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

In [2]:
```
df = pd.read_csv('netflix_titles.csv')
```

In [3]:
```
df.head()
```

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

In [93]:
```
df.describe()
```

Out[93]:

	date_added	release_year	rating	date_added_year
count	5328	5332.000000	0.0	5328.000000
mean	2019-04-29 02:27:01.621621504	2012.742123	NaN	2018.826764
min	2008-01-01 00:00:00	1942.000000	NaN	2008.000000
25%	2018-04-06 18:00:00	2011.000000	NaN	2018.000000
50%	2019-06-18 00:00:00	2016.000000	NaN	2019.000000
75%	2020-06-27 06:00:00	2018.000000	NaN	2020.000000
max	2021-09-24 00:00:00	2021.000000	NaN	2021.000000
std	NaN	9.625831	NaN	1.540584

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

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

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

Out[5]:
```
105684
```

In [6]:
```
df.columns
```

Out[6]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

In [7]:
```
df.dtypes
```

Out[7]:
```
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 [10]:
```
df[df.duplicated()]
```

Out[10]:

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
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 [18]:
```
df.dropna(inplace=True)
```

In [20]:
```
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce', infer_datetime_format=True)
```

In [24]:
```
df.info()
```

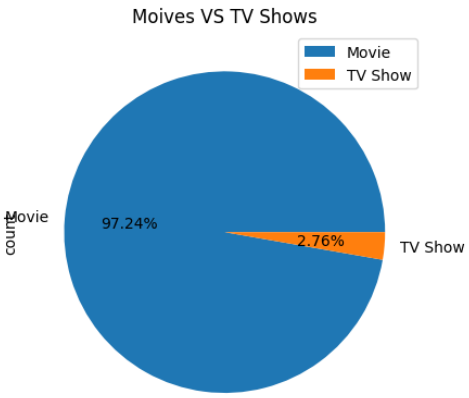
```
<class 'pandas.core.frame.DataFrame'>
Index: 5332 entries, 7 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       5332 non-null   object        
 1   type          5332 non-null   object        
 2   title         5332 non-null   object        
 3   director      5332 non-null   object        
 4   cast          5332 non-null   object        
 5   country       5332 non-null   object        
 6   date_added    5328 non-null   datetime64[ns]
 7   release_year  5332 non-null   int64         
 8   rating        5332 non-null   object        
 9   duration      5332 non-null   object        
 10  listed_in     5332 non-null   object        
 11  description   5332 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(10)
memory usage: 541.5+ KB
```

In [123…
```
# 0 number of types
types =  df['type'].value_counts().reset_index()
types
```

Out[123…

	type	count
0	Movie	5185
1	TV Show	147

In [135…
```
# visualizing the types of shows
types.set_index('type', inplace=True)
types.plot.pie(y = 'count', autopct='%.2f%%', legend='type')
plt.title("Moives VS TV Shows")
plt.show()
```



In [ ]:
```
#Insight: The dataset contains a larger proportion of movies compared to TV shows,
#indicating that Netflix offers a wider variety of movies to its subscribers
```

In [31]:
```
directors = df.groupby(['director', 'type'])['director'].value_counts().reset_index()
directors
```

Out[31]:

	director	type	count
0	A. L. Vijay	Movie	2
1	A. Raajdheep	Movie	1
2	A. Salaam	Movie	1
3	A.R. Murugadoss	Movie	1
4	Aadish Keluskar	Movie	1
...	...	...	...
3964	Çagan Irmak	Movie	1
3965	Ísold Uggadóttir	Movie	1
3966	Óskar Thór Axelsson	Movie	1
3967	Ömer Faruk Sorak	Movie	2
3968	Şenol Sönmez	Movie	2

3969 rows × 3 columns

In [32]:
```
# 1.  Top 10 directors from movies
top_10_directors_movie = df['director'][df['type'] == 'Movie'].value_counts().sort_values(ascending=False).iloc[0:10].reset_index()
top_10_directors_movie
```

Out[32]:

	director	count
0	Raúl Campos, Jan Suter	18
1	Marcus Raboy	14
2	Jay Karas	14

3	Cathy Garcia-Molina	13
4	Jay Chapman	12
5	Youssef Chahine	12
6	Martin Scorsese	12
7	Steven Spielberg	11
8	Don Michael Paul	10
9	David Dhawan	9

In [47]:
```
# 2 top 10 directors TVShow
top_10_directors_TVShow = df['director'][df['type'] == 'TV Show'].value_counts().sort_values(ascending=False).iloc[0:10].reset_index()
top_10_directors_TVShow
```

Out[47]:

	director	count
0	Alastair Fothergill	3
1	Iginio Straffi	2
2	Rob Seidenglanz	2
3	Shin Won-ho	2
4	Stan Lathan	2
5	Simon Frederick	1
6	Daniel Minahan	1
7	Takuya Igarashi	1
8	Ally Pankiw	1
9	Jay Oliva	1

In [48]:
```
# 3 top 10 countries Movies
top_10_countries_movie = df['country'][df['type'] == 'Movie'].value_counts().sort_values(ascending=False).iloc[0:10].reset_index()
top_10_countries_movie
```

Out[48]:

	country	count
0	United States	1819
1	India	868
2	United Kingdom	164
3	Canada	104
4	Egypt	90
5	Nigeria	84
6	Spain	84
7	Indonesia	76
8	Turkey	74
9	Japan	73

In [ ]:
```

```

In [49]:
```
# 4 top 10 countries TV Show
top_10_countries_TVShow = df['country'][df['type'] == 'TV Show'].value_counts().sort_values(ascending=False).iloc[0:10].reset_index()
top_10_countries_TVShow
```

Out[49]:

	country	count
0	United States	27
1	United Kingdom	19
2	Japan	10
3	South Korea	10
4	Spain	7
5	India	7
6	Taiwan	7
7	France	5
8	Turkey	5
9	Thailand	5

In [ ]:
```

```

In [50]:
```
# 5 top_10_rating_movie
top_10_rating_movie = df['rating'][df['type'] == 'Movie'].value_counts().sort_values(ascending=False).iloc[0:10].reset_index()
top_10_rating_movie
```

Out[50]:

	rating	count
0	TV-MA	1741
1	TV-14	1177
2	R	778
3	PG-13	470
4	TV-PG	416
5	PG	275
6	TV-G	81

7	TV-Y	71
8	TV-Y7	70
9	NR	58

In [51]:
```
# 6 top_10_rating_tvshow
top_10_rating_tvshow = df['rating'][df['type'] == 'TV Show'].value_counts().sort_values(ascending=False).iloc[0:10].reset_index()
top_10_rating_tvshow
```

Out[51]:

	rating	count
0	TV-MA	81
1	TV-14	37
2	TV-PG	15
3	TV-Y7	6
4	TV-Y	5
5	TV-G	3

In [54]:
```
df.rename(columns={'listed_in' : 'category'}, inplace=True)
```

In [55]:
```
# 7 top_10_category_moive
top_10_category_moive = df['category'][df['type'] == 'Movie'].value_counts().sort_values(ascending=False).iloc[0:10].reset_index()
top_10_category_moive
```

Out[55]:

	category	count
0	Dramas, International Movies	336
1	Stand-Up Comedy	286
2	Comedies, Dramas, International Movies	257
3	Dramas, Independent Movies, International Movies	243
4	Children & Family Movies, Comedies	179
5	Dramas, International Movies, Romantic Movies	160
6	Documentaries	156
7	Comedies, International Movies	152
8	Comedies, International Movies, Romantic Movies	143
9	Dramas	133

In [56]:
```
df['date_added_year'] = df['date_added'].dt.year
```

In [58]:
```
df.head(3)
```

Out[58]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	category	description	date_added_year
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...	2021.0
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...	2021.0
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...	2021.0

In [59]:
```
date_added =  df['date_added_year'].value_counts()
date_added
```

Out[59]:
```
date_added_year
2019.0    1264
2020.0    1194
2018.0    1100
2021.0     755
2017.0     722
2016.0     202
2015.0      50
2014.0      14
2011.0      13
2013.0       7
2012.0       3
2009.0       2
2008.0       1
2010.0       1
Name: count, dtype: int64
```

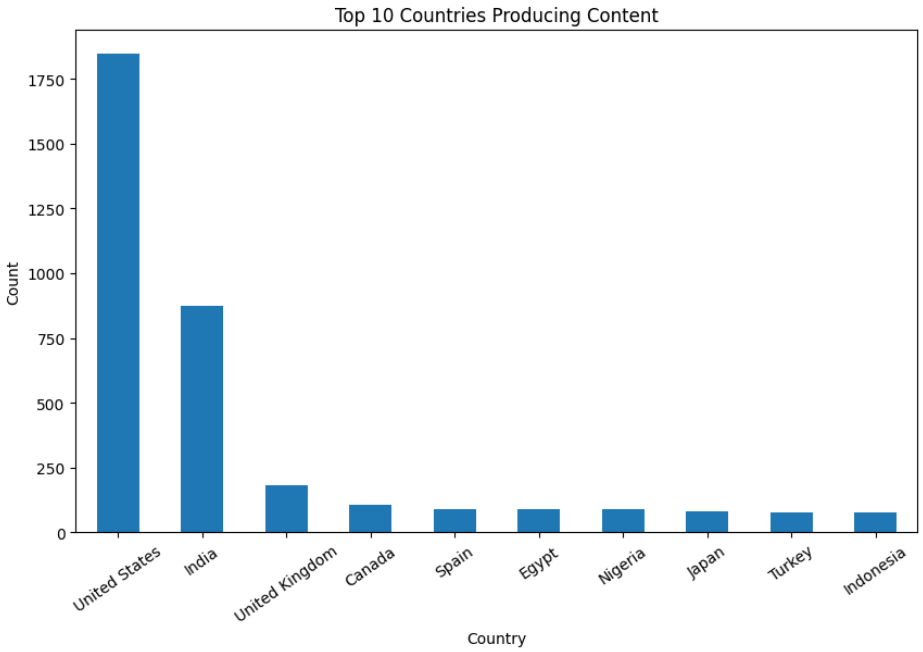
In [82]:
```
df['rating'] = pd.to_numeric(df['rating'], errors='coerce')
```

In [199…
```
# 9 Countries producing the most content
top_countries = df['country'].value_counts().head(10)
print("Top producing countries:")
print(top_countries)
```

```
Top producing countries:
country
United States     1846
India              875
United Kingdom     183
Canada             107
```

```
Spain              91
Egypt              90
Nigeria            88
Japan              83
Turkey             79
Indonesia          76
Name: count, dtype: int64
```

In [178…
```
top_countries = df['country'].value_counts().head(10)
plt.figure(figsize=(10, 6))
top_countries.plot(kind='bar')
plt.title('Top 10 Countries Producing Content')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=35)
plt.show()
```

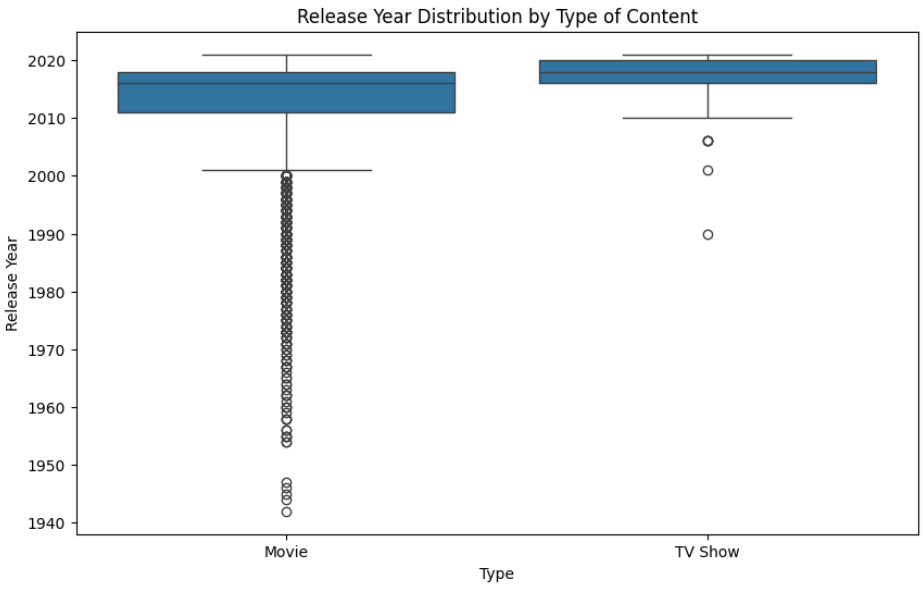


In [ ]:
```
#INSIGHT: The United States, followed by the United Kingdom and India,
#are the top three countries producing content on Netflix.
```

In [101…
```
# 10 Relationship between release year and type of content
movie_release_years = df[df['type'] == 'Movie']['release_year']
tv_show_release_years = df[df['type'] == 'TV Show']['release_year']
print("Mean release year for movies:", np.mean(movie_release_years))
print("Mean release year for TV shows:", np.mean(tv_show_release_years))
```

```
Mean release year for movies: 2012.6133076181293
Mean release year for TV shows: 2017.2857142857142
```

In [144…
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='type', y='release_year', data=df)
plt.title('Release Year Distribution by Type of Content')
plt.xlabel('Type')
plt.ylabel('Release Year')
plt.show()
```

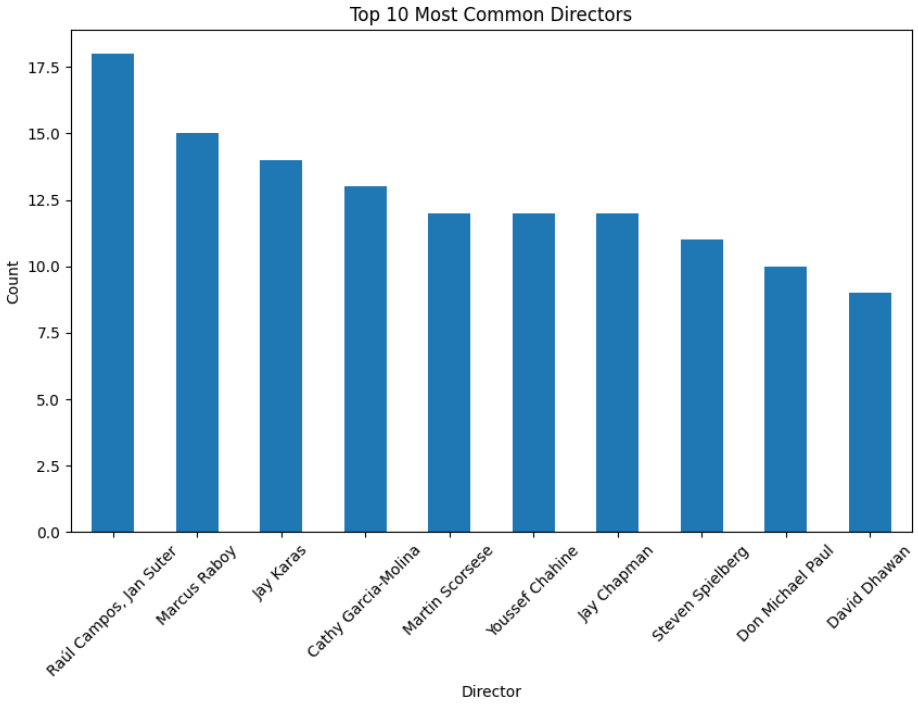


In [ ]:
```
#INSIGHT: he United States, followed by the United Kingdom and India,
#are the top three countries producing content on Netflix.
```

In [189…
```
# 11 Most common directors
common_directors = df['director'].value_counts().head(10)
print("Most common directors:")
print(common_directors)
```

```
Most common directors:
director
Raúl Campos, Jan Suter    18
Marcus Raboy              15
Jay Karas                 14
Cathy Garcia-Molina       13
Martin Scorsese           12
Youssef Chahine           12
Jay Chapman               12
Steven Spielberg          11
Don Michael Paul          10
David Dhawan               9
Name: count, dtype: int64
```

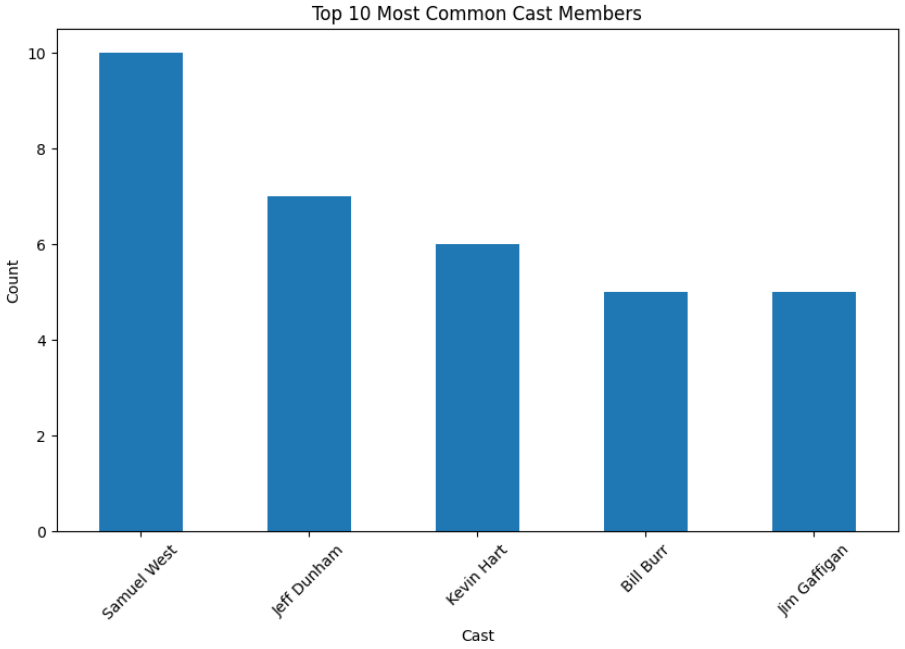
In [188…
```
common_directors = df['director'].value_counts().head(10)
plt.figure(figsize=(10, 6))
common_directors.plot(kind='bar')
plt.title('Top 10 Most Common Directors')
plt.xlabel('Director')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



In [103…
```
# 12. Most common actors/actresses
common_cast = df['cast'].value_counts().head(3)
print("Most common cast members:")
print(common_cast)
```

```
Most common cast members:
cast
Samuel West    10
Jeff Dunham     7
Kevin Hart      6
Name: count, dtype: int64
```

In [145…
```
common_cast = df['cast'].value_counts().head(5)
plt.figure(figsize=(10, 6))
common_cast.plot(kind='bar')
plt.title('Top 10 Most Common Cast Members')
plt.xlabel('Cast')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

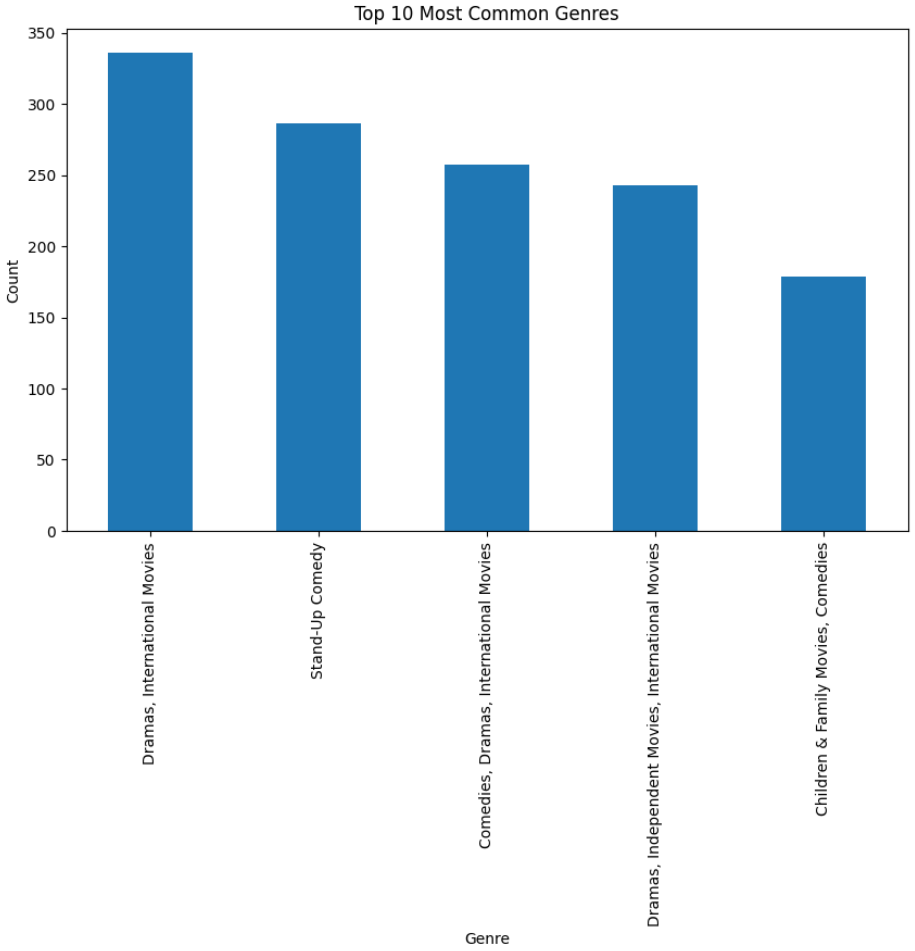


In [184…
```
# 13. Most common genres
common_genres = df['category'].value_counts().head(5).reset_index()
common_genres
```

Out[184…

	category	count
0	Dramas, International Movies	336
1	Stand-Up Comedy	286
2	Comedies, Dramas, International Movies	257
3	Dramas, Independent Movies, International Movies	243
4	Children & Family Movies, Comedies	179

In [185…
```
common_genres = df['category'].value_counts().head(5)
plt.figure(figsize=(10, 6))
common_genres.plot(kind='bar')
plt.title('Top 10 Most Common Genres')
plt.xlabel('Genre')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```

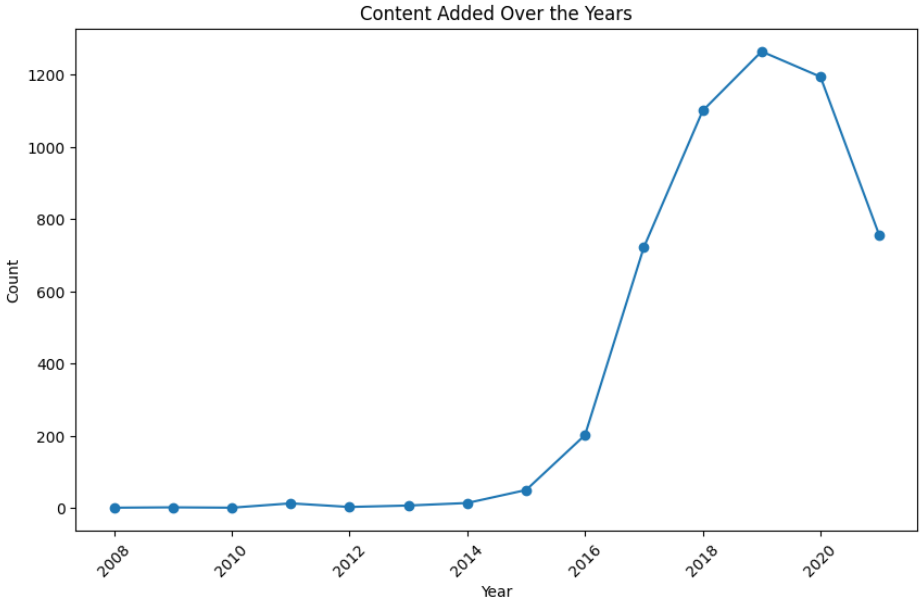


In [ ]:
```
#INSIGHT: Insight: Genres such as Drama, Comedy, and Thriller are the most common on Netflix.
```

In [105…
```
# 14. Amount of content added over the years
content_added_years = df['date_added_year'].value_counts().sort_index()
print("Content added over the years:")
print(content_added_years)
```

```
Content added over the years:
date_added_year
2008.0       1
2009.0       2
2010.0       1
2011.0      13
2012.0       3
2013.0       7
2014.0      14
2015.0      50
2016.0     202
2017.0     722
2018.0    1100
2019.0    1264
2020.0    1194
2021.0     755
Name: count, dtype: int64
```

In [133…
```
content_added_years = df['date_added_year'].value_counts().sort_index()
plt.figure(figsize=(10, 6))
content_added_years.plot(kind='line', marker='o')
plt.title('Content Added Over the Years')
plt.xlabel('Year')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



In [ ]: