What Makes a Netflix Show Popular?

Loading The Dataset

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('netflix_titles.csv')
df.head(5)
```

Out[2]:

	show_id	type	title	director	cast	country	date_added	release_year	rati
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	T M
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	T M
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	T M
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	T M

Insight Fetching

In [3]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64 
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
None
```

Cleaning Data

Null value proccessing

In [4]:
```
print(df.isnull().sum())
```

```
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 [5]:
```
df.fillna({
    "director": "unknown",
    "cast": "unknown",
    "country": "unknown"
}, inplace=True)
```

In [6]:
```
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
df.dropna(subset=['date_added'], inplace=True)
# df['year_added'] = df['date_added'].dt.year
```

In [7]:
```
mode_tv = df[df["type"]=="TV Show"]["rating"].mode()[0]
mode_movie = df[df['type'] == 'Movie']['rating'].mode()[0]

df.loc[(df["rating"].isna()) & (df["type"]== "TV Show"), "rating"] = mode_tv
df.loc[(df["rating"].isna())& (df["type"]== "Movie"), "rating"] = mode_movie
```

In [8]:
```
mode_tv = df[df['type'] == 'TV Show']['duration'].mode()[0]
mode_movie = df[df['type'] == 'Movie']['duration'].mode()[0]

df.loc[(df['duration'].isna()) & (df['type'] == 'TV Show'), 'duration'] = mode_tv
df.loc[(df['duration'].isna()) & (df['type'] == 'Movie'), 'duration'] = mode_movie
```

In [9]:
```
print(df.isnull().sum())
```

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

Selecting necessary coloumns

In [10]:
```
df = df.drop('show_id', axis=1)
df = df.drop('description', axis=1)
```

In [11]:
```
df.head(5)
```

Out[11]:

	type	title	director	cast	country	date_added	release_year	rating	dura
0	Movie	Dick Johnson Is Dead	Kirsten Johnson	unknown	United States	2021-09-25	2020	PG-13	90
1	TV Show	Blood & Water	unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	Sea
2	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	unknown	2021-09-24	2021	TV-MA	1 Se
3	TV Show	Jailbirds New Orleans	unknown	unknown	unknown	2021-09-24	2021	TV-MA	1 Se
4	TV Show	Kota Factory	unknown	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	Sea

In [12]:
```
df.info()
```

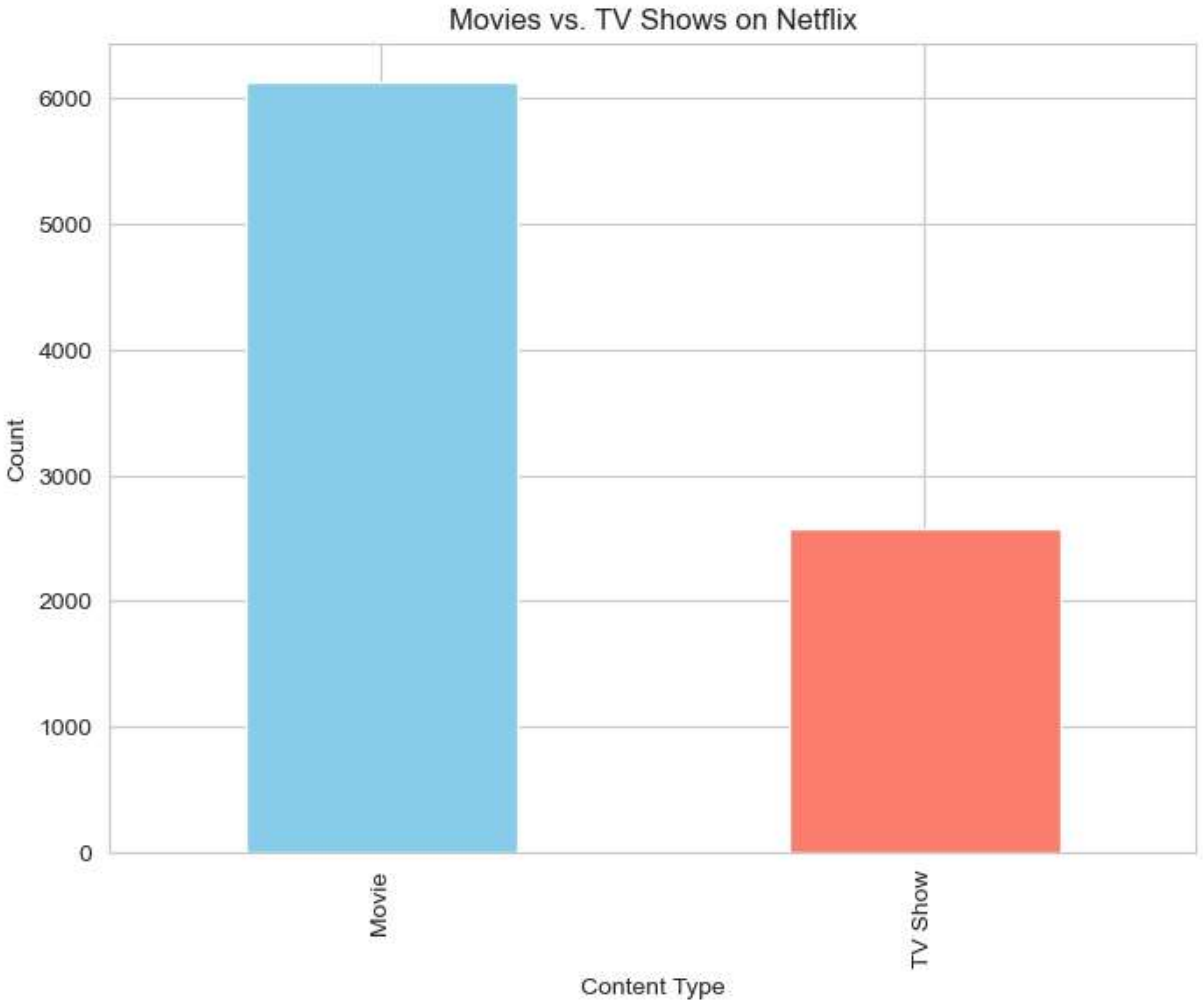
```
<class 'pandas.core.frame.DataFrame'>
Index: 8709 entries, 0 to 8806
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   type          8709 non-null   object        
 1   title         8709 non-null   object        
 2   director      8709 non-null   object        
 3   cast          8709 non-null   object        
 4   country       8709 non-null   object        
 5   date_added    8709 non-null   datetime64[ns]
 6   release_year  8709 non-null   int64         
 7   rating        8709 non-null   object        
 8   duration      8709 non-null   object        
 9   listed_in     8709 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 748.4+ KB
```

Exploratory Data Analysis (EDA)

EDA: Movies vs. TV Shows

Question 1: Does Netflix have more movies or TV shows?
Why: Understanding content distribution helps infer what Netflix prioritizes, which may reflect viewer popularity.

In [13]:
```
sns.set_style('whitegrid')

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

plt.figure(figsize=(8, 6))
type_counts.plot(kind='bar', color=['skyblue', 'salmon'])
plt.title('Movies vs. TV Shows on Netflix')
plt.xlabel('Content Type')
plt.ylabel('Count')
plt.show()

# Print the counts for clarity
print("Content Type Counts:")
print(type_counts)
```




```
Content Type Counts:
type
Movie      6131
TV Show    2578
Name: count, dtype: int64
```

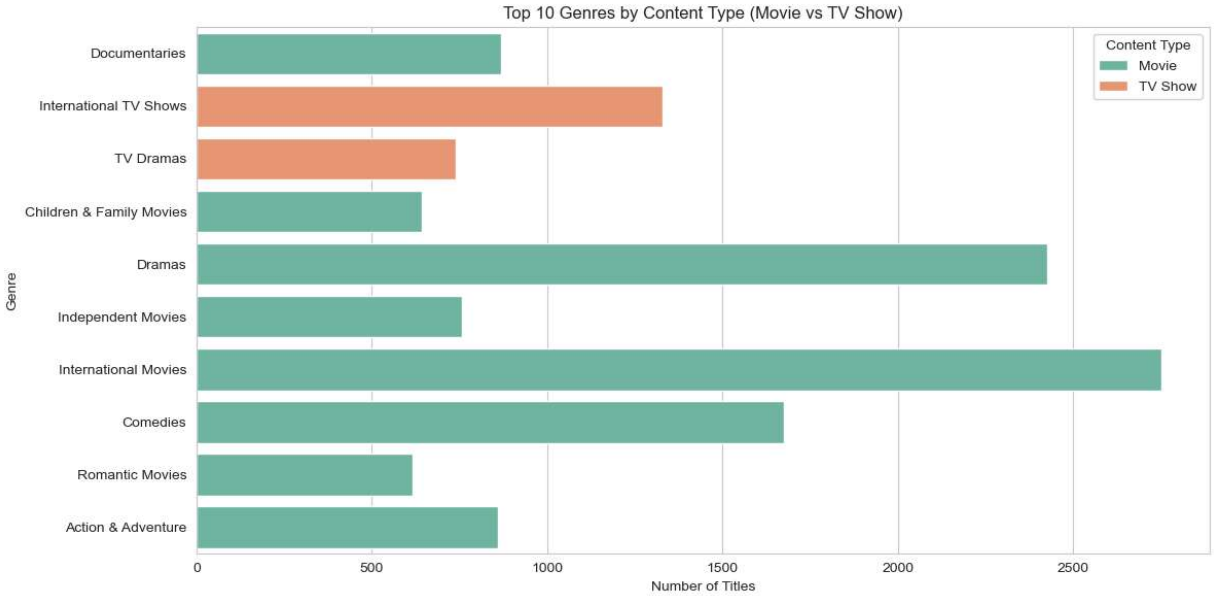
EDA: Top Genres

Question 2: What are the most common genres on Netflix?

Why: Frequent genres may indicate what's popular with viewers, as Netflix likely prioritizes high-demand content.

In [14]:
```
sns.set_style('whitegrid')

top_10_genres = df['listed_in'].str.split(', ').explode().value_counts().head(10).i

# Step 2: Prepare genre column
df_genre = df.copy()
df_genre['genre'] = df_genre['listed_in'].str.split(', ')
df_genre = df_genre.explode('genre')

# Step 3: Filter only top 10 genres
df_genre = df_genre[df_genre['genre'].isin(top_10_genres)]
```

```
# Step 4: Plot
plt.figure(figsize=(12, 6))
sns.countplot(data=df_genre, y='genre', hue='type', palette='Set2')
plt.title('Top 10 Genres by Content Type (Movie vs TV Show)')
plt.xlabel('Number of Titles')
plt.ylabel('Genre')
plt.legend(title='Content Type')
plt.tight_layout()
plt.show()
```

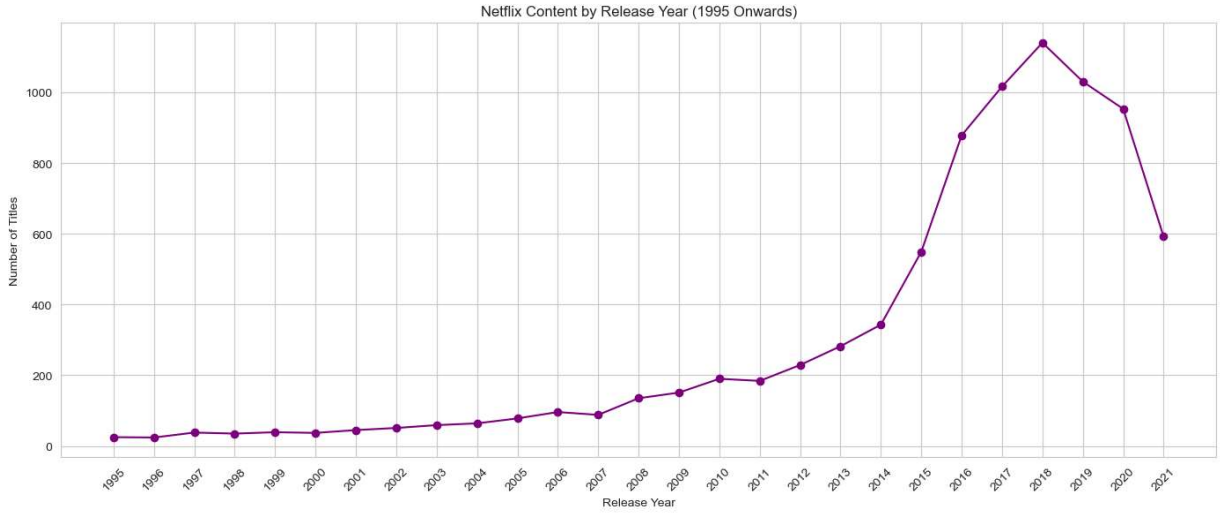


EDA: Content Over Time

Question 3: How has Netflix's content grown over the years?
Why: A rise in recent years may indicate Netflix's growth or focus on new, potentially popular content.

In [15]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Clean style
sns.set_style('whitegrid')

# Filter for years >= 1995
filtered_df = df[df['release_year'] >= 1995]

# Count titles by release year and sort
release_trends = filtered_df['release_year'].value_counts().sort_index()

# Create line chart
plt.figure(figsize=(14, 6))
release_trends.plot(kind='line', marker='o', color='purple')

# Customize x-ticks
years = np.arange(1995, filtered_df['release_year'].max() + 1)
plt.xticks(years, rotation=45)
```

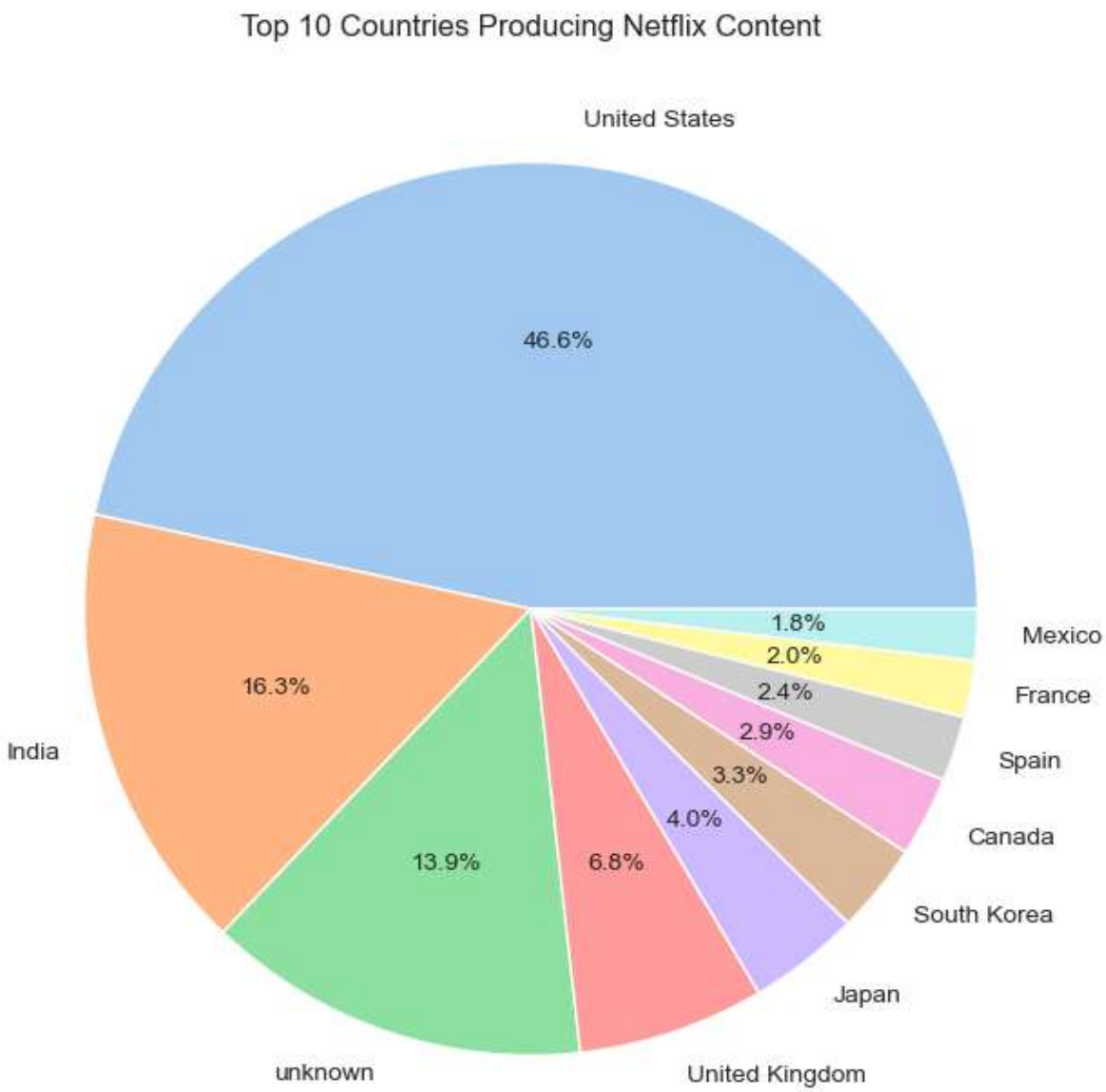
```
plt.title('Netflix Content by Release Year (1995 Onwards)')
plt.xlabel('Release Year')
plt.ylabel('Number of Titles')
plt.tight_layout()
plt.show()
```



EDA: Country-Wise Distribution

Question 4: Which countries produce the most Netflix content?
Why: Dominant countries may indicate Netflix's focus on popular markets or regional viewer preferences.

In [16]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Keep the clean plot style
sns.set_style('whitegrid')

# Count titles by country and get the top 10
countries = df['country'].value_counts().head(10)

# Create a pie chart
plt.figure(figsize=(8, 8))
countries.plot(kind='pie', autopct='%1.1f%%', colors=sns.color_palette('pastel'))
plt.title('Top 10 Countries Producing Netflix Content')
plt.ylabel('')  # Remove y-label for cleaner look
plt.show()

# Print the country counts for clarity
print("Top 10 Countries:")
print(countries)
```




```
Top 10 Countries:
country
United States     2778
India              971
unknown            827
United Kingdom     403
Japan              241
South Korea        195
Canada             173
Spain              141
France             122
Mexico             110
Name: count, dtype: int64
```

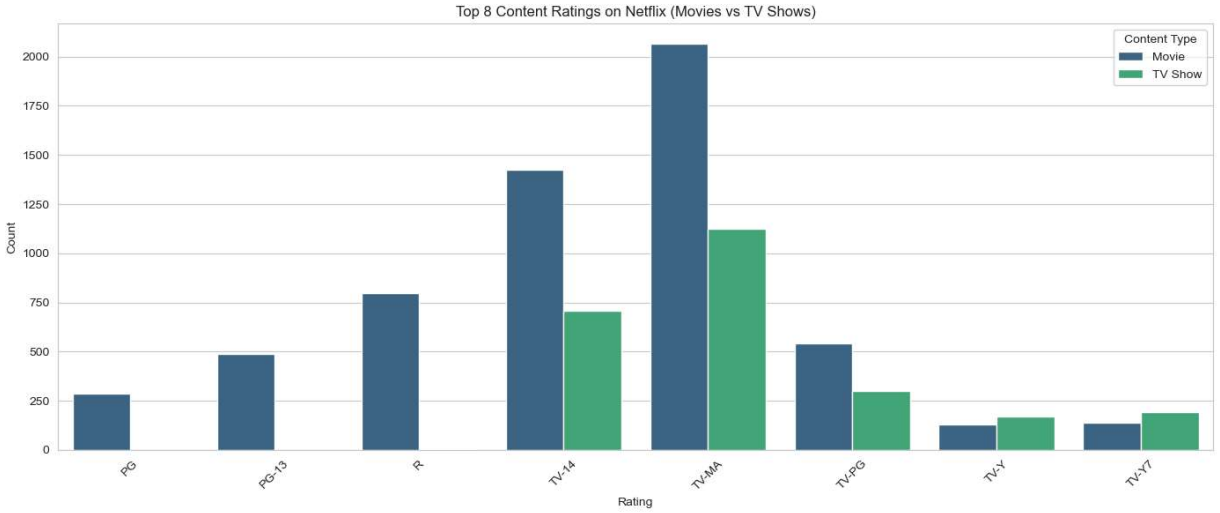
EDA: Content by Rating

Question 5: What are the most common content ratings on Netflix?

Why: Popular ratings may reflect the target audience (e.g., adults vs. families), indicating

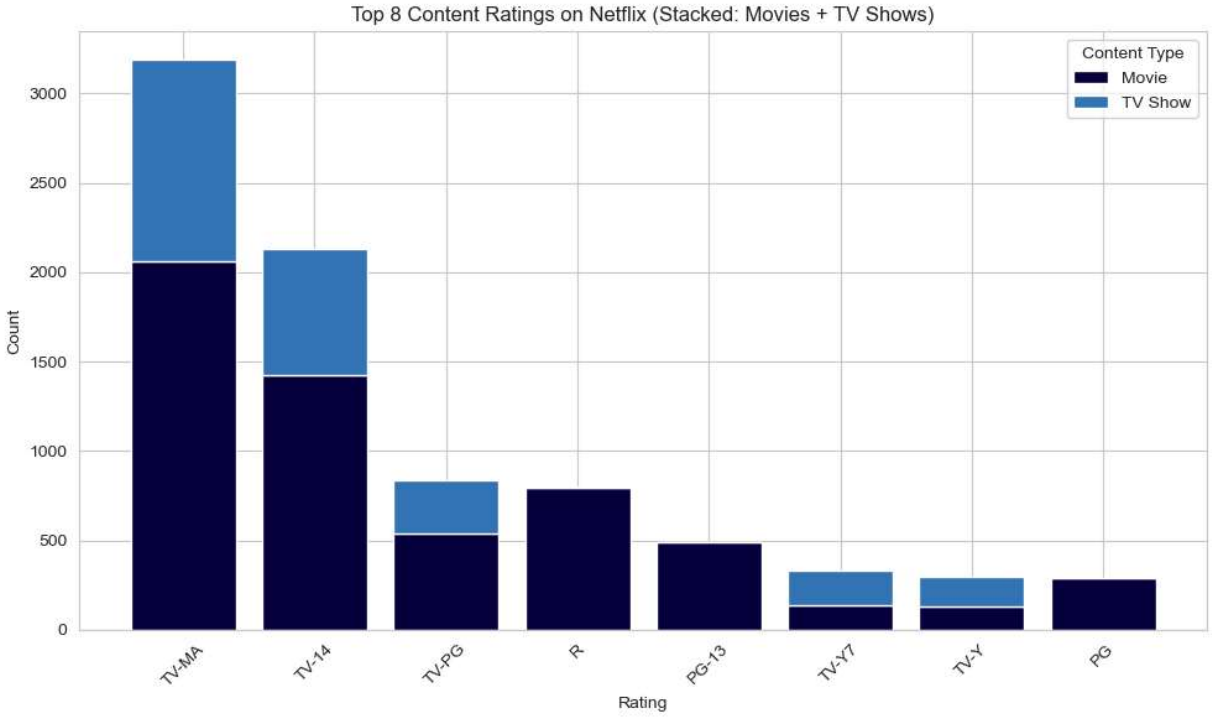
viewer preferences.

In [17]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Keep the clean plot style
sns.set_style('whitegrid')

# Count titles by rating and type, then get the top 8 ratings
ratings_by_type = df.groupby(['type', 'rating']).size().reset_index(name='count')
top_ratings = ratings_by_type.groupby('rating')['count'].sum().nlargest(8).index
ratings_filtered = ratings_by_type[ratings_by_type['rating'].isin(top_ratings)]

# Create a bar chart
plt.figure(figsize=(14, 6))
sns.barplot(data=ratings_filtered, x='rating', y='count', hue='type', palette='viri
plt.title('Top 8 Content Ratings on Netflix (Movies vs TV Shows)')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Content Type')
plt.tight_layout()
plt.show()
```



In [18]:
```
sns.set_style('whitegrid')
ratings_by_type = df.groupby(['rating', 'type']).size().unstack(fill_value=0)

# Keep only top 8 ratings by total count
top_ratings = ratings_by_type.sum(axis=1).nlargest(8).index
ratings_top8 = ratings_by_type.loc[top_ratings]

# Updated colors
movie_color = '#090040'    # Dark navy for Movies
tvshow_color = '#3674B5'   # Bright yellow for TV Shows

# Plot
plt.figure(figsize=(10, 6))
bar1 = plt.bar(ratings_top8.index, ratings_top8['Movie'], label='Movie', color=movi
bar2 = plt.bar(ratings_top8.index, ratings_top8['TV Show'], bottom=ratings_top8['Mo
```

```
# Labels and styling
plt.title('Top 8 Content Ratings on Netflix (Stacked: Movies + TV Shows)')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Content Type')
plt.tight_layout()
plt.show()

# Print for clarity
print("Top 8 Ratings by Type (Stacked View):")
print(ratings_top8)
```

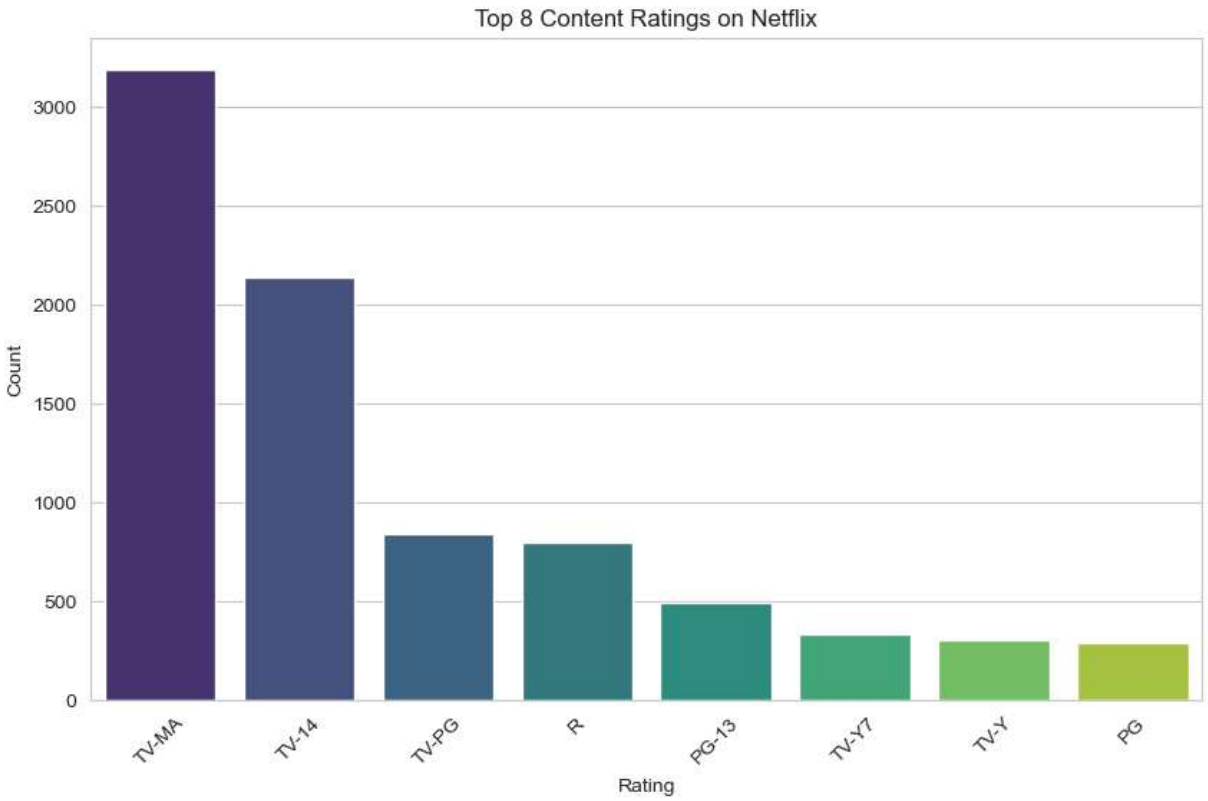



```
Top 8 Ratings by Type (Stacked View):
type    Movie  TV Show
rating
TV-MA    2064     1123
TV-14    1427      706
TV-PG     540      298
R         797        2
PG-13     490        0
TV-Y7     139      191
TV-Y      131      169
PG        287        0
```

In [19]:

```
sns.set_style('whitegrid')

# Count titles by rating and get the top 8 (to avoid clutter)
ratings = df['rating'].value_counts().head(8)

# Create a bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x=ratings.index, y=ratings.values,hue=ratings.index, palette='viridis')
plt.title('Top 8 Content Ratings on Netflix')
```

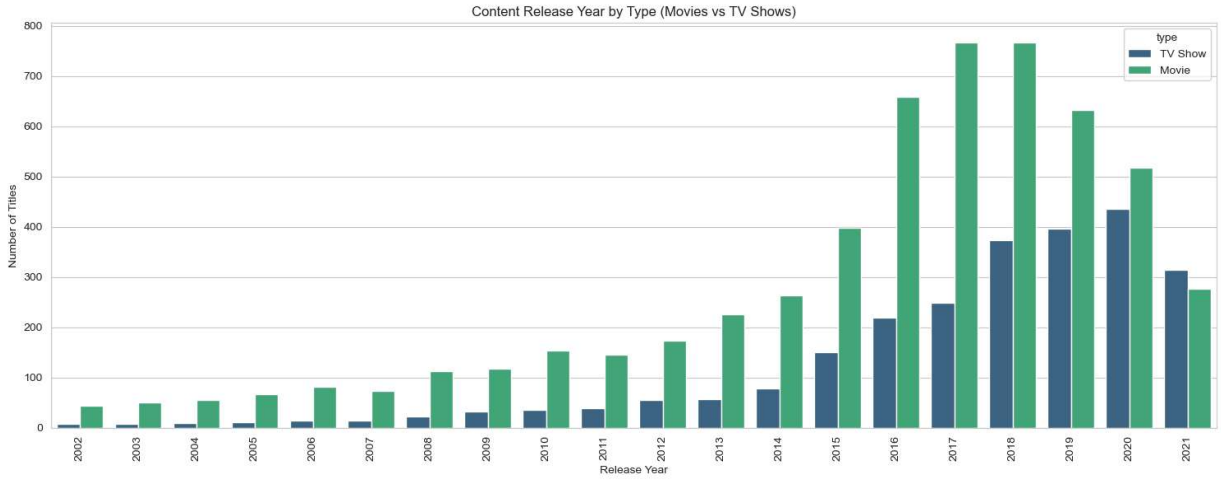
```
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



EDA: Content Type by Release Year

Question 6: What were the most popular content in the individual years by type on Netflix?
Why: Popular ratings may reflect the target audience (e.g., adults vs. families), indicating viewer preferences.

In [20]:
```
plt.figure(figsize=(15, 6))
sns.countplot(data=df, x='release_year', hue='type', palette='viridis', order=sorte
plt.title('Content Release Year by Type (Movies vs TV Shows)')
plt.xticks(rotation=90)
plt.xlabel('Release Year')
plt.ylabel('Number of Titles')
plt.tight_layout()
plt.show()
```

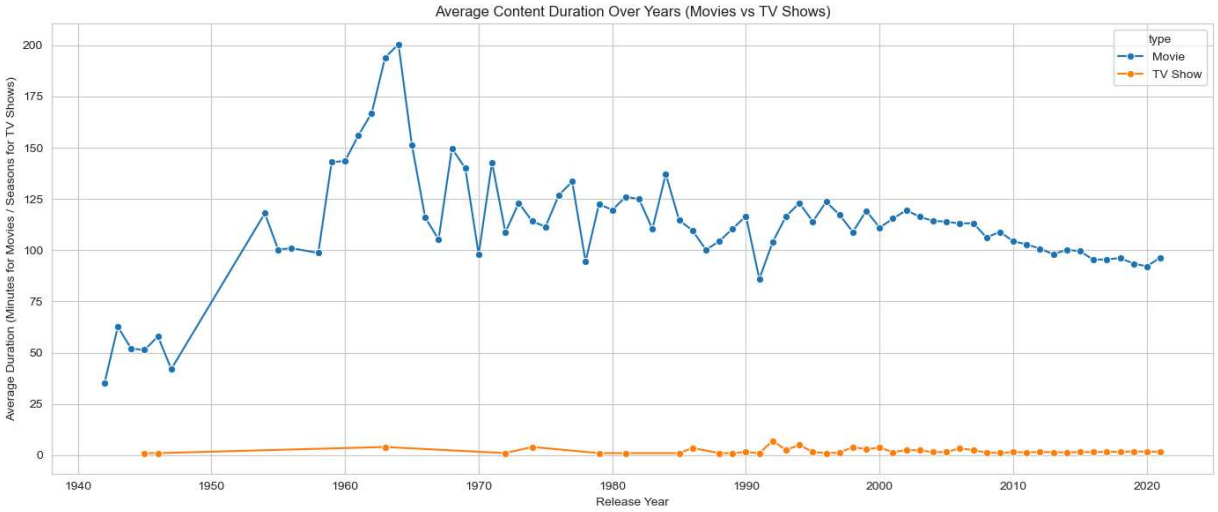


EDA: Content Duration Analysis

Question 7: How does the duration of Movies and TV Shows vary over time or by rating?
Why: This analysis reveals trends in content length (e.g., are Movies getting shorter, or are TV Show seasons longer?) and how it correlates with ratings (e.g., longer dramas vs. short comedies), providing insights into production strategies and audience preferences.

In [21]:
```
# --- Prep Movies Data ---
movies_df = df[df['type'] == 'Movie'].copy()
movies_df['duration'] = movies_df['duration'].astype(str)
movies_df['duration_minutes'] = movies_df['duration'].str.extract(r'(\d+)').astype(

movie_trend = movies_df.groupby('release_year')['duration_minutes'].mean().reset_in
movie_trend['type'] = 'Movie'

# --- Prep TV Shows Data ---
tv_df = df[df['type'] == 'TV Show'].copy()
tv_df['duration'] = tv_df['duration'].astype(str)
tv_df['num_seasons'] = tv_df['duration'].str.extract(r'(\d+)').astype(float)

tv_trend = tv_df.groupby('release_year')['num_seasons'].mean().reset_index()
tv_trend['type'] = 'TV Show'

# --- Combine both ---
duration_trend = pd.concat([
    movie_trend.rename(columns={'duration_minutes': 'avg_duration'}),
    tv_trend.rename(columns={'num_seasons': 'avg_duration'})
])

# --- Filter from 1940 onwards ---
duration_trend = duration_trend[duration_trend['release_year'] >= 1940]

# --- Plot ---
plt.figure(figsize=(14, 6))
sns.lineplot(data=duration_trend, x='release_year', y='avg_duration', hue='type', m
plt.title('Average Content Duration Over Years (Movies vs TV Shows)')
plt.xlabel('Release Year')
plt.ylabel('Average Duration (Minutes for Movies / Seasons for TV Shows)')
```

```
plt.grid(True)
plt.tight_layout()
plt.show()
```



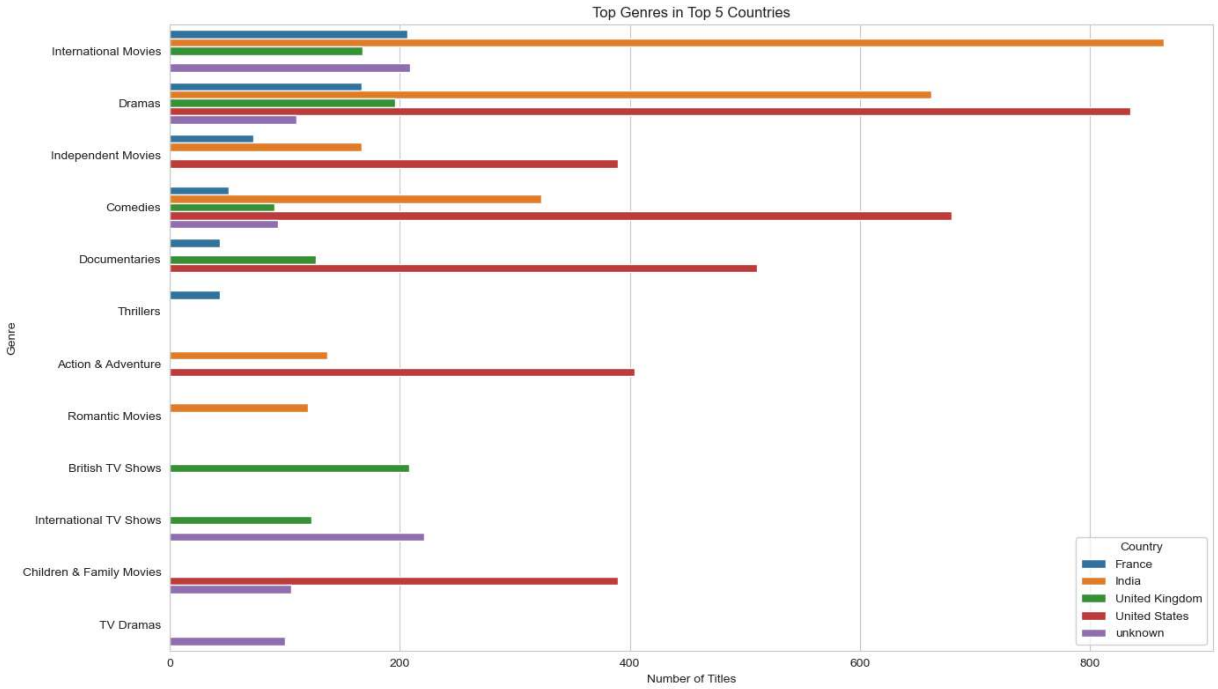
EDA: Country vs. Genre Distribution

Question: Which genres are most popular in different top countries?
Why: This highlights regional content preferences (e.g., India for Dramas, USA for Comedies), offering insights into Netflix's market focus and viewer tastes.

In [22]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Prepare country-genre exploded dataframe
df_genre_country = df.copy()
df_genre_country['country'] = df_genre_country['country'].astype(str).str.split(',
df_genre_country['genre'] = df_genre_country['listed_in'].astype(str).str.split(',
df_genre_country = df_genre_country.explode('country')
df_genre_country = df_genre_country.explode('genre')

# 2. Filter top 5 countries
top_5 = df_genre_country['country'].value_counts().head(5).index
df_top = df_genre_country[df_genre_country['country'].isin(top_5)]

# 3. Group by country and genre
genre_counts = df_top.groupby(['country', 'genre']).size().reset_index(name='count'

# 4. Optional: Limit to top 6 genres per country
top_genres_per_country = (
    genre_counts.groupby('country')
    .apply(lambda x: x.nlargest(6, 'count'))
    .reset_index(drop=True)
)

# 5. Plot
plt.figure(figsize=(14, 8))
sns.barplot(data=top_genres_per_country, y='genre', x='count', hue='country')
```

```
plt.title('Top Genres in Top 5 Countries')
plt.xlabel('Number of Titles')
plt.ylabel('Genre')
plt.legend(title='Country')
plt.tight_layout()
plt.show()
```

```
C:\Users\arnab\AppData\Local\Temp\ipykernel_11824\3908031646.py:22: DeprecationWarni
ng: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprec
ated, and in a future version of pandas the grouping columns will be excluded from t
he operation. Either pass `include_groups=False` to exclude the groupings or explici
tly select the grouping columns after groupby to silence this warning.
  .apply(lambda x: x.nlargest(6, 'count'))
```

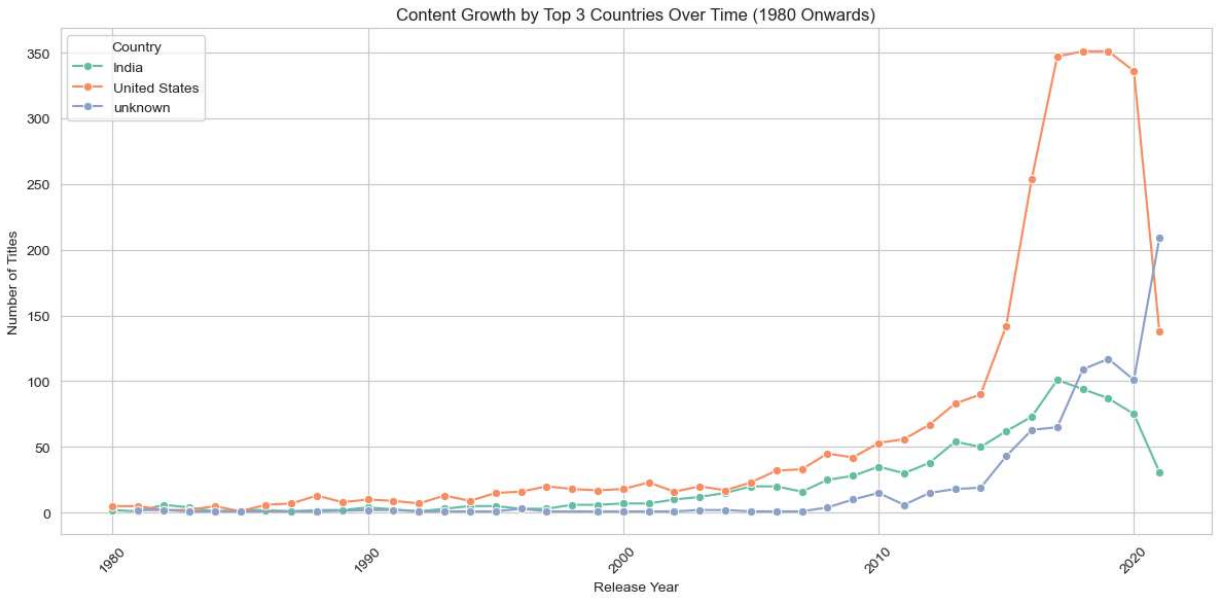


EDA: Content Growth by Country

Question: How has content from top countries grown over time?
Why: This shows if Netflix is expanding into new markets (e.g., India's rise), reflecting global strategy shifts.

In [23]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Keep the clean plot style
sns.set_style('whitegrid')

# Select top 3 countries
top_countries = df['country'].value_counts().index[:3]

# Filter data for those countries
df_top = df[df['country'].isin(top_countries)]
```

```
# Group by country and release year and count titles
count_by_year_country = df_top.groupby(['country', 'release_year']).size().reset_in

# Filter for release_year >= 1980
count_by_year_country = count_by_year_country[count_by_year_country['release_year']

# Plot with seaborn lineplot
plt.figure(figsize=(12, 6))
sns.lineplot(data=count_by_year_country, x='release_year', y='count', hue='country'
plt.title('Content Growth by Top 3 Countries Over Time (1980 Onwards)')
plt.xlabel('Release Year')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.legend(title='Country')
plt.tight_layout()
plt.show()

# Print total counts per country (from original data, no filter)
print("Total Titles by Top 3 Countries:")
print(df_top['country'].value_counts())
```

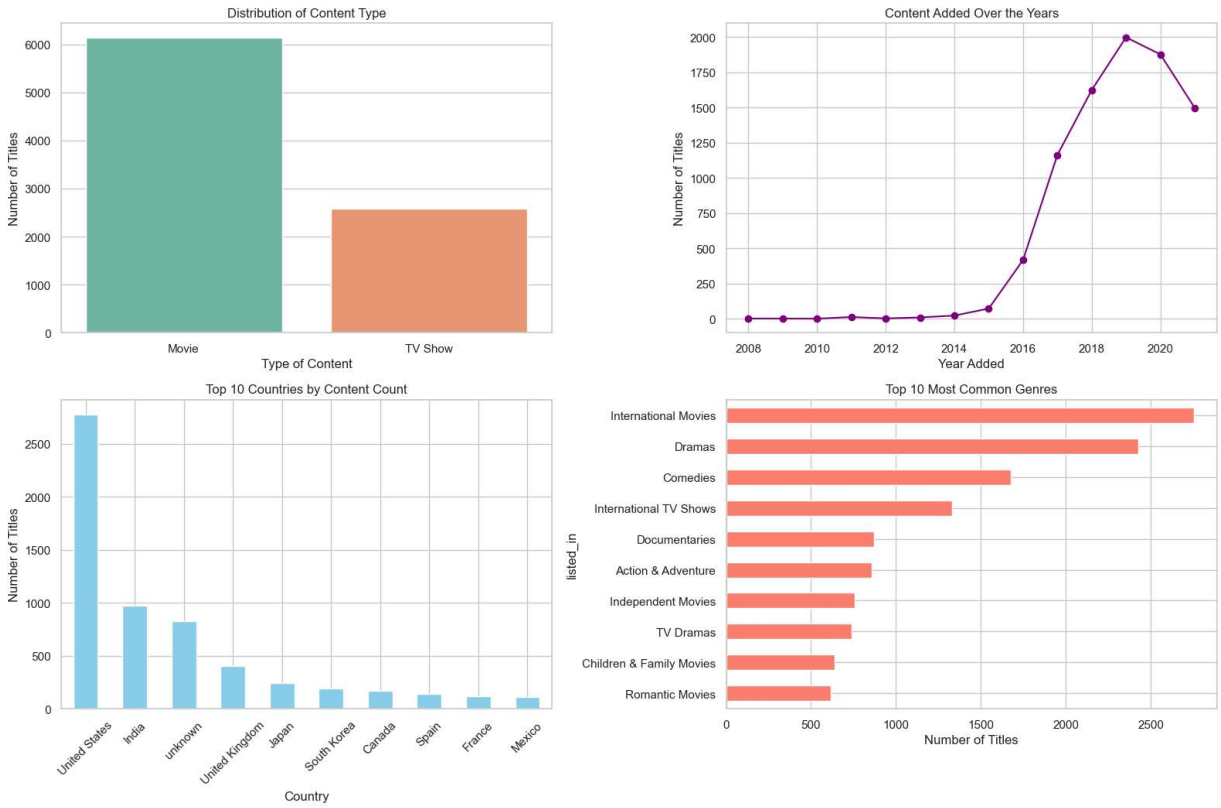



```
Total Titles by Top 3 Countries:
country
United States    2778
India             971
unknown           827
Name: count, dtype: int64
```

Netflix EDA Dashboard

In [24]:
```
sns.set(style="whitegrid")

fig, axs = plt.subplots(2, 2, figsize=(16, 12))
fig.suptitle("Netflix EDA Dashboard", fontsize=18)

# ----------------------------------------
# 1. Content Type Distribution
```

```
sns.countplot(data=df, x='type', hue='type', ax=axs[0, 0], palette='Set2', legend=F
axs[0, 0].set_title("Distribution of Content Type")
axs[0, 0].set_xlabel("Type of Content")
axs[0, 0].set_ylabel("Number of Titles")

# ----------------------------------------
# 2. Titles Added by Year
df['year_added'] = df['date_added'].dt.year
df['year_added'].value_counts().sort_index().plot(
    kind='line', marker='o', ax=axs[0, 1], color='purple')
axs[0, 1].set_title("Content Added Over the Years")
axs[0, 1].set_xlabel("Year Added")
axs[0, 1].set_ylabel("Number of Titles")
axs[0, 1].grid(True)

# ----------------------------------------
# 3. Top 10 Countries Producing Content
df['country'].value_counts().head(10).plot(
    kind='bar', ax=axs[1, 0], color='skyblue')
axs[1, 0].set_title("Top 10 Countries by Content Count")
axs[1, 0].set_xlabel("Country")
axs[1, 0].set_ylabel("Number of Titles")
axs[1, 0].tick_params(axis='x', rotation=45)

# ----------------------------------------
# 4. Top 10 Genres (listed_in)
top_genres = df['listed_in'].str.split(', ').explode().value_counts().head(10)
top_genres.plot(kind='barh', ax=axs[1, 1], color='salmon')
axs[1, 1].set_title("Top 10 Most Common Genres")
axs[1, 1].set_xlabel("Number of Titles")
axs[1, 1].invert_yaxis()  # optional: show most popular genre at top

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```



In [28]:
```
# Clean duration column
df['duration'] = df['duration'].astype(str).str.extract(r'(\d+)').astype(float)
df.loc[df['type'] == 'TV Show', 'duration'] = df.loc[df['type'] == 'TV Show', 'dura

# Create genre column by splitting and exploding listed_in
df = df.assign(genre=df['listed_in'].str.split(', ')).explode('genre')

# Export cleaned DataFrame to CSV
df.to_csv('netflix_cleaned.csv', index=False)
print("Cleaned data exported to 'netflix_cleaned.csv'!")
```

```
Cleaned data exported to 'netflix_cleaned.csv'!
```

In [ ]: