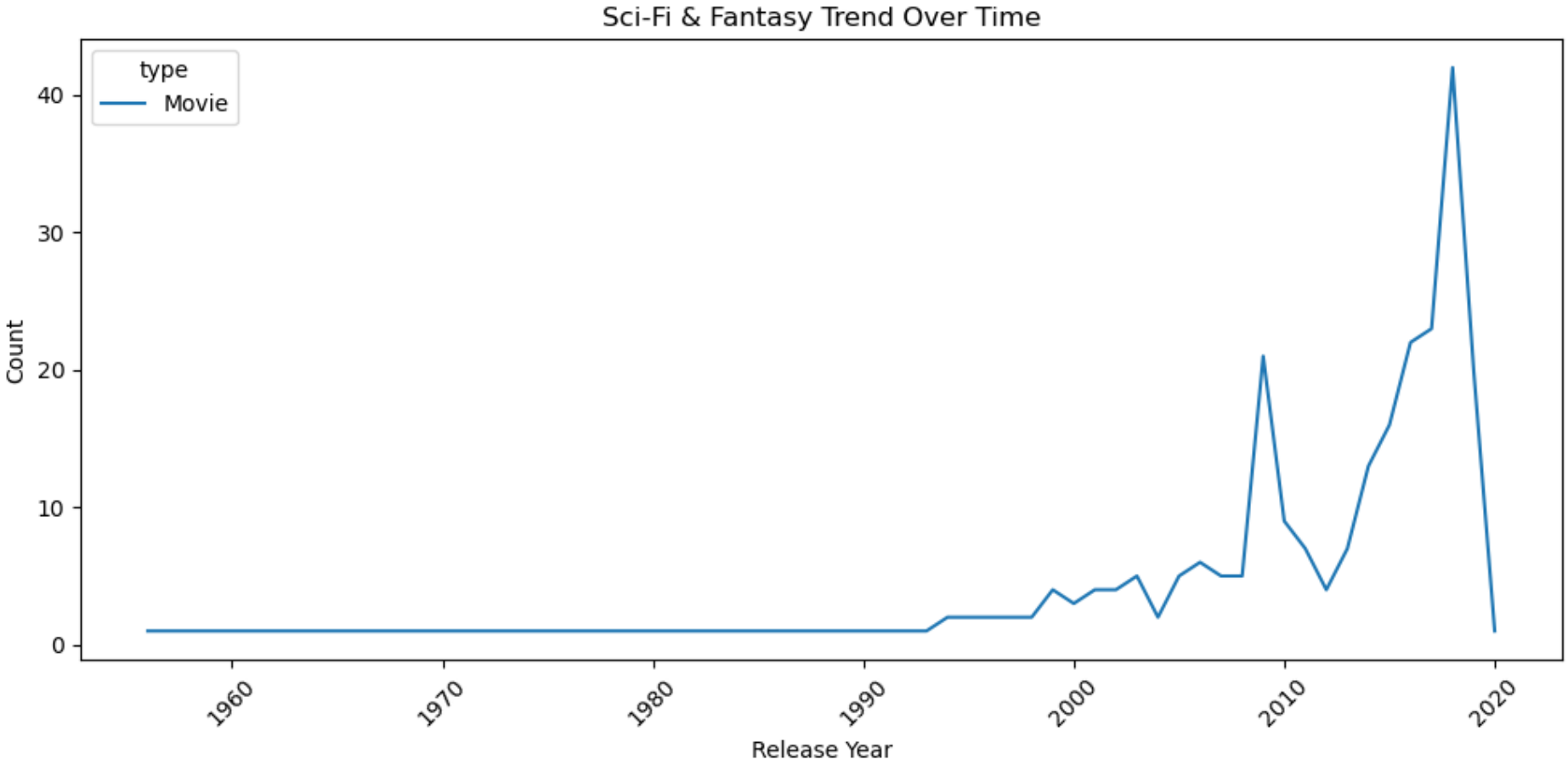

Spanish-Language TV Shows Trend Over Time
type
TV Show
Count
0
5
10
15
20
25
30
1980
1990
2000
2010
2020
Release Year

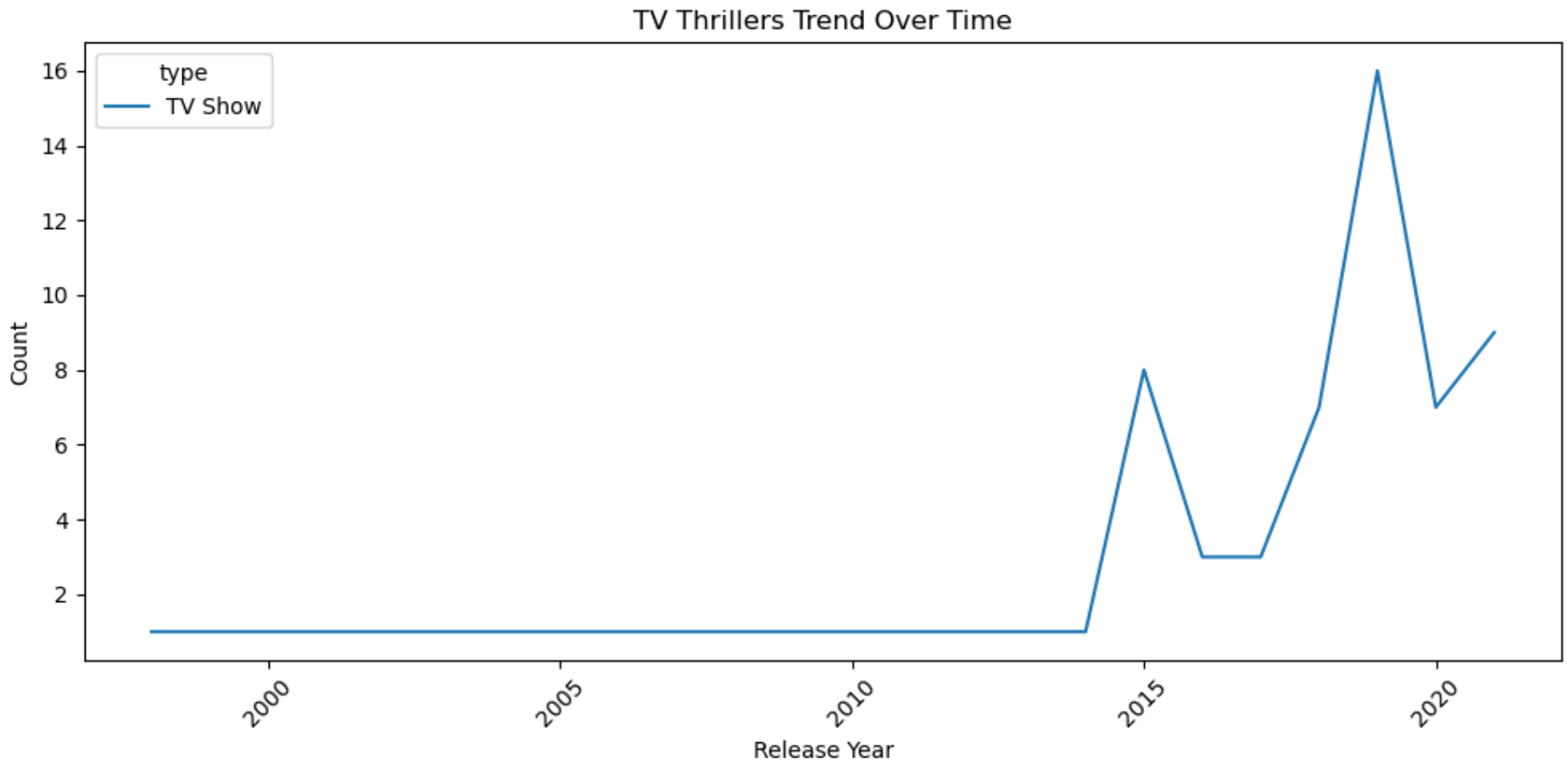

Thrillers Trend Over Time
type
Movie
Count
0
20
40
60
80
1970
1980
1990
2000
2010
2020
Release Year

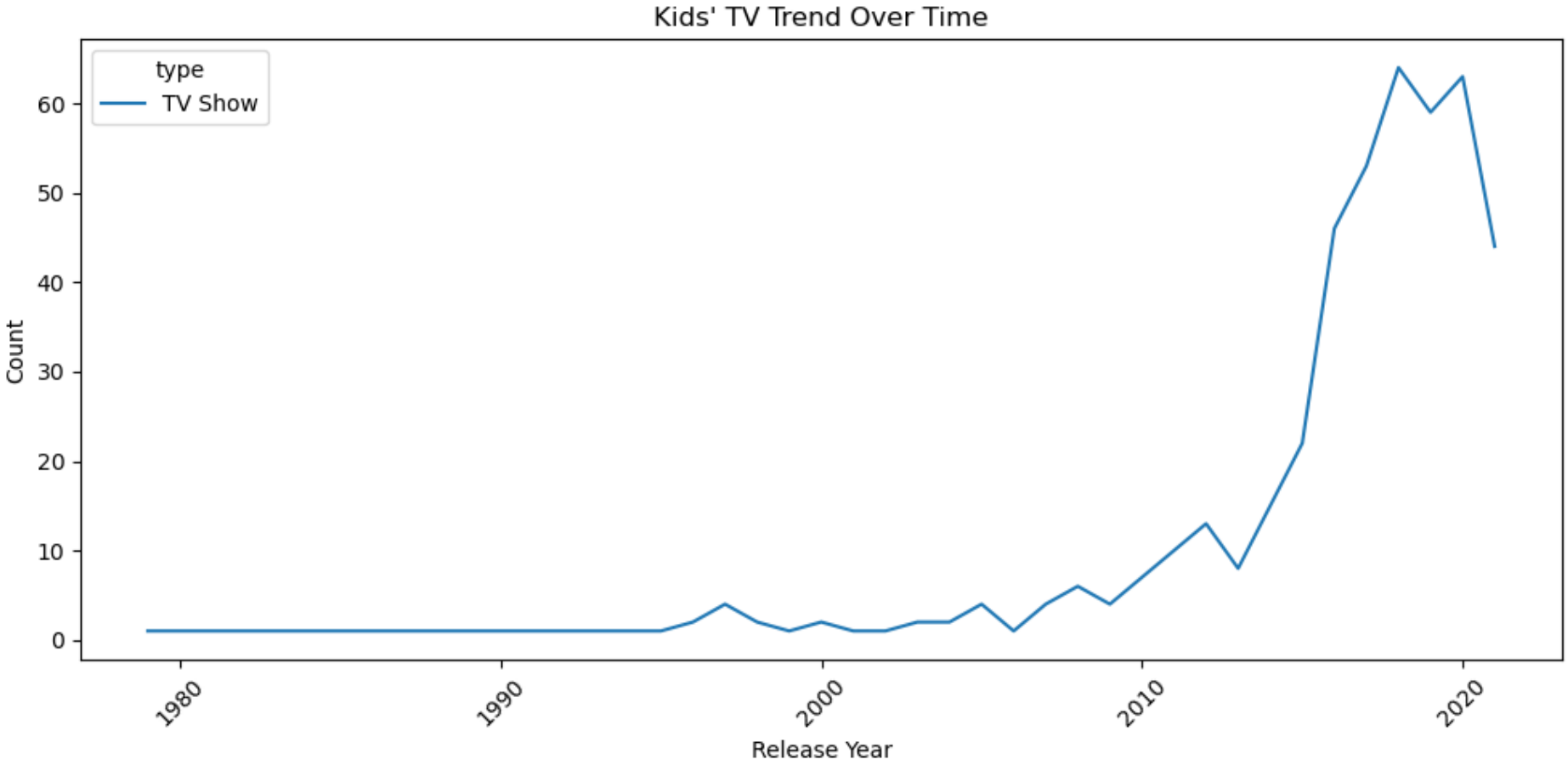

Romantic Movies Trend Over Time
type
Movie
Count
0
10
20
30
40
50
60
1960
1970
1980
1990
2000
2010
2020
Release Year

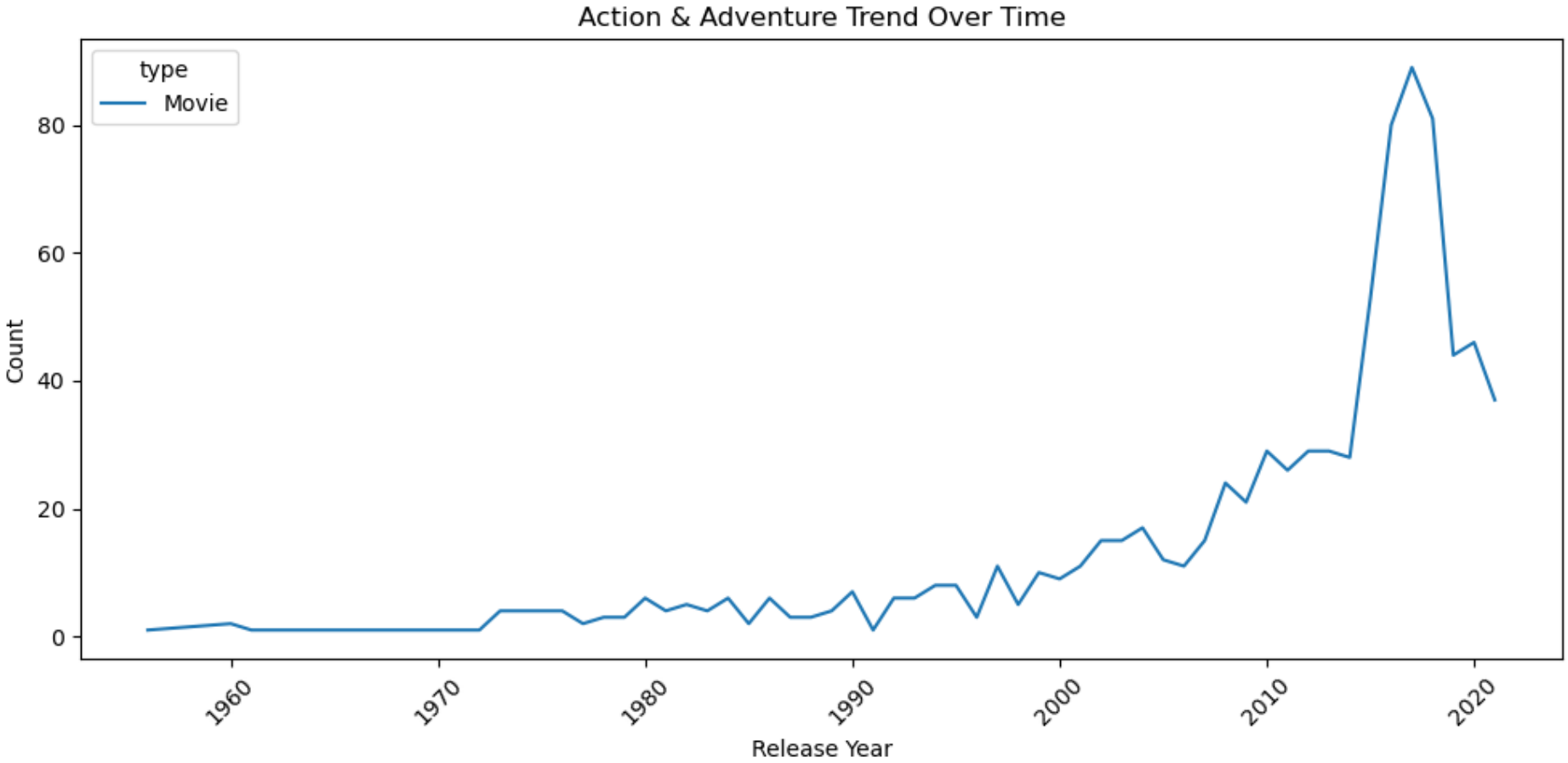

Music & Musicals Trend Over Time
type
Movie
Count
Release Year
0
10
20
30
40
50
1960
1970
1980
1990
2000
2010
2020

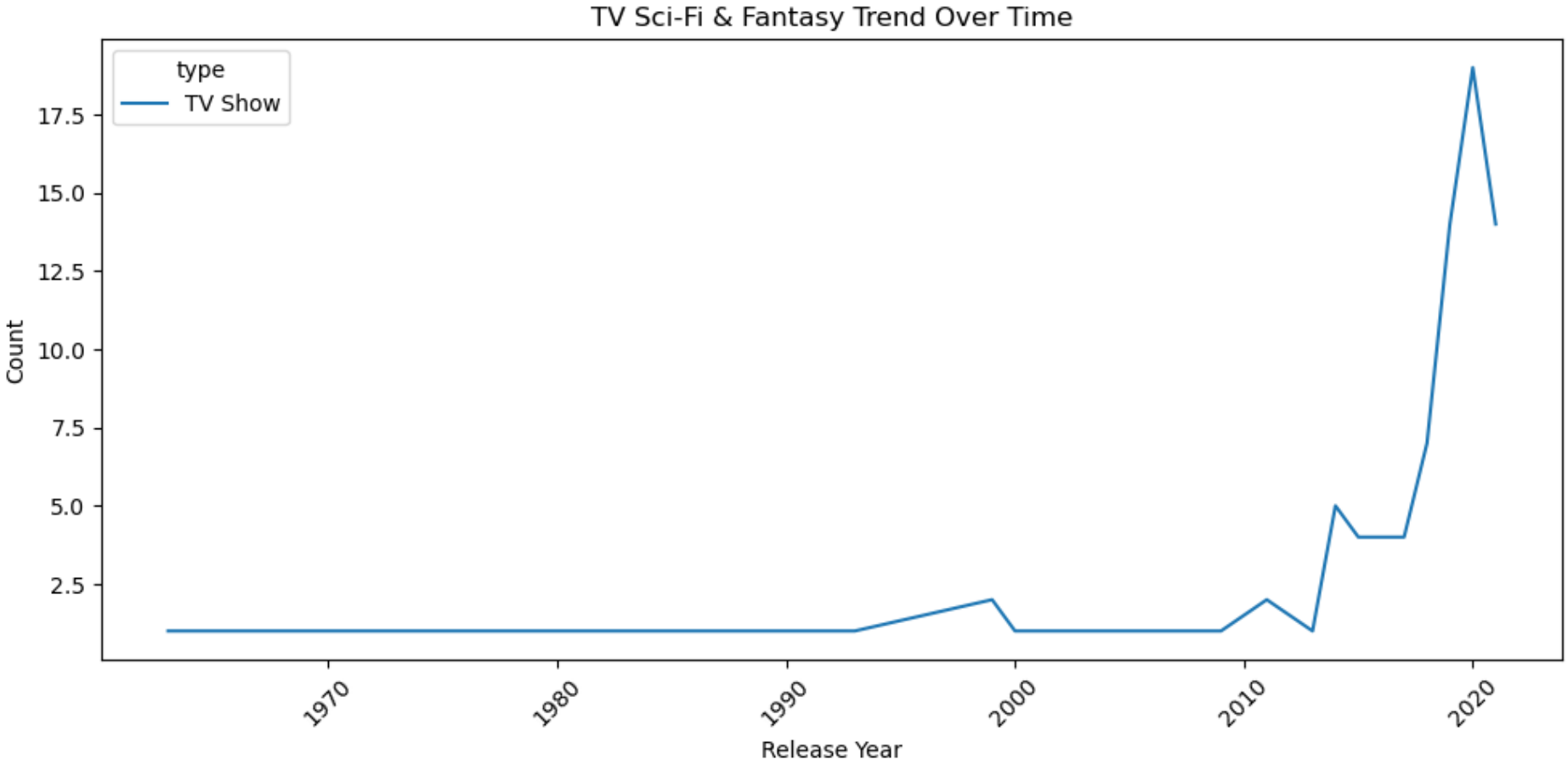

Horror Movies Trend Over Time
type
Movie
Count
0
10
20
30
40
50
1970
1980
1990
2000
2010
2020
Release Year

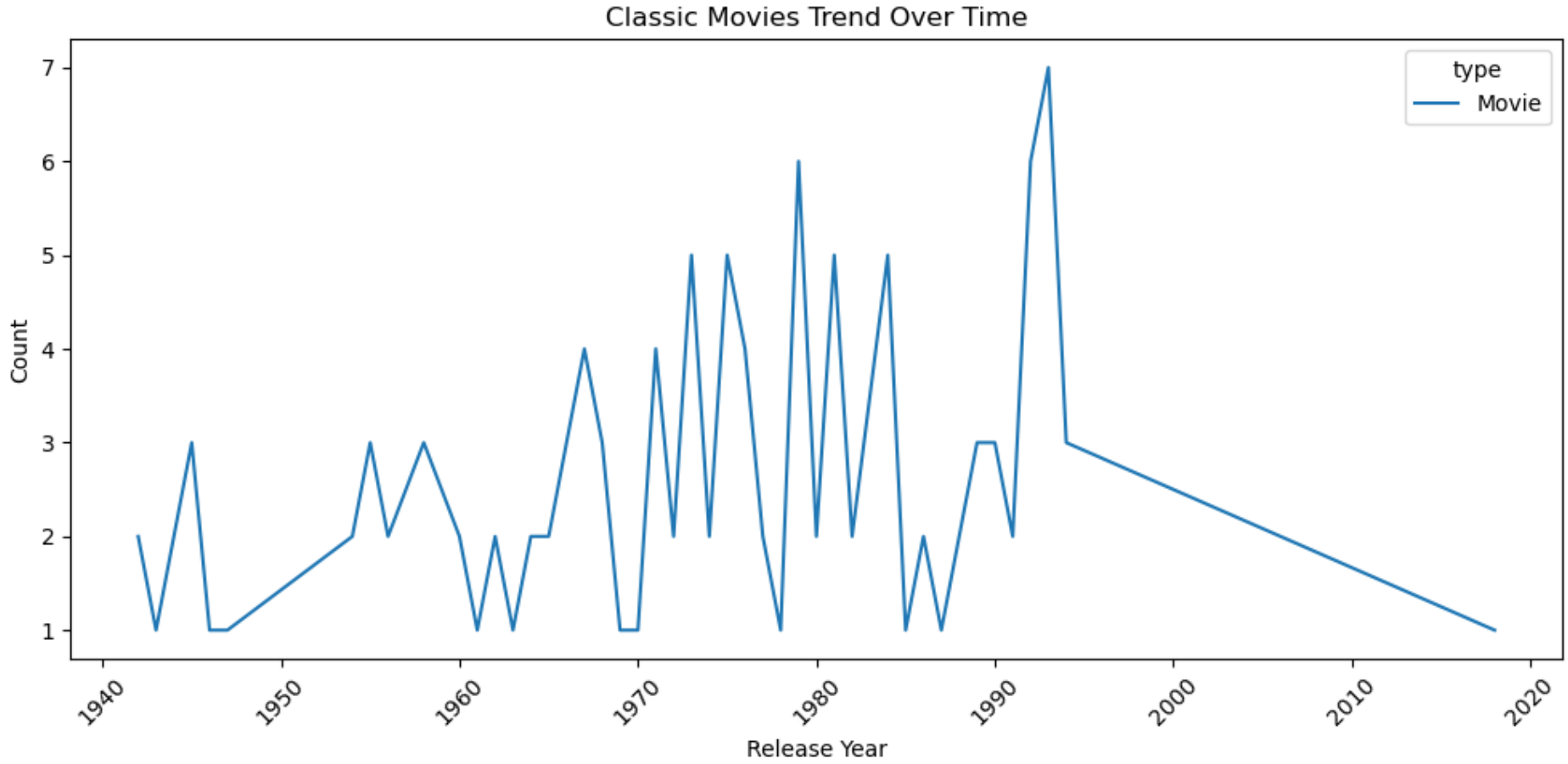

Sci-Fi & Fantasy Trend Over Time
type
Movie
Count
0
10
20
30
40
1960
1970
1980
1990
2000
2010
2020
Release Year

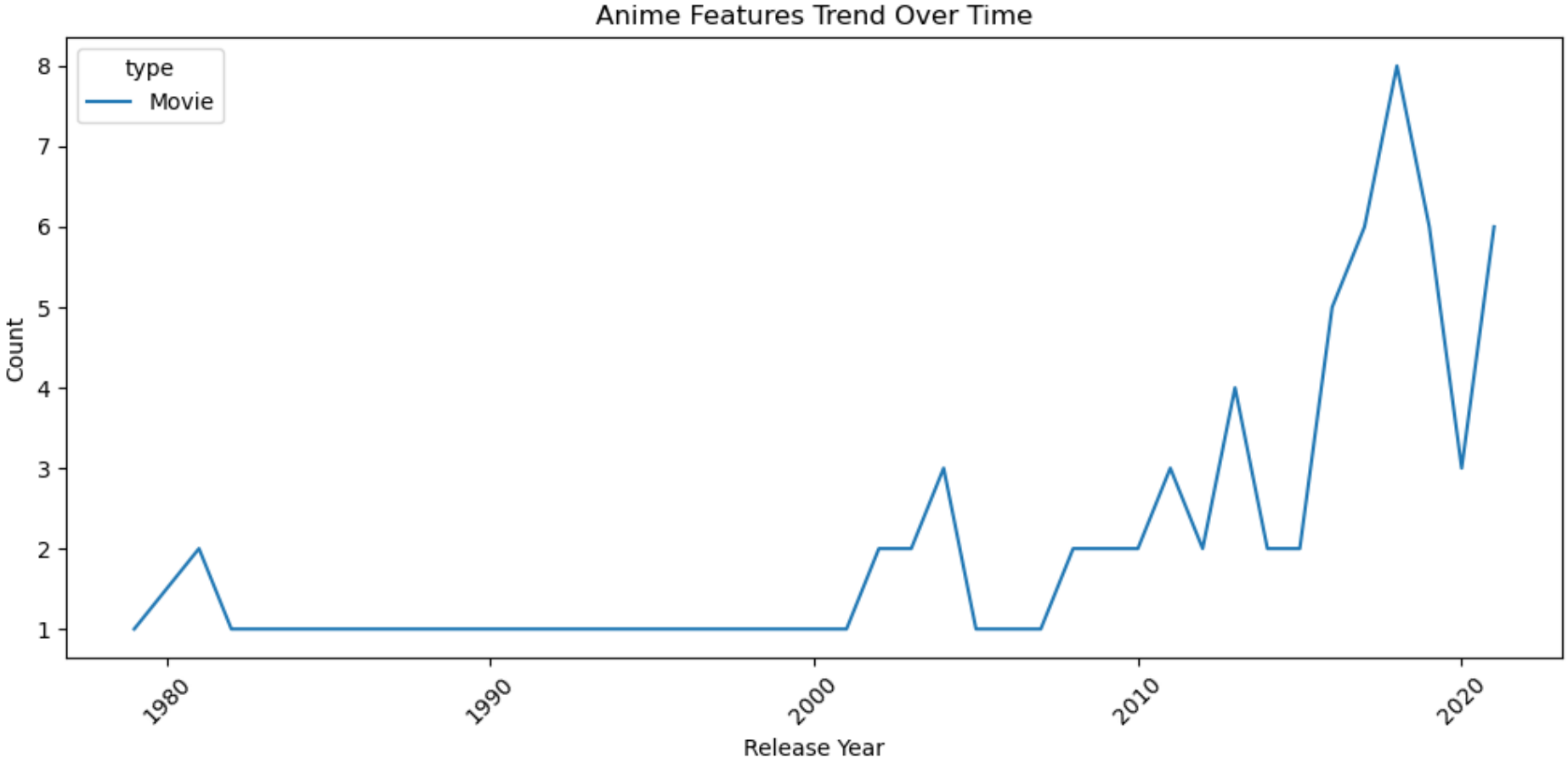

TV Thrillers Trend Over Time
type
TV Show
Count
16
14
12
10
8
6
4
2
2000
2005
2010
2015
2020
Release Year

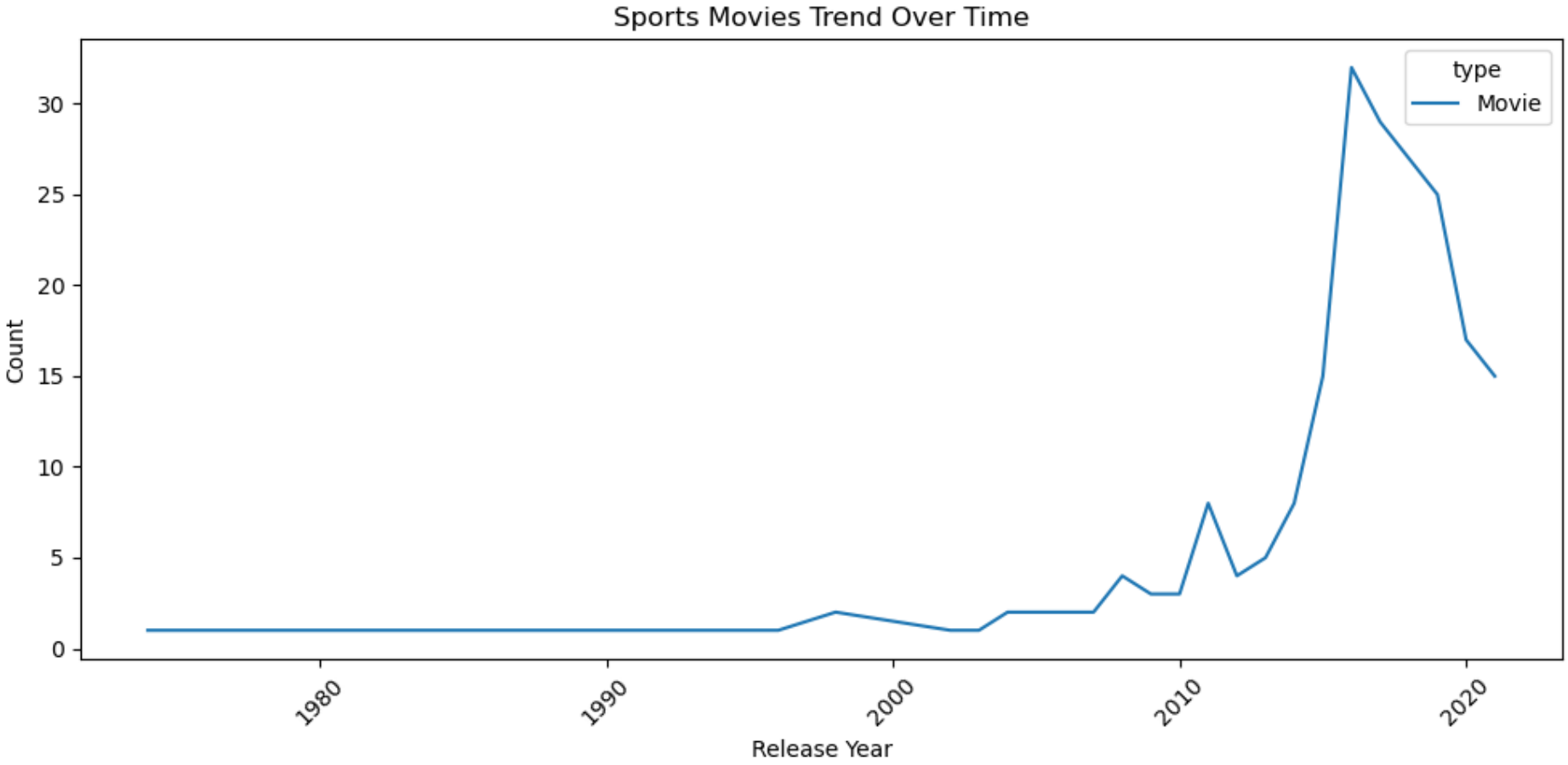

Kids' TV Trend Over Time
type
TV Show
Count
0
10
20
30
40
50
60
1980
1990
2000
2010
2020
Release Year

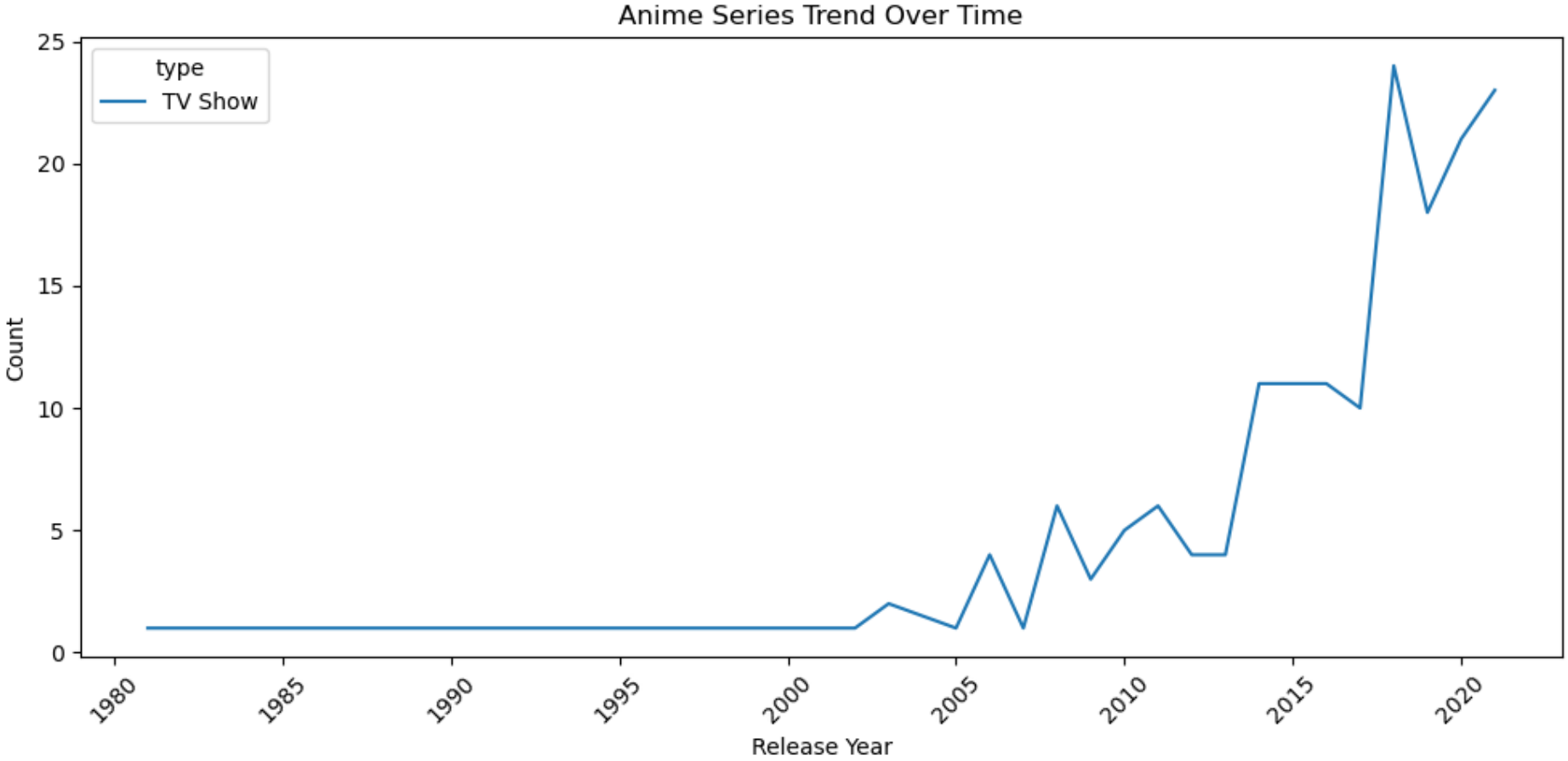

Action & Adventure Trend Over Time
type
Movie
Count
0
20
40
60
80
1960
1970
1980
1990
2000
2010
2020
Release Year

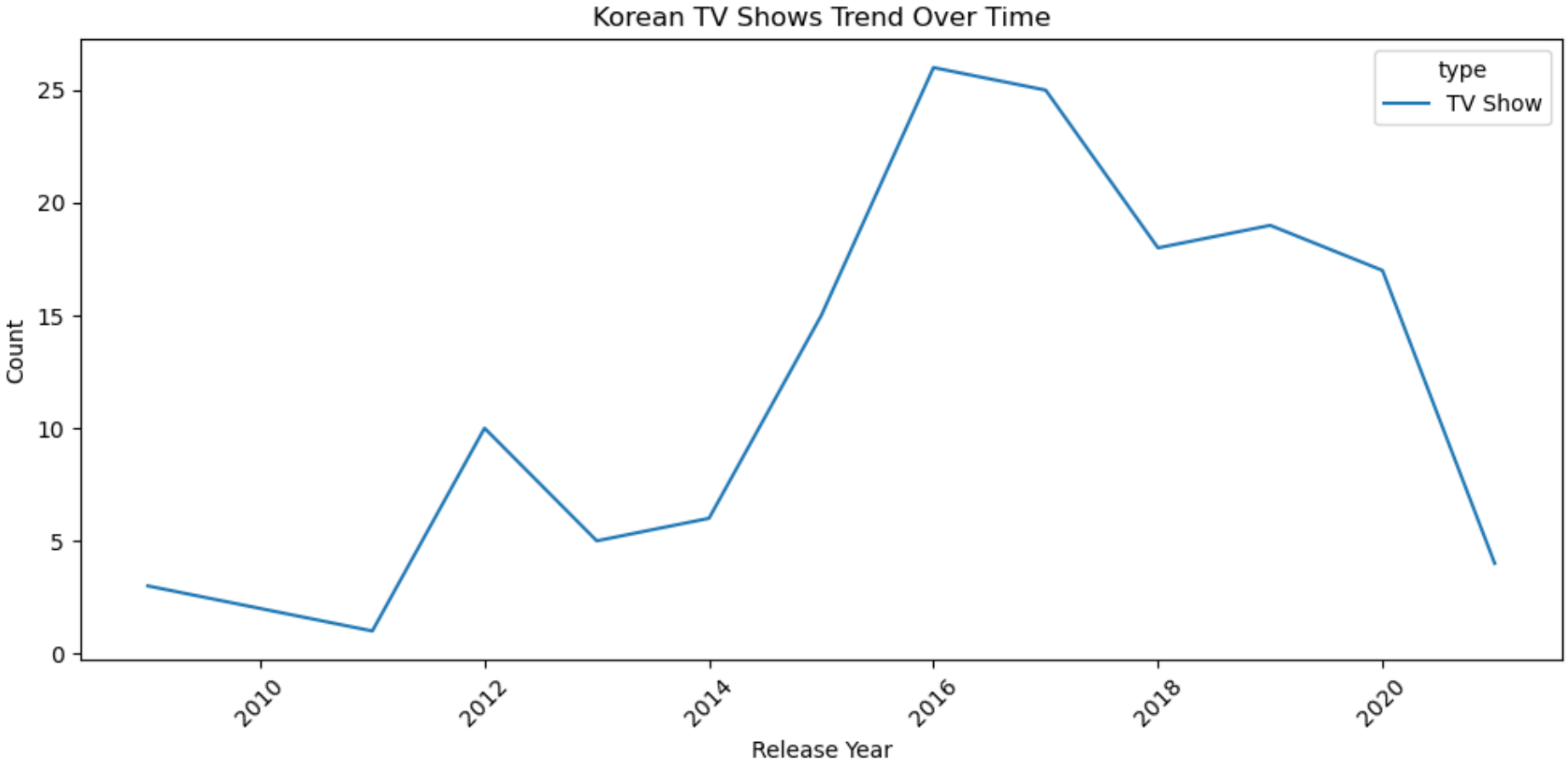

TV Sci-Fi & Fantasy Trend Over Time
type
TV Show
17.5
15.0
12.5
10.0
7.5
5.0
2.5
Count
1970
1980
1990
2000
2010
2020
Release Year

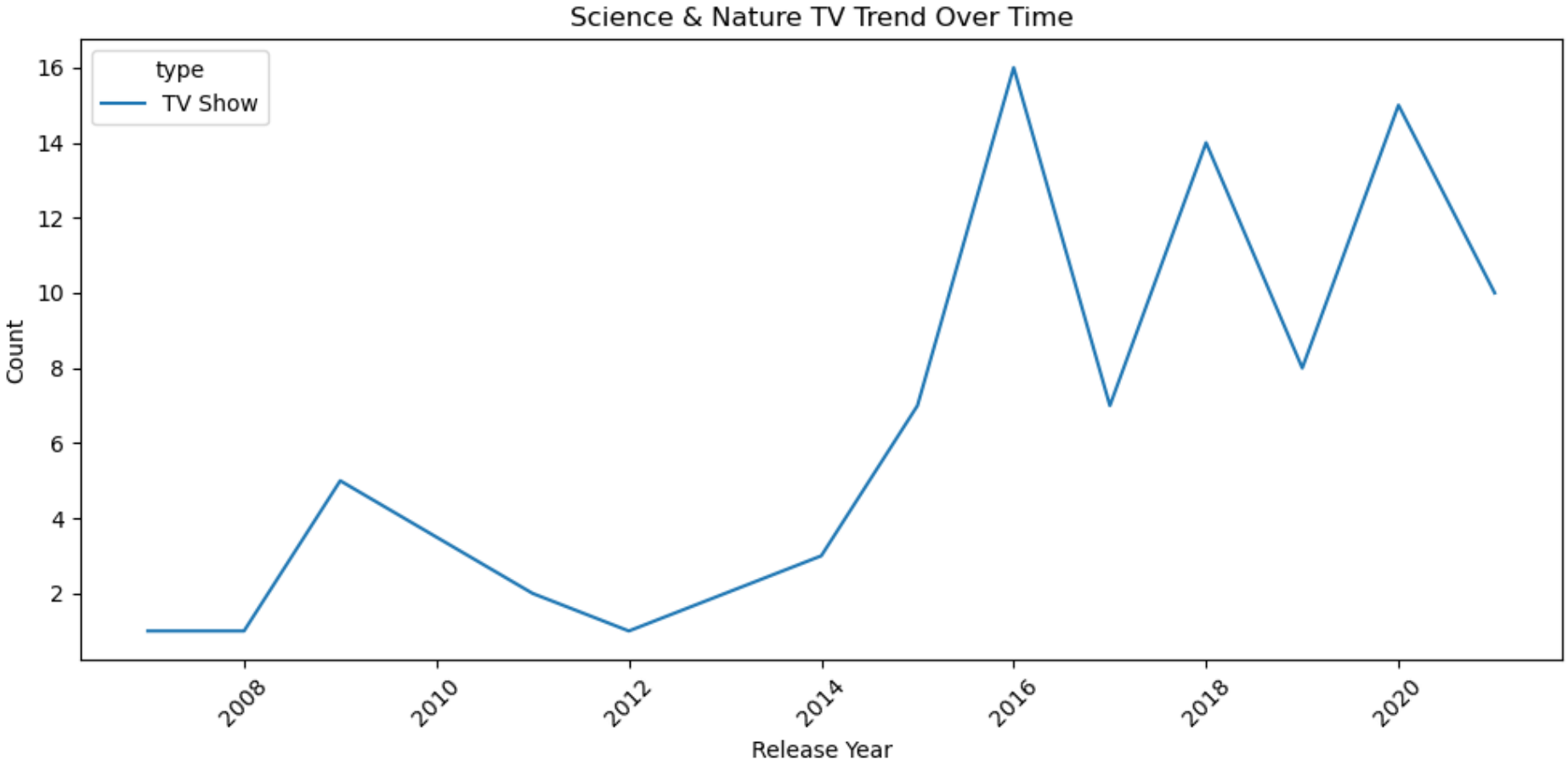

Classic Movies Trend Over Time
type
Movie
Count
Release Year
1
2
3
4
5
6
7
1940
1950
1960
1970
1980
1990
2000
2010
2020

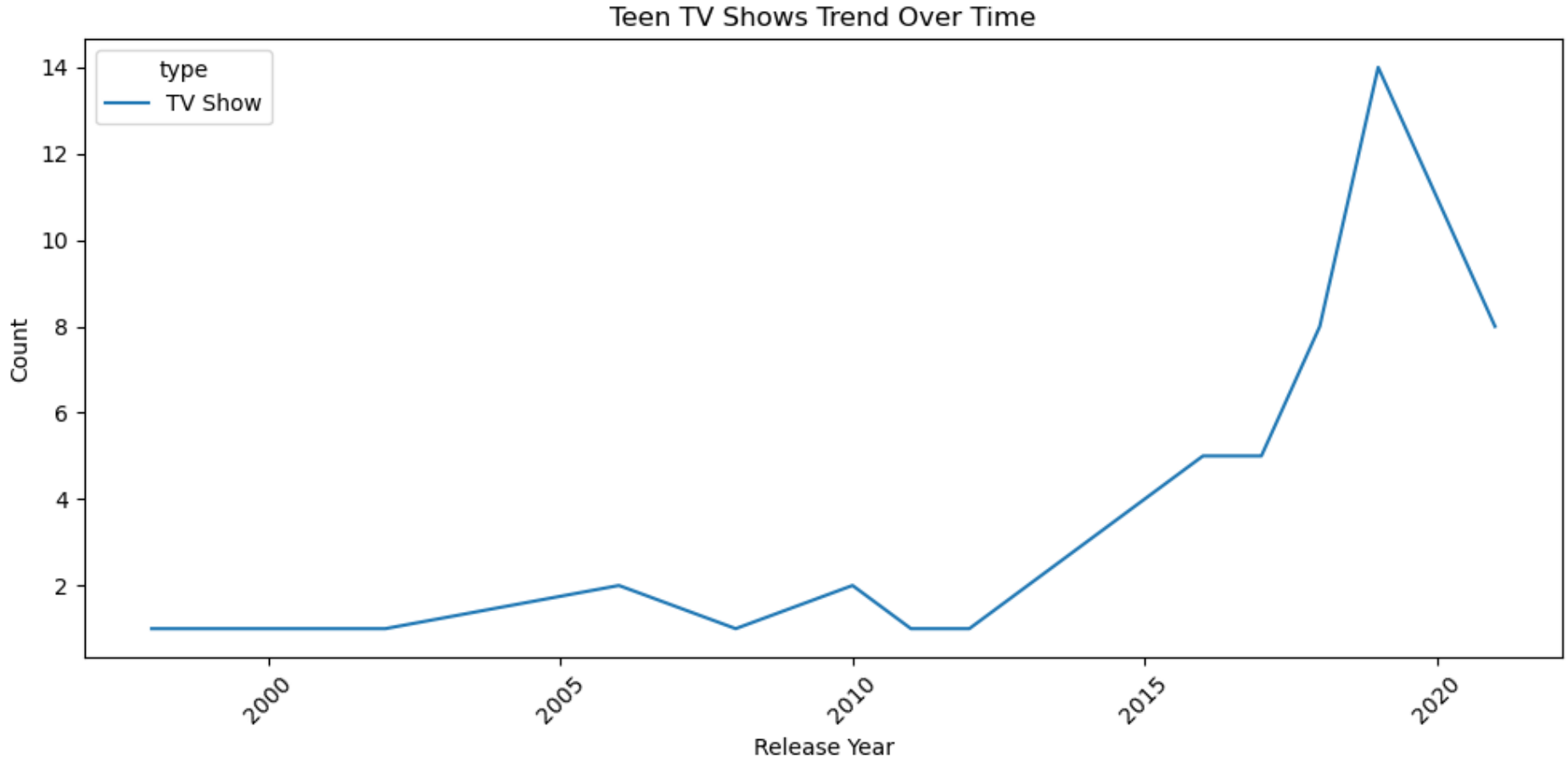

Anime Features Trend Over Time
type
Movie
Count
Release Year
1
2
3
4
5
6
7
8
1980
1990
2000
2010
2020

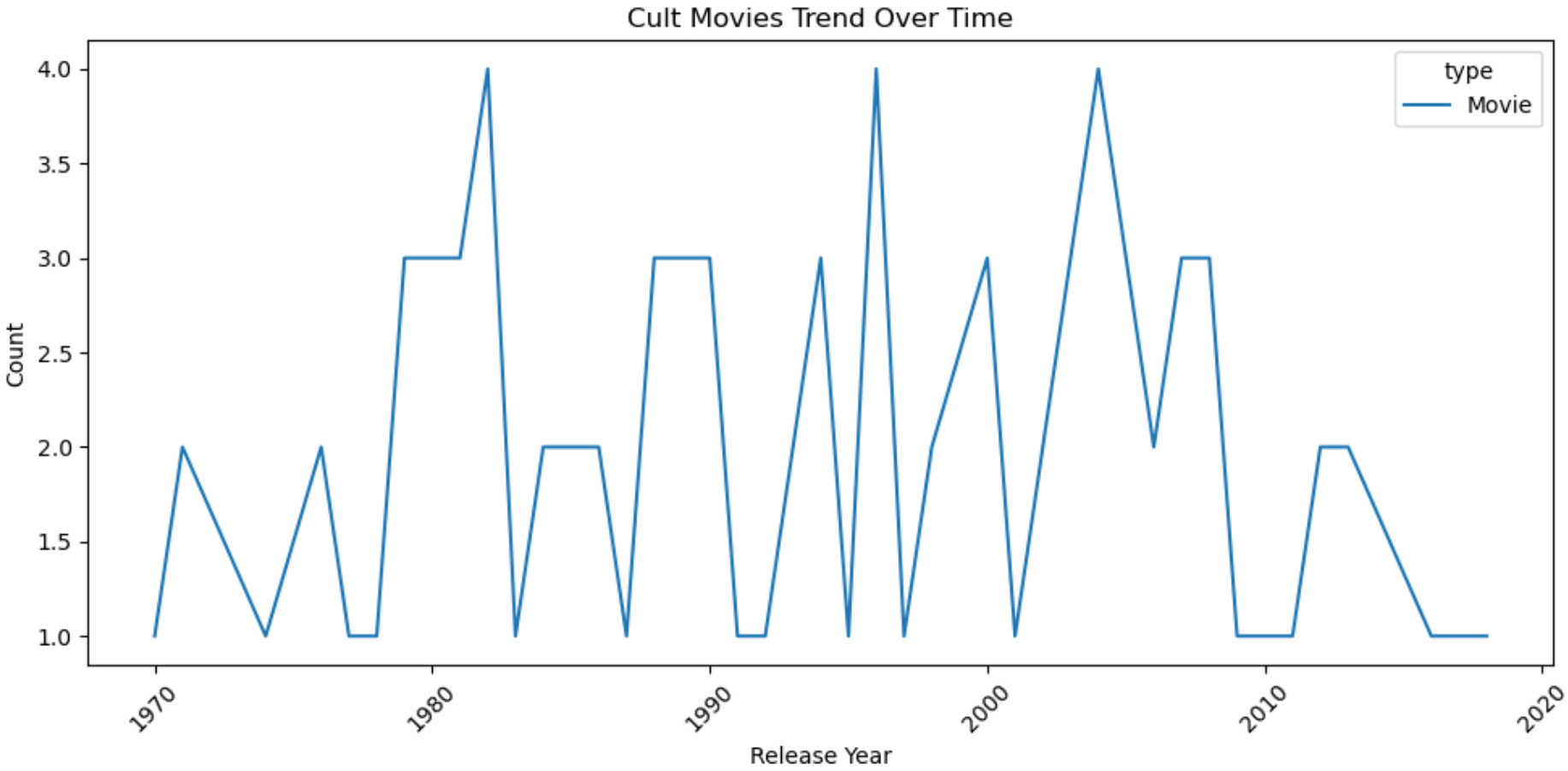

Sports Movies Trend Over Time
type
Movie
Count
0
5
10
15
20
25
30
1980
1990
2000
2010
2020
Release Year

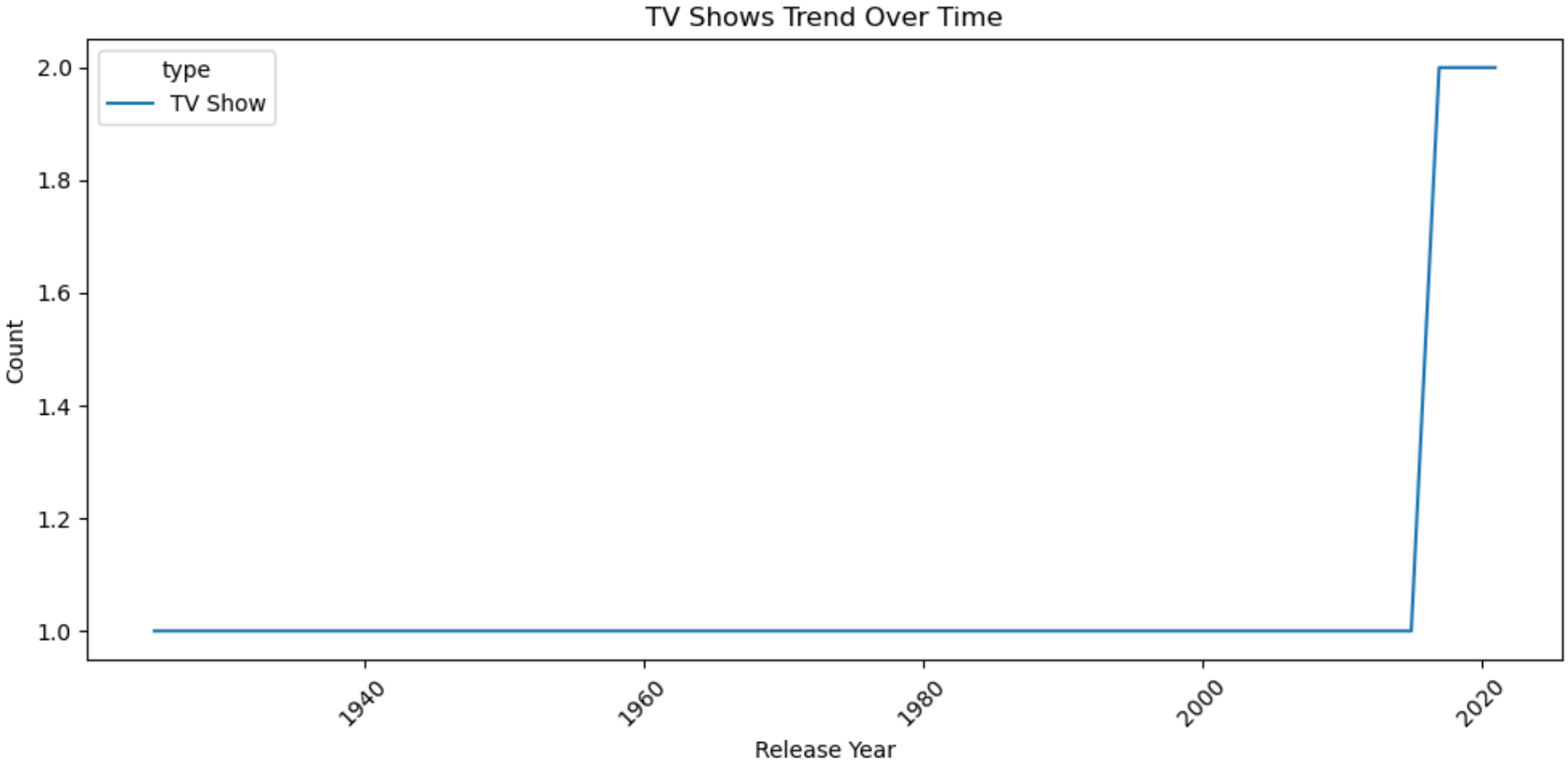

Anime Series Trend Over Time
type
TV Show
Count
Release Year
0
5
10
15
20
25
1980
1985
1990
1995
2000
2005
2010
2015
2020

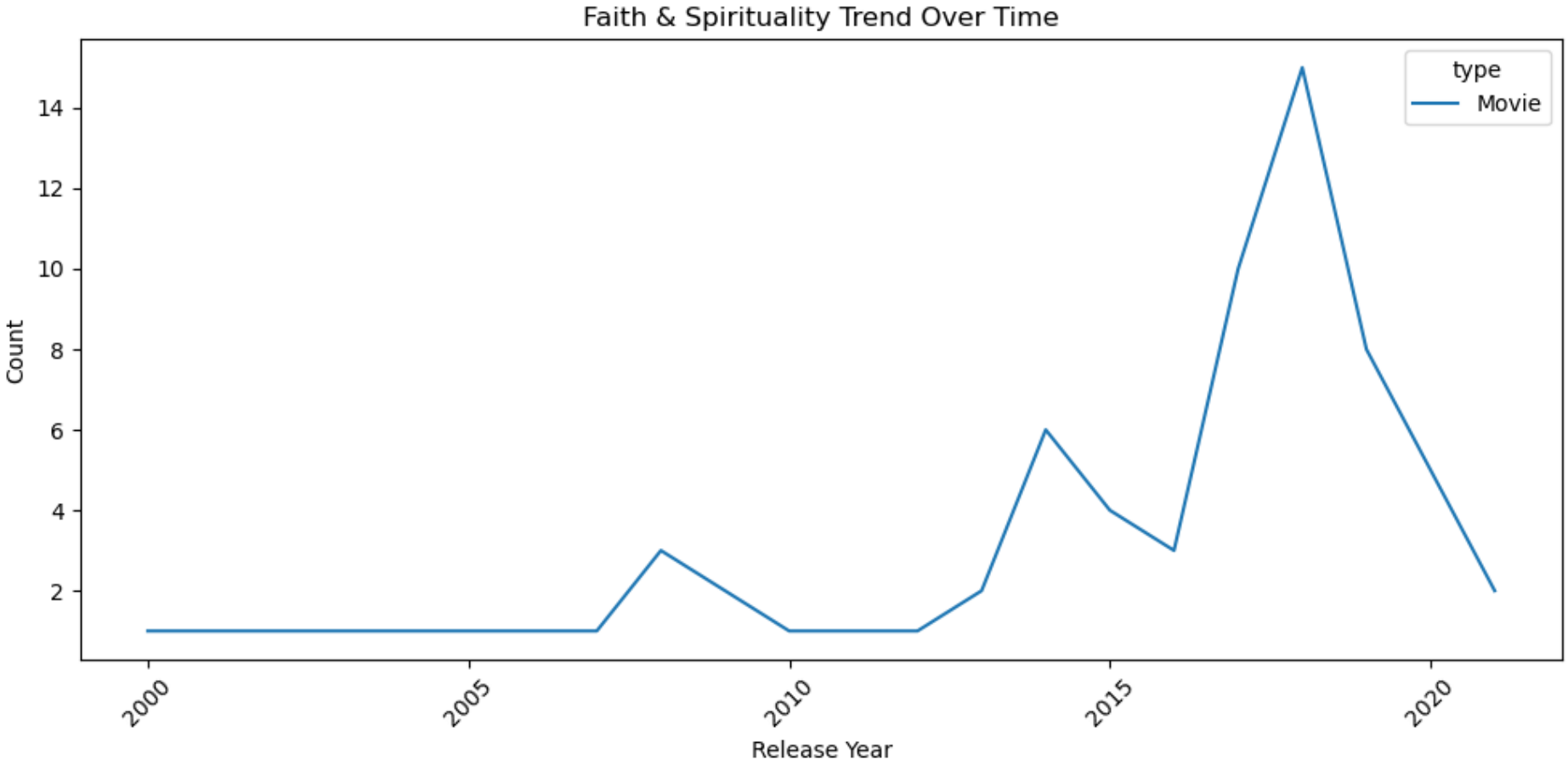

Korean TV Shows Trend Over Time
type
TV Show
Count
0
5
10
15
20
25
2010
2012
2014
2016
2018
2020
Release Year

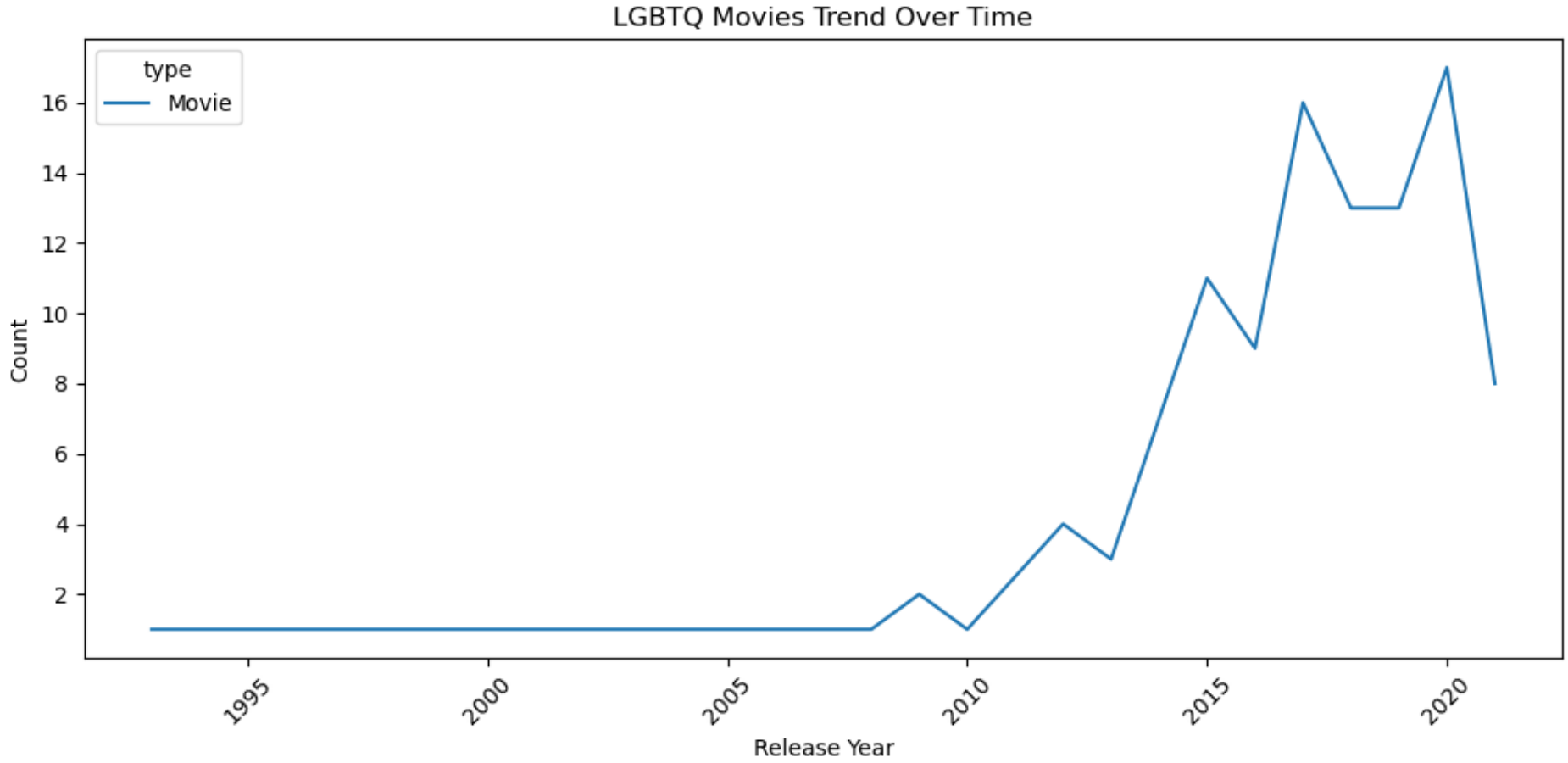

Science & Nature TV Trend Over Time
type
TV Show
Count
16
14
12
10
8
6
4
2
2008
2010
2012
2014
2016
2018
2020
Release Year

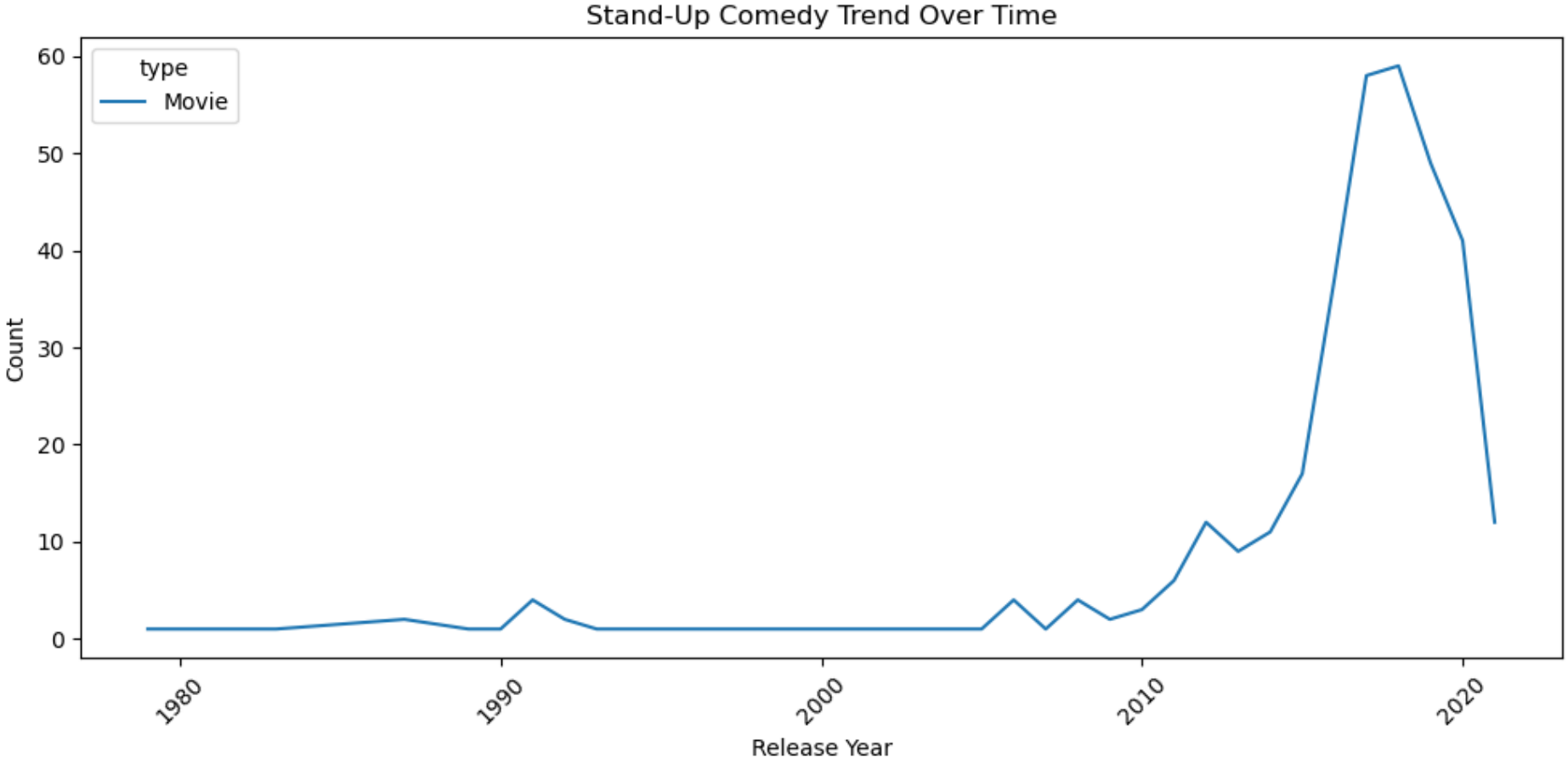

Teen TV Shows Trend Over Time
type
TV Show
Count
14
12
10
8
6
4
2
2000
2005
2010
2015
2020
Release Year

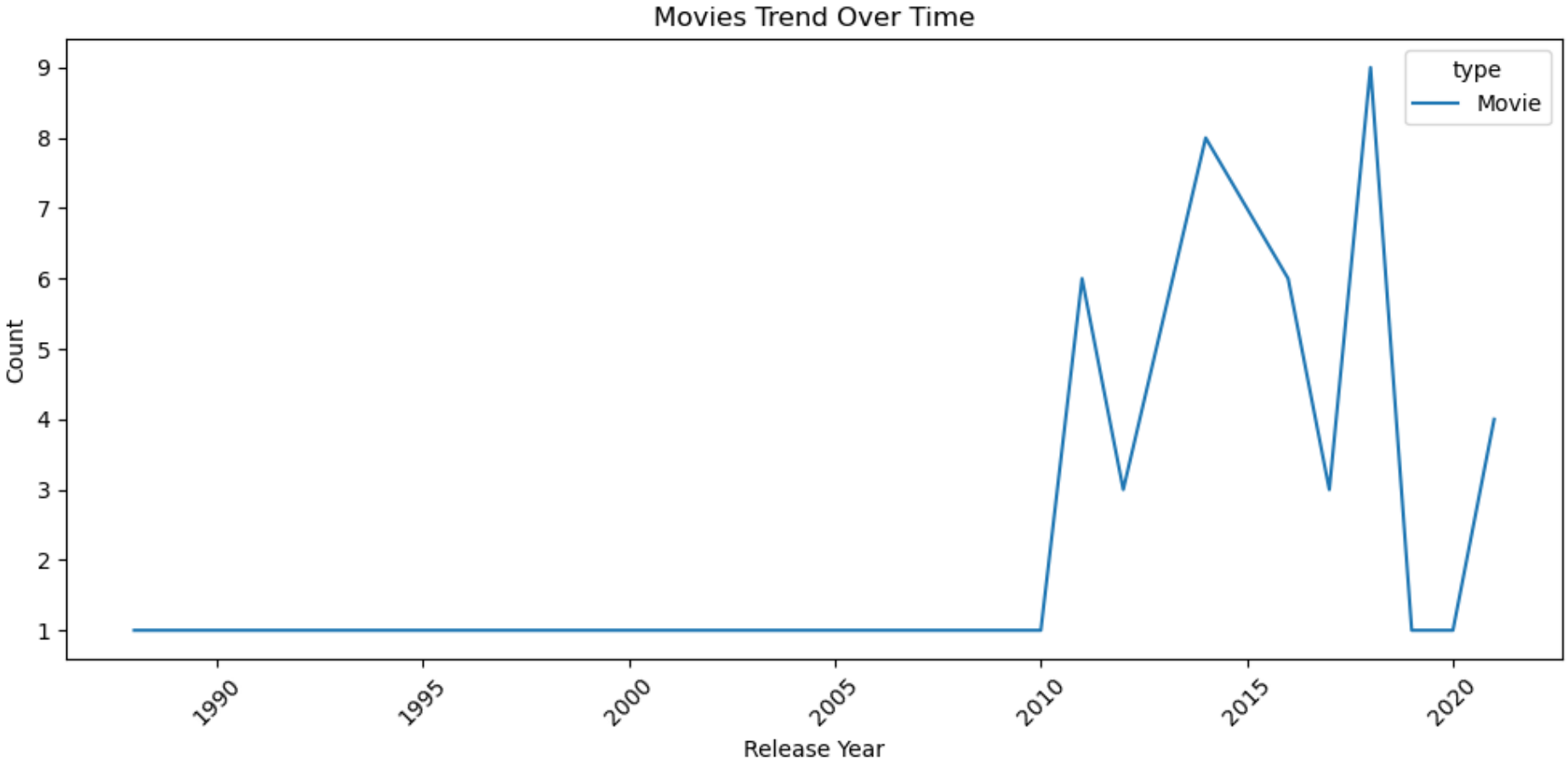

Cult Movies Trend Over Time
type
Movie
Count
4.0
3.5
3.0
2.5
2.0
1.5
1.0
1970
1980
1990
2000
2010
2020
Release Year

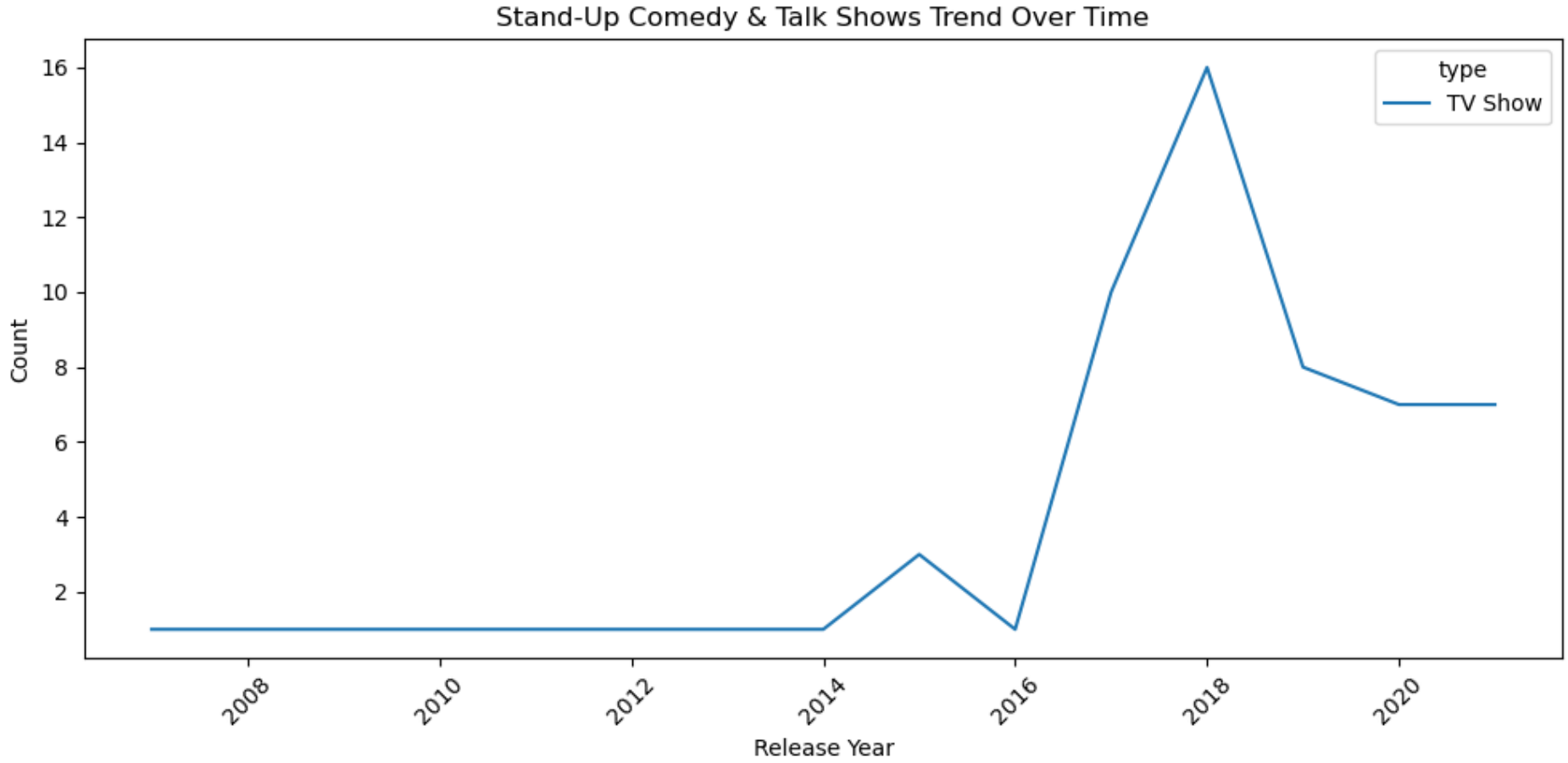

TV Shows Trend Over Time
type
TV Show
2.0
1.8
1.6
1.4
1.2
1.0
Count
1940
1960
1980
2000
2020
Release Year

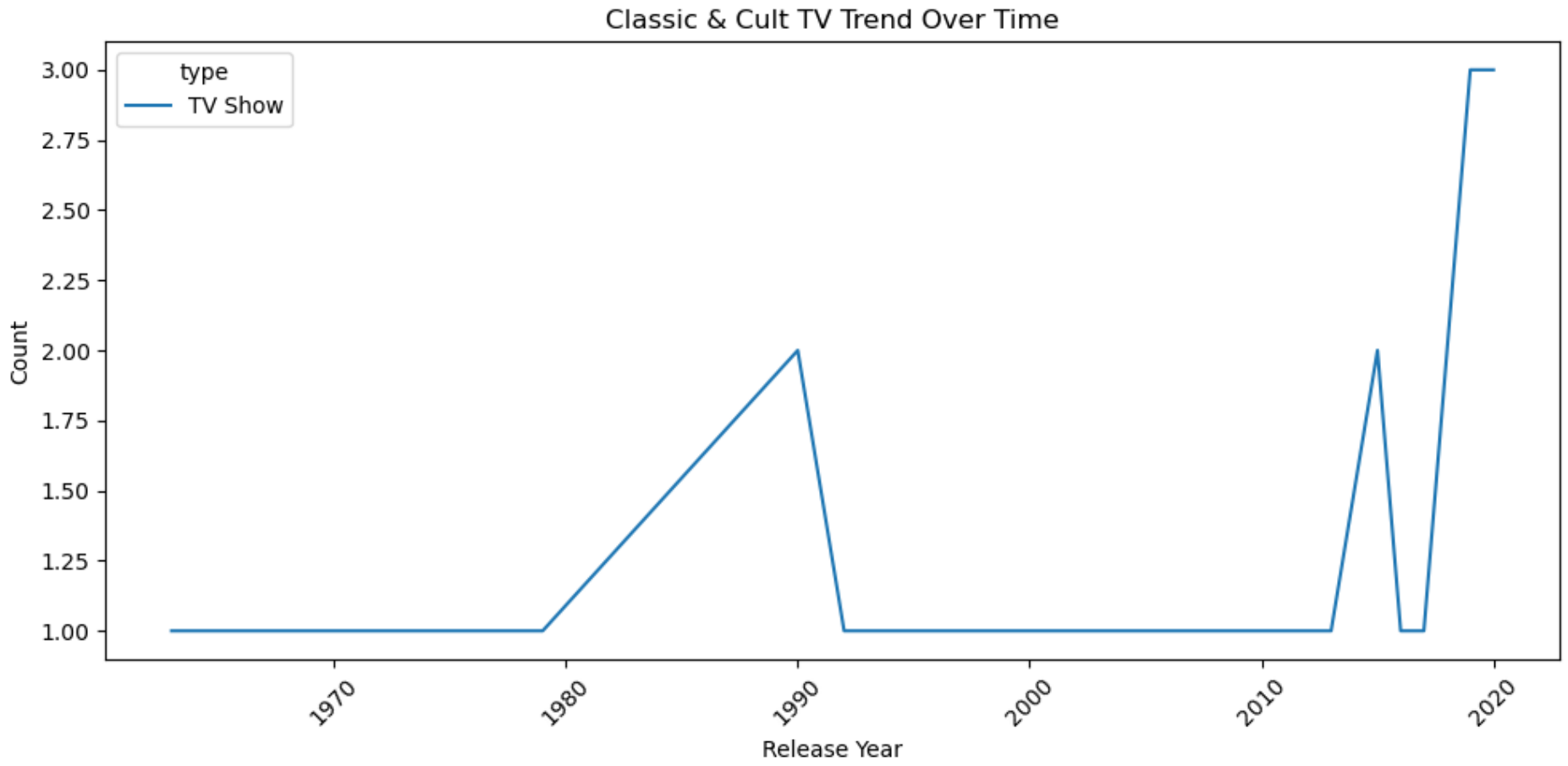

Faith & Spirituality Trend Over Time
type
Movie
Count
14
12
10
8
6
4
2
2000
2005
2010
2015
2020
Release Year


LGBTQ Movies Trend Over Time
type
Movie
Count
16
14
12
10
8
6
4
2
1995
2000
2005
2010
2015
2020
Release Year


Stand-Up Comedy Trend Over Time
type
Movie
Count
Release Year
0
10
20
30
40
50
60
1980
1990
2000
2010
2020

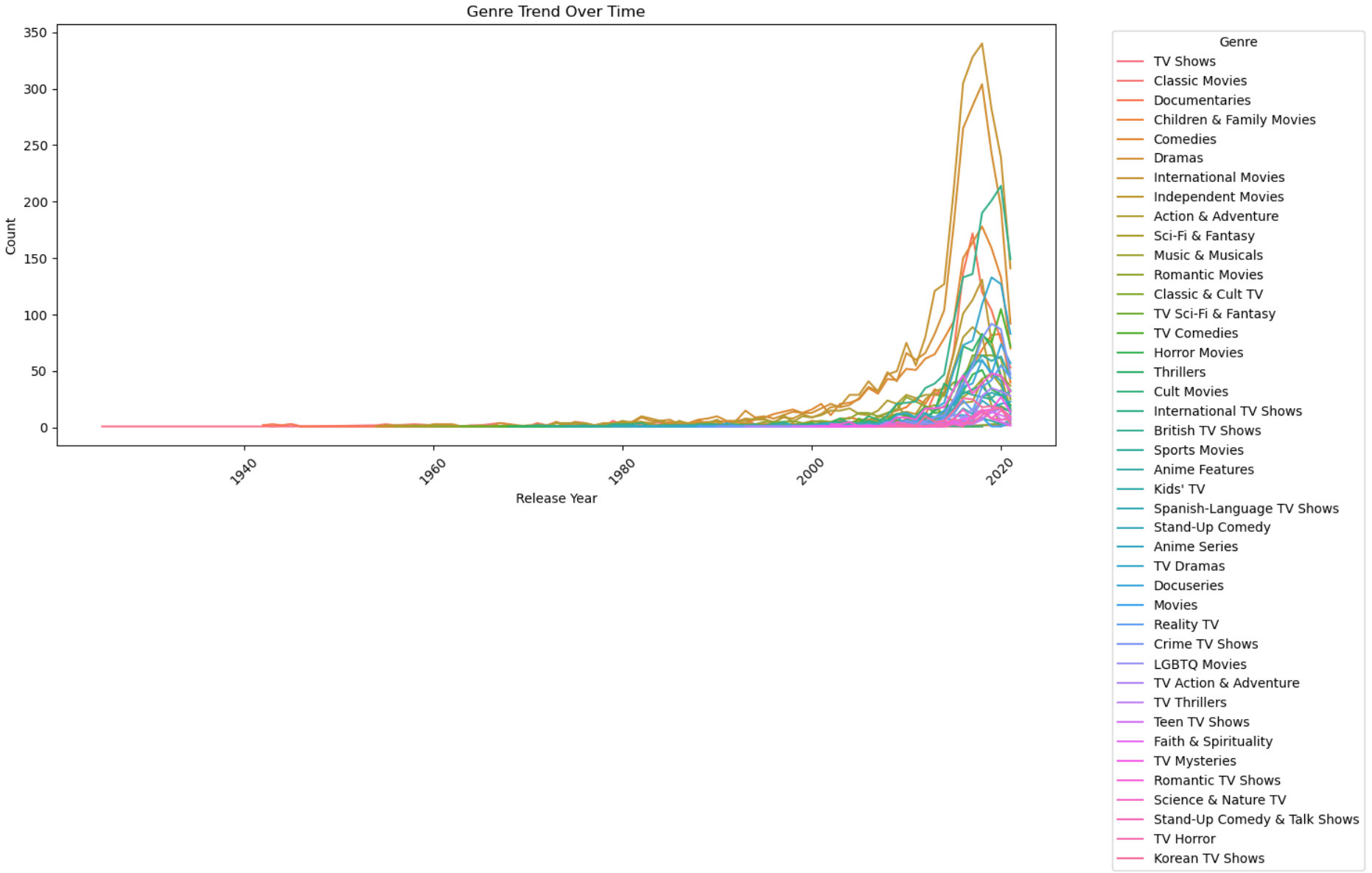

Movies Trend Over Time
type
Movie
Count
Release Year
1
2
3
4
5
6
7
8
9
1990
1995
2000
2005
2010
2015
2020


Stand-Up Comedy & Talk Shows Trend Over Time
type
TV Show
Count
16
14
12
10
8
6
4
2
2008
2010
2012
2014
2016
2018
2020
Release Year



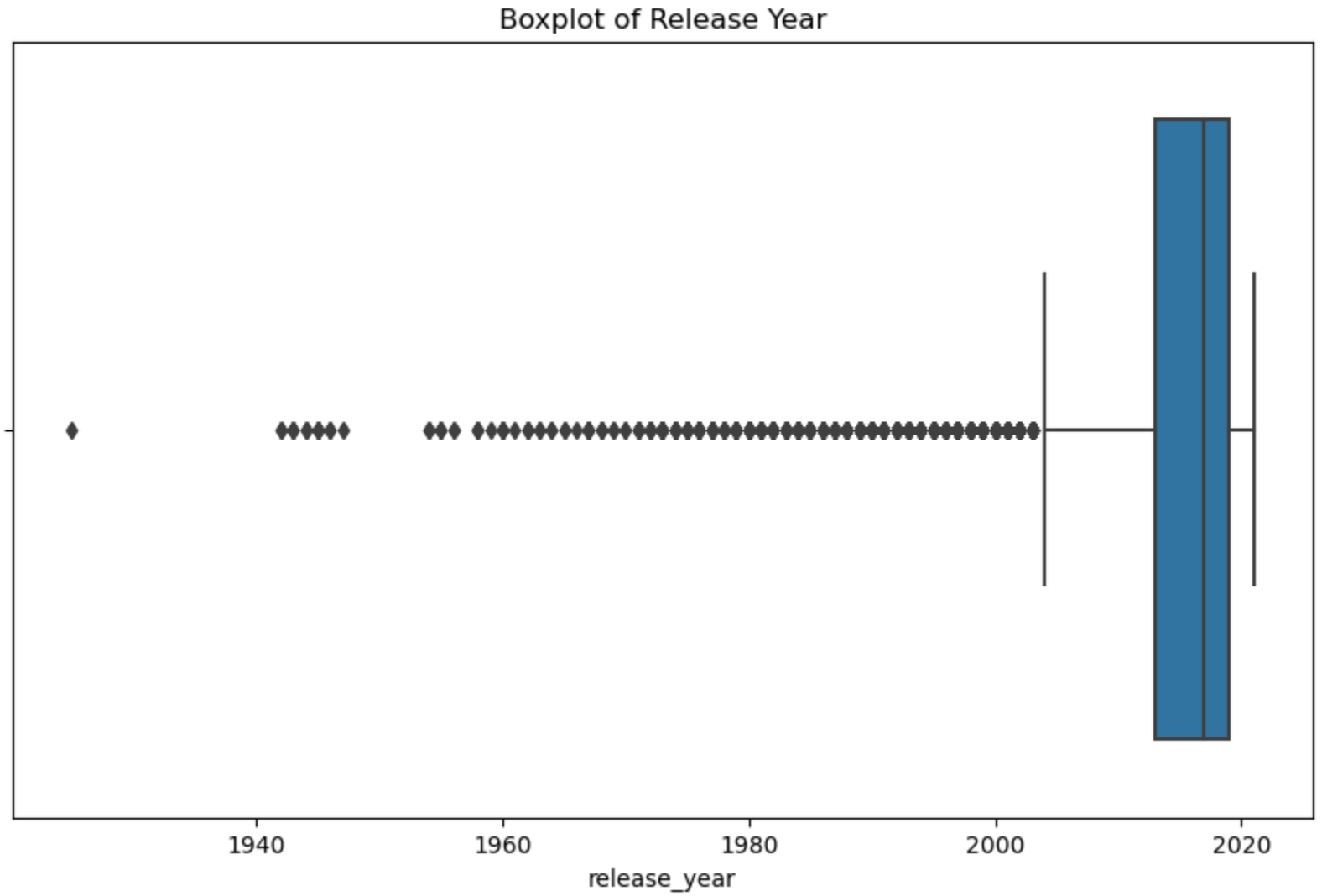
In [109…

```
# Genre Trend Over Time
df_exploded = df.assign(listed_in=df['listed_in'].str.split(', ')).explode('listed_in')
df_grouped = df_exploded.groupby(['release_year', 'listed_in']).size().reset_index(name='count')
plt.figure(figsize=(15, 8))
sns.lineplot(x='release_year', y='count', hue='listed_in', data=df_grouped)
plt.title('Genre Trend Over Time')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.legend(title='Genre', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [110…

```
#Outliers
plt.figure(figsize=(10, 6))
sns.boxplot(x='release_year', data=df)
plt.title('Boxplot of Release Year')
plt.show()
```



In [113…

```
#Outliers
#The Z-score is the number of standard deviations a data point is from the mean.
#A Z-score higher than 3 or lower than -3 usually indicates an outlier.

z_scores = np.abs(stats.zscore(df['release_year']))
outliers = z_scores > 3
total_outliers = np.sum(outliers)
print(f'Total outliers: {total_outliers}')
```

```
Total outliers: 216
```

In [115…

```
#Treatment of Outliers
#Method 1 - By removing the outliers
```

```
df_filtered = df[~outliers]
```

In [116…

```
#Treatment of Outliers
#Method 2 - By replacing with mean or median
median = df['release_year'].median()
df['release_year'] = np.where(outliers, median, df['release_year'])
```

In [117…

```
#Treatment of Outliers
#Method 3 - By capping the outliers
upper_bound = df['release_year'].quantile(0.99) # 99th percentile
df['release_year'] = np.where(df['release_year'] > upper_bound, upper_bound, df['release_year'])
```

In [119…

```
#Exploding the cast column to facilitate filtration on the basis of actors
df['cast'] = df['cast'].astype(str)
df_exploded = df.assign(cast=df['cast'].str.split(', ')).explode('cast')
df_exploded.reset_index(drop=True, inplace=True)
df_exploded['cast'].replace('nan', np.nan, inplace=True)
df_exploded
```

Out[119]:

	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	No Cast	United States	25-Sep-21	2020.0	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	No Director	Ama Qamata	South Africa	24-Sep-21	2021.0	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s2	TV Show	Blood & Water	No Director	Khosi Ngema	South Africa	24-Sep-21	2021.0	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
3	s2	TV Show	Blood & Water	No Director	Gail Mabalane	South Africa	24-Sep-21	2021.0	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
4	s2	TV Show	Blood & Water	No Director	Thabang Molaba	South Africa	24-Sep-21	2021.0	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
...	...	...	...	...	...	...	...	...	...	...	...	...
64835	s8807	Movie	Zubaan	Mozez Singh	Manish Chaudhary	India	2-Mar-19	2015.0	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...
64836	s8807	Movie	Zubaan	Mozez Singh	Meghna Malik	India	2-Mar-19	2015.0	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...
64837	s8807	Movie	Zubaan	Mozez Singh	Malkeet Rauni	India	2-Mar-19	2015.0	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...
64838	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	2-Mar-19	2015.0	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...
64839	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2-Mar-19	2015.0	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

64840 rows × 12 columns

In [120…

```
# Getting the number of Movies and TV Show for each Cast
cast_type_count = df_exploded.groupby(['cast', 'type']).size().unstack(fill_value=0)
cast_type_count.columns = [f'Number of {col}' for col in cast_type_count.columns]
cast_type_count.sort_values(by=['Number of Movie', 'Number of TV Show'], ascending=[False, False], inplace=True)
cast_type_count.reset_index(inplace=True)
cast_type_count
```

Out[120]:

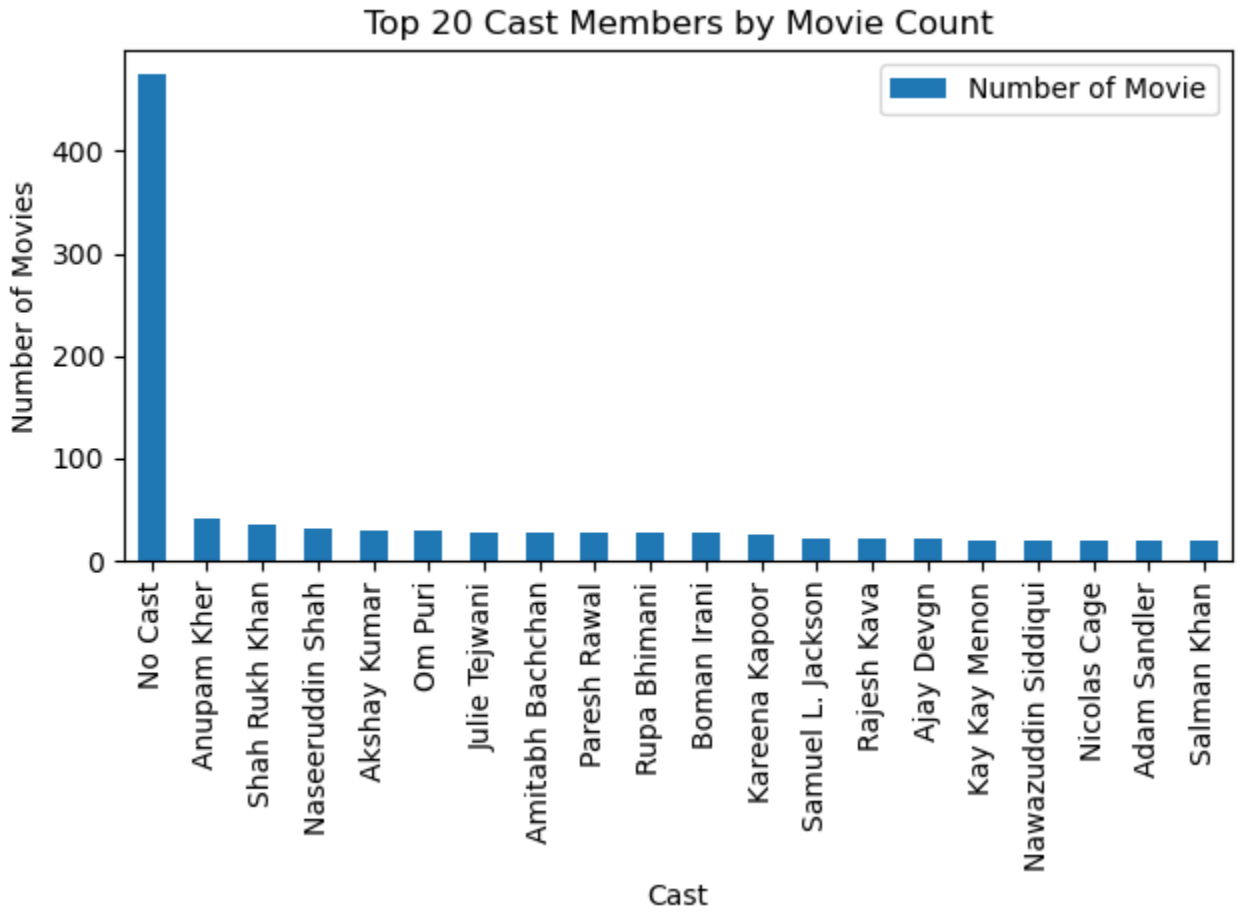
	cast	Number of Movie	Number of TV Show
0	No Cast	474	350
1	Anupam Kher	42	1
2	Shah Rukh Khan	35	0
3	Naseeruddin Shah	32	0
4	Akshay Kumar	30	0
...	...	...	...
36388	İpek Filiz Yazıcı	0	1
36389	İsmail Filiz	0	1
36390	Şafak Başkaya	0	1
36391	Şehsuvar Aktaş	0	1
36392	Şenay Gürler	0	1

36393 rows × 3 columns

In [122…

```
# Filtering for Movies
cast_movies_count = cast_type_count[['cast', 'Number of Movie']].sort_values(by='Number of Movie', ascending=False)
# Plotting the top 20 cast members by movie count
plt.figure(figsize=(12, 8))
cast_movies_count.head(20).plot(x='cast', kind='bar')
plt.title('Top 20 Cast Members by Movie Count')
plt.xlabel('Cast')
plt.ylabel('Number of Movies')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



In [123…

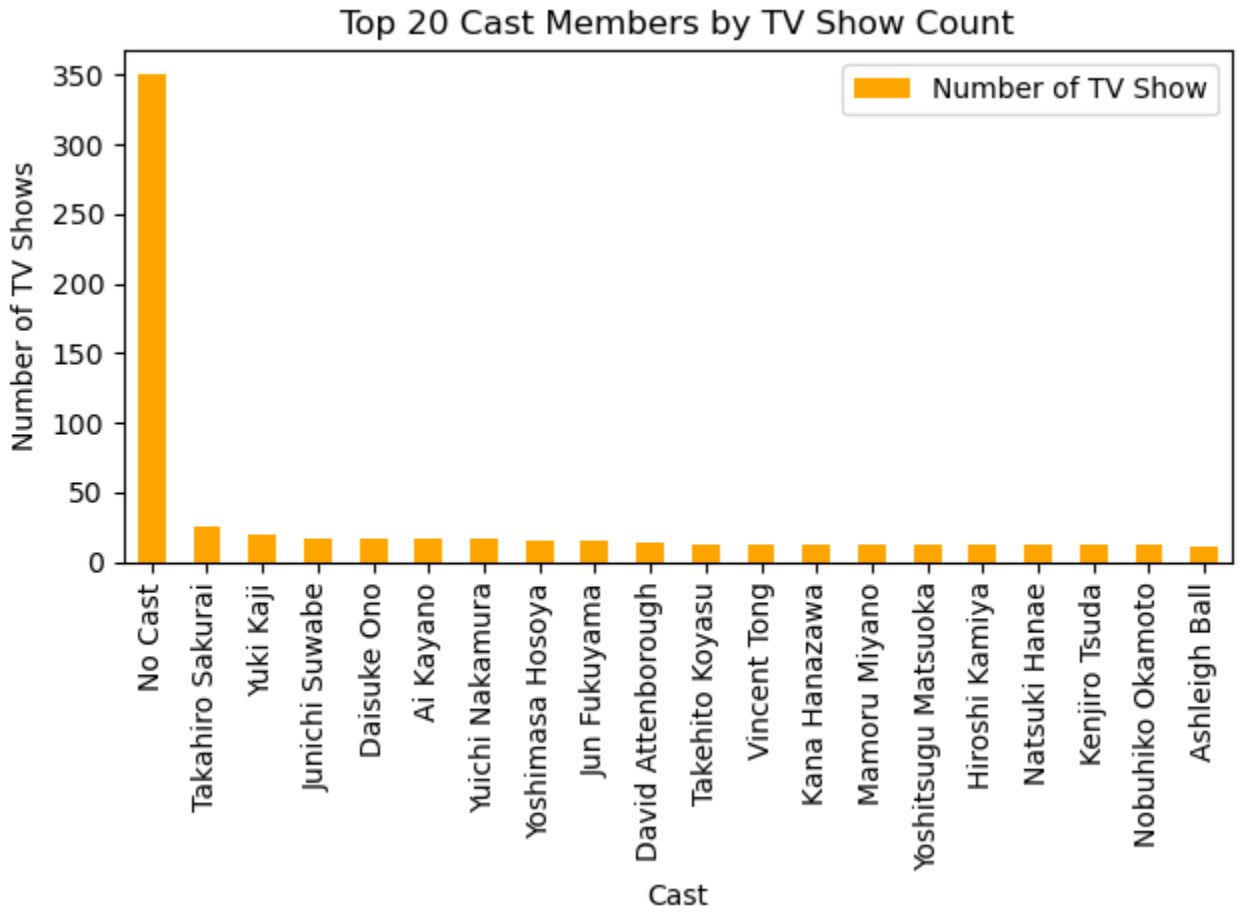
```
cast_movies_count = cast_type_count[['cast', 'Number of Movie']].sort_values(by='Number of Movie', ascending=False)
print(cast_movies_count.head(20))
```

	cast	Number of Movie
0	No Cast	474
1	Anupam Kher	42
2	Shah Rukh Khan	35
3	Naseeruddin Shah	32
4	Akshay Kumar	30
5	Om Puri	30
6	Julie Tejwani	28
7	Amitabh Bachchan	28
8	Paresh Rawal	28
9	Rupa Bhimani	27
10	Boman Irani	27
11	Kareena Kapoor	25
12	Samuel L. Jackson	22
13	Rajesh Kava	21
14	Ajay Devgn	21
15	Kay Kay Menon	20
16	Nawazuddin Siddiqui	20
17	Nicolas Cage	20
18	Adam Sandler	20
19	Salman Khan	20

In [124…

```
cast_tv_shows_count = cast_type_count[['cast', 'Number of TV Show']].sort_values(by='Number of TV Show', ascending=False)
plt.figure(figsize=(12, 8))
cast_tv_shows_count.head(20).plot(x='cast', kind='bar', color='orange')
plt.title('Top 20 Cast Members by TV Show Count')
plt.xlabel('Cast')
plt.ylabel('Number of TV Shows')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



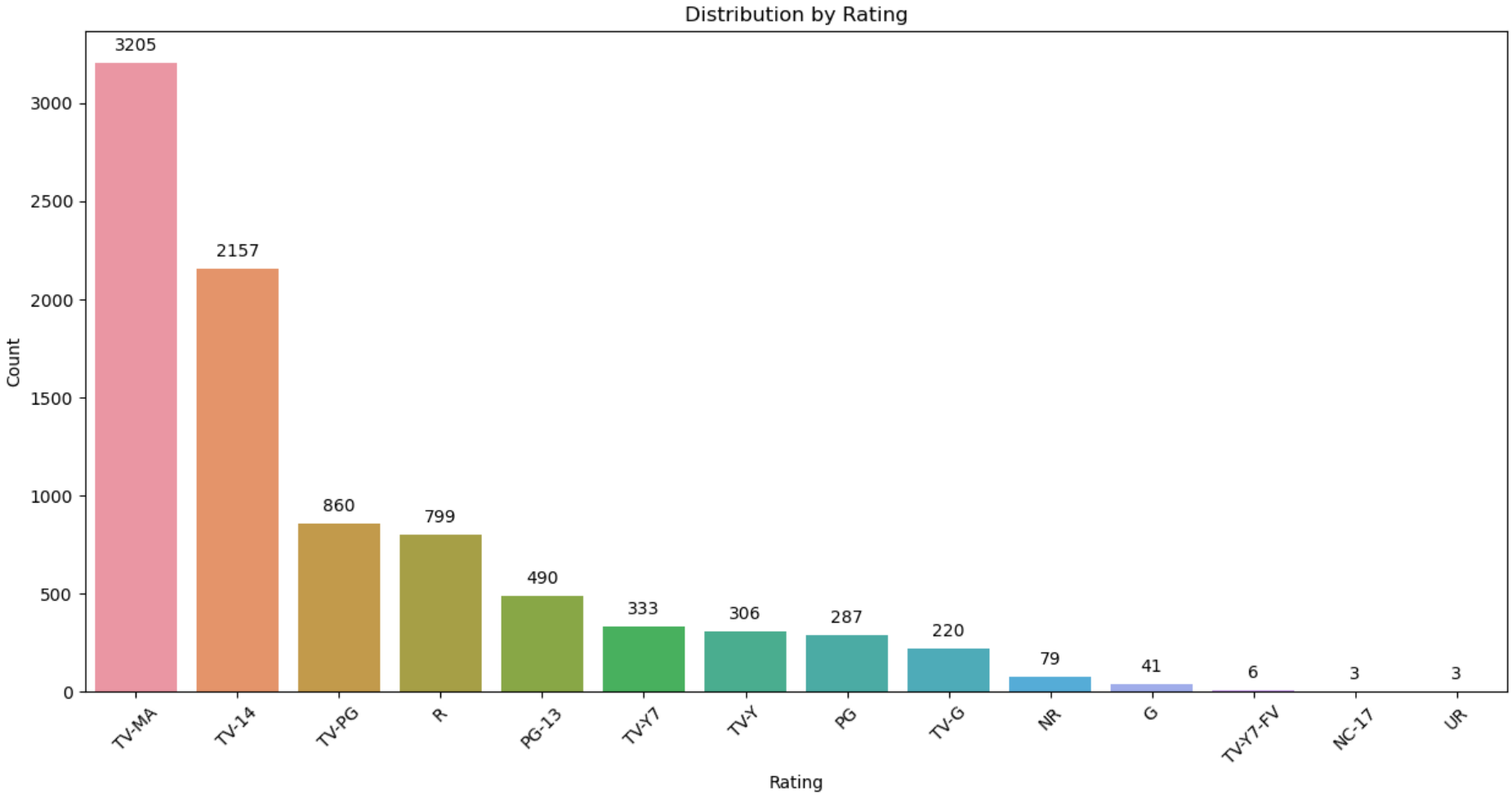
In [125…

```
cast_tv_shows_count = cast_type_count[['cast', 'Number of TV Show']].sort_values(by='Number of TV Show', ascending=False)
print(cast_tv_shows_count.head(20))
```

	cast	Number of TV Show
0	No Cast	350
538	Takahiro Sakurai	25
197	Yuki Kaji	19
1469	Junichi Suwabe	17
986	Daisuke Ono	17
3724	Ai Kayano	16
2254	Yuichi Nakamura	16
7321	Yoshimasa Hosoya	15
7320	Jun Fukuyama	15
738	David Attenborough	14
3725	Takehito Koyasu	13
79	Vincent Tong	13
987	Kana Hanazawa	13
739	Mamoru Miyano	13
7322	Yoshitsugu Matsuoka	13
2255	Hiroshi Kamiya	13
3727	Natsuki Hanae	12
3726	Kenjiro Tsuda	12
25945	Nobuhiko Okamoto	12
198	Ashleigh Ball	11

In [126…

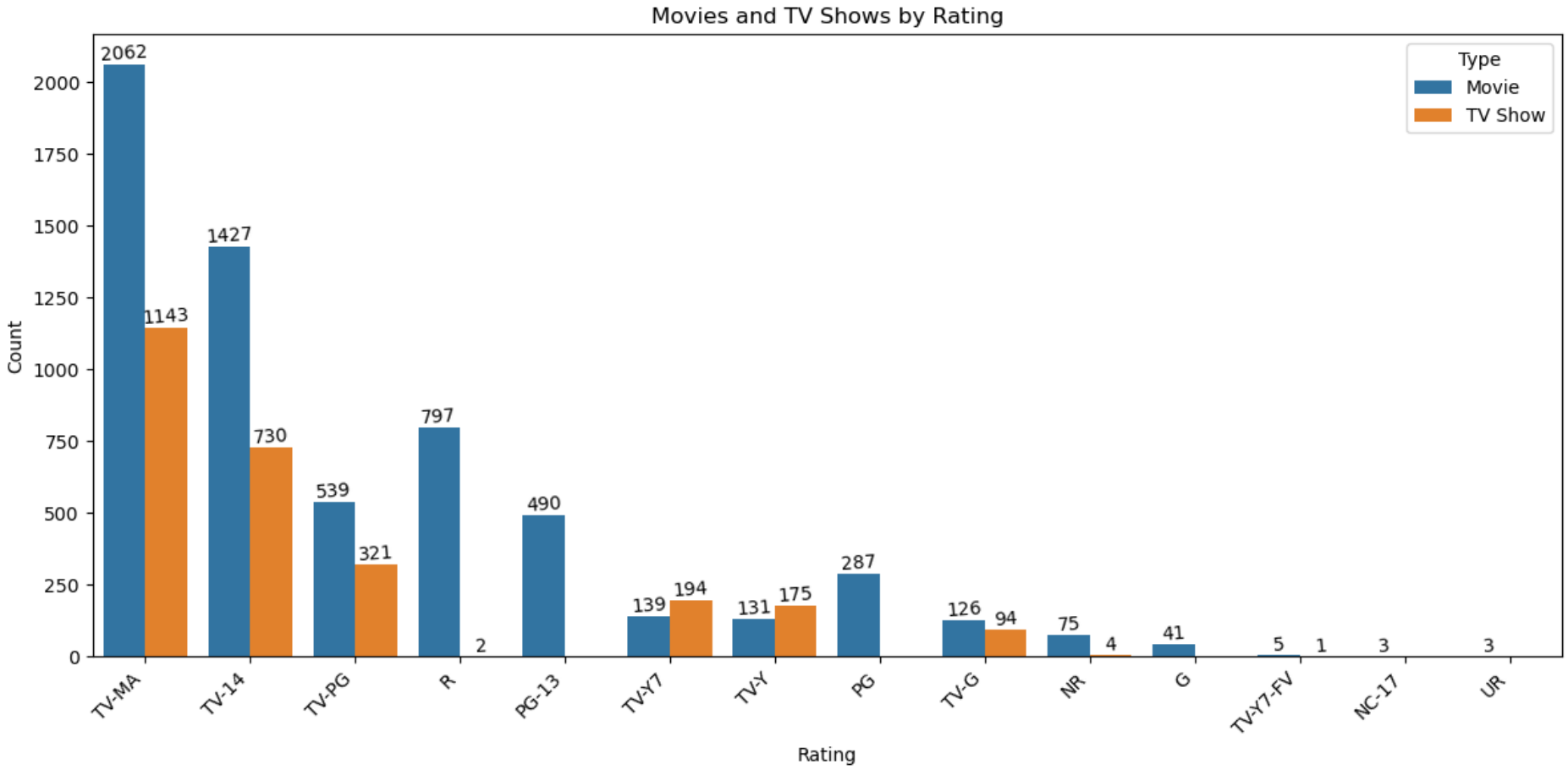
```
plt.figure(figsize=(15, 7))
ax = sns.countplot(data=df, x='rating', order=df['rating'].value_counts().index)
plt.title('Distribution by Rating')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
for p in ax.patches:
 ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', xytext=(0, 10), textcoords='offset points')
plt.show()
```



In [133…

```
tv_shows_df = df[df['type'] == 'TV Show']
movies_df = df[df['type'] == 'Movie']
# Combining the two dataframes
combined_df = pd.concat([movies_df, tv_shows_df])
# Plotting TV shows and movies based on ratings with bin labels at the top
plt.figure(figsize=(12, 6))
ax = sns.countplot(data=combined_df, x='rating', hue='type', order=combined_df['rating'].value_counts().index)
plt.title('Movies and TV Shows by Rating')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
# Adding bin labels at the top of the bins at an angle of 45 degrees (as integers)
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(str(int(height)), (p.get_x() + p.get_width() / 2., height), ha='center', va='bottom', fontsize=10, rotation=4)
```

```
plt.legend(title='Type')
plt.tight_layout()
plt.show()
```

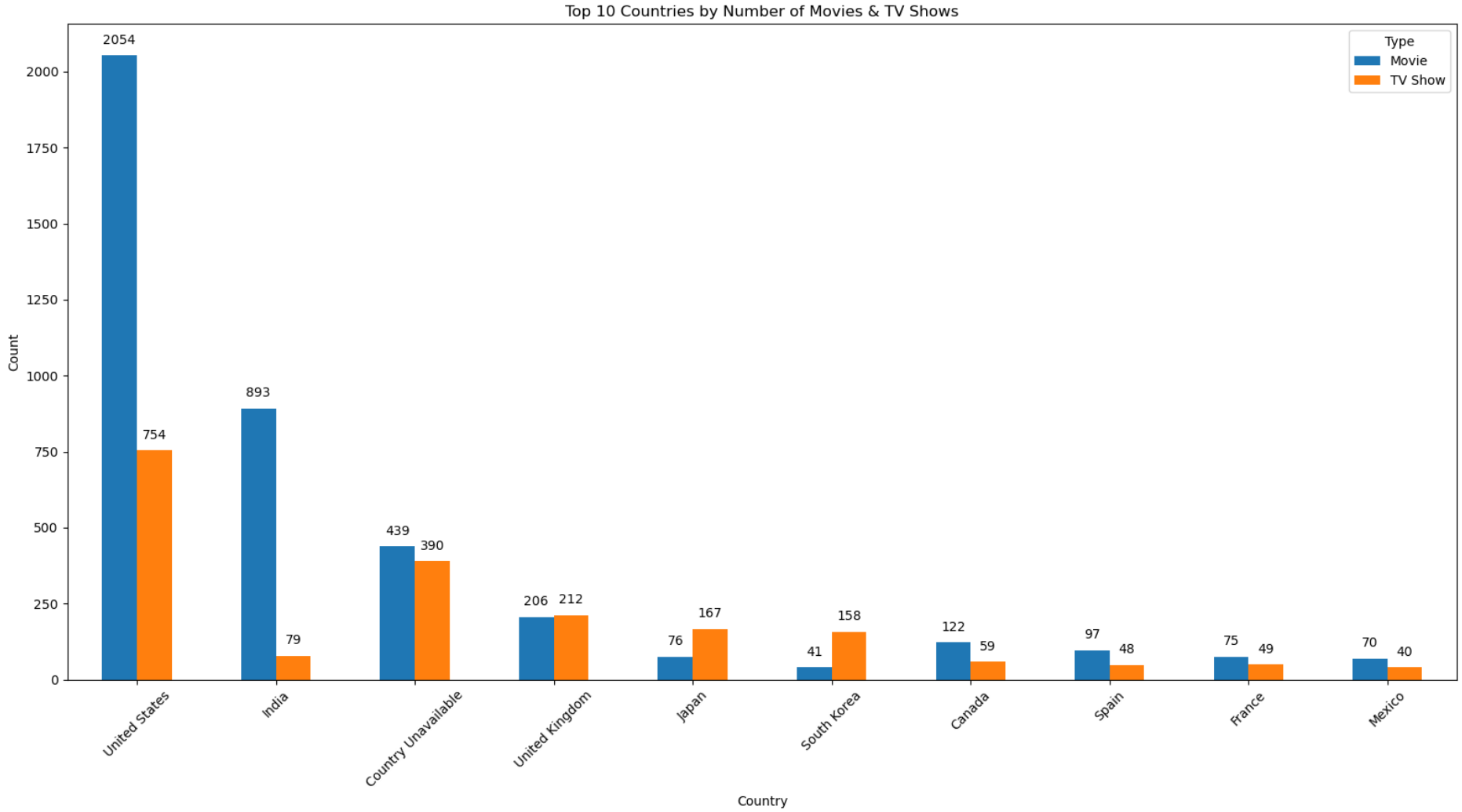


In [137…

```
country_type_counts = df.groupby(['country', 'type']).size().unstack().fillna(0)
country_type_counts['Movie'] = country_type_counts['Movie'].astype(int)
country_type_counts['TV Show'] = country_type_counts['TV Show'].astype(int)
country_type_counts['total'] = country_type_counts.sum(axis=1)
sorted_countries = country_type_counts.sort_values(by='total', ascending=False)
sorted_countries['total'] = sorted_countries['total'].astype(int)
top_10 = sorted_countries.head(10)
plt.figure(figsize=(16, 9))
ax = top_10[['Movie', 'TV Show']].plot(kind='bar', stacked=False, figsize=(16, 9))
plt.title('Top 10 Countries by Number of Movies & TV Shows')
plt.xlabel('Country')
plt.ylabel('Count')
plt.legend(title='Type')
plt.xticks(rotation=45)
```

```
for p in ax.patches:
 width, height = p.get_width(), p.get_height()
 x, y = p.get_xy()
 ax.text(x + width / 2,
 y + height + 50,
 '{:.0f}'.format(height),
 horizontalalignment='center',
 verticalalignment='center')
plt.tight_layout()
plt.show()
```

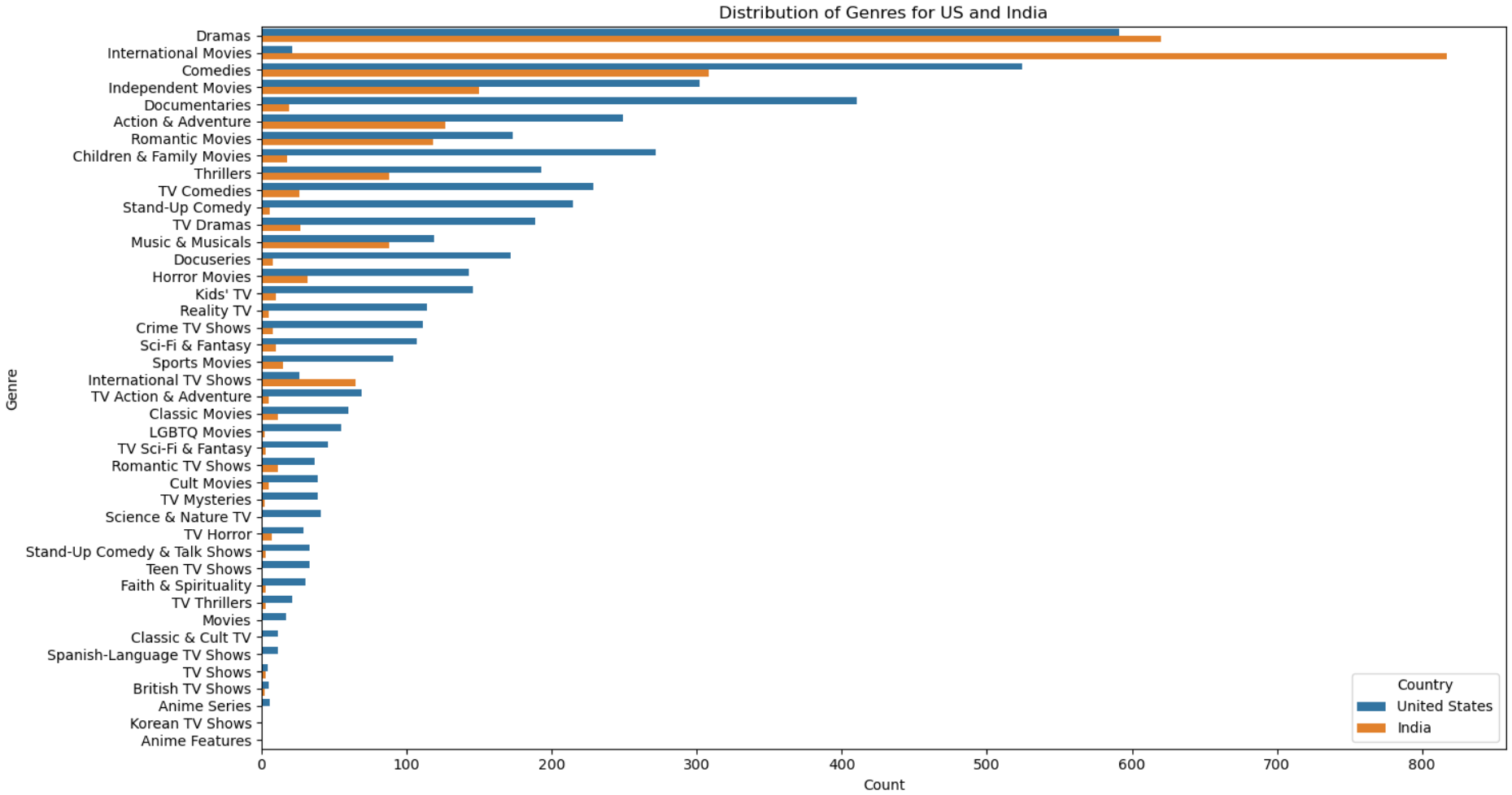
```
<Figure size 1600x900 with 0 Axes>
```



In [138…

```
us_india_df = df[df['country'].isin(['United States', 'India'])]
us_india_df = us_india_df.assign(genre=us_india_df['listed_in'].str.split(', ')).explode('genre')
```

```
plt.figure(figsize=(15, 8))
sns.countplot(data=us_india_df, y='genre', hue='country', order=us_india_df['genre'].value_counts().index)
plt.title('Distribution of Genres for US and India')
plt.xlabel('Count')
plt.ylabel('Genre')
plt.legend(title='Country')
plt.tight_layout()
plt.show()
```

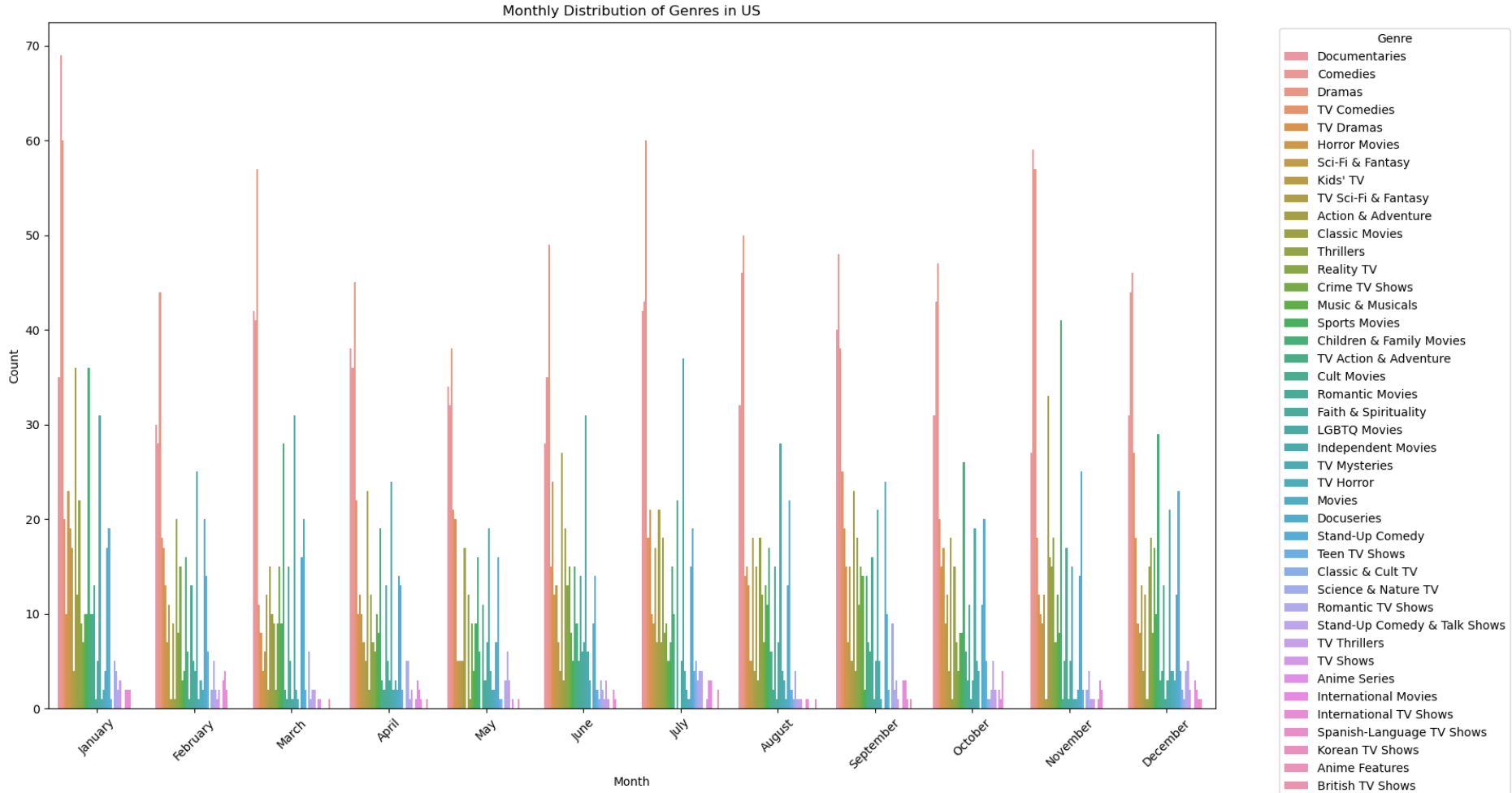


In [141…

```
us_df = df[df['country'] == 'United States']
us_df['month_added'] = pd.to_datetime(us_df['date_added']).dt.month_name()
us_df = us_df.assign(genre=us_df['listed_in'].str.split(', ')).explode('genre')
plt.figure(figsize=(18, 10))
sns.countplot(data=us_df, x='month_added', hue='genre', order=["January", "February", "March", "April", "May", "June", "July", "Au
plt.title('Monthly Distribution of Genres in US')
plt.xlabel('Month')
```

```
plt.ylabel('Count')
plt.legend(title='Genre', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_14228\1865093125.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  us_df['month_added'] = pd.to_datetime(us_df['date_added']).dt.month_name()
```

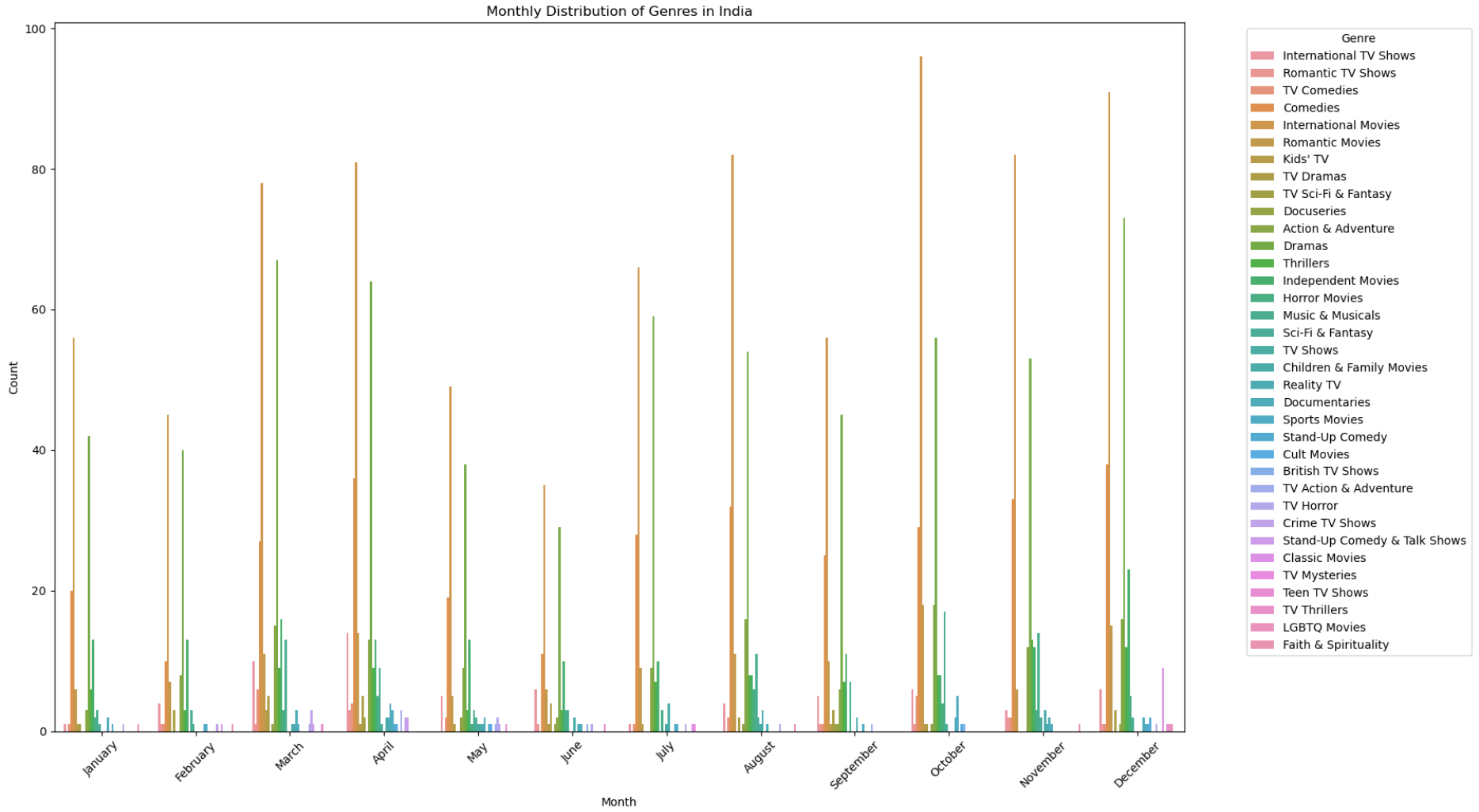


In [142…

```
india_df = df[df['country'] == 'India']
india_df['month_added'] = pd.to_datetime(india_df['date_added']).dt.month_name()
india_df = india_df.assign(genre=india_df['listed_in'].str.split(', ')).explode('genre')
```

```
plt.figure(figsize=(18, 10))
sns.countplot(data=india_df, x='month_added', hue='genre', order=["January", "February", "March", "April", "May", "June", "July",
plt.title('Monthly Distribution of Genres in India')
plt.xlabel('Month')
plt.ylabel('Count')
plt.legend(title='Genre', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

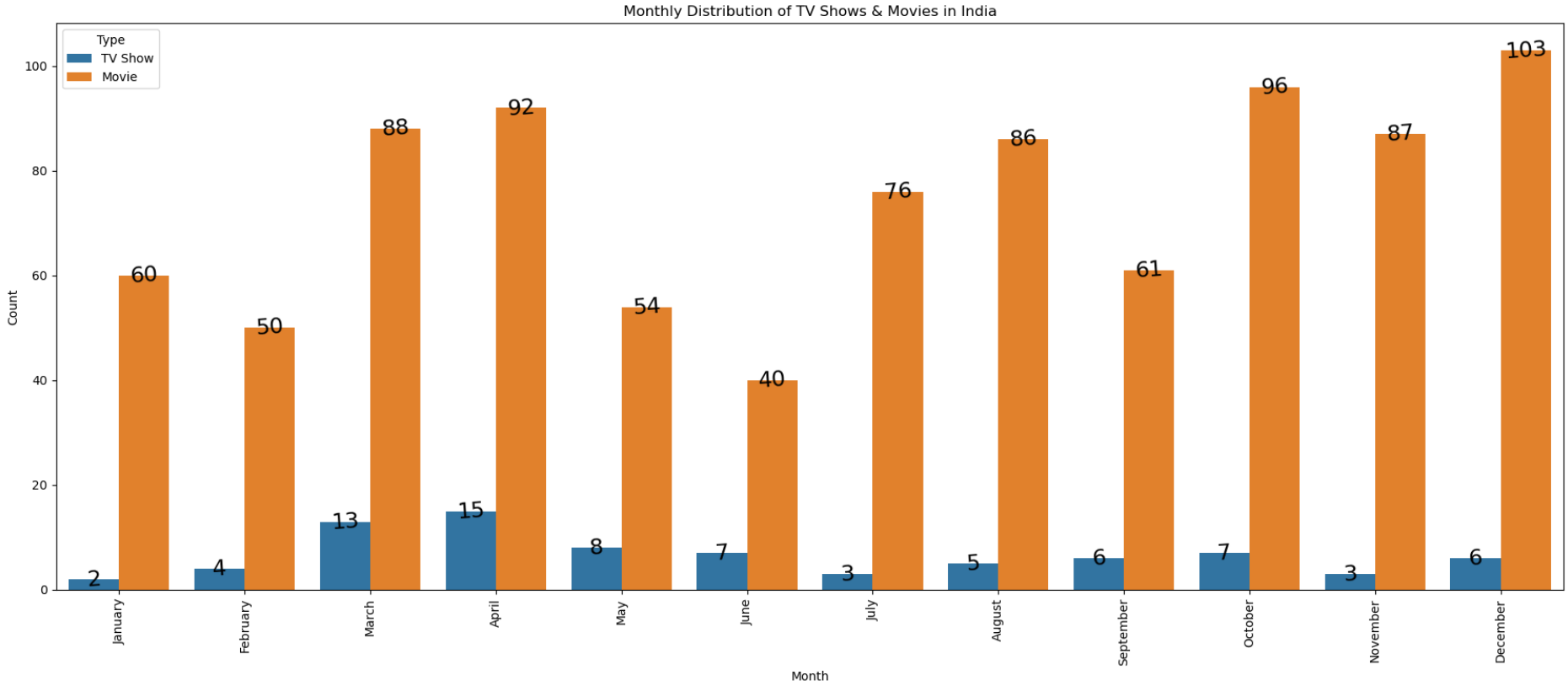
```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_14228\2866049256.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  india_df['month_added'] = pd.to_datetime(india_df['date_added']).dt.month_name()
```



In [149…

```
india_df = df[df['country'] == 'India']
india_df['month_added'] = pd.to_datetime(india_df['date_added']).dt.month_name()
plt.figure(figsize=(18, 8))
ax_india = sns.countplot(data=india_df, x='month_added', hue='type', order=["January", "February", "March", "April", "May", "June"
plt.title('Monthly Distribution of TV Shows & Movies in India')
plt.xlabel('Month')
plt.ylabel('Count')
plt.legend(title='Type')
plt.xticks(rotation=90)
plt.tight_layout()
for p in ax_india.patches:
 ax_india.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='center',fontsize=1
plt.show()
```

```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_14228\3724353305.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  india_df['month_added'] = pd.to_datetime(india_df['date_added']).dt.month_name()
```



Data Analysis

Movies Vs. TV Shows

Netflix contains more movies than TV shows.

It indicates that Movies has more audience than TV Shows.

Releasing Year Analysis

Most no:of releases are during the period 2000-2020

Highest no: of releases are in the year 2018

Released country Analysis

Netflix has a greater no: of contents from United States.

India stands out in the second position.

Top Geners of TV Shows / Movies

Netflix has most number of TV Shows and Movies under International Gener followed by Dramas

Top Ratings Movies / TV Shows

TV-MA has most number of Movies and TV Shows followed by TV-14

Top Directors on Netflix in terms of releasing number of Movies / TV Shows

Rajiv Chilaka released most number of Movies and TV Shows followed by Jan Suter

Netflix age rating for U. S.

For Kids

TV-Y | This category is appropriate for all kids.

TV-Y7 | This category is appropriate for all kids above the age of 7.

G | It means it is suitable for all general audience

TV-G | It is suitable for the general audience

PG | It means the movie\series under this category requires parental guidance

TV-PG | Again, this means the movie\ series requires parental guidance.

For Teenagers

PG-13 | It means the series\ movie may not be suitable for teens below 12

TV-14 | It means the series may not be suitable for teens under the age of 14.

For Adults

R | R stands for restricted. It may not be suitable for people under 17.

TV-MA | Suitable for a mature audience, not suitable for people under 17

NC-17 | Not suitable for ages under 17.