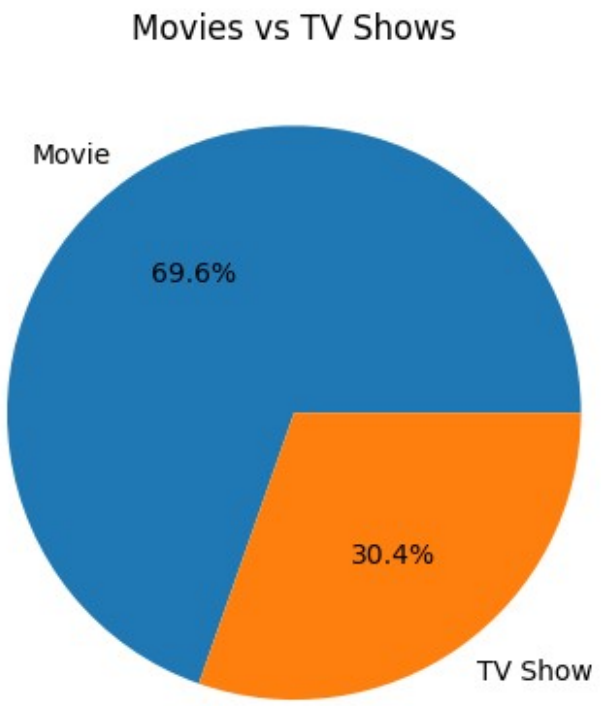
```python
# max movie year vs budgeting

import pandas as pd
df = pd.read_csv("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...
```

```python
#1: What is the ratio of Movies vs TV Shows on Netflix?

type_counts = df['type'].value_counts()
print(type_counts)


# answer for ratio between movies and tv shows
movie_count = type_counts.get('Movie', 0)
tv_count = type_counts.get('TV Show', 0)
```

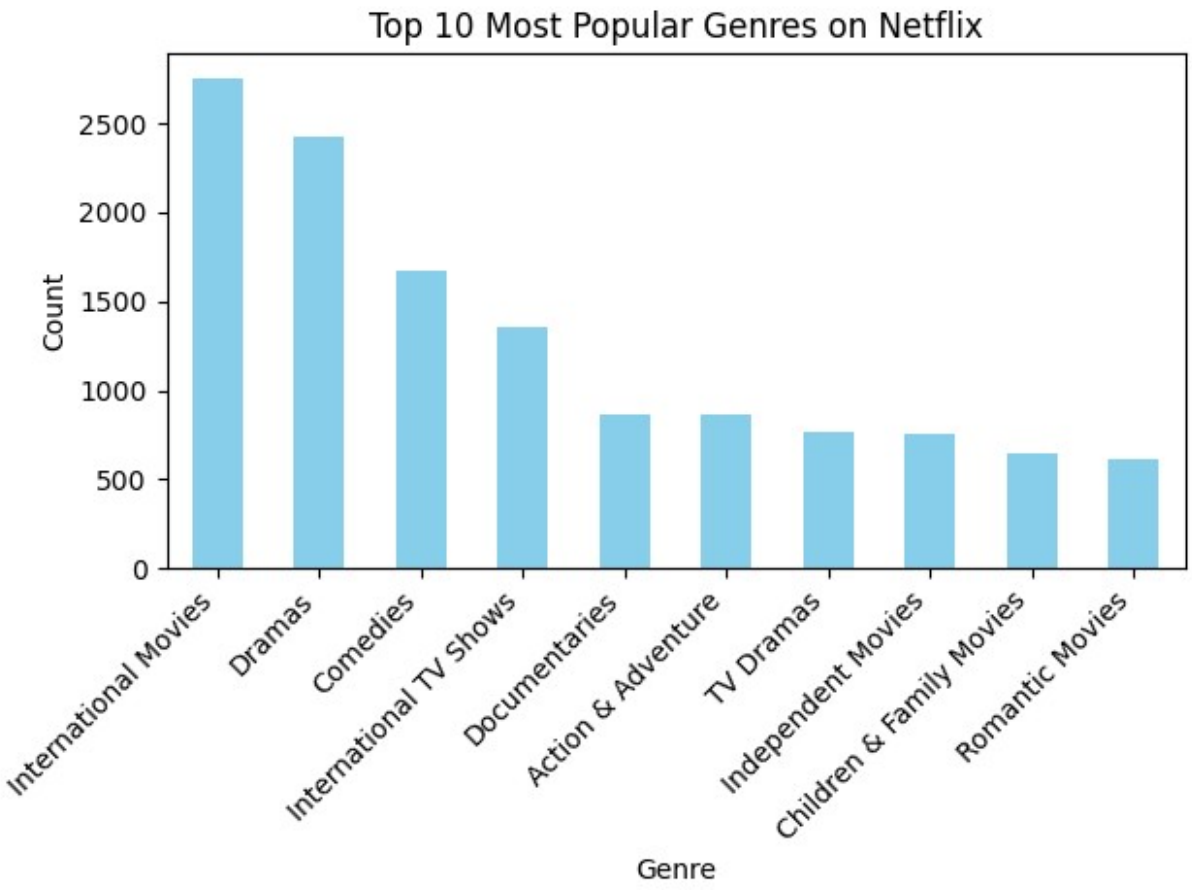
```
ratio = movie_count / tv_count
print(f"Ratio of Movies to TV Shows: {ratio:.2f} : 1")


total = movie_count + tv_count
print(f"Movies: {movie_count/total*100:.1f}%")
print(f"TV Shows: {tv_count/total*100:.1f}%")


import matplotlib.pyplot as plt

type_counts.plot(kind='pie', autopct='%1.1f%%', title='Movies vs TV
Shows')
plt.ylabel('')
plt.show()
Insight: Movies are the dominant content type,accounting for 69.6% of
the entire catalog, while TV Shows make up the remaining 30.4%. This
translates to a Movie to TV Show ratio of 2.29:1. This disparity
strongly suggests that Netflix's content investment priorities are
heavily skewed toward films.

type
Movie      6131
TV Show    2676
Name: count, dtype: int64
Ratio of Movies to TV Shows: 2.29 : 1
Movies: 69.6%
TV Shows: 30.4%
```




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


# Split genres correctly
df_genres = df['listed_in'].str.split(', ',
expand=True).stack().reset_index(drop=True)

# Count frequency
genre_counts = df_genres.value_counts()
print(genre_counts)

# Plot top 10 genres
import matplotlib.pyplot as plt

top_genres = genre_counts.head(10)
top_genres.plot(kind='bar', color='skyblue', title='Top 10 Most
Popular Genres on Netflix')
plt.xlabel('Genre')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
Insight: The top-ranking genre, international Movies (2,752 titles),
highlights Netflix's critical strategy of catering to a global
audience and aggressively acquiring non-English language films. This
is immediately followed by Dramas (2,427 titles) and Comedies (1,674
```

titles), suggesting these two universal genres are fundamental to the platform's core offering.

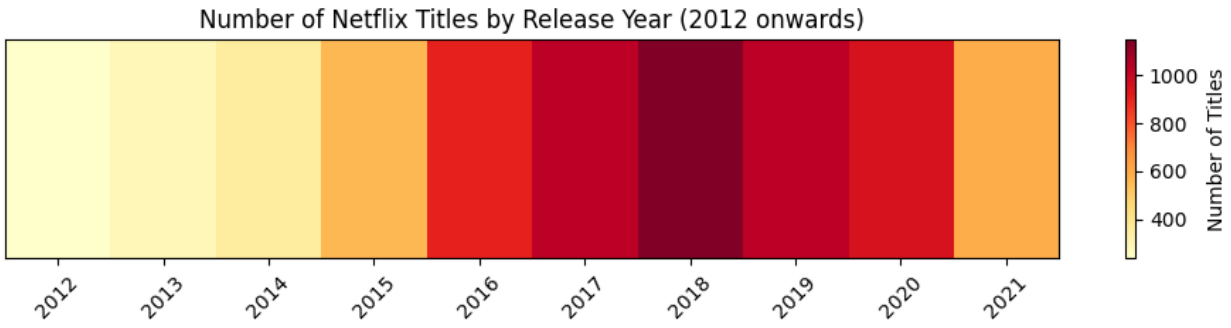
```
International Movies            2752
Dramas                          2427
Comedies                        1674
International TV Shows          1351
Documentaries                    869
Action & Adventure               859
TV Dramas                        763
Independent Movies               756
Children & Family Movies         641
Romantic Movies                  616
TV Comedies                      581
Thrillers                        577
Crime TV Shows                   470
Kids' TV                         451
Docuseries                       395
Music & Musicals                 375
Romantic TV Shows                370
Horror Movies                    357
Stand-Up Comedy                  343
Reality TV                       255
British TV Shows                 253
Sci-Fi & Fantasy                 243
Sports Movies                    219
Anime Series                     176
Spanish-Language TV Shows        174
TV Action & Adventure            168
Korean TV Shows                  151
Classic Movies                   116
LGBTQ Movies                     102
TV Mysteries                      98
Science & Nature TV               92
TV Sci-Fi & Fantasy               84
TV Horror                         75
Anime Features                    71
Cult Movies                       71
Teen TV Shows                     69
Faith & Spirituality              65
TV Thrillers                      57
Movies                            57
Stand-Up Comedy & Talk Shows      56
Classic & Cult TV                 28
TV Shows                          16
Name: count, dtype: int64
```




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

# Count number of titles per release year
year_counts = df['release_year'].value_counts().sort_index()
print(year_counts.tail(10))

import matplotlib.pyplot as plt
import numpy as np

# Filter for release_year >= 2012
df_recent = df[df['release_year'] >= 2012]

# Count number of titles per year
year_counts = df_recent['release_year'].value_counts().sort_index()

# Convert to 2D array for heatmap-like plot
heatmap_data = np.array([year_counts.values])

plt.figure(figsize=(12,2))
plt.imshow(heatmap_data, cmap='YlOrRd', aspect='auto')
plt.colorbar(label='Number of Titles')

# Show only every other year to reduce clutter
```

```
years = year_counts.index
plt.xticks(ticks=np.arange(0, len(years), 1), labels=years,
rotation=45)
plt.yticks([])

plt.title('Number of Netflix Titles by Release Year (2012 onwards)')
plt.show()
Insight: The dataset clearly shows a massive surge in content
production and acquisition between 2016 and 2019. The number of
released titles in the library grew dramatically, peaking in 2018 with
1,147 titles. This four-year period—2016, 2017, 2018, and 2019—saw
Netflix at its most aggressive in building its catalog with newly
released content. Following this peak, there's a noticeable decline in
volume in the years 2020 (953 titles) and 2021 (592 titles).
```

```
release_year
2012     237
2013     288
2014     352
2015     560
2016     902
2017    1032
2018    1147
2019    1030
2020     953
2021     592
Name: count, dtype: int64
```

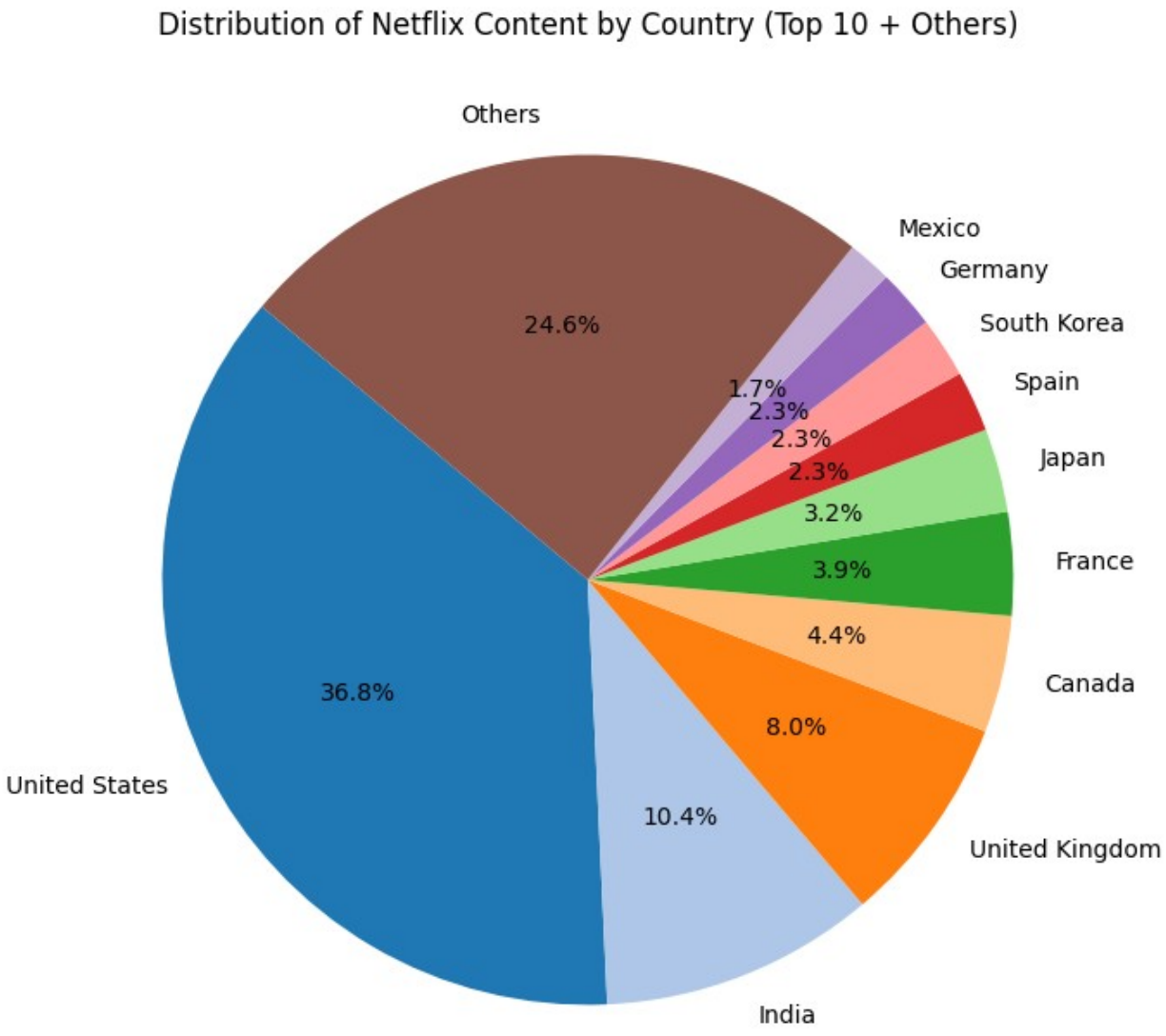



```
#4: Which countries produce the most Netflix content?

country_counts = df['country'].value_counts()
print(country_counts.head(10))  # top 10 countries
```

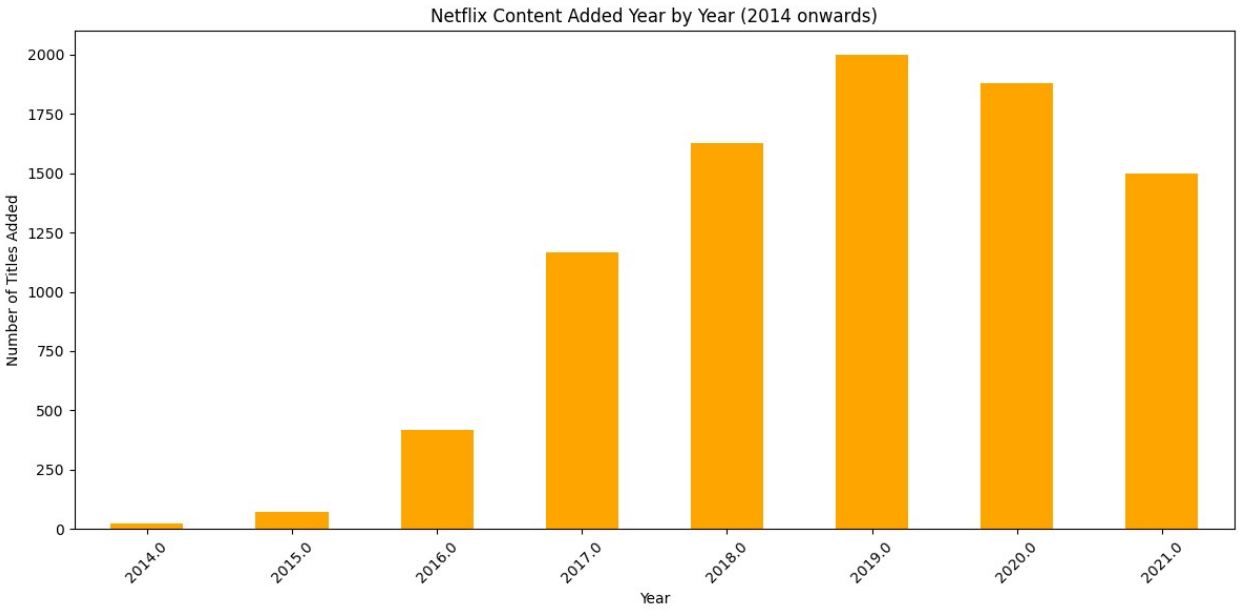
```
country
United States     2818
India              972
United Kingdom     419
Japan              245
South Korea        199
```

```
Canada              181
Spain               145
France              124
Mexico              110
Egypt               106
Name: count, dtype: int64

# Some rows might have missing or NaN country values — they are
ignored by value_counts()
# Some titles have multiple countries separated by commas

# Split multiple countries into separate rows
df_countries = df['country'].dropna().str.split(', ',
expand=True).stack().reset_index(drop=True)

# Count frequency
country_counts = df_countries.value_counts()
print(country_counts.head(10))

United States     3689
India             1046
United Kingdom     804
Canada             445
France             393
Japan              318
Spain              232
South Korea        231
Germany            226
Mexico             169
Name: count, dtype: int64

import matplotlib.pyplot as plt
import pandas as pd

# Split multiple countries into separate rows
df_countries = df['country'].dropna().str.split(', ',
expand=True).stack().reset_index(drop=True)

# Count frequency of each country
country_counts = df_countries.value_counts()

# Select top 10 countries
top_countries = country_counts.head(10)
others_count = country_counts[10:].sum()  # combine remaining
countries into "Others"

# Combine top 10 and "Others" using pd.concat
pie_data = pd.concat([top_countries, pd.Series({'Others':
others_count})])

# Plot pie chart
```

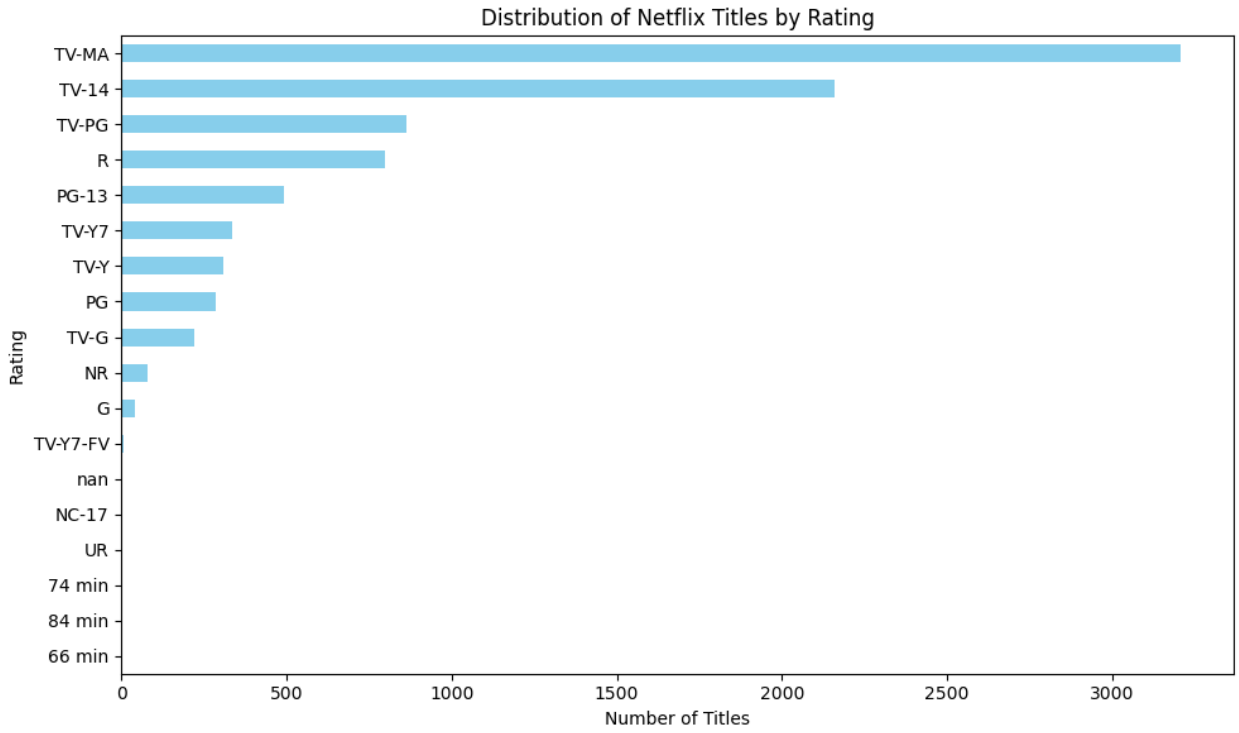
```
plt.figure(figsize=(8,8))
plt.pie(pie_data, labels=pie_data.index, autopct='%1.1f%%',
startangle=140, colors=plt.cm.tab20.colors)
plt.title('Distribution of Netflix Content by Country (Top 10 +
Others)')
plt.show()
Insight: The United States is by far the most dominant content
producer, contributing 3,689 titles, which underscores the platform's
foundation in American content and its continued investment in
Hollywood and domestic production. Following the US, there is a
significant investment in key international markets: India (1,046
titles) and the United Kingdom (804 titles).
```



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

df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')

df['added_year'] = df['date_added'].dt.year

added_counts = df['added_year'].value_counts().sort_index()
print(added_counts)

# Filter for years >= 2014
added_counts_recent = added_counts[added_counts.index >= 2014]

# Plot bar chart
plt.figure(figsize=(12,6))
added_counts_recent.plot(kind='bar', color='orange')
plt.title('Netflix Content Added Year by Year (2014 onwards)')
plt.xlabel('Year')
plt.ylabel('Number of Titles Added')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
Insight: The volume of new titles added to Netflix exploded between
2016 and 2019, growing from 418 titles in 2016 to a peak of 1,999
titles in 2019. This period marks Netflix's most ambitious push to
rapidly expand its catalog to dominate the streaming market. Since
this peak, however, the trend has reversed: content additions declined
to 1,878 in 2020 and further to 1,498 in 2021.

added_year
2008.0       2
2009.0       2
2010.0       1
2011.0      13
2012.0       3
2013.0      10
2014.0      23
2015.0      73
2016.0     418
2017.0    1164
2018.0    1625
2019.0    1999
2020.0    1878
2021.0    1498
Name: count, dtype: int64
```

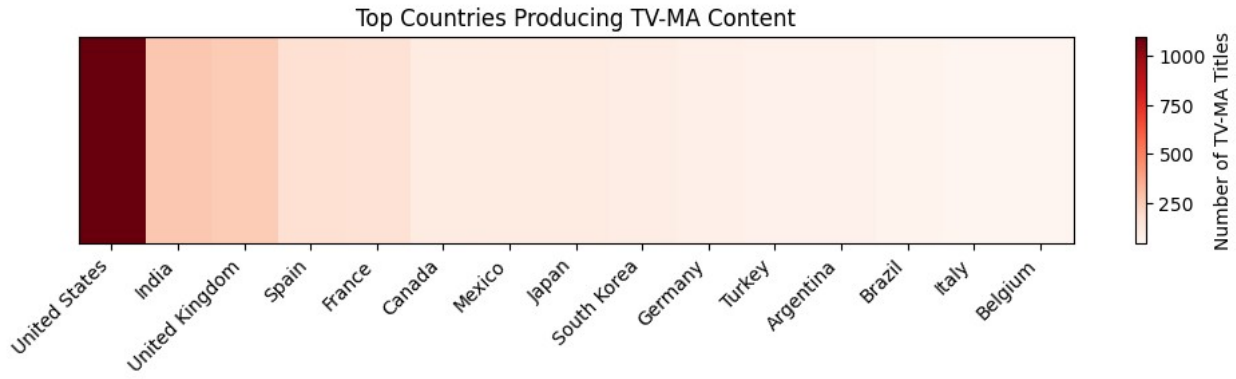



```
#6: Which ratings (e.g., TV-MA, PG, etc.) are most frequent on
Netflix?

rating_counts = df['rating'].value_counts(dropna=False)

rating_counts = rating_counts.rename({pd.NA: "Unknown", None:
"Unknown"})

plt.figure(figsize=(10,6))
rating_counts.plot(kind='barh', color='skyblue')
plt.title('Distribution of Netflix Titles by Rating')
plt.xlabel('Number of Titles')
plt.ylabel('Rating')
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```



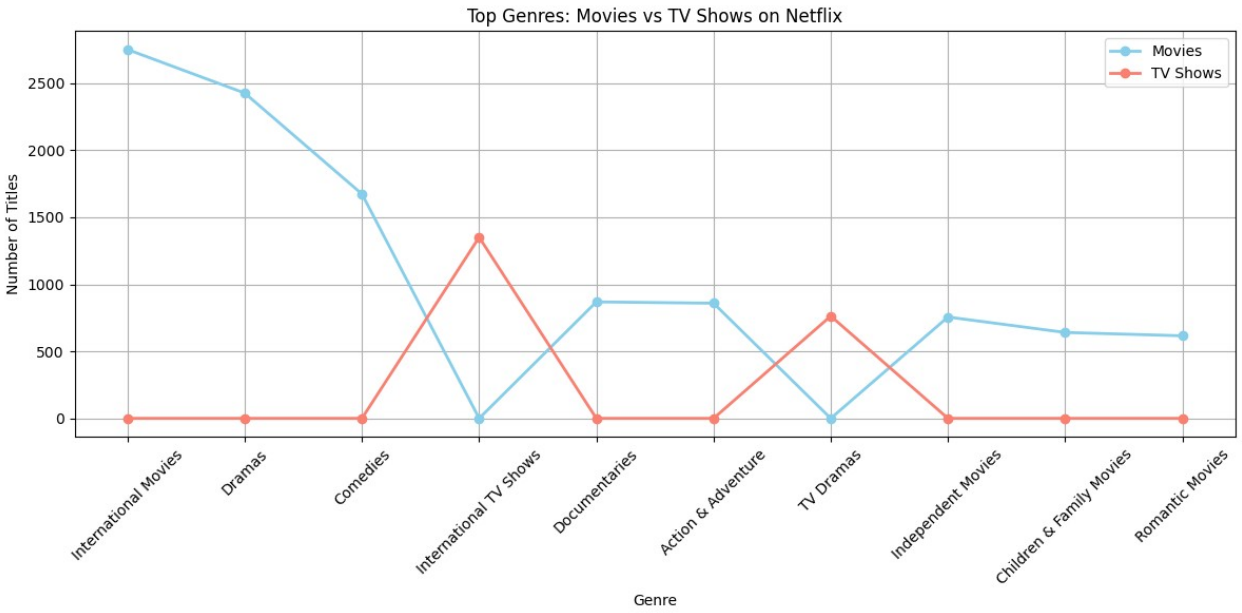

```
#7:  Do some countries tend to produce more mature content (TV-MA)?

import numpy as np

# Filter for TV-MA titles
tv_ma_df = df[df['rating'] == 'TV-MA']

# Split multiple countries into separate rows
tv_ma_countries = tv_ma_df['country'].dropna().str.split(', ',
expand=True).stack().reset_index(drop=True)

# Count TV-MA titles per country
tv_ma_counts = tv_ma_countries.value_counts()

# Optionally, include only top 15 countries for a cleaner heatmap
top_countries = tv_ma_counts.head(15)
heatmap_data = np.array([top_countries.values])

plt.figure(figsize=(12,2))
plt.imshow(heatmap_data, cmap='Reds', aspect='auto')
plt.colorbar(label='Number of TV-MA Titles')

# Label the countries
plt.xticks(ticks=np.arange(len(top_countries)),
labels=top_countries.index, rotation=45, ha='right')
plt.yticks([])  # no y-axis needed

plt.title('Top Countries Producing TV-MA Content')
```

```
plt.show()
Insight: The analysis of the TV-MA (Mature Audience) content, which
comprises the largest content rating category on Netflix, highlights
the key markets driving the platform's adult-oriented programming.
The United States is the overwhelming leader in producing TV-MA
content, with 1,100 titles, cementing its role as the primary source
for mature movies and TV shows. However, the data also reveals
significant international investment in adult content, particularly
from India (266 titles) and the United Kingdom (251 titles).
```




```
#8: Which genres are more associated with TV Shows vs Movies?

# Split genres into separate rows
df_genres = df[['type', 'listed_in']].dropna()
df_genres =
df_genres.assign(listed_in=df_genres['listed_in'].str.split(',
')).explode('listed_in')

# Count genres for each type
genre_type_counts = df_genres.groupby(['type',
'listed_in']).size().unstack(fill_value=0)

# Sum across Movies + TV Shows to find top genres
top_genres =
genre_type_counts.sum().sort_values(ascending=False).head(10).index

# Filter the DataFrame to only include top genres
genre_type_counts_top = genre_type_counts[top_genres]

import matplotlib.pyplot as plt

plt.figure(figsize=(12,6))

# Plot Movies line
plt.plot(top_genres, genre_type_counts_top.loc['Movie'], marker='o',
label='Movies', color='skyblue', linewidth=2)
```

```
# Plot TV Shows line
plt.plot(top_genres, genre_type_counts_top.loc['TV Show'], marker='o',
label='TV Shows', color='salmon', linewidth=2)

plt.title('Top Genres: Movies vs TV Shows on Netflix')
plt.xlabel('Genre')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

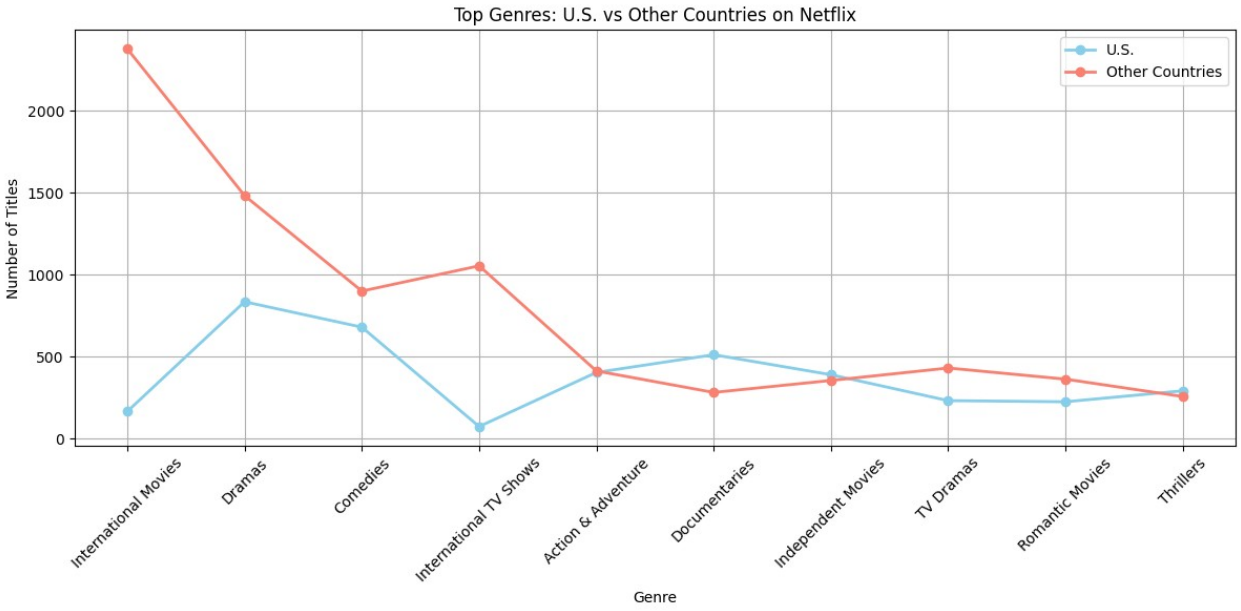



```
#9:  Which genres dominate the U.S. vs other countries?

import pandas as pd

# Split genres into separate rows
df_genres = df[['country', 'listed_in']].dropna()
df_genres =
df_genres.assign(listed_in=df_genres['listed_in'].str.split(',
')).explode('listed_in')

# Separate U.S. and other countries
us_genres = df_genres[df_genres['country'].str.contains('United
States', na=False)]
other_genres = df_genres[~df_genres['country'].str.contains('United
States', na=False)]

# Count genres
```

```
us_counts = us_genres['listed_in'].value_counts()
other_counts = other_genres['listed_in'].value_counts()

# Select top 10 genres based on combined counts
top_genres = (us_counts +
other_counts).sort_values(ascending=False).head(10).index

# Filter counts for top genres
us_top = us_counts.reindex(top_genres, fill_value=0)
other_top = other_counts.reindex(top_genres, fill_value=0)

import pandas as pd
from IPython.display import display

# Combine counts into a DataFrame
genre_comparison = pd.DataFrame({
    'Genre': top_genres,
    'U.S.': us_top.values,
    'Other Countries': other_top.values
})

# Set Genre as index
genre_comparison.set_index('Genre', inplace=True)

# Display the table
display(genre_comparison)



import matplotlib.pyplot as plt

plt.figure(figsize=(12,6))

# U.S. line
plt.plot(top_genres, us_top.values, marker='o', label='U.S.',
color='skyblue', linewidth=2)

# Other countries line
plt.plot(top_genres, other_top.values, marker='o', label='Other
Countries', color='salmon', linewidth=2)

plt.title('Top Genres: U.S. vs Other Countries on Netflix')
plt.xlabel('Genre')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
Insight: Netflix uses its Movie strategy to quickly secure vast
```

volumes of diverse international and niche-interest content (Documentaries, Independent Films), while it leverages its TV Show strategy to build committed, serial viewership through long-running International TV Shows and platform-exclusive TV Dramas and TV Comedies.

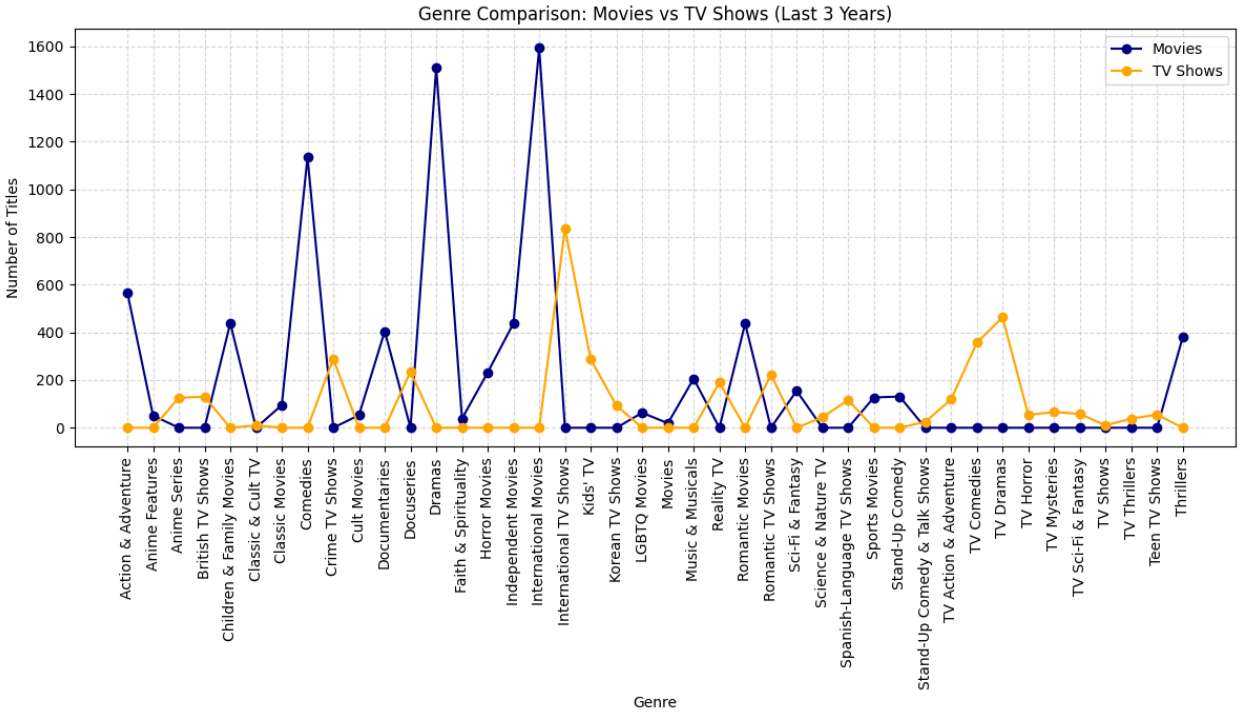
Genre	U.S.	Other Countries
International Movies	166	2377
Dramas	835	1482
Comedies	680	900
International TV Shows	74	1054
Action & Adventure	404	413
Documentaries	512	282
Independent Movies	390	355
TV Dramas	232	431
Romantic Movies	225	363
Thrillers	292	257




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

import pandas as pd
import matplotlib.pyplot as plt

# Load your dataset again (make sure the file name is correct)
df = pd.read_csv("netflix.csv")

# Ensure 'date_added' is datetime
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')

# Continue from here
```

```
latest_year = df['date_added'].dt.year.max()
recent_df = df[df['date_added'].dt.year >= latest_year - 2]  # last 3
years

# Ensure date_added is datetime
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')

latest_year = df['date_added'].dt.year.max()
recent_df = df[df['date_added'].dt.year >= latest_year - 2]  # last 3
years

# Drop rows with missing values
recent_genres = recent_df[['type', 'listed_in']].dropna()

# Split and explode genres
recent_genres =
recent_genres.assign(listed_in=recent_genres['listed_in'].str.split(',
')).explode('listed_in')

# Count genre frequencies separately
movies_genres = recent_genres[recent_genres['type'] == 'Movie']
['listed_in'].value_counts()
tv_genres = recent_genres[recent_genres['type'] == 'TV Show']
['listed_in'].value_counts()

genre_compare = pd.DataFrame({
    'Movies': movies_genres,
    'TV Shows': tv_genres
}).fillna(0)

from IPython.display import display
display(genre_compare)

plt.figure(figsize=(12,7))
plt.plot(genre_compare.index, genre_compare['Movies'], color='navy',
marker='o', label='Movies')
plt.plot(genre_compare.index, genre_compare['TV Shows'],
color='orange', marker='o', label='TV Shows')

plt.title('Genre Comparison: Movies vs TV Shows (Last 3 Years)')
plt.xlabel('Genre')
plt.ylabel('Number of Titles')
plt.xticks(rotation=90)
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
Insight: The recent popularity of genres like Reality TV or K-Dramas
suggests a current trend that Netflix should heavily invest in to
satisfy changing consumer tastes.
```

listed_in	Movies	TV Shows
Action & Adventure	568.0	0.0
Anime Features	51.0	0.0
Anime Series	0.0	126.0
British TV Shows	0.0	130.0
Children & Family Movies	439.0	0.0
Classic & Cult TV	0.0	10.0
Classic Movies	95.0	0.0
Comedies	1135.0	0.0
Crime TV Shows	0.0	289.0
Cult Movies	53.0	0.0
Documentaries	405.0	0.0
Docuseries	0.0	235.0
Dramas	1511.0	0.0
Faith & Spirituality	38.0	0.0
Horror Movies	232.0	0.0
Independent Movies	438.0	0.0
International Movies	1593.0	0.0
International TV Shows	0.0	836.0
Kids' TV	0.0	287.0
Korean TV Shows	0.0	93.0
LGBTQ Movies	63.0	0.0
Movies	19.0	0.0
Music & Musicals	206.0	0.0
Reality TV	0.0	190.0
Romantic Movies	437.0	0.0
Romantic TV Shows	0.0	223.0
Sci-Fi & Fantasy	155.0	0.0
Science & Nature TV	0.0	44.0
Spanish-Language TV Shows	0.0	116.0
Sports Movies	127.0	0.0
Stand-Up Comedy	131.0	0.0
Stand-Up Comedy & Talk Shows	0.0	25.0
TV Action & Adventure	0.0	119.0
TV Comedies	0.0	357.0
TV Dramas	0.0	463.0
TV Horror	0.0	54.0
TV Mysteries	0.0	66.0
TV Sci-Fi & Fantasy	0.0	58.0
TV Shows	0.0	10.0
TV Thrillers	0.0	38.0
Teen TV Shows	0.0	55.0
Thrillers	380.0	0.0




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

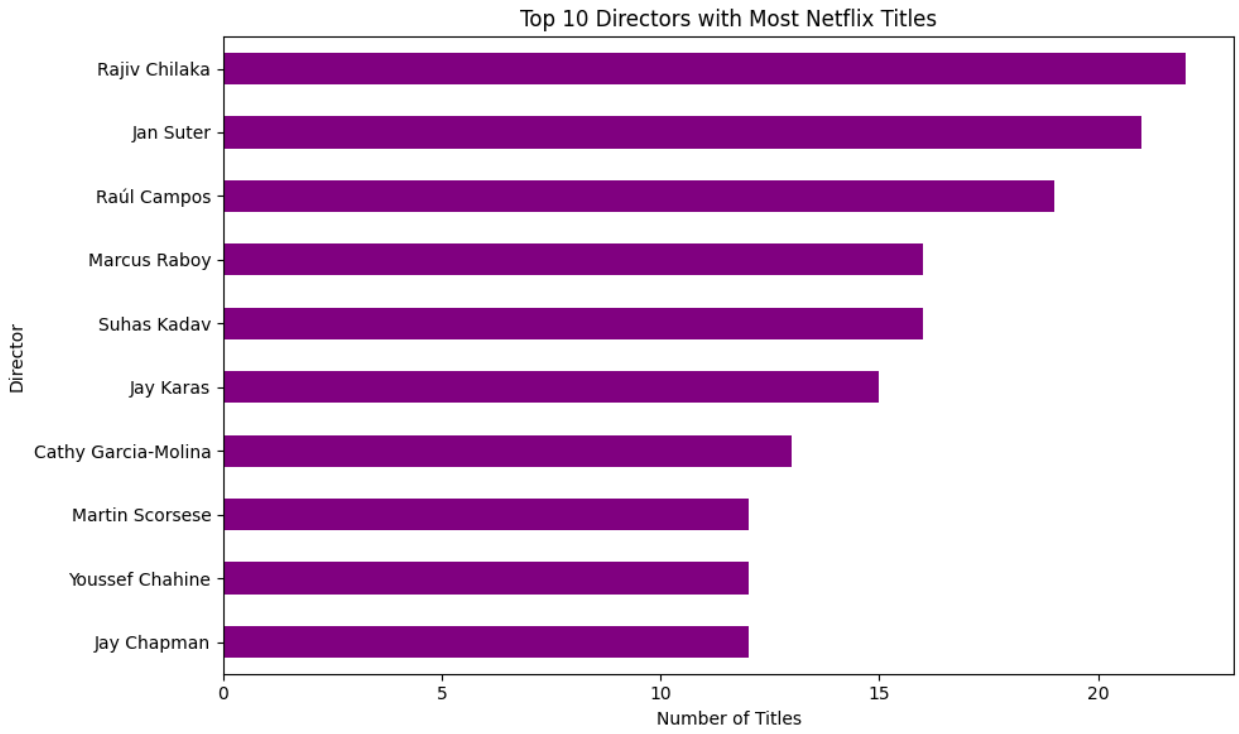
Insight: Count the frequency of names in the Director column. The top 10 names are valuable for forming exclusive or priority-access partnerships, as they are proven content creators for the platform.

```
import pandas as pd
import matplotlib.pyplot as plt

# Drop missing director names
directors = df['director'].dropna()

# Split multiple directors (some entries have multiple names separated
by commas)
all_directors = directors.str.split(',',
expand=True).stack().str.strip()

# Count how many titles each director has
director_counts = all_directors.value_counts().head(10)

# Display as table
print("Top 10 Directors with Most Netflix Titles:")
display(director_counts.to_frame(name='Number of Titles'))

# Plot horizontal bar chart
plt.figure(figsize=(10,6))
director_counts.sort_values().plot(kind='barh', color='purple')
```

```
plt.title('Top 10 Directors with Most Netflix Titles')
plt.xlabel('Number of Titles')
plt.ylabel('Director')
plt.tight_layout()
plt.show()
```

Top 10 Directors with Most Netflix Titles:

	Number of Titles
Rajiv Chilaka	22
Jan Suter	21
Raúl Campos	19
Suhas Kadav	16
Marcus Raboy	16
Jay Karas	15
Cathy Garcia-Molina	13
Jay Chapman	12
Youssef Chahine	12
Martin Scorsese	12



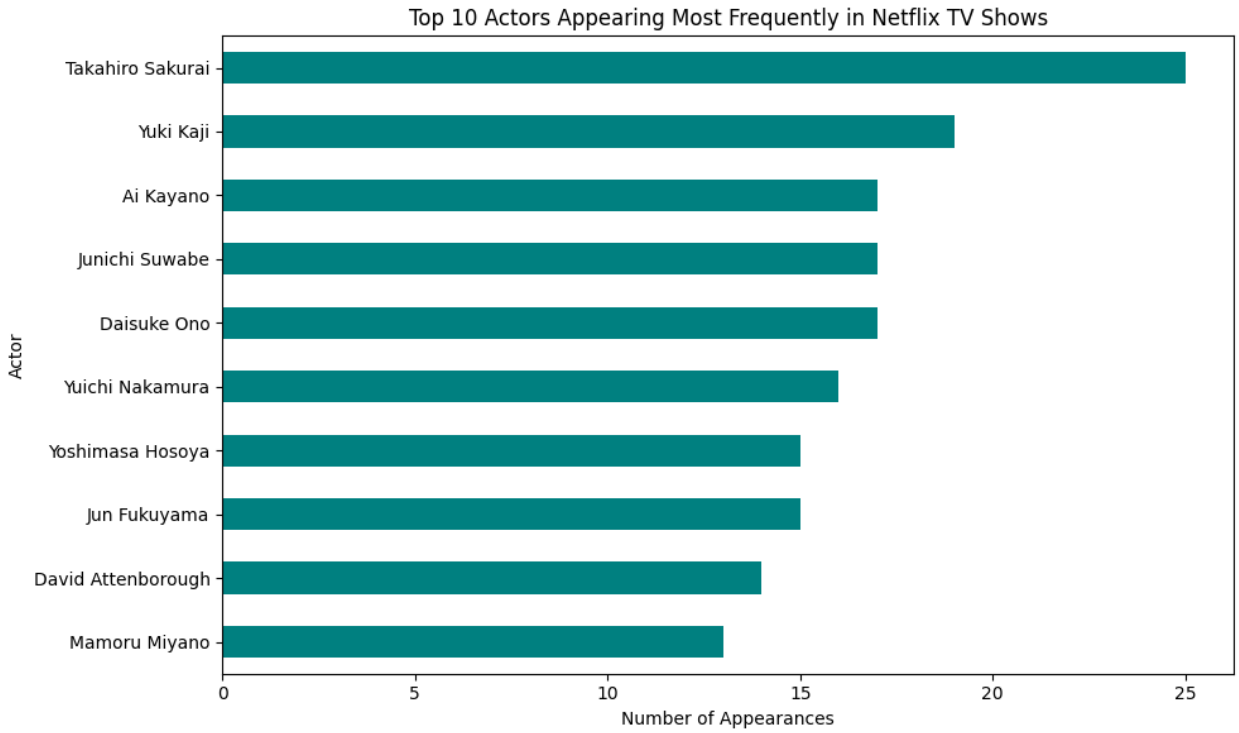
#12 : Which actors appear most frequently in Netflix shows?

insight: This requires splitting the entries in the Cast column and counting individual actor frequencies. The most frequent actors are key talent for new projects, as their presence may attract existing viewership and improve retention

```python
import pandas as pd
import matplotlib.pyplot as plt

# Filter only TV Shows
tv_shows = df[df['type'] == 'TV Show']

# Drop missing cast entries
actors = tv_shows['cast'].dropna()

# Split multiple actors in a single row (comma-separated)
all_actors = actors.str.split(',', expand=True).stack().str.strip()

# Count actor appearances
actor_counts = all_actors.value_counts().head(10)

# Display as table
print("Top 10 Most Frequent Actors in Netflix TV Shows:")
display(actor_counts.to_frame(name='Number of Appearances'))

# Plot horizontal bar chart
plt.figure(figsize=(10,6))
actor_counts.sort_values().plot(kind='barh', color='teal')
plt.title('Top 10 Actors Appearing Most Frequently in Netflix TV Shows')
plt.xlabel('Number of Appearances')
plt.ylabel('Actor')
plt.tight_layout()
plt.show()
```

```
Top 10 Most Frequent Actors in Netflix TV Shows:
```

	Number of Appearances
Takahiro Sakurai	25
Yuki Kaji	19
Daisuke Ono	17
Junichi Suwabe	17
Ai Kayano	17
Yuichi Nakamura	16
Jun Fukuyama	15
Yoshimasa Hosoya	15
David Attenborough	14
Mamoru Miyano	13




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

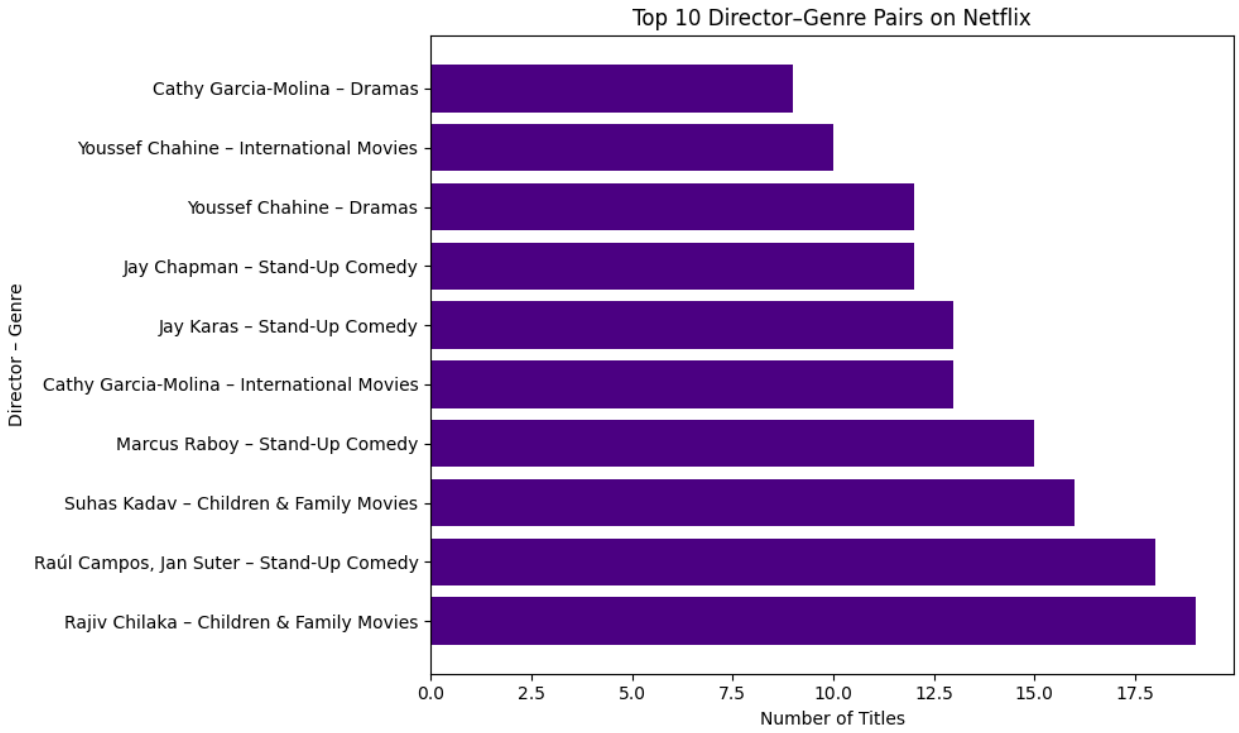
Insight: Create a combined key of Director and one of the genres from Listed_in. A high-frequency pair (e.g., Director X and "International Comedies") reveals a successful, proven creative formula that Netflix can replicate or expand upon.

```
import pandas as pd
import matplotlib.pyplot as plt

# Drop missing directors or genres
df_pairs = df.dropna(subset=['director', 'listed_in'])

# Split multiple genres per title
df_pairs =
df_pairs.assign(listed_in=df_pairs['listed_in'].str.split(','))

# Explode so each (director, genre) combination becomes one row
df_pairs = df_pairs.explode('listed_in')

# Clean whitespace
df_pairs['listed_in'] = df_pairs['listed_in'].str.strip()

# Count frequency of each (director, genre) pair
pair_counts = (
    df_pairs.groupby(['director', 'listed_in'])
    .size()
    .reset_index(name='count')
```

```
    .sort_values(by='count', ascending=False)
    .head(10)
)

# Display as table
print("Top 10 Most Frequent Director–Genre Pairs on Netflix:")
display(pair_counts)

# Plot
plt.figure(figsize=(10,6))
plt.barh(
    pair_counts['director'] + ' – ' + pair_counts['listed_in'],
    pair_counts['count'],
    color='indigo'
)
plt.title('Top 10 Director–Genre Pairs on Netflix')
plt.xlabel('Number of Titles')
plt.ylabel('Director – Genre')
plt.tight_layout()
plt.show()
```

```
Top 10 Most Frequent Director–Genre Pairs on Netflix:
```

	director	listed_in	count
8097	Rajiv Chilaka	Children & Family Movies	19
8227	Raúl Campos, Jan Suter	Stand-Up Comedy	18
9725	Suhas Kadav	Children & Family Movies	16
6150	Marcus Raboy	Stand-Up Comedy	15
1659	Cathy Garcia-Molina	International Movies	13
4251	Jay Karas	Stand-Up Comedy	13
4242	Jay Chapman	Stand-Up Comedy	12
10802	Youssef Chahine	Dramas	12
10804	Youssef Chahine	International Movies	10
1658	Cathy Garcia-Molina	Dramas	9




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

Insight: Count the number of entries where Director or Cast are null/missing. A high number of missing values indicates either a data cleanup need or a large volume of non-star-driven content, suggesting that marketing may need to focus on the title/genre/description rather than talent.

```
# Count titles with missing directors or cast
unknown_directors = df['director'].isna().sum()
unknown_cast = df['cast'].isna().sum()

# Total records
total_titles = len(df)

# Display results
print(f"Total Titles: {total_titles}")
print(f"Titles with Unknown Directors: {unknown_directors}
({(unknown_directors/total_titles)*100:.2f}%)")
print(f"Titles with Unknown Cast Members: {unknown_cast}
({(unknown_cast/total_titles)*100:.2f}%)")
```

```
Total Titles: 8807
Titles with Unknown Directors: 2634 (29.91%)
Titles with Unknown Cast Members: 825 (9.37%)
```

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

Insight: Filter for Type = 'Movie' and calculate the average of the Duration (in minutes). If the average is ~97 minutes, this suggests content around this length is the current platform standard and potentially optimal for viewer attention.

```
# Filter only movies (use .copy() to avoid warning)
movies = df[df['type'] == 'Movie'].copy()

# Extract numeric duration
movies['duration_num'] = movies['duration'].str.replace(' min', '',
regex=False).astype(float)

# Calculate average
avg_duration = movies['duration_num'].mean()

print(f"□ Average Movie Duration on Netflix: {avg_duration:.2f}
minutes")
```

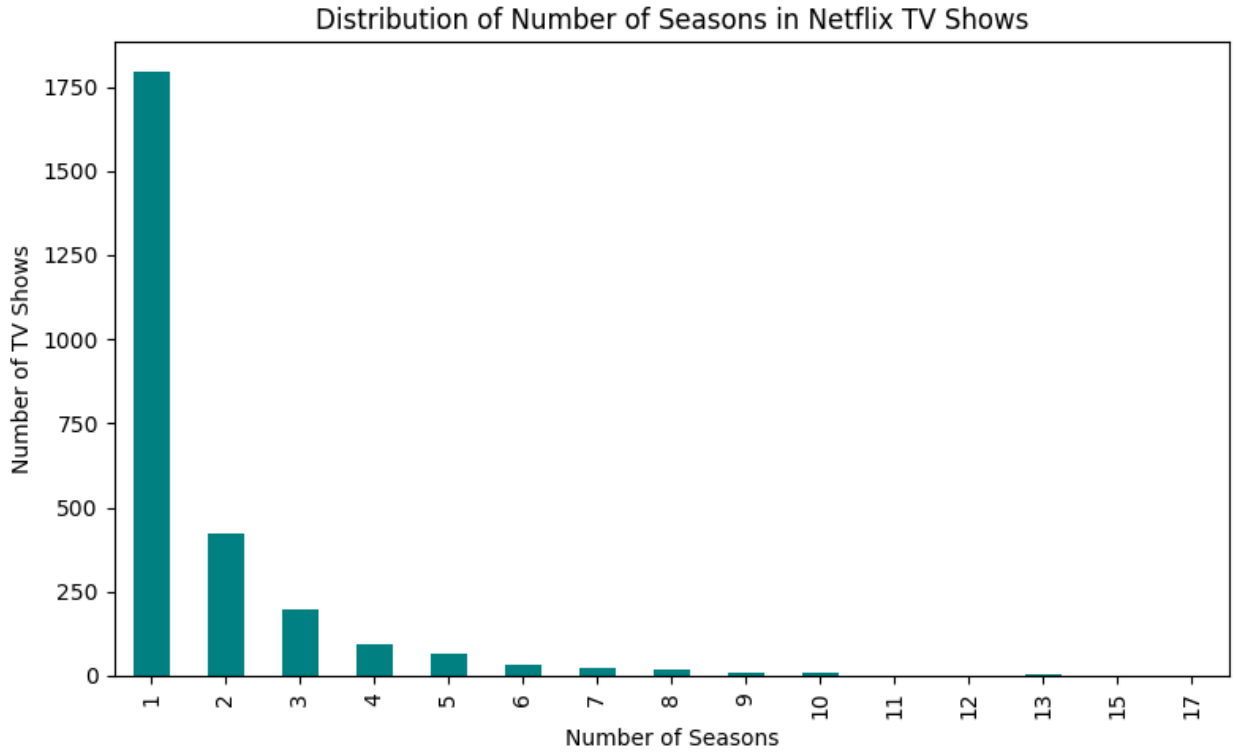
```
□ Average Movie Duration on Netflix: 99.58 minutes
```

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

Insight: Filter for Type = 'TV Show' and find the mode of the Duration (in seasons). The high frequency of '1 Season' suggests many shows are either limited series or are often canceled/not renewed after the first season, indicating a high-volume, short-life-cycle strategy for TV content.

```
# Filter only TV Shows
tv_shows = df[df['type'] == 'TV Show'].copy()

# Extract numeric number of seasons
tv_shows['seasons_num'] = tv_shows['duration'].str.replace(' Season',
'', regex=False).str.replace('s', '', regex=False).astype(int)

# Find most common number of seasons
most_common_seasons = tv_shows['seasons_num'].mode()[0]
count_common = (tv_shows['seasons_num'] == most_common_seasons).sum()

print(f"□ Most Common Number of Seasons: {most_common_seasons}")
print(f"Number of Shows with {most_common_seasons} Season(s):
{count_common}")

# Plot distribution
tv_shows['seasons_num'].value_counts().sort_index().plot(
    kind='bar',
    color='teal',
    figsize=(8,5),
    title='Distribution of Number of Seasons in Netflix TV Shows'
)
plt.xlabel('Number of Seasons')
plt.ylabel('Number of TV Shows')
```

```
plt.tight_layout()
plt.show()

 Most Common Number of Seasons: 1
Number of Shows with 1 Season(s): 1793
```

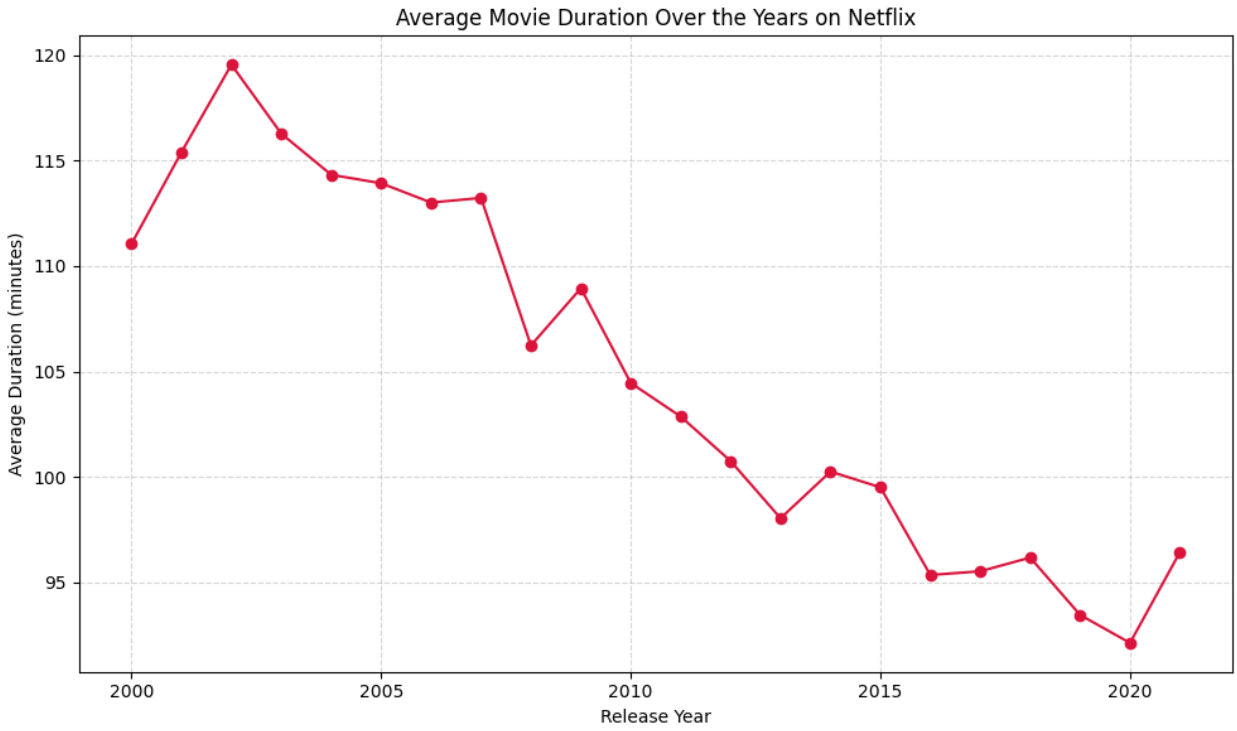



```
#17: Is there a trend in movie durations over the years?
```

Insight: Plot the average Duration (minutes) of Movies against the Release_year. If the plot shows a slight decrease in average movie length in recent years, it could reflect a strategic move towards shorter content due to changing viewer attention spans.

```
import matplotlib.pyplot as plt
import pandas as pd

movies['duration_num'] = movies['duration'].str.replace(' min', '',
regex=False).astype(float)

# Drop missing release years or durations
movies = movies.dropna(subset=['release_year', 'duration_num'])

# Group by release year and calculate average duration
yearly_duration = movies.groupby('release_year')
['duration_num'].mean().reset_index(

# Filter to recent years (optional, e.g., after 2000 for clarity)
```

```
yearly_duration = yearly_duration[yearly_duration['release_year'] >=
2000]

# Plot trend
plt.figure(figsize=(10,6))
plt.plot(yearly_duration['release_year'],
yearly_duration['duration_num'], color='crimson', marker='o')
plt.title('Average Movie Duration Over the Years on Netflix')
plt.xlabel('Release Year')
plt.ylabel('Average Duration (minutes)')
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```



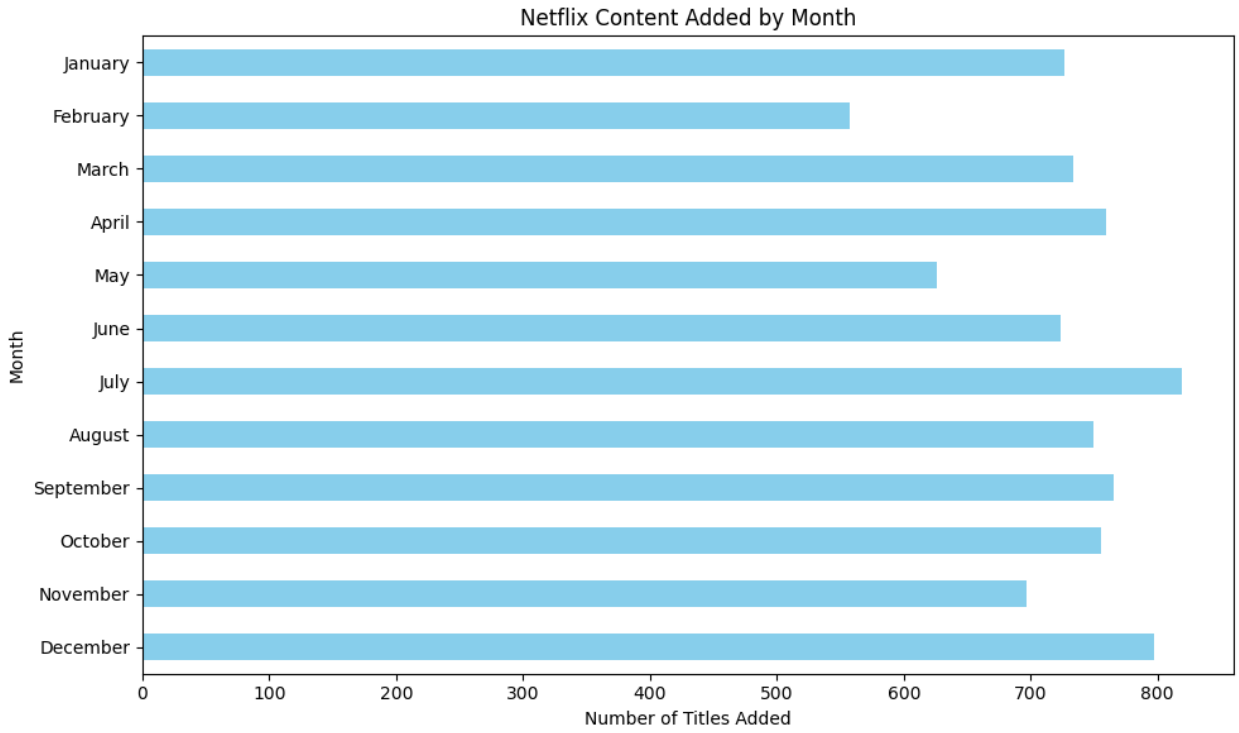

```
#18: In which months does Netflix add the most content?
```

Insight: Extract the month from the Date_added column and count the frequencies. If October, November, and December are the peak months, this suggests a strategy to release major content during holiday seasons or end-of-year viewing spikes, guiding future launch calendar decisions.

```
import pandas as pd
import matplotlib.pyplot as plt

# Ensure date_added is datetime
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
```

```
# Drop missing dates
df_month = df.dropna(subset=['date_added']).copy()

# Extract month name
df_month['month_added'] = df_month['date_added'].dt.month_name()

# Count number of titles added per month
month_counts = df_month['month_added'].value_counts()

# Optional: order months correctly
months_order = ['January', 'February', 'March', 'April', 'May',
'June',
                'July', 'August', 'September', 'October', 'November',
'December']
month_counts = month_counts.reindex(months_order)

# Display table
from IPython.display import display
display(month_counts.to_frame(name='Number of Titles Added'))

# Plot horizontal bar chart
plt.figure(figsize=(10,6))
month_counts.plot(kind='barh', color='skyblue')
plt.title('Netflix Content Added by Month')
plt.xlabel('Number of Titles Added')
plt.ylabel('Month')
plt.gca().invert_yaxis()  # highest month on top
plt.tight_layout()
plt.show()
```

month_added	Number of Titles Added
January	727
February	557
March	734
April	759
May	626
June	724
July	819
August	749
September	765
October	755
November	697
December	797




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

Insight: Analyze the percentage breakdown of Listed_in genres for each Release_year or Date_added year. A trend showing a rise in "Anime" and "Reality TV" since 2018, while "Classic Movies" decline, indicates a shift in content strategy towards contemporary and binge-worthy formats.

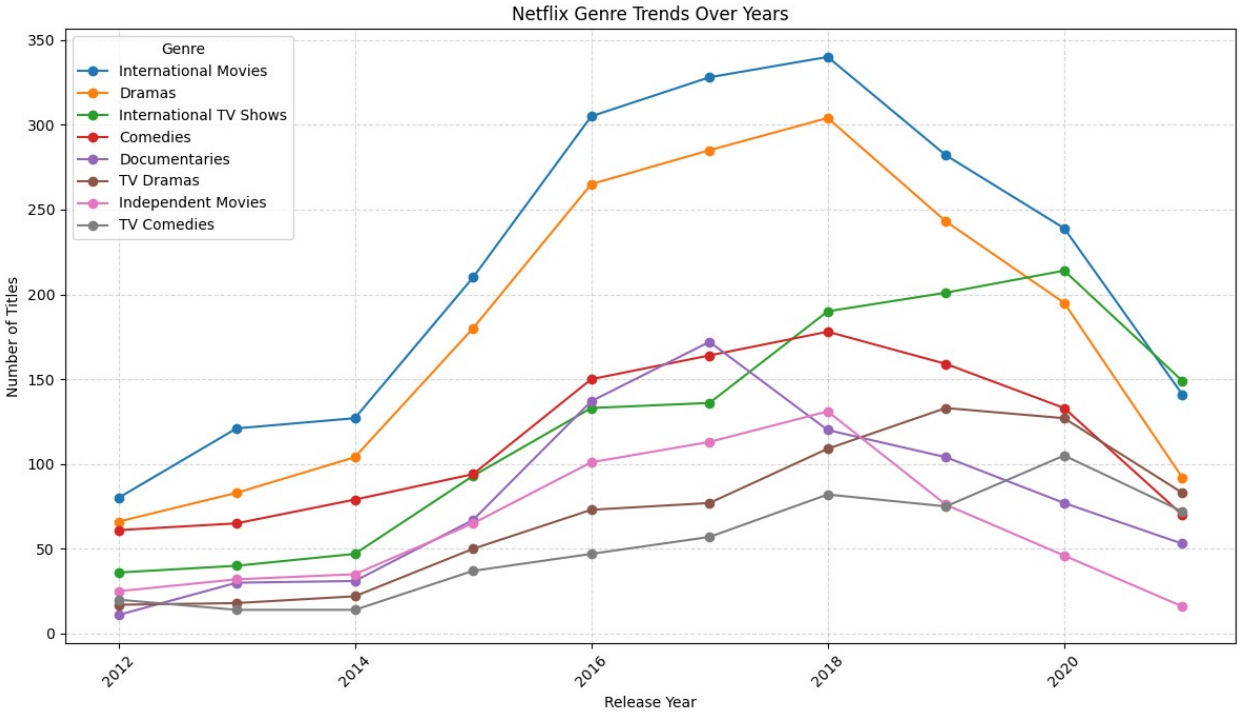
```
import pandas as pd
import matplotlib.pyplot as plt

# Drop rows with missing genre or release year
df_genre_year = df.dropna(subset=['listed_in', 'release_year']).copy()

# Split multiple genres into separate rows
df_genre_year =
df_genre_year.assign(listed_in=df_genre_year['listed_in'].str.split(',
')).explode('listed_in')
df_genre_year['listed_in'] = df_genre_year['listed_in'].str.strip()

# Optional: focus on recent years for clarity
df_genre_year = df_genre_year[df_genre_year['release_year'] >= 2012]

# Create pivot table: rows=year, columns=genre, values=count of titles
genre_year_counts = df_genre_year.pivot_table(index='release_year',
                                              columns='listed_in',
                                              aggfunc='size',
                                              fill_value=0)
```

```
# Display table
from IPython.display import display
display(genre_year_counts)

plt.figure(figsize=(12,7))

# Option: plot only top 8 genres by total count for clarity
top_genres =
genre_year_counts.sum().sort_values(ascending=False).head(8).index
for genre in top_genres:
    plt.plot(genre_year_counts.index, genre_year_counts[genre],
marker='o', label=genre)

plt.title('Netflix Genre Trends Over Years')
plt.xlabel('Release Year')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.legend(title='Genre')
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```

```
listed_in     Action & Adventure  Anime Features  Anime Series  \
release_year
2012                          29               2             4
2013                          29               4             5
2014                          28               2            11
2015                          53               2            11
2016                          80               5            11
2017                          89               6            10
2018                          81               8            24
2019                          44               6            18
2020                          46               3            21
2021                          37               6            23

listed_in     British TV Shows  Children & Family Movies  Classic &
Cult TV  \
release_year

2012                         9                        21
0
2013                        10                        34
1
2014                         7                        29
0
2015                        22                        23
2
2016                        30                        45
1
2017                        34                        55
```

```
1
2018                          37                          69
2
2019                          26                          82
3
2020                          33                          83
3
2021                          17                          40
0

listed_in     Classic Movies  Comedies  Crime TV Shows  Cult
Movies  ...  \
release_year                                                 .
..
2012                          0        61                6
2  ...
2013                          0        65                9
2  ...
2014                          0        79               11
0  ...
2015                          0        94               25
0  ...
2016                          0       150               39
1  ...
2017                          0       164               54
1  ...
2018                          1       178               79
1  ...
2019                          0       159               92
0  ...
2020                          0       133               87
0  ...
2021                          0        70               47
0  ...

listed_in     TV Action & Adventure  TV Comedies  TV Dramas  TV Horror
\
release_year

2012                              1           20         17           0

2013                              2           14         18           0

2014                              4           14         22           3

2015                              5           37         50           8

2016                              9           47         73           4

2017                              8           57         77           4
```

```
2018                              28            82         109          11

2019                              35            75         133          16

2020                              32           105         127          17

2021                              28            72          83          10


listed_in     TV Mysteries  TV Sci-Fi & Fantasy  TV Shows  TV
Thrillers  \
release_year

2012                         0                    0         0
0
2013                         1                    1         0
1
2014                         1                    5         1
1
2015                         6                    5         1
8
2016                         7                    4         0
3
2017                         9                    4         2
3
2018                        15                    7         0
7
2019                        16                   14         2
16
2020                        27                   19         2
7
2021                        14                   14         2
9

listed_in     Teen TV Shows  Thrillers
release_year
2012                          1          9
2013                          0         15
2014                          3         39
2015                          4         32
2016                          5         72
2017                          5         68
2018                          8         83
2019                         14         71
2020                         11         45
2021                          8         33

[10 rows x 42 columns]
```




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

Insight: Group the data by Listed_in (genre) and then find the dominant Country for each genre group. For "Anime," the top country is likely Japan. For "Bollywood Movies," it's India. This highlights niche-specific production hubs for targeted international content licensing

```
# Drop rows with missing country or genre
df_country_genre = df.dropna(subset=['country', 'listed_in']).copy()

# Split multiple genres
df_country_genre =
df_country_genre.assign(listed_in=df_country_genre['listed_in'].str.sp
lit(',')).explode('listed_in')
df_country_genre['listed_in'] =
df_country_genre['listed_in'].str.strip()

# Split multiple countries per title
df_country_genre =
df_country_genre.assign(country=df_country_genre['country'].str.split(
',')).explode('country')
df_country_genre['country'] = df_country_genre['country'].str.strip()

# Pivot table: rows=genre, columns=country, values=count
country_genre_counts = df_country_genre.pivot_table(index='listed_in',
                                                    columns='country',
                                                    aggfunc='size',
                                                    fill_value=0)
```

```
# Display table
from IPython.display import display


# For clarity, choose top 5 genres and top 5 countries overall
top_genres =
country_genre_counts.sum(axis=1).sort_values(ascending=False).head(5).
index
top_countries =
country_genre_counts.sum(axis=0).sort_values(ascending=False).head(5).
index

subset = country_genre_counts.loc[top_genres, top_countries]

# Stacked bar chart
subset.plot(kind='bar', figsize=(12,7), stacked=True)
plt.title('Top Countries Producing Content for Top Genres on Netflix')
plt.xlabel('Genre')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.legend(title='Country')
plt.tight_layout()
plt.show()
```