Netflix Data Analysis Project

Project Analyst

SWAPNISH PANDEY

(Master Of Computer Applications)

Project Summary & Key Insights

- This project presents a comprehensive analysis of Netflix’s content catalogusingPython and data visualization techniques. The dataset includes thousandsof titles spanning movies and TV shows, with metadata such as genre, duration, country of origin, release year, and date added to the platform. The goal of thisanalysis was to uncover patterns in content strategy, viewer engagement, andplatform evolution over time.

- The study begins by examining the distribution of content types. It was observedthat movies dominate the platform, but TV shows have shown consistent growth, especially in international markets. The average duration of movies was foundtobe approximately 90 minutes, aligning with standard feature-length expectations. Interestingly, a trend analysis revealed that movie durations have slightly declined in recent years, possibly reflecting changing viewer attention spansandthe rise of mobile-first consumption.

- For TV shows, the most common number of seasons is one, indicating ahighvolume of limited series or pilot content. This suggests that Netflix frequentlyexperiments with new formats and concepts, using viewer feedback to determinerenewals. Genre distribution over the years showed that Drama and Comedyremain dominant, while Documentaries and International content have grownsignificantly, reflecting Netflix’s global expansion and diversified audiencebase.

- The analysis also explored content launch strategy. It was found that July andDecember are peak months for content additions, aligning with summer breaks and holiday seasons. This insight is valuable for planning promotional campaigns and major releases. Additionally, the genre-country relationshiprevealed that the United States leads in Drama and Comedy, while IndiaandSouth Korea are prominent in Romance and Action genres. This supportsNetflix’s strategy of tailoring content to regional preferences and leveraginglocal production hubs.

- Overall, this project demonstrates how data-driven insights can informcontent acquisition, production planning, and viewer engagement strategies. By understanding trends in duration, genre, and geography, Netflix can optimizeitscatalog to meet evolving audience demands. The analysis not only highlightscurrent strengths but also uncovers opportunities for future growth, especiallyinemerging markets and underrepresented genres.

```
!pip install pandas
```

```
Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: pandas in c:\users\hp\appdata\roaming\
python\python313\site-packages (2.3.3)
Requirement already satisfied: numpy>=1.26.0 in c:\users\hp\appdata\
roaming\python\python313\site-packages (from pandas) (2.3.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hp\
appdata\roaming\python\python313\site-packages (from pandas)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\appdata\
roaming\python\python313\site-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\hp\appdata\
roaming\python\python313\site-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\hp\appdata\
roaming\python\python313\site-packages (from python-dateutil>=2.8.2-
>pandas) (1.17.0)
```

```
import pandas as pd
print(pd.__version__)
```

```
2.3.3
```

```
df = pd.read_csv(r"C:\Users\HP\Downloads\netflix.csv")
df.head()
```

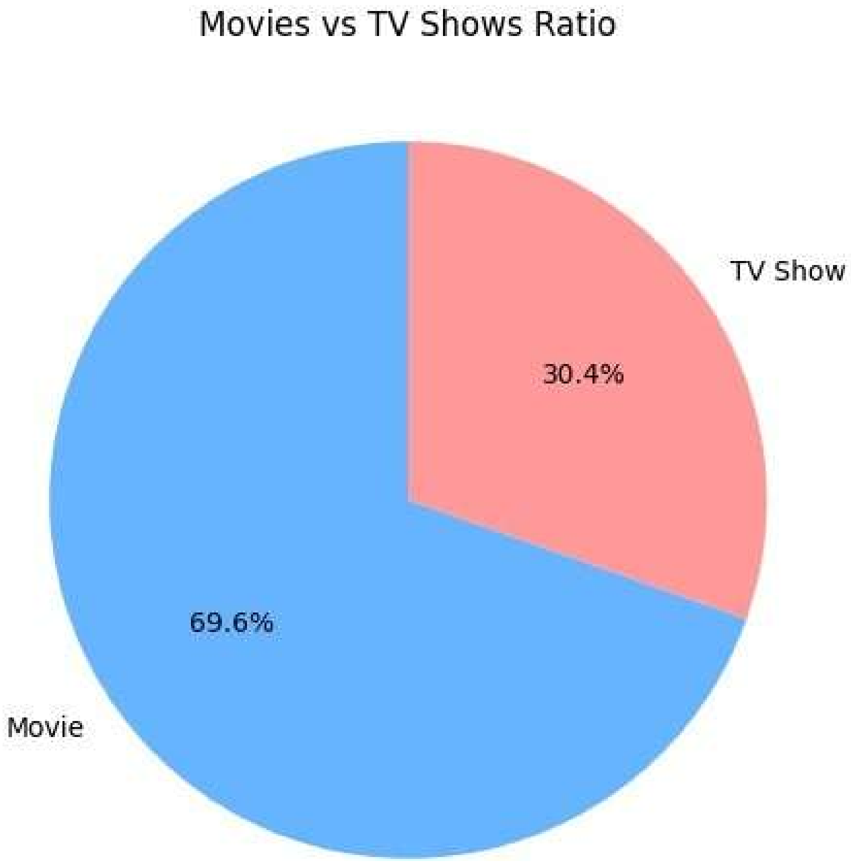
```
  show_id     type                  title         director  \
0      s1    Movie   Dick Johnson Is Dead  Kirsten Johnson   
1      s2  TV Show          Blood & Water              NaN   
2      s3  TV Show              Ganglands  Julien Leclercq   
3      s4  TV Show  Jailbirds New Orleans              NaN   
4      s5  TV Show           Kota Factory              NaN   


                                                cast        country  \
0                                                NaN  United States   
1  Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...   South Africa   
2  Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...            NaN   
3                                                NaN            NaN   
4  Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...          India   

           date_added  release_year rating   duration  \
0  September 25, 2021          2020  PG-13     90 min   
1  September 24, 2021          2021  TV-MA  2 Seasons   
2  September 24, 2021          2021  TV-MA   1 Season   
3  September 24, 2021          2021  TV-MA   1 Season   
4  September 24, 2021          2021  TV-MA  2 Seasons   

                                            listed_in  \
0                                       Documentaries   
1     International TV Shows, TV Dramas, TV Mysteries   
```

```
2  Crime TV Shows, International TV Shows, TV Act...
3                           Docuseries, Reality TV
4  International TV Shows, Romantic TV Shows, TV ...

                                         description
0  As her father nears the end of his life, filmm...
1  After crossing paths at a party, a Cape Town t...
2  To protect his family from a powerful drug lor...
3  Feuds, flirtations and toilet talk go down amo...
4  In a city of coaching centers known to train I...
```

```
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
df['month_added'] = df['date_added'].dt.month
df['year_added'] = df['date_added'].dt.year

df['duration_type'] = df['duration'].apply(lambda x: 'Season' if
'Season' in str(x) else 'Minutes')
df['duration_value'] = df['duration'].str.extract(r'(\
d+)').astype(float)

df = df.copy()
df['director'] = df['director'].fillna('Unknown')
df['cast'] = df['cast'].fillna('Unknown')
df['country'] = df['country'].fillna('Unknown')

## 1:- Content Strategy

## Q1 :- What is the ratio of movies vs TV shows on Netflix?

type_counts = df['type'].value_counts()
type_percent = df['type'].value_counts(normalize=True) * 100

print("Content Type Counts:\n", type_counts)
print("\nContent Type Percentages:\n", type_percent)

type_counts.plot(kind='pie', autopct='%1.1f%%', startangle=90,
colors=['#66b3ff','#ff9999'], figsize=(6,6))
plt.title('Movies vs TV Shows Ratio')
plt.ylabel('')
plt.show()
```

```
Content Type Counts:
 type
Movie      6131
TV Show    2676
Name: count, dtype: int64

Content Type Percentages:
 type
Movie      69.615079
```

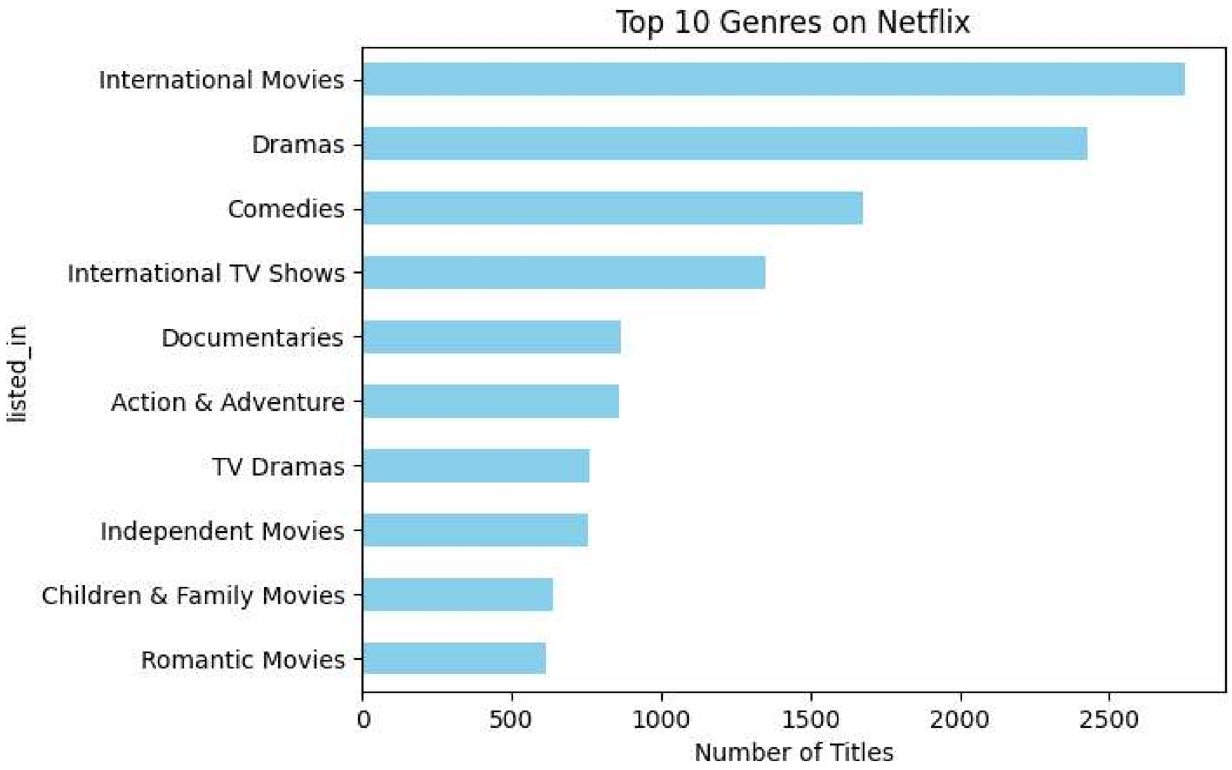
```
TV Show     30.384921
Name: proportion, dtype: float64
```



```
##Q2:- Which genres are most popular on Netflix globally?


genre_series =
df['listed_in'].dropna().str.split(',').explode().str.strip()
top_genres = genre_series.value_counts().head(10)

print("Top Genres:\n", top_genres)

top_genres.plot(kind='barh', color='skyblue')
plt.title('Top 10 Genres on Netflix')
plt.xlabel('Number of Titles')
plt.gca().invert_yaxis()
plt.show()
```

```
Top Genres:
 listed_in
```

```
International Movies            2752
Dramas                          2427
Comedies                        1674
International TV Shows          1351
Documentaries                    869
Action & Adventure               859
TV Dramas                        763
Independent Movies               756
Children & Family Movies         641
Romantic Movies                  616
Name: count, dtype: int64
```

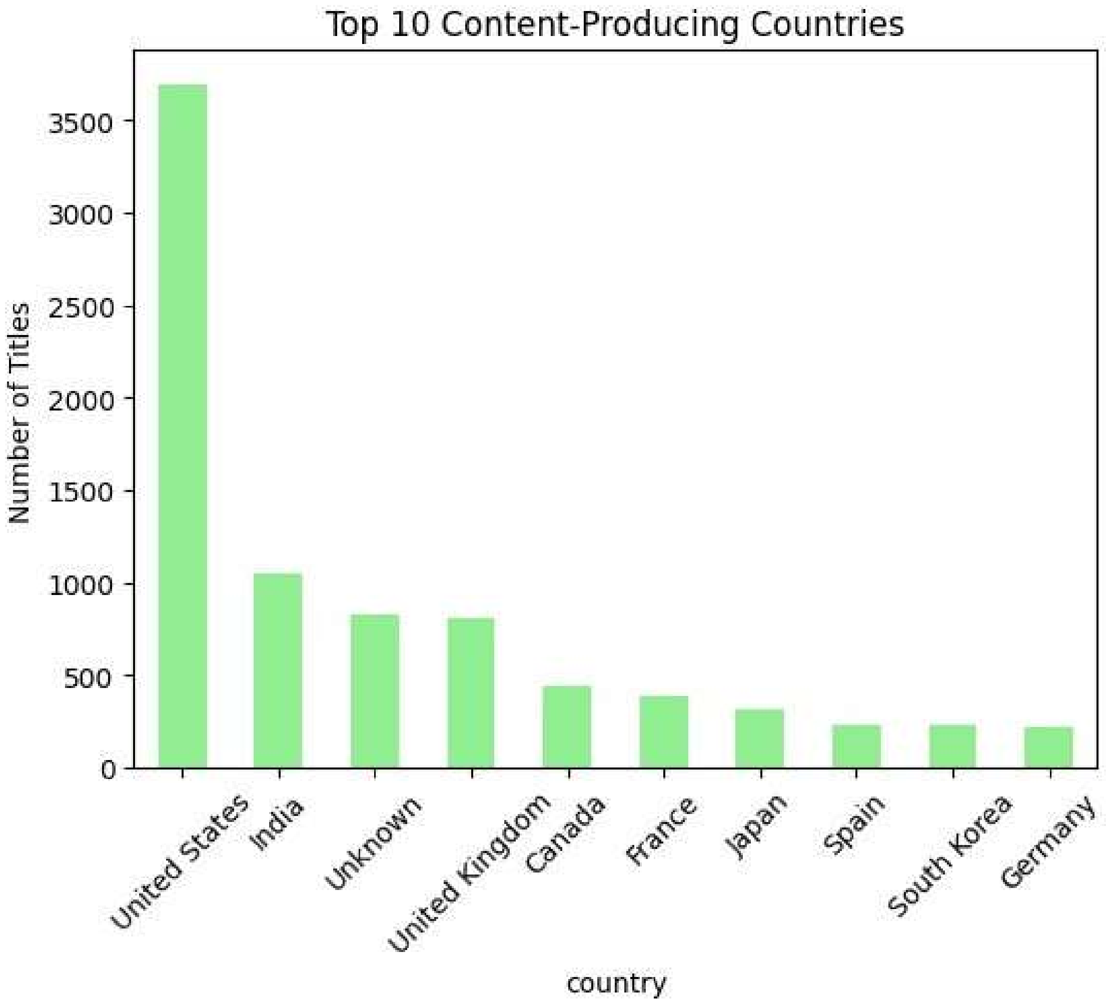


```
##Q3:- Which years saw the highest release of content on Netflix ?

release_years = df['release_year'] =
pd.to_datetime(df['release_year'], errors='coerce').dt.year
year_counts = df['release_year'].value_counts().sort_index()

year_counts.plot(kind='bar', figsize=(12,6), color='coral')
plt.title('Content Releases by Year')
plt.xlabel('Release Year')
plt.ylabel('Number of Titles')
plt.show()
```

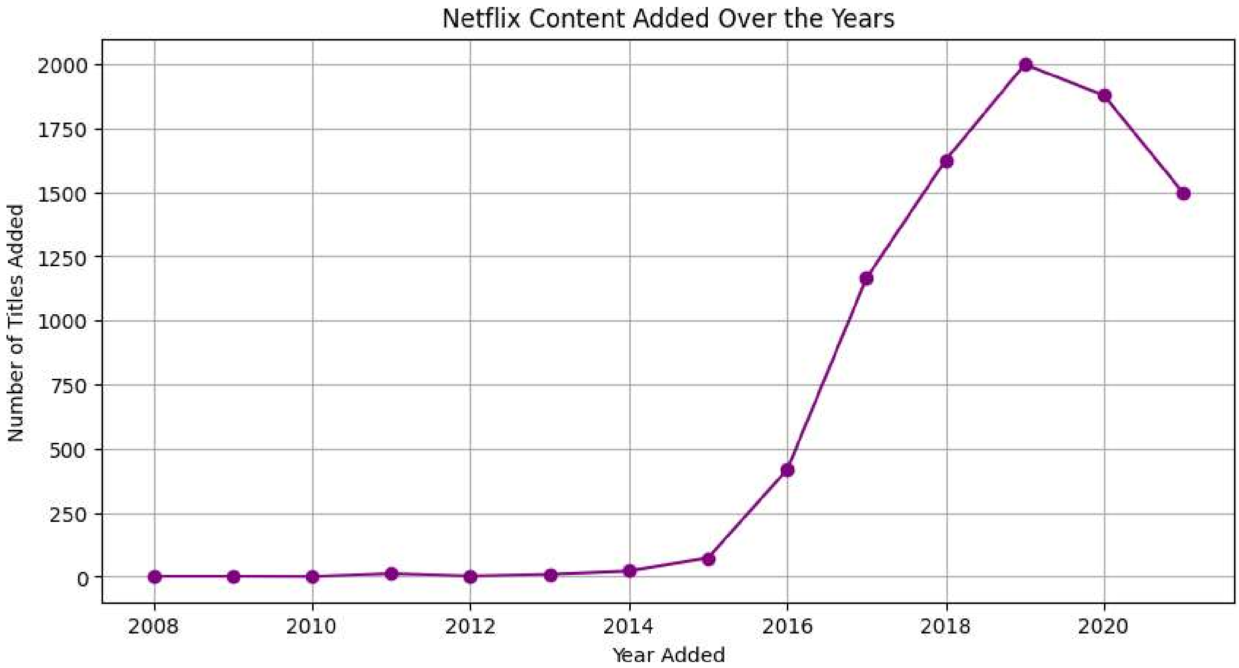
```
-----------------------------------------------------------------------
------
NameError                                 Traceback (most recent call
last)
Cell In[5], line 1
----> 1 release_years = df['release_year'] =
pd.to_datetime(df['release_year'], errors='coerce').dt.year
      2 year_counts = df['release_year'].value_counts().sort_index()
      4 year_counts.plot(kind='bar', figsize=(12,6), color='coral')

NameError: name 'pd' is not defined
```

```
#Q4:- Which Countries Produce the most Netflix content ?

country_series =
df['country'].dropna().str.split(',').explode().str.strip()
top_countries = country_series.value_counts().head(10)

print("Top Countries:\n", top_countries)

top_countries.plot(kind='bar', color='lightgreen')
plt.title('Top 10 Content-Producing Countries')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.show()
```

```
Top Countries:
 country
United States     3690
India             1046
Unknown            831
United Kingdom     806
Canada             445
France             393
Japan              318
Spain              232
South Korea        231
Germany            226
Name: count, dtype: int64
```

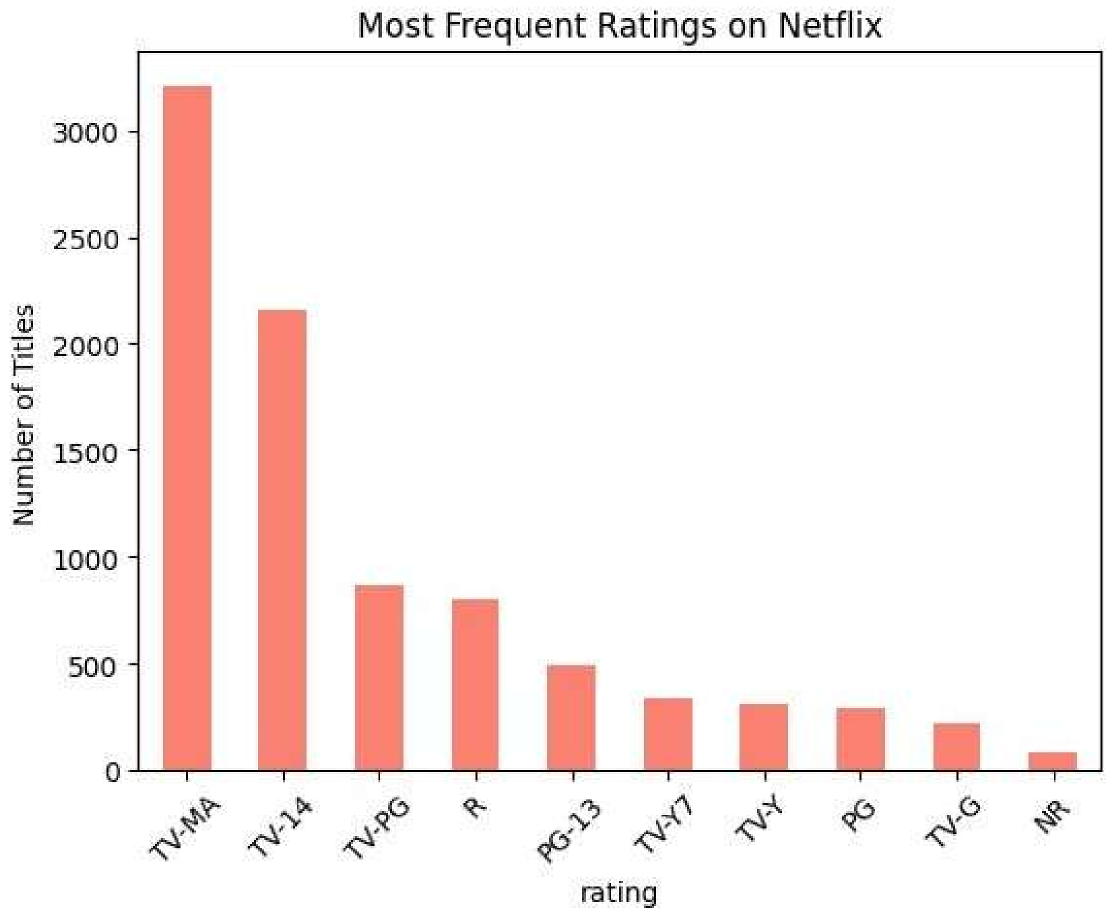


```
##Q5:-How has the trend of adding new content evolved year by year?

year_added_counts = df['year_added'].value_counts().sort_index()

year_added_counts.plot(kind='line', marker='o', color='purple',
figsize=(10,5))
plt.title('Netflix Content Added Over the Years')
plt.xlabel('Year Added')
plt.ylabel('Number of Titles Added')
plt.grid(True)
plt.show()
```



```
##6:-6.Which rangs (e.g., TV-MA, PG, etc.) are most frequent on
Netflix?

rating_counts = df['rating'].value_counts().head(10)

print("Top Ratings:\n", rating_counts)

rating_counts.plot(kind='bar', color='salmon')
plt.title('Most Frequent Ratings on Netflix')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.show()
```

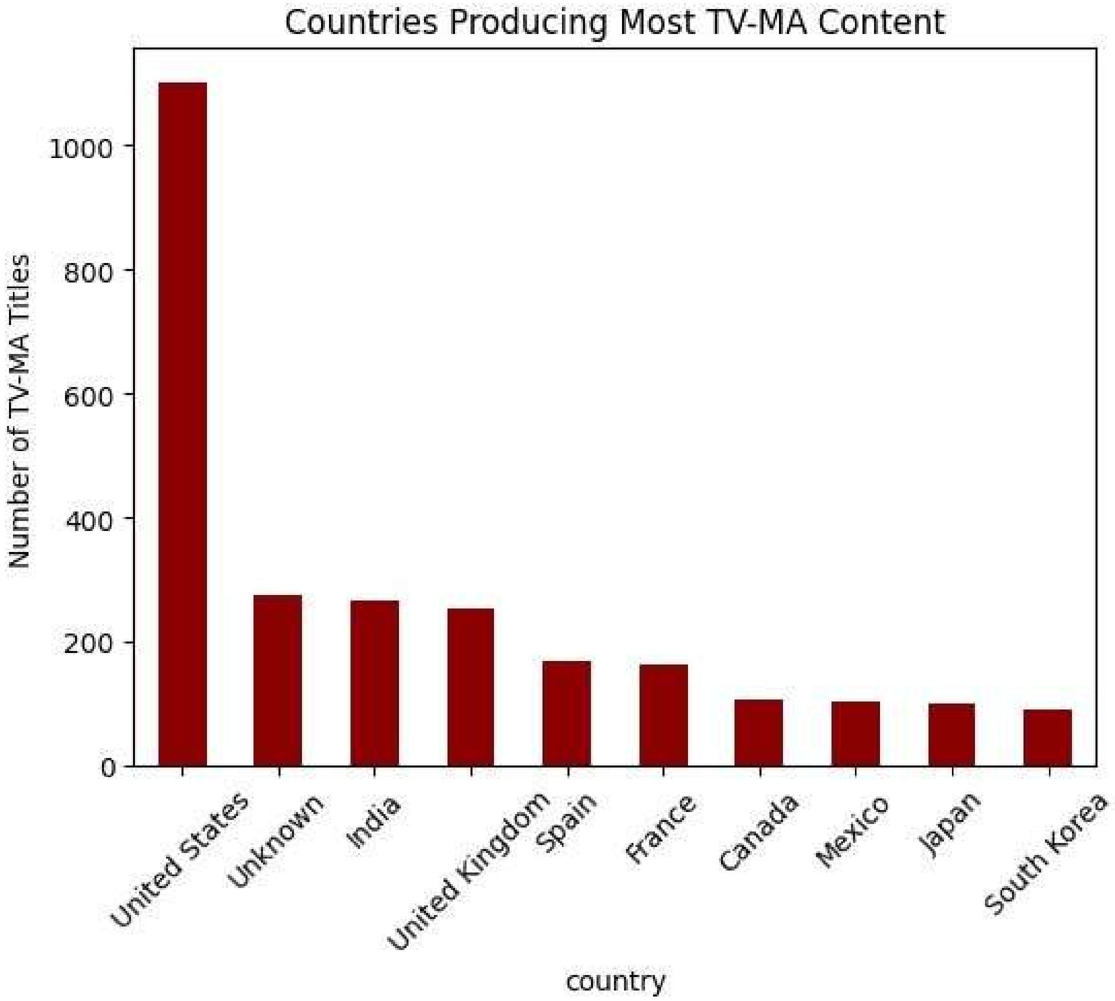
```
Top Ratings:
 rating
TV-MA    3207
TV-14    2160
TV-PG     863
R         799
PG-13     490
TV-Y7     334
TV-Y      307
PG        287
TV-G      220
NR         80
Name: count, dtype: int64
```



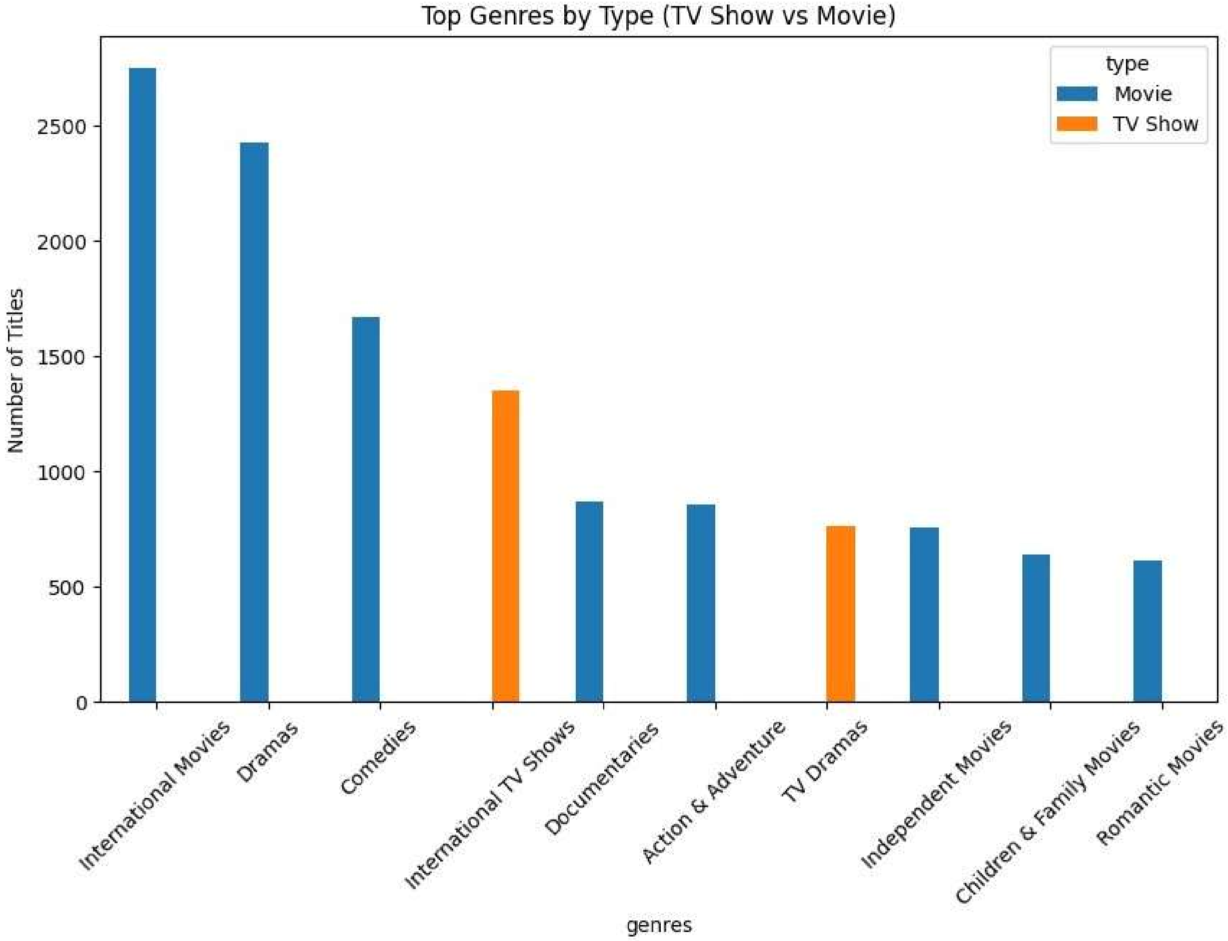
```
#Q7:-Do some countries tend to produce more mature content (TV-MA)?

tvma_df = df[df['rating'] == 'TV-MA']
tvma_countries =
tvma_df['country'].dropna().str.split(',').explode().str.strip().value
_counts().head(10)

print("Top TV-MA Producing Countries:\n", tvma_countries)

tvma_countries.plot(kind='bar', color='darkred')
plt.title('Countries Producing Most TV-MA Content')
plt.ylabel('Number of TV-MA Titles')
plt.xticks(rotation=45)
plt.show()
```

```
Top TV-MA Producing Countries:
 country
United States     1101
Unknown            276
India              266
United Kingdom     253
Spain              170
```

```
France                163
Canada                107
Mexico                102
Japan                 101
South Korea            92
Name: count, dtype: int64
```

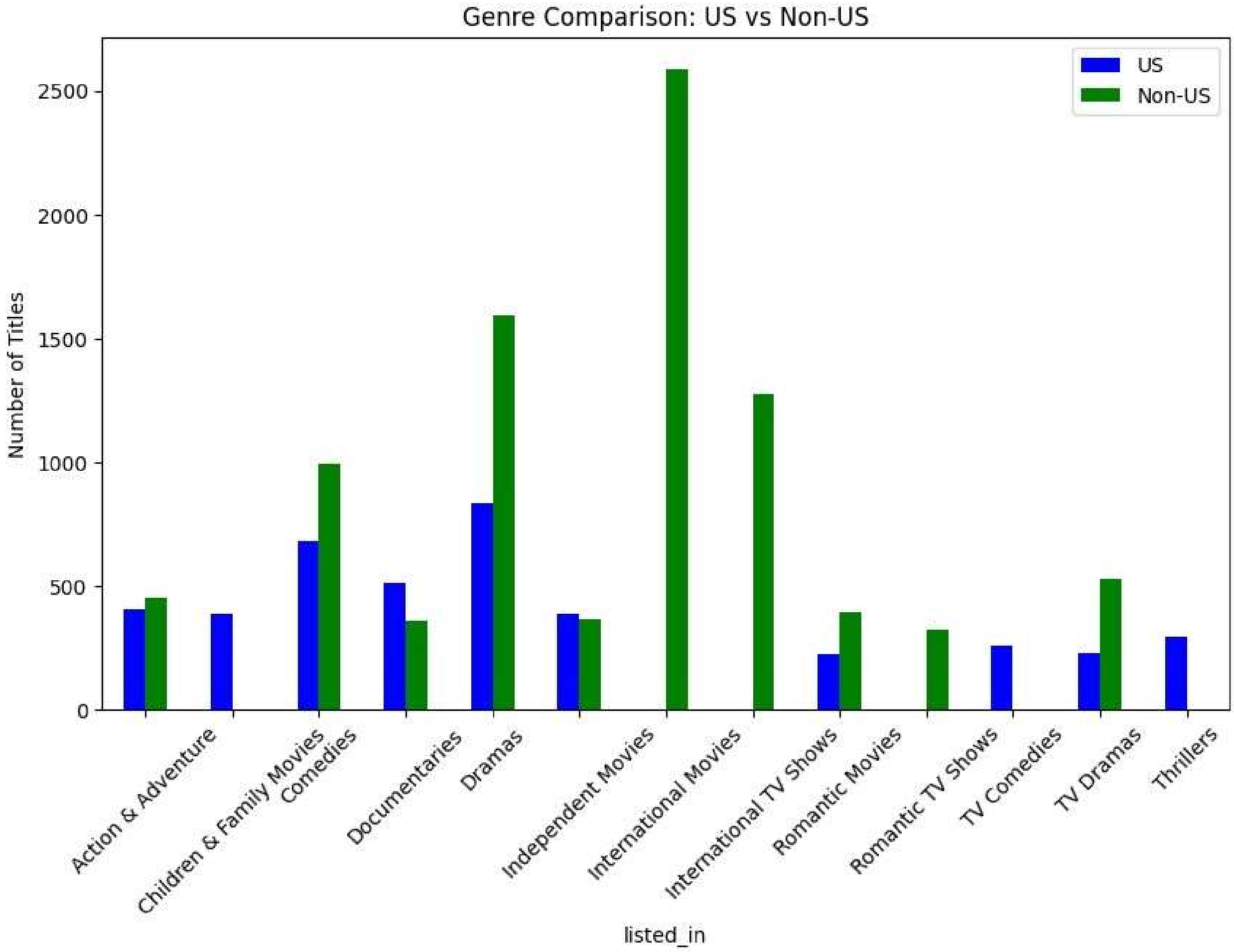


```
#Q8:-Which genres are more associated with TV shows .

genre_split = df[['type', 'listed_in']].dropna()
genre_split =
genre_split.assign(genres=genre_split['listed_in'].str.split(',')).exp
lode('genres')
genre_split['genres'] = genre_split['genres'].str.strip()

genre_pivot = genre_split.groupby(['type',
'genres']).size().unstack(fill_value=0)
genre_pivot = genre_pivot.loc[:,
genre_pivot.sum().sort_values(ascending=False).head(10).index]

genre_pivot.T.plot(kind='bar', figsize=(10,6))
plt.title('Top Genres by Type (TV Show vs Movie)')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.show()
```



```
#9:-Which genres dominate the U.S. vs other countries?

us_df = df[df['country'].str.contains('United States', na=False)]
non_us_df = df[~df['country'].str.contains('United States', na=False)]

us_genres =
us_df['listed_in'].dropna().str.split(',').explode().str.strip().value
_counts().head(10)
non_us_genres =
non_us_df['listed_in'].dropna().str.split(',').explode().str.strip().v
alue_counts().head(10)
```

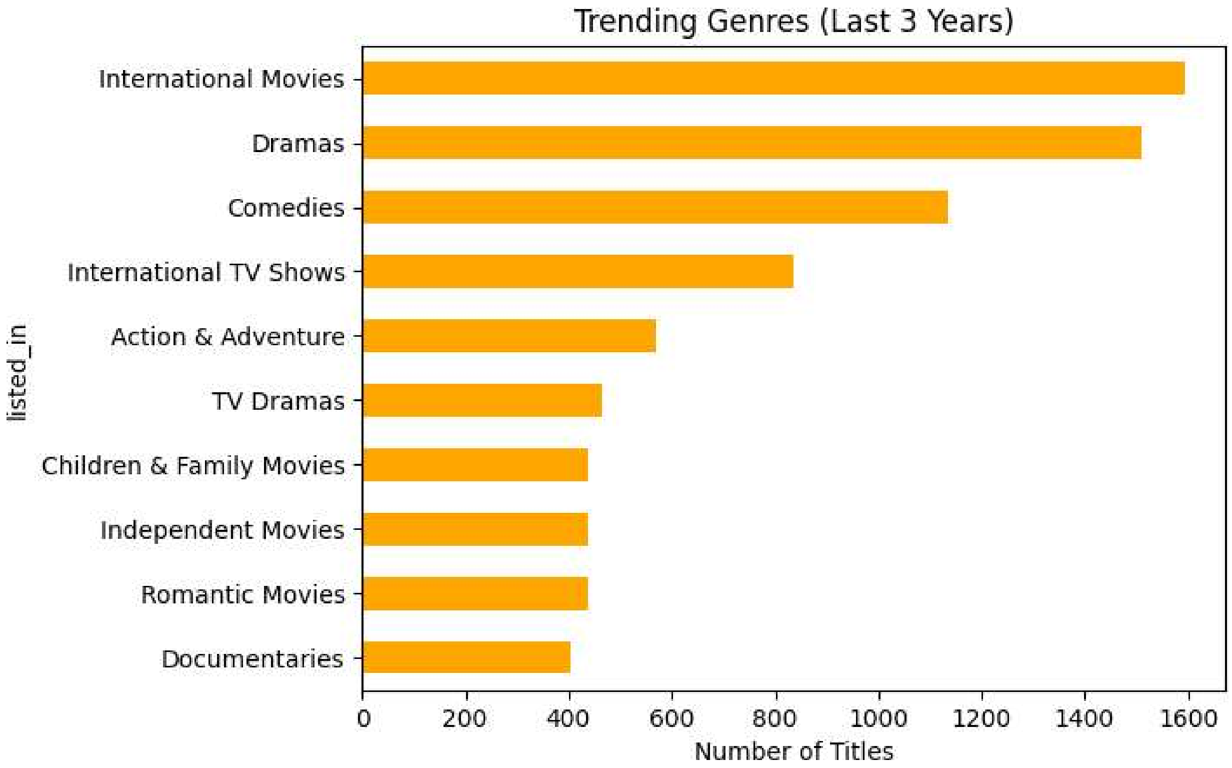
```
# Combine for comparison
genre_compare = pd.DataFrame({'US': us_genres, 'Non-US':
non_us_genres}).fillna(0)

genre_compare.plot(kind='bar', figsize=(10,6), color=['blue',
'green'])
plt.title('Genre Comparison: US vs Non-US')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.show()
```



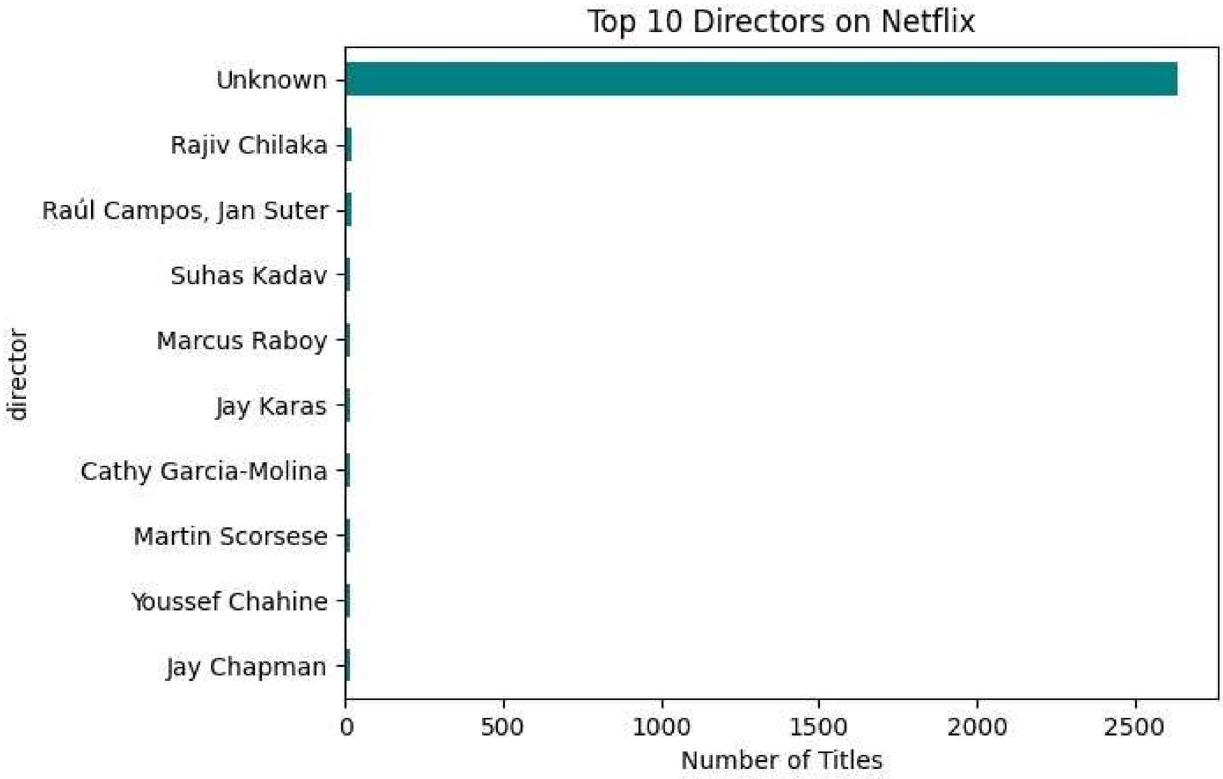
```
#10:-What genres are most popular in the last 3 years?

recent_df = df[df['year_added'] >= (df['year_added'].max() - 2)]
recent_genres =
recent_df['listed_in'].dropna().str.split(',').explode().str.strip().v
alue_counts().head(10)

print("Top Genres (Last 3 Years):\n", recent_genres)
```

```
recent_genres.plot(kind='barh', color='orange')
plt.title('Trending Genres (Last 3 Years)')
plt.xlabel('Number of Titles')
plt.gca().invert_yaxis()
plt.show()
```

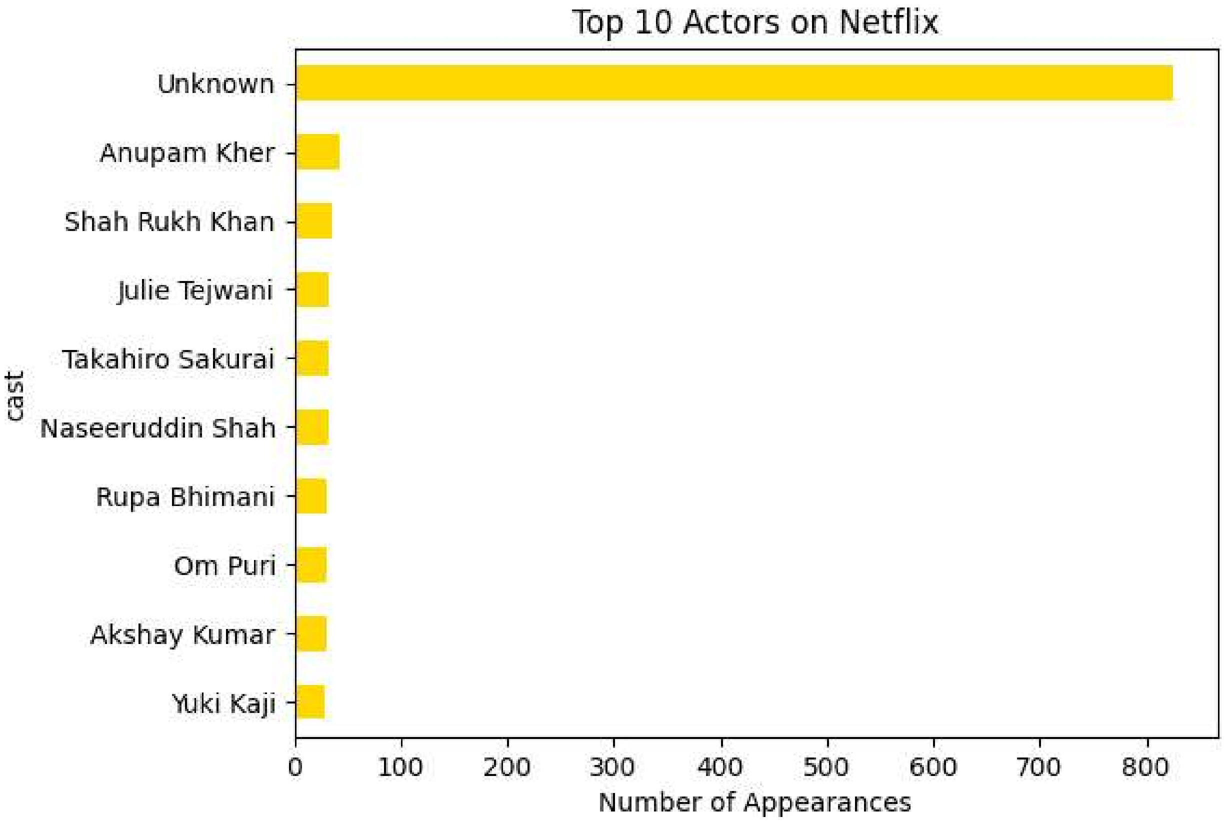
```
Top Genres (Last 3 Years):
 listed_in
International Movies          1593
Dramas                        1511
Comedies                      1135
International TV Shows         836
Action & Adventure             568
TV Dramas                      463
Children & Family Movies       439
Independent Movies             438
Romantic Movies                437
Documentaries                  405
Name: count, dtype: int64
```



```
#11:-Who are the top 10 directors with the most Netflix content?

top_directors = df['director'].dropna().value_counts().head(10)
top_directors.plot(kind='barh', color='teal')
plt.title('Top 10 Directors on Netflix')
plt.xlabel('Number of Titles')
plt.gca().invert_yaxis()
plt.show()
```



```
#12:-Which actors appear most frequently in Netflix shows?

actor_series =
df['cast'].dropna().str.split(',').explode().str.strip()
top_actors = actor_series.value_counts().head(10)
top_actors.plot(kind='barh', color='gold')
plt.title('Top 10 Actors on Netflix')
plt.xlabel('Number of Appearances')
plt.gca().invert_yaxis()
plt.show()
```



```
#13:-Which director-genre pairs are most frequent?

director_genre = df[['director', 'listed_in']].dropna()
director_genre =
director_genre.assign(genres=director_genre['listed_in'].str.split(','
)).explode('genres')
director_genre['genres'] = director_genre['genres'].str.strip()
pair_counts = director_genre.groupby(['director',
'genres']).size().sort_values(ascending=False).head(10)
print(pair_counts)
```

```
director  genres
Unknown   International TV Shows    1223
          TV Dramas                  702
          TV Comedies                539
          Kids' TV                   433
          Crime TV Shows             401
          Romantic TV Shows          341
          Docuseries                 335
          Reality TV                 249
          British TV Shows           228
          Anime Series               165
dtype: int64
```

```
#14:-How many Titles have unknown directors or cast members?
```

```
unknown_directors = df['director'].isna().sum()
unknown_cast = df['cast'].isna().sum()
print(f"Titles with Unknown Director: {unknown_directors}")
print(f"Titles with Unknown Cast: {unknown_cast}")
```

```
Titles with Unknown Director: 0
Titles with Unknown Cast: 0
```

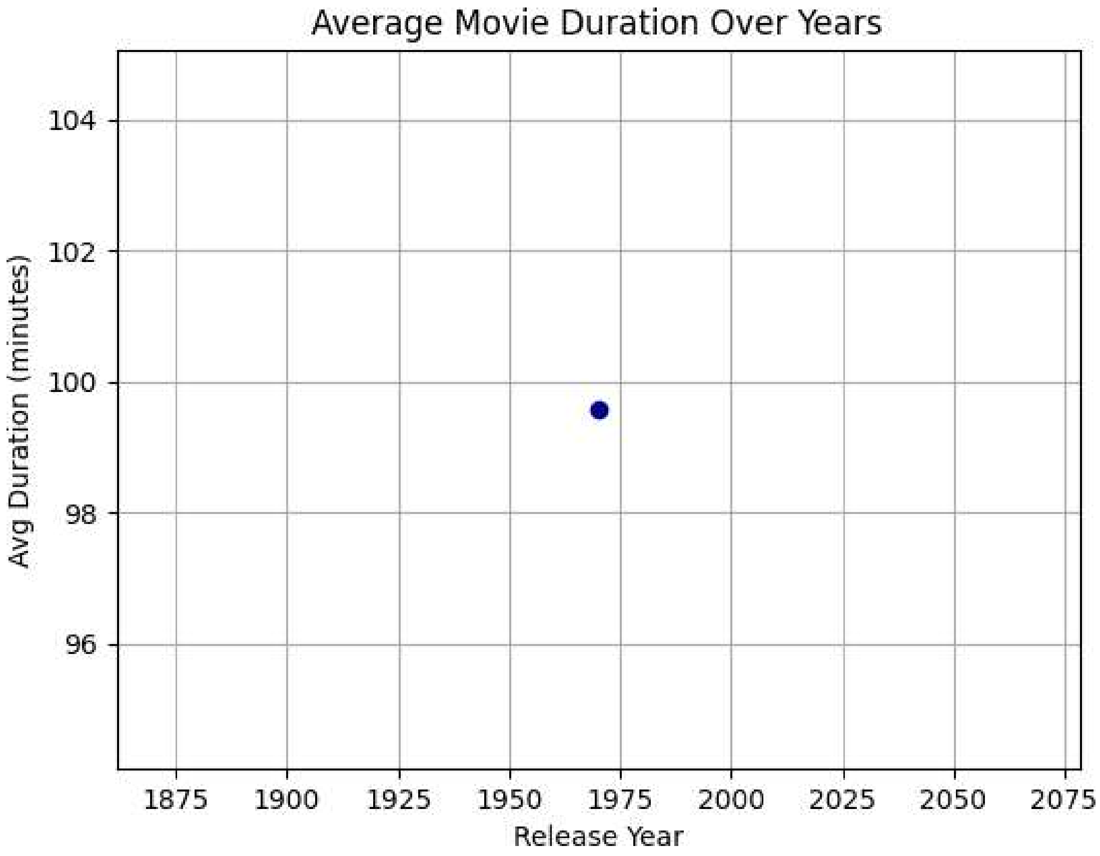
```
##15:-What is the average duraton of Movies on Netflix?

avg_duration = df[df['type'] == 'Movie']['duration_value'].mean()
print(f"Average Movie Duration: {avg_duration:.2f} minutes")
```

```
Average Movie Duration: 99.58 minutes
```

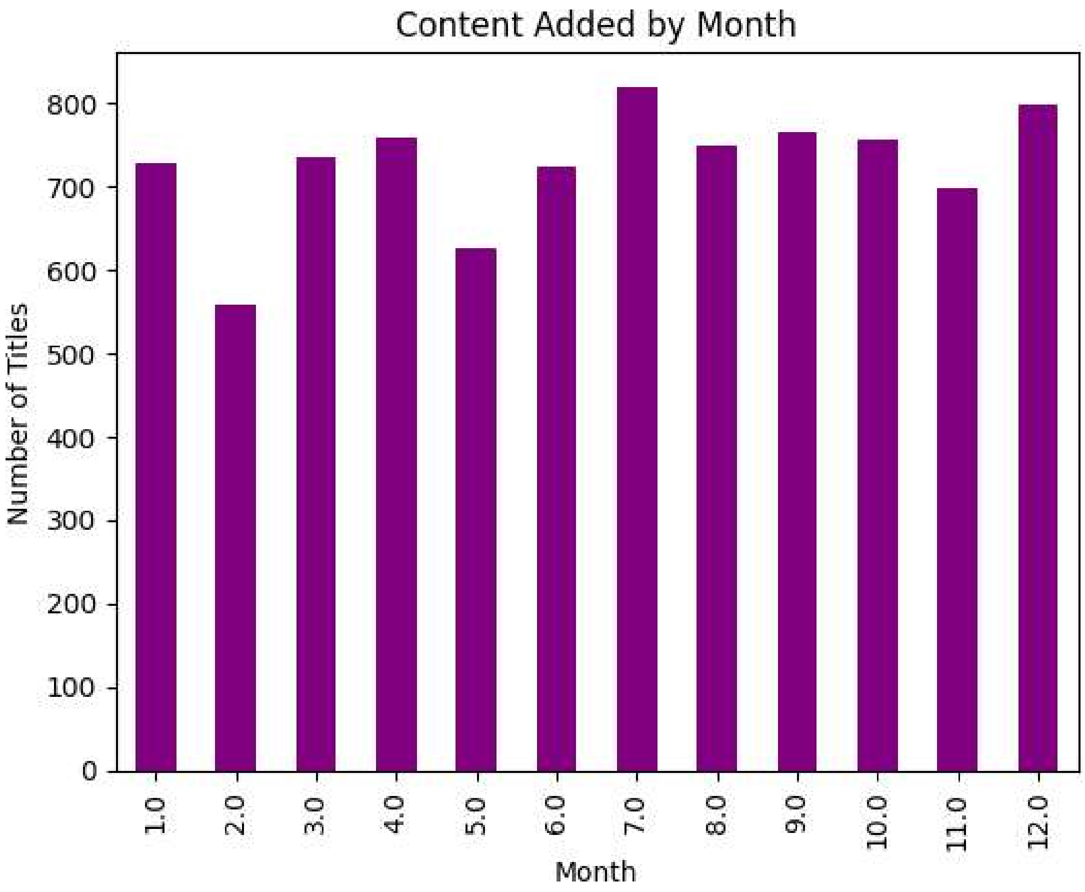
```
##16:-What’s the most common number for seasons for TV shows?

season_counts = df[df['type'] == 'TV Show']
['duration_value'].value_counts().head(1)
print("Most Common Season Count:\n", season_counts)
```

```
Most Common Season Count:
 duration_value
1.0    1793
Name: count, dtype: int64
```

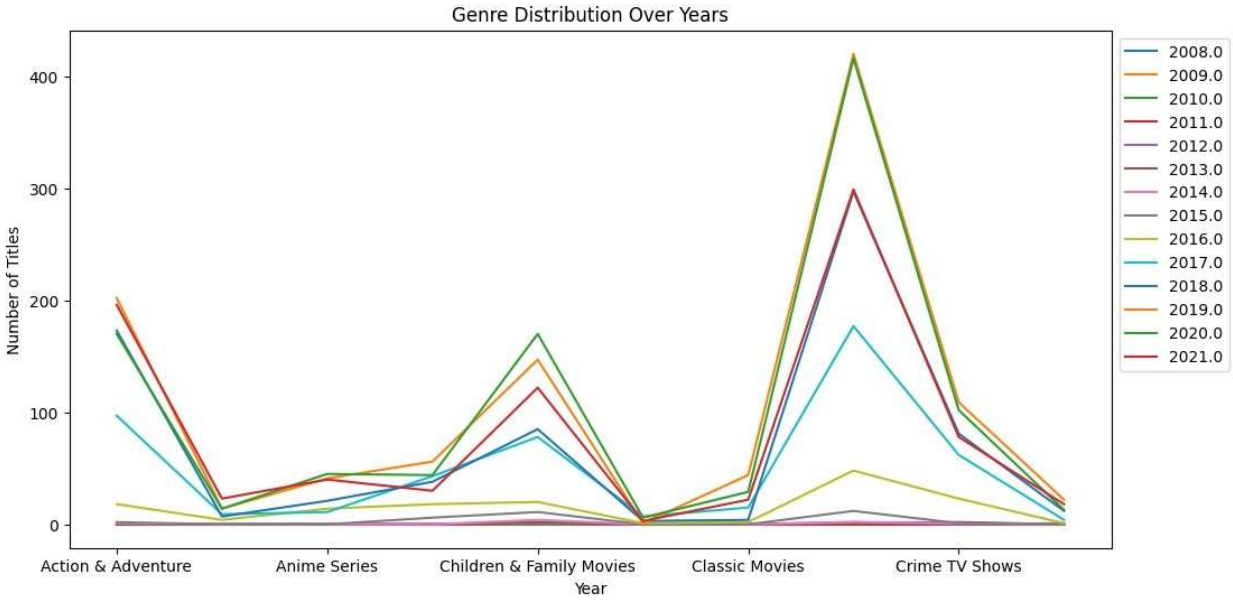
```
#17:-Is there a trend in movie durations over the years?

movie_df = df[df['type'] == 'Movie']
duration_trend = movie_df.groupby('release_year')
['duration_value'].mean()

duration_trend.plot(kind='line', marker='o', color='darkblue')
plt.title('Average Movie Duration Over Years')
plt.xlabel('Release Year')
plt.ylabel('Avg Duration (minutes)')
plt.grid(True)
plt.show()
```



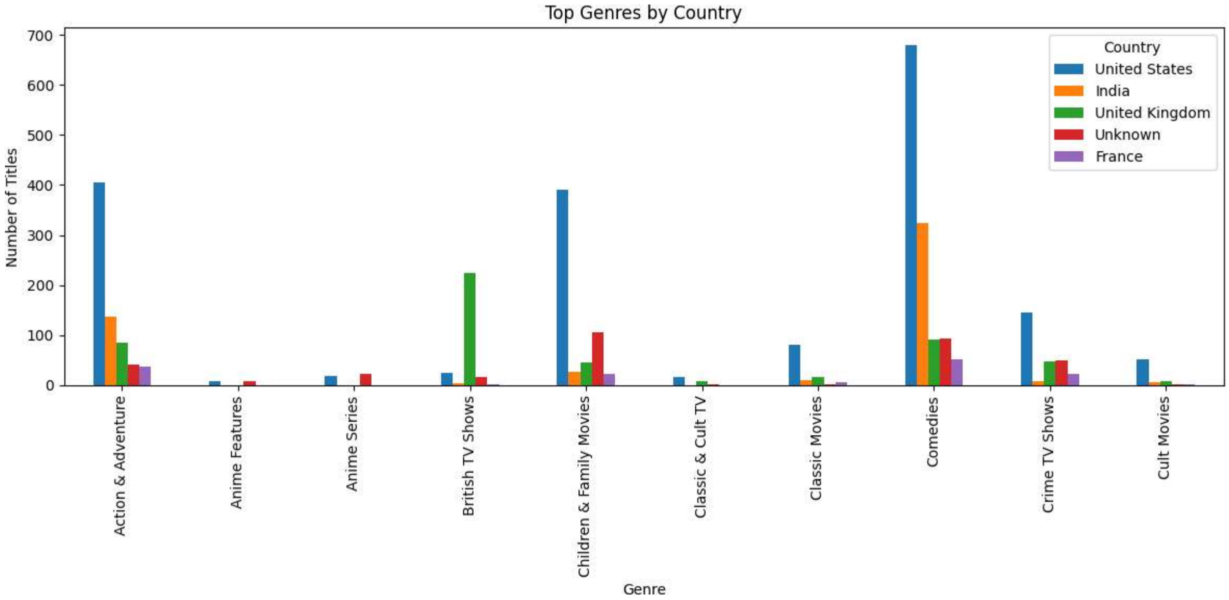
```
#18:-In which months does Netflix add the most content?

month_counts = df['month_added'].value_counts().sort_index()
month_counts.plot(kind='bar', color='purple')
plt.title('Content Added by Month')
plt.xlabel('Month')
plt.ylabel('Number of Titles')
plt.show()
```



```
#19:-How does the genre distribution vary across different years?

genre_year = df[['year_added', 'listed_in']].dropna()
genre_year =
genre_year.assign(genres=genre_year['listed_in'].str.split(',')).explo
de('genres')
genre_year['genres'] = genre_year['genres'].str.strip()

genre_trend = genre_year.groupby(['year_added',
'genres']).size().unstack(fill_value=0)
genre_trend.T.head(10).plot(figsize=(12,6))
plt.title('Genre Distribution Over Years')
plt.ylabel('Number of Titles')
plt.xlabel('Year')
plt.legend(loc='upper left', bbox_to_anchor=(1,1))
plt.show()
```



```
#20:-Which countries produce the most content in each genre?


country_genre = df[['country', 'listed_in']].dropna()
country_genre =
country_genre.assign(genres=country_genre['listed_in'].str.split(','))
.explode('genres')
country_genre['genres'] = country_genre['genres'].str.strip()
country_genre =
country_genre.assign(country_split=country_genre['country'].str.split(
','))
country_genre = country_genre.explode('country_split')
country_genre['country_split'] =
country_genre['country_split'].str.strip()
matrix = country_genre.groupby(['country_split',
'genres']).size().unstack(fill_value=0)
top_countries =
country_genre['country_split'].value_counts().head(5).index
matrix.loc[top_countries].T.head(10).plot(kind='bar', figsize=(12,6))
plt.title('Top Genres by Country')
plt.ylabel('Number of Titles')
plt.xlabel('Genre')
plt.legend(title='Country', bbox_to_anchor=(1,1))
plt.tight_layout()
plt.show()
```


Top Genres by Country
Number of Titles
0
100
200
300
400
500
600
700
Country
United States
India
United Kingdom
Unknown
France
Action & Adventure
Anime Features
Anime Series
British TV Shows
Children & Family Movies
Classic & Cult TV
Classic Movies
Comedies
Crime TV Shows
Cult Movies
Genre