In [1]:

```
#NETFLIX DATA ANALYSIS
#NAME-DINESH KUMAR YADAV
#ROLL_NO-1240259022
#SUBMITTED TO:MR ANKIT VERMA

#STEP-1
# Import libraries import
pandas as pd import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings("ignore")

# Set style for seaborn plots sns.set(style="whitegrid")
```

In [2]:

```
#STEP-2
# Load Netflix dataset
data = pd.read_csv("../data/netflix_cleaned.csv")

# first 5 rows to check dataset data.head()
```

Out[2]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
			Dick									As her father nears

0	s1	Movie	Johnson Is	Kirsten NaN	2020	PG-13	United 90 min	September Documentaries	the end of Johnson	States	25, 2021 Dead	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 [3]:  # Insight: Dataset loaded successfully. We can see columns like type, title, director, cast, #
In [4]:  country, release_year, rating, duration, listed_in (genre), and description.
         # This will allow us to analyze content strategy, audience, and talent.
```

```
#STEP-3
#NOW WE CHECK BASIC INFORMATION data.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
```

```
In [5]:  #STEP-4
         #then we will go for missing values data.isnull().sum()
```

Out[5]:

```
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 [6]:

```
# Insight:
          # - Missing values in 'director' or 'cast' indicate unknown talent.
            # - 'country' ,'duration','rating','date_added'and 'rating' also have nulls.
# - These need handling before some analyses.
```

In [7]:

```
#STEP-5
# HANDLE ALL MISSING VALUES
# Fill text columns with 'Unknown'
text_columns = ['director', 'cast', 'country', 'rating']
for col in text_columns:     data[col].fillna('Unknown',
inplace=True)

# Fill date_added with most frequent date
data['date_added'].fillna(data['date_added'].mode()[0], inplace=True)

# Fill duration with 0
data['duration'].fillna(0, inplace=True)

# Verify missing values
print("Missing values after handling:\n", data.isnull().sum())
```

```
Missing values after handling:
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
```

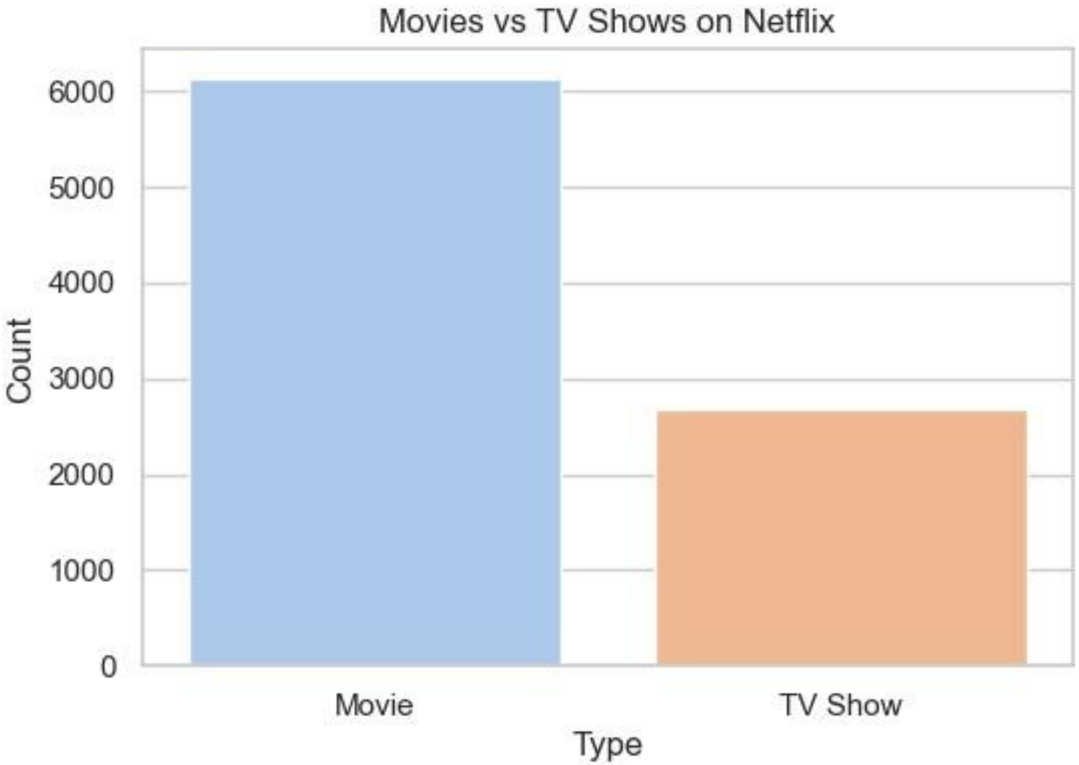
```
0 description     0 dtype:
int64
```

In [8]:

```
# Insight:
# - All missing values are handled.
# - Text columns now have 'Unknown' for missing entries.
# - Missing dates are filled with the most frequent date.
# - Missing durations are set to 0.
# - This ensures all analysis runs without errors.
```

In [9]:

```
#BUSINESS-ORIENTED ANALYSIS QUESTIONS
# CONTENT STRATEGY
#1-MOVIES VS TV SHOWS plt.figure(figsize=(6,4))
sns.countplot(x='type', data=data, palette='pastel')
plt.title('Movies vs TV Shows on Netflix')
plt.xlabel('Type') plt.ylabel('Count') plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



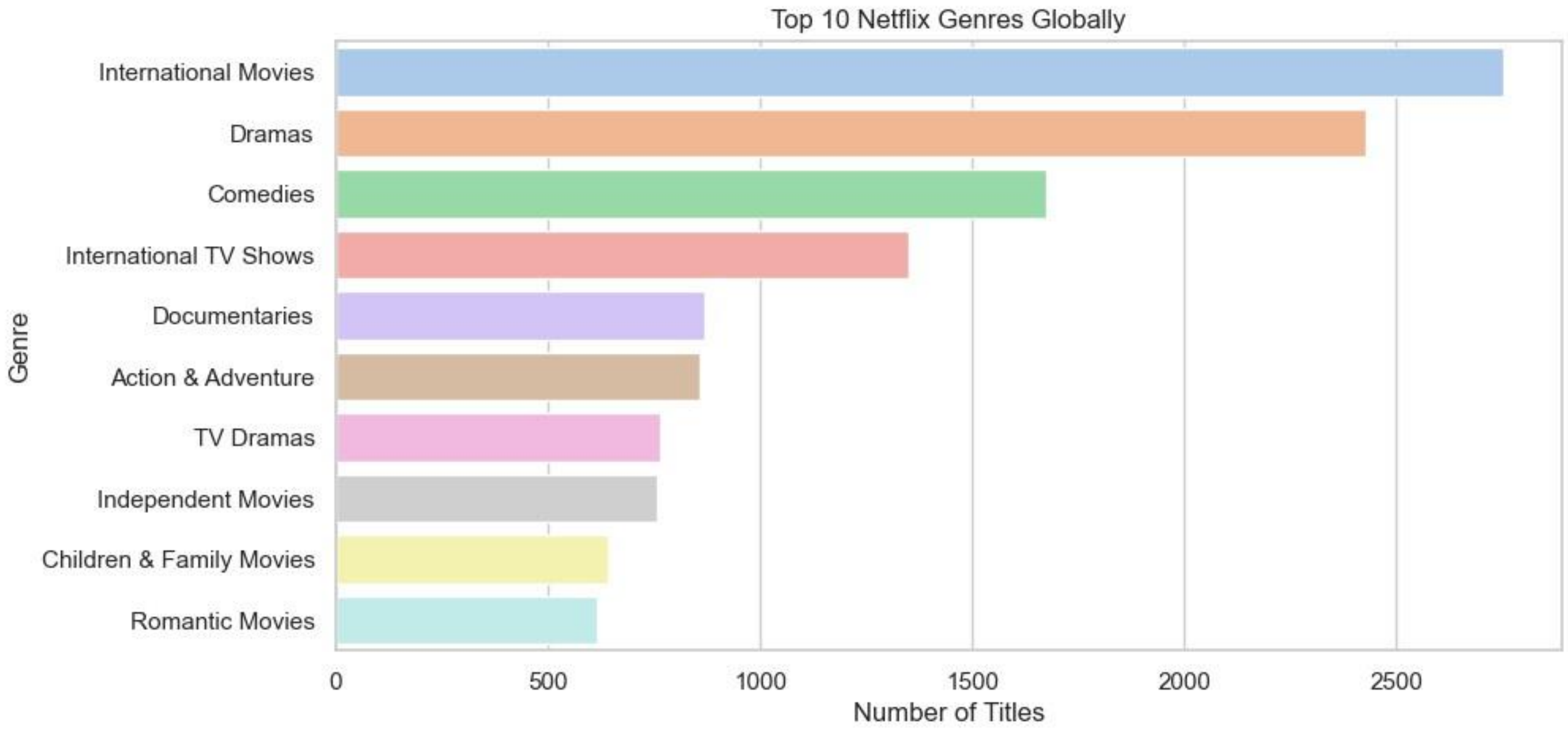
```
<Figure size 640x480 with 0 Axes> In
```

[10]:

```
#insihts:
#- The chart shows the distribution of Movies and TV Shows on Netflix.
#- Observation: Netflix has historically produced more Movies than TV Shows.
#- Significance: Indicates a past focus on short-form content, but TV Shows are growing.
#- Recommendation: Invest more in limited-series TV Shows to increase viewer engagement over longer periods and retain subscri
```

In [11]:

```
#2- MOST POPULAR GENRES
# Split genres (comma-separated) and explode into separate rows all_genres
= data['listed_in'].str.split(',').explode().str.strip()

# Count top 10 genres
top_genres = all_genres.value_counts().head(10)

# Plot top 10 genres plt.figure(figsize=(10,5))


sns.barplot(x=top_genres.values, y=top_genres.index, palette='pastel')
plt.title('Top 10 Netflix Genres Globally') plt.xlabel('Number of Titles')
plt.ylabel('Genre') plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



```
<Figure size 640x480 with 0 Axes> In
```

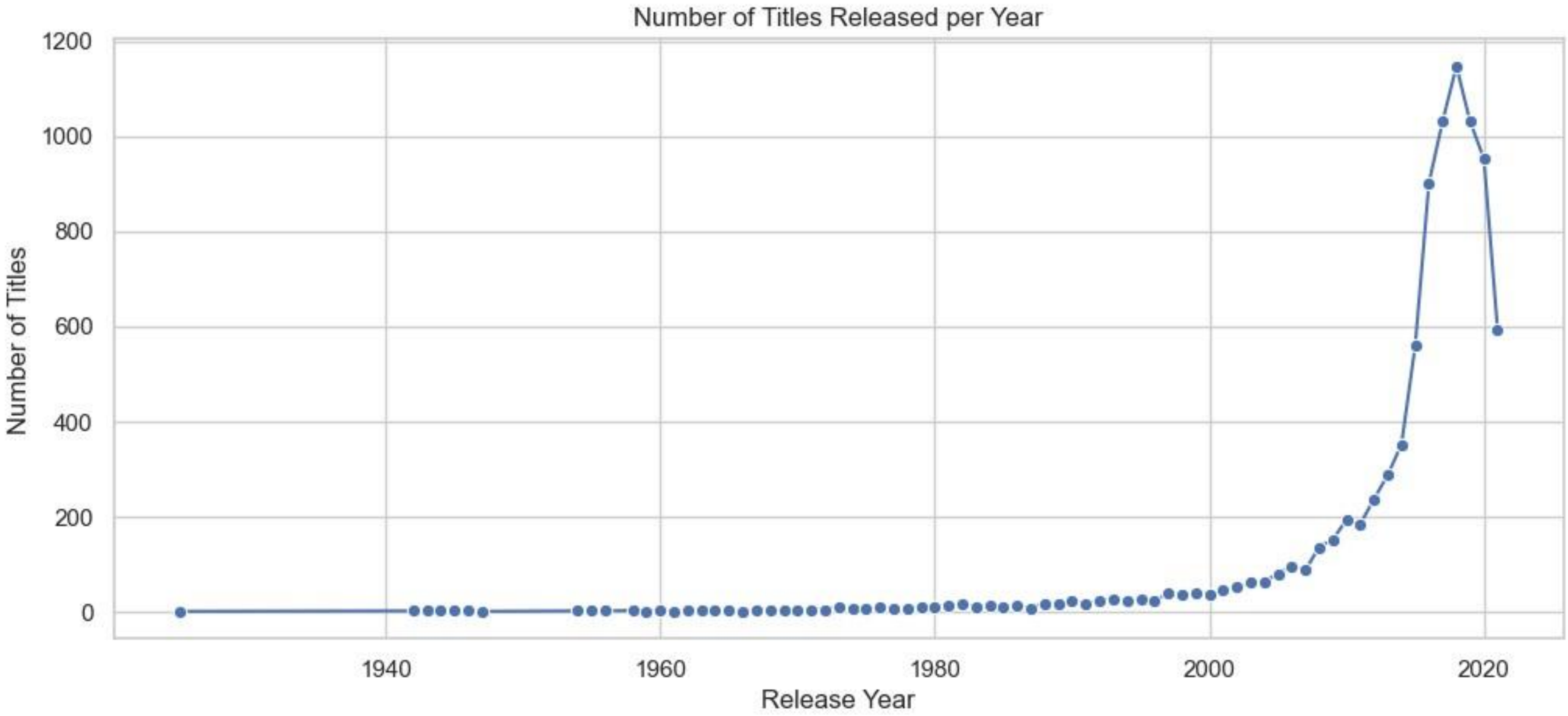
[12]:

```
# Insight:
#- The top 10 genres indicate the most frequently produced content on Netflix.
#- Observation: Genres like Drama, Comedy, and Documentary dominate globally.
```

```
#-
Significance: These genres attract a wide audience across multiple demographics.
    #- Recommendation: Prioritize acquisition or production in these popular genres, while exploring niche genres for targeted cam
```

In [13]:

```
#3- CONTENT RELEASE OVER YEARS
year_count = data['release_year'].value_counts().sort_index()

plt.figure(figsize=(12,5)) sns.lineplot(x=year_count.index,
y=year_count.values, marker='o') plt.title('Number of Titles
Released per Year') plt.xlabel('Release Year') plt.ylabel('Number
of Titles') plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



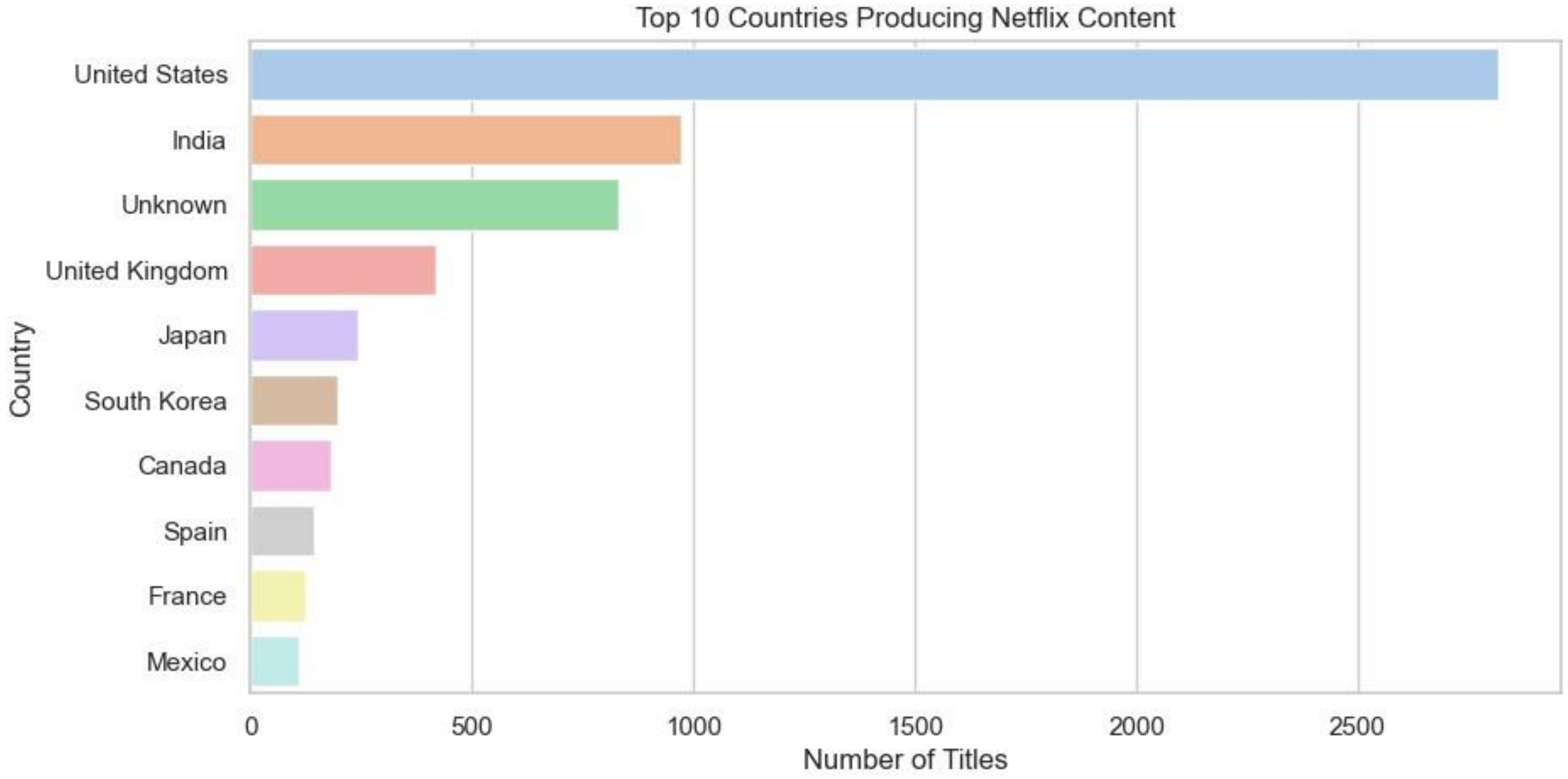
```
<Figure size 640x480 with 0 Axes> In
```

[14]:

```
#insights #- Shows how many titles were released
each year.
#- Observation: Certain years saw a significant spike in content production.
#- Significance: Indicates periods of aggressive content acquisition by Netflix.
#- Recommendation: Analyze trends to forecast content investment and decide on production schedules.
```

In [15]:

```
# 4-COUNTRIES PRODUCING MOST CONTENT
top_countries = data['country'].value_counts().head(10)

plt.figure(figsize=(10,5)) sns.barplot(x=top_countries.values,
y=top_countries.index, palette='pastel') plt.title('Top 10 Countries Producing
Netflix Content') plt.xlabel('Number of Titles')


plt.ylabel('Country')
plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



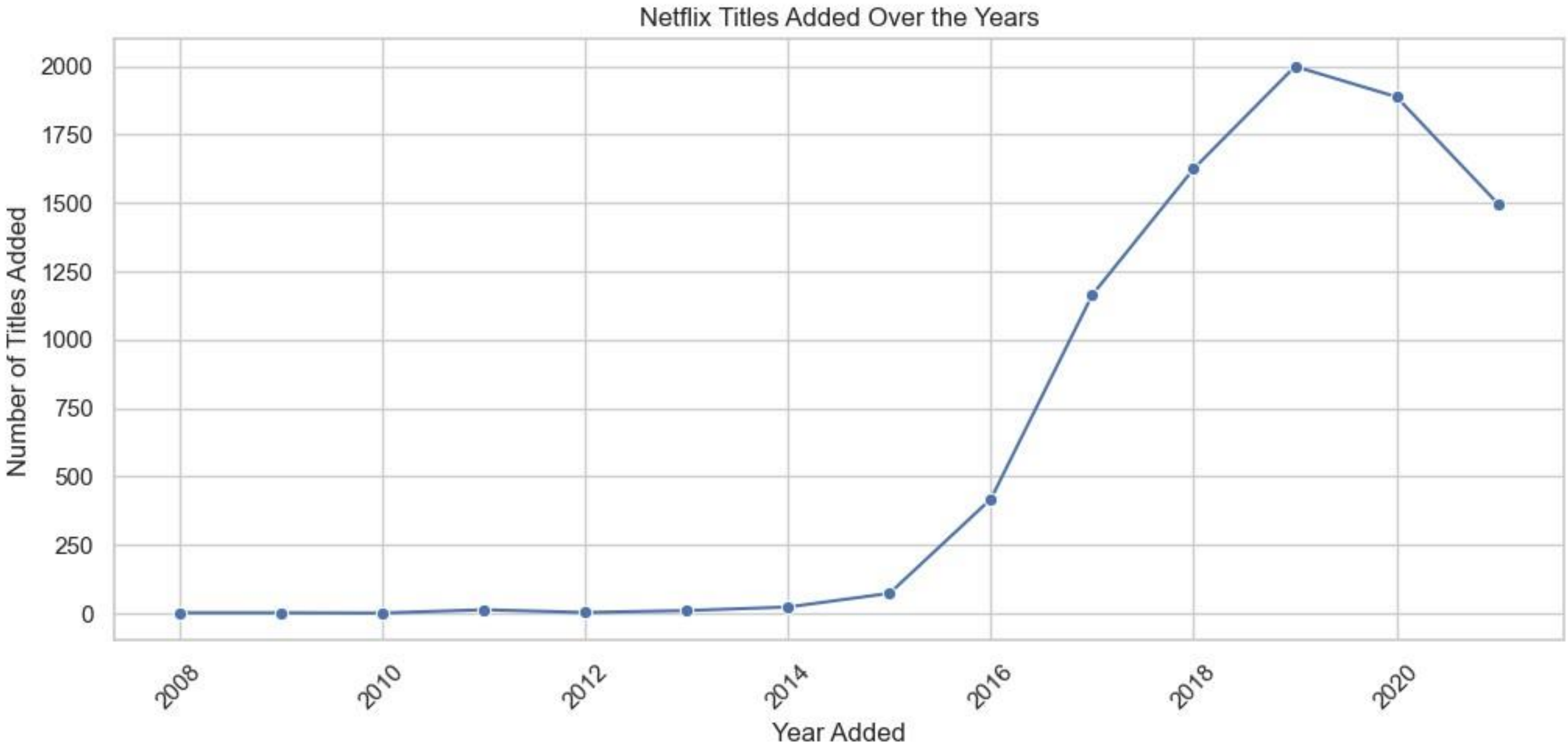
```
<Figure size 640x480 with 0 Axes> In
```

[16]:

In [17]:

```
# Insight:
#- Highlights countries contributing most Netflix content.
#- Observation: USA leads, followed by India, UK, etc. #-
Significance: Shows key markets for content production.
#- Recommendation: Strengthen partnerships in high-producing countries; explore emerging markets for local content to expand g
```

```
# 5-TREND OF CONTENT ADDED TO NETFLIX
# Convert date_added to datetime
data['date_added'] = pd.to_datetime(data['date_added'], errors='coerce')

# Drop rows with missing date_added
data_cleaned = data.dropna(subset=['date_added']).copy()

# Extract year and month added
data_cleaned['year_added'] = data_cleaned['date_added'].dt.year

# Count number of titles added per year
trend_added = data_cleaned['year_added'].value_counts().sort_index()

plt.figure(figsize=(12,5)) sns.lineplot(x=trend_added.index,
y=trend_added.values, marker='o') plt.title('Netflix Titles Added
Over the Years') plt.xlabel('Year Added') plt.ylabel('Number of
Titles Added') plt.xticks(rotation=45) plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



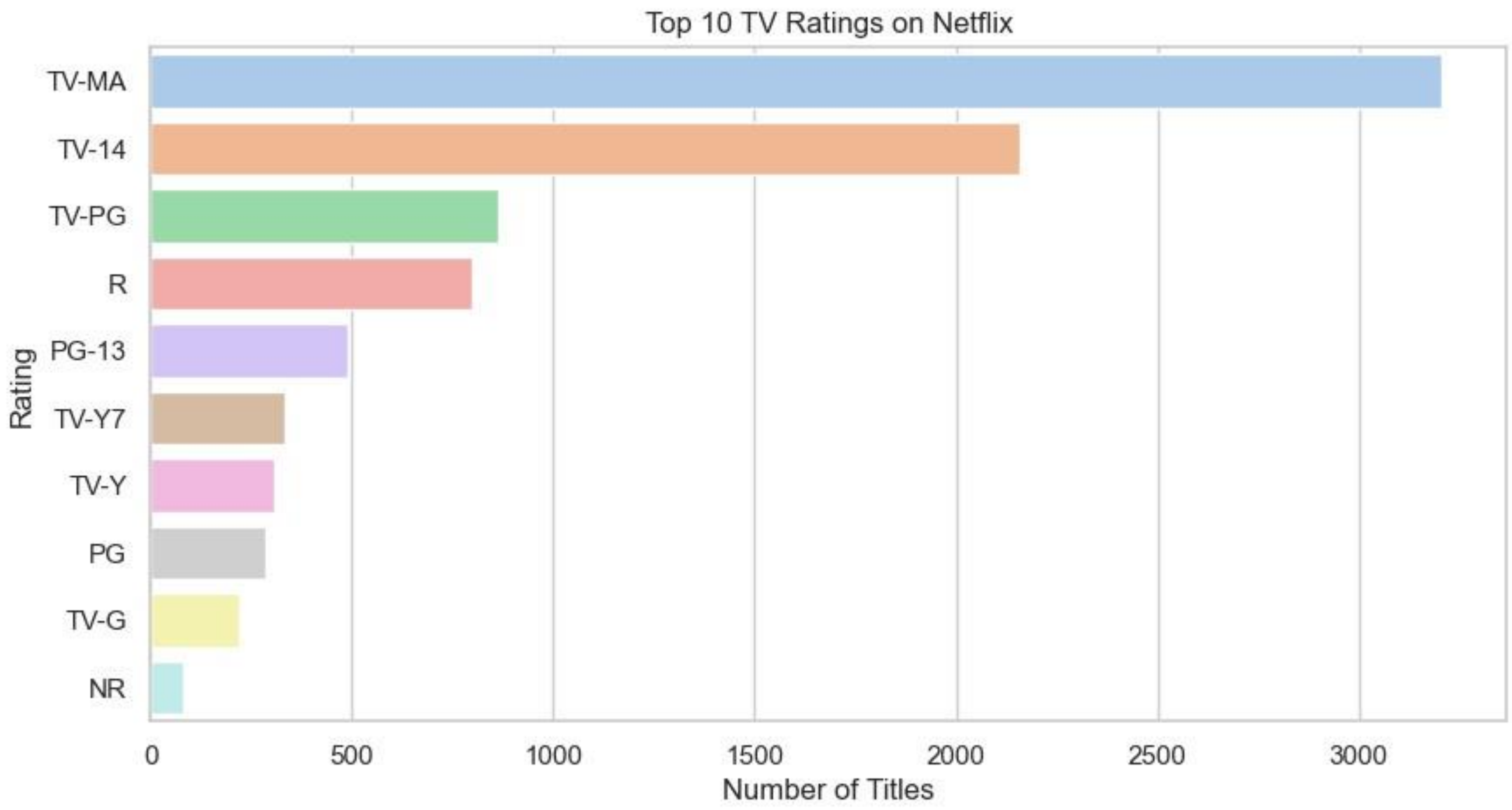
```
<Figure size 640x480 with 0 Axes>
```

In [18]:

```
#insights:
#- Tracks the growth of Netflix content library.
#- Observation: Major spikes in 2016-2019 indicate strategic expansion periods.
#- Significance: Understanding addition trends helps predict future content growth.
#- Recommendation: Plan marketing campaigns around high-addition periods; consider seasonality for new releases.
```

In [19]:

```
#USER DEMOGRAPHICS AND TARGETING
#6- MOST FREQUENT RATINGS
rating_count = data['rating'].value_counts().head(10)

plt.figure(figsize=(10,5)) sns.barplot(x=rating_count.values,
y=rating_count.index, palette='pastel')

plt.title('Top 10 TV Ratings on Netflix')
plt.xlabel('Number of Titles')
plt.ylabel('Rating') plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



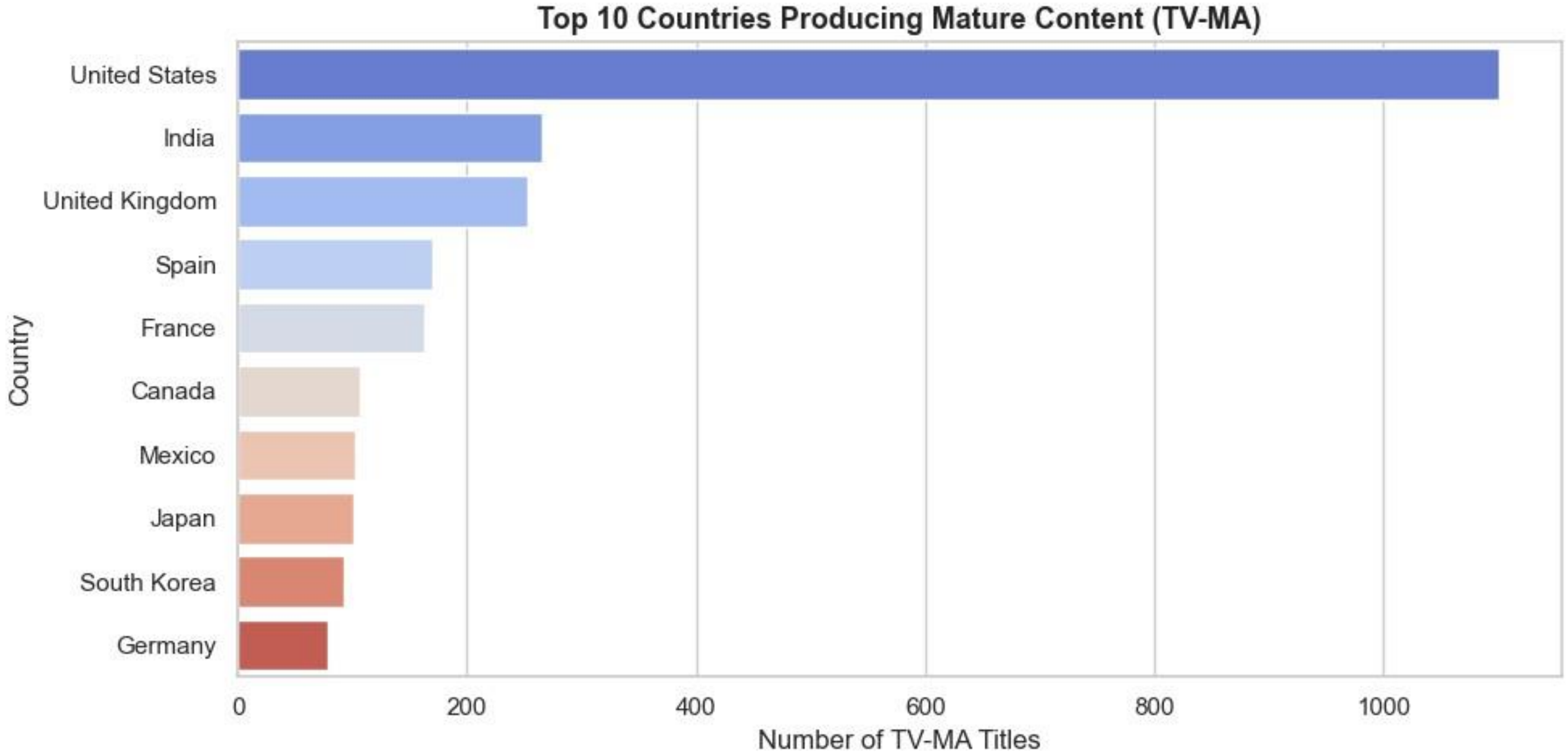
```
<Figure size 640x480 with 0 Axes>
```

In [20]:

```
# Insight:
# - Shows which audience ratings are most frequent.
#- Observation: TV-MA, TV-14, and PG are the most common.
#- Significance: Helps align content with target age groups. #- Recommendation: Focus on producing more content for popular
rating categories while diversifying for younger audiences.
```

In [21]:

```
# Q7: Do some countries tend to produce more mature content (TV-MA)?
# Purpose: Understand which countries create more adult-oriented content.
data['rating'] = data['rating'].astype(str) data['country']
= data['country'].astype(str)
tvma_data = data[data['rating'].str.upper().str.contains('TV-MA', na=False)].copy()

tvma_data = tvma_data.assign(country_split=tvma_data['country'].str.split(',')) tvma_data =
tvma_data.explode('country_split')   # Split multiple countries into separate rows
tvma_data['country_split'] = tvma_data['country_split'].str.strip()   # Remove extra spaces
tvma_data = tvma_data[~tvma_data['country_split'].isin(['', 'Unknown', 'nan', 'None'])]

top_tvma_countries = tvma_data['country_split'].value_counts().head(10)
plt.figure(figsize=(10, 5)) sns.barplot(
x=top_tvma_countries.values,     y=top_tvma_countries.index,
palette='coolwarm'
)
plt.title('Top 10 Countries Producing Mature Content (TV-MA)', fontsize=13, weight='bold')
plt.xlabel('Number of TV-MA Titles') plt.ylabel('Country') plt.tight_layout()
plt.savefig('../output/tvma_by_country.png', dpi=150, bbox_inches='tight') plt.show()
display(top_tvma_countries) # Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



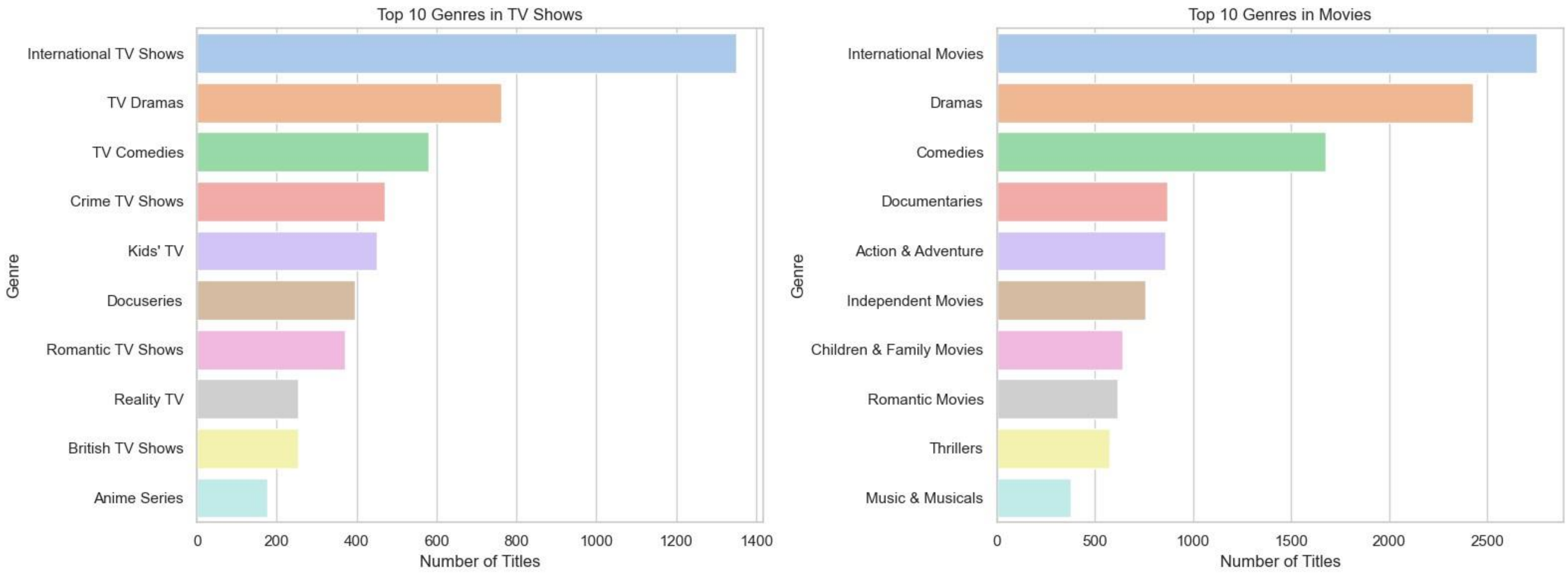
```
country_split    United
States     1101
India             266
United Kingdom    253
Spain             170
France            163
Canada            107
Mexico            102
Japan             101
South Korea        92
Germany            79
Name: count, dtype: int64
```

```
#Insight:
#USA: Focuses heavily on Dramas, Comedies, and Documentaries.
#Other countries: Lean more toward International Dramas, Romantic titles, and Action/Adventure.
#Recommendation: Netflix should regionalize its content strategy — e.g., promote romantic K-dramas in Asia, thrillers in Europ
```

```
#8-genres more associated with TV Shows vs Movies
# Split and count genres for TV Shows
tv_genres = data[data['type'] == 'TV Show']['listed_in'].str.split(',').explode().str.strip().value_counts().head(10)

# Split and count genres for Movies
movie_genres = data[data['type'] == 'Movie']['listed_in'].str.split(',').explode().str.strip().value_counts().head(10)

# Plot side by side
fig, axes = plt.subplots(1, 2, figsize=(16,6)) sns.barplot(x=tv_genres.values,
y=tv_genres.index, ax=axes[0], palette='pastel') axes[0].set_title('Top 10 Genres
in TV Shows') axes[0].set_xlabel('Number of Titles') axes[0].set_ylabel('Genre')

sns.barplot(x=movie_genres.values, y=movie_genres.index, ax=axes[1], palette='pastel')
axes[1].set_title('Top 10 Genres in Movies') axes[1].set_xlabel('Number of Titles')
axes[1].set_ylabel('Genre')

plt.tight_layout()
plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```

```
<Figure size 640x480 with 0 Axes> In
```

```
[22]: In [23]:
```



Top 10 Genres in TV Shows

Top 10 Genres in Movies

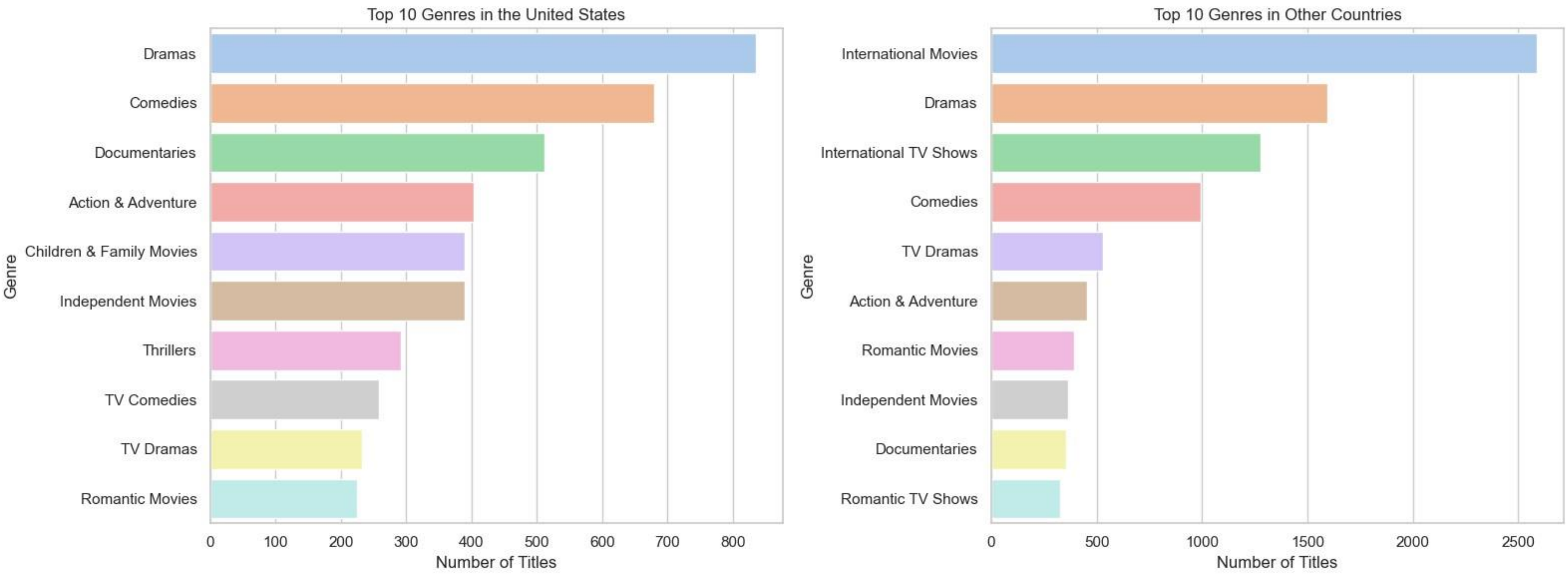
```
<Figure size 640x480 with 0 Axes> In
```

[24]:

```
#Insight:
#TV Shows: Dominated by Drama, Reality, and Documentary formats — appealing for binge-watching.
#Movies: Led by Drama, Comedy, and Action & Adventure — fitting shorter viewing patterns.
#Recommendation: Marketing for TV Shows should highlight emotional depth and continuity, while Movies should focus on quick en
```

In [25]:

```
#9-Which genres dominate the U.S. vs other countries
# Split genres by country
us_data = data[data['country'].str.contains('United States', na=False)] other_data
= data[~data['country'].str.contains('United States', na=False)]

# Most common genres
us_genres = us_data['listed_in'].str.split(',').explode().str.strip().value_counts().head(10) other_genres
= other_data['listed_in'].str.split(',').explode().str.strip().value_counts().head(10)

# Plot
fig, axes = plt.subplots(1, 2, figsize=(16,6)) sns.barplot(x=us_genres.values,
y=us_genres.index, ax=axes[0], palette='pastel') axes[0].set_title('Top 10 Genres
in the United States')

axes[0].set_xlabel('Number of Titles') axes[0].set_ylabel('Genre')

sns.barplot(x=other_genres.values, y=other_genres.index, ax=axes[1], palette='pastel')
axes[1].set_title('Top 10 Genres in Other Countries') axes[1].set_xlabel('Number of
Titles') axes[1].set_ylabel('Genre')

plt.tight_layout()
plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



```
<Figure size 640x480 with 0 Axes>
```

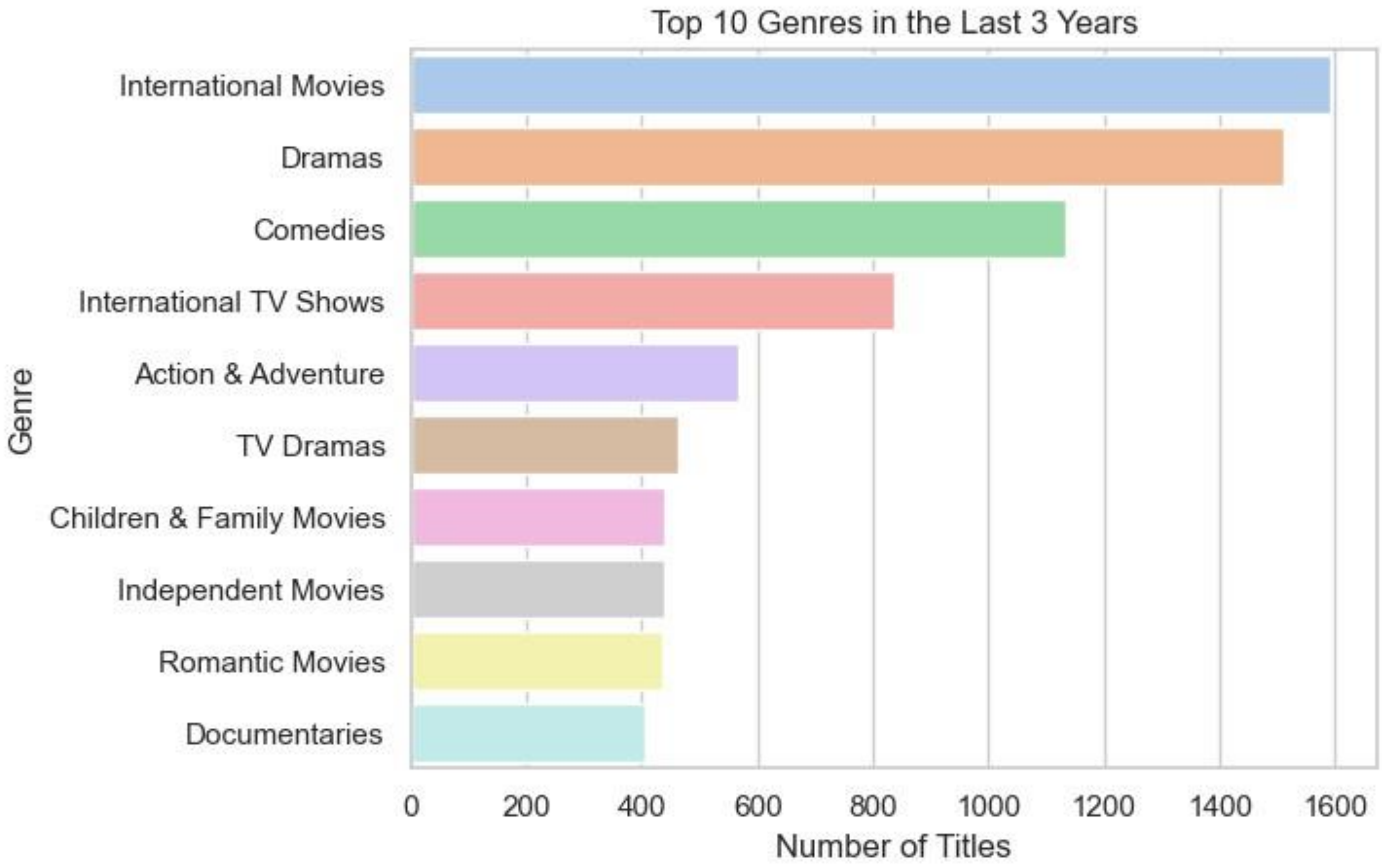
In [26]:

```
#Insight:
#USA:Focuses heavily on Dramas, Comedies, and Documentaries.
#Other countries: Lean more toward International Dramas, Romantic titles, and Action/Adventure.
#Recommendation: Netflix should regionalize its content strategy — e.g., promote romantic K-dramas in Asia, thrillers in Europ
```

```
  Cell In[26], line 1
💡 #Insight:
    ^
SyntaxError: invalid character '💡' (U+1F4A1)
```

In [27]:

```
#10-What genres are most popular in the last 3 years #
Filter data for last 3 years (based on date_added)
latest_year = data_cleaned['year_added'].max()
recent_data = data_cleaned[data_cleaned['year_added'] >= latest_year - 2]

# Extract and count genres
recent_genres = recent_data['listed_in'].str.split(',').explode().str.strip().value_counts().head(10)

# Plot
sns.barplot(x=recent_genres.values, y=recent_genres.index, palette='pastel')
plt.title('Top 10 Genres in the Last 3 Years') plt.xlabel('Number of Titles')
plt.ylabel('Genre') plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



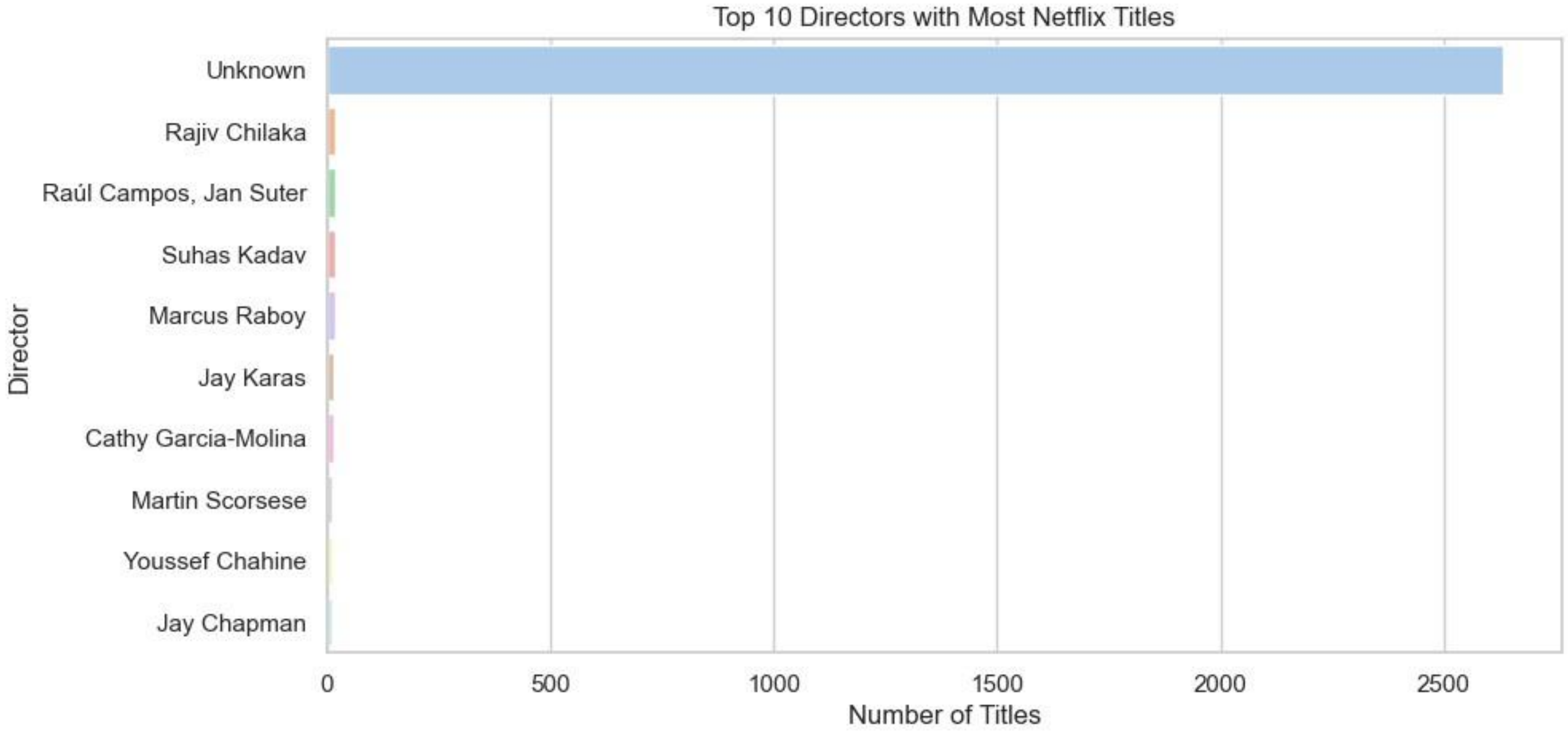
```
<Figure size 640x480 with 0 Axes>
```

In [ ]:

```
#Insight: #Recent trends: Drama, International TV, and Documentaries
have surged.
#Shift: Indicates that audiences now prefer realistic stories and global storytelling over traditional Hollywood genres.
#Recommendation: Continue investing in localized dramas and docu-series, particularly in markets like India, Korea, and Spain.
```

In [28]:

```
#TALENT ACQUISITION AND PARTNERSHIP QUESTIONS
#11- TOP 10 DIRECTORS
top_directors = data['director'].value_counts().head(10)

plt.figure(figsize=(10,5)) sns.barplot(x=top_directors.values,
y=top_directors.index, palette='pastel')

plt.title('Top 10 Directors with Most Netflix Titles')
plt.xlabel('Number of Titles') plt.ylabel('Director')
plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```

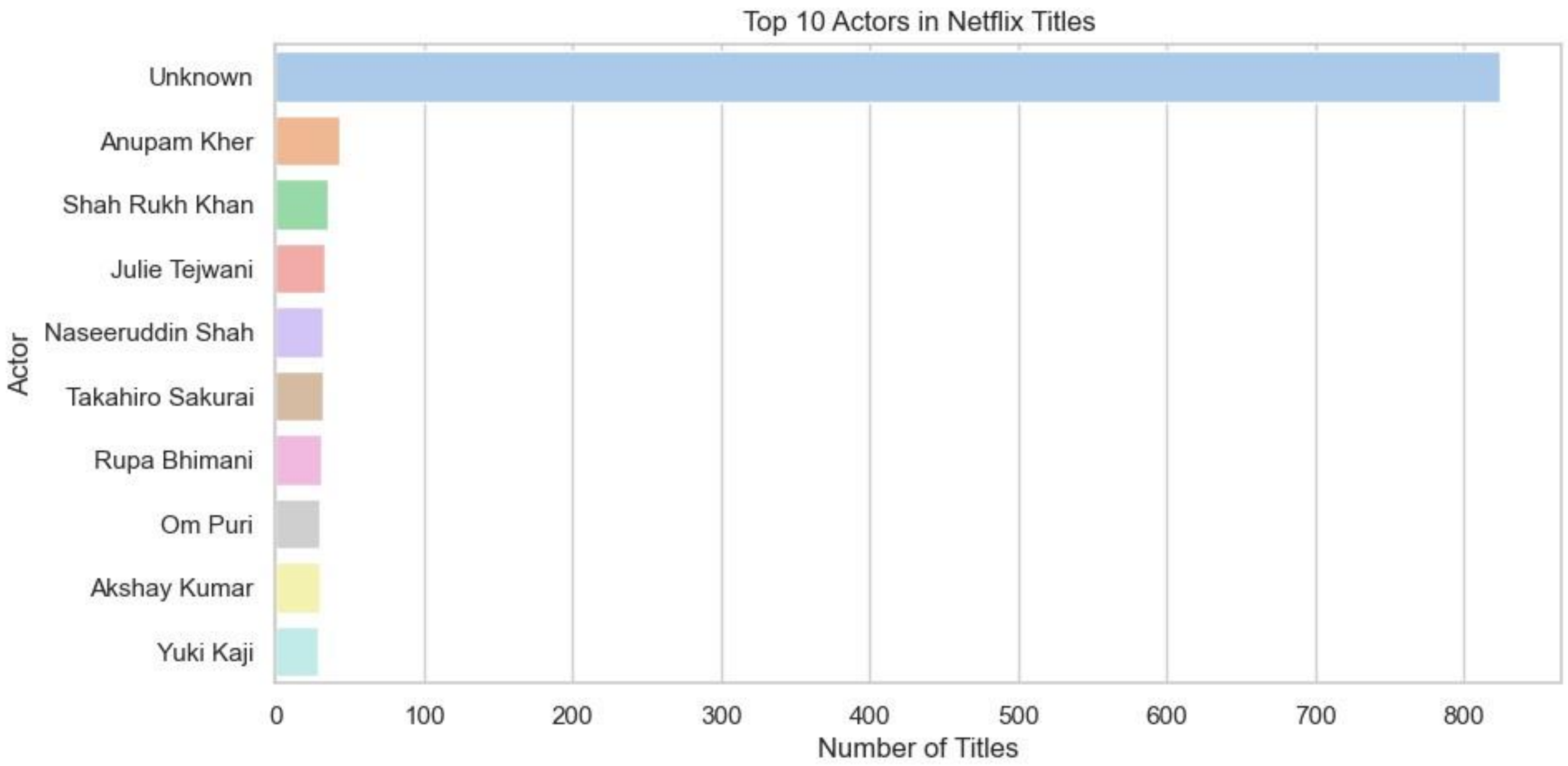


```
# Insight:
#- Identifies directors with the most Netflix content.
#- Observation: Certain directors dominate content production. #-
Significance: Indicates reliable creative talent.
#- Recommendation: Strengthen collaboration with prolific directors; explore emerging directors for new creative directions.
```

```
#12- TOP 10 ACTORS
all_actors = data['cast'].str.split(',').explode().str.strip() top_actors
= all_actors.value_counts().head(10)

plt.figure(figsize=(10,5)) sns.barplot(x=top_actors.values,
y=top_actors.index, palette='pastel') plt.title('Top 10 Actors in
Netflix Titles') plt.xlabel('Number of Titles') plt.ylabel('Actor')
plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```

```
<Figure size 640x480 with 0 Axes> In
```

```
[ ]: In [29]:
```



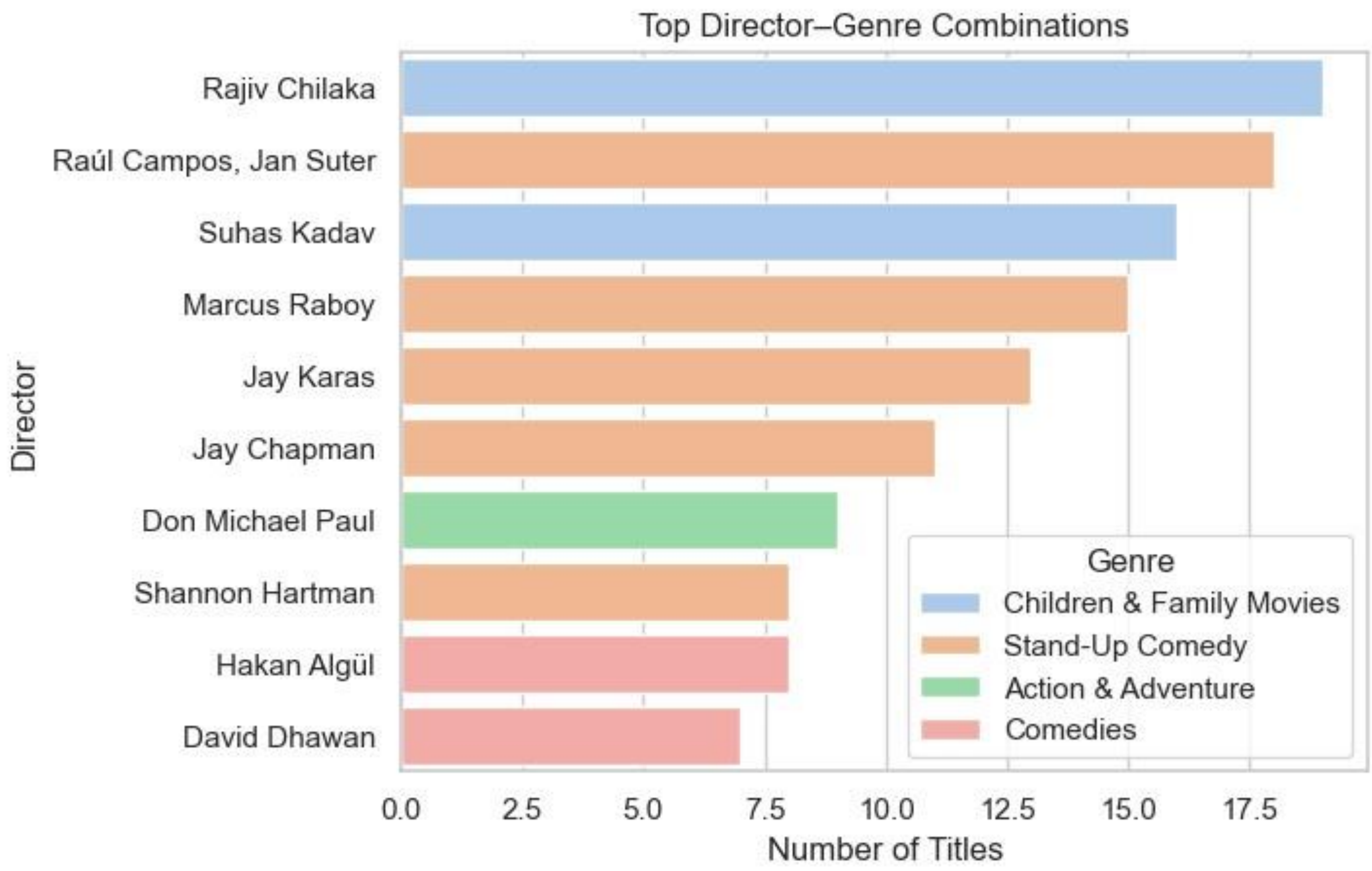
```
<Figure size 640x480 with 0 Axes>
```

In [ ]:

```
# Insight:
#- Highlights actors appearing most frequently.
#- Observation: Popular actors appear in multiple shows and movies.
#- Significance: Casting these actors may attract loyal viewers.
#- Recommendation: Consider recurring collaborations with popular actors to boost engagement.
```

In [30]:

```
#13- MOST FREQUENT DIRECTOR-GENRE PAIRS
# Explode genres director_genre = (
data[data['director'] != 'Unknown']
    .assign(genre=data['listed_in'].str.split(',').str[0])


    .groupby(['director', 'genre'])
    .size()
    .reset_index(name='count')
    .sort_values('count', ascending=False)
    .head(10)
)

sns.barplot(x='count', y='director', hue='genre', data=director_genre, palette='pastel')
plt.title('Top Director-Genre Combinations') plt.xlabel('Number of Titles')
plt.ylabel('Director') plt.legend(title='Genre') plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



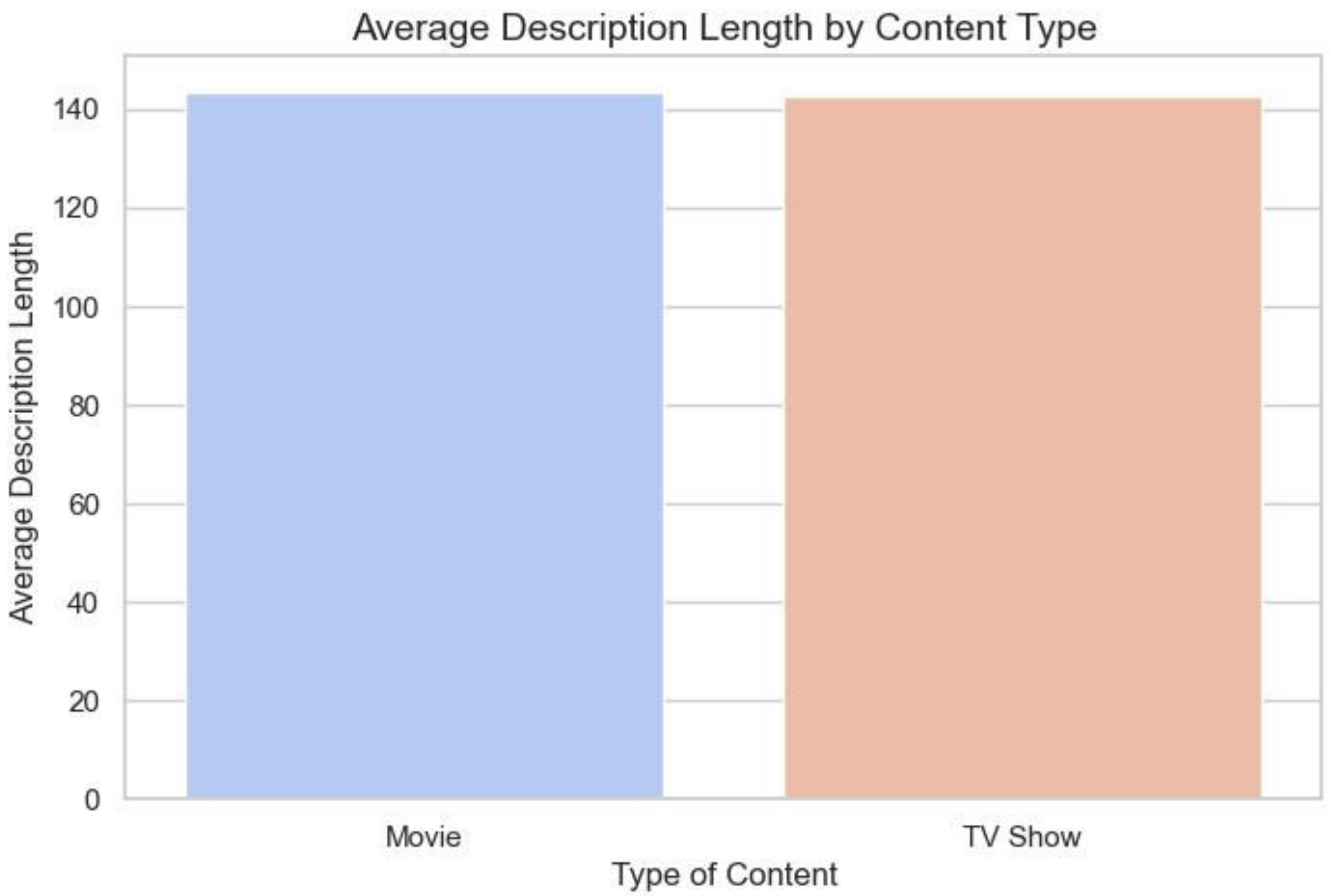
```
<Figure size 640x480 with 0 Axes> In
```

[ ]:

```
# Insight:
#- Shows which directors frequently work in which genres.
#- Observation: Some directors consistently produce content in Drama, Comedy, or Documentary genres.
#- Significance: Indicates creative trends and successful director-genre combinations.
#- Recommendation: Use this data to identify reliable talent for specific genre productions and partnerships.
```

In [31]:

```
#14-Which types of shows (Movies/TV Shows) have longer descriptions
# Calculate average description length by type
data['description_length'] = data['description'].astype(str).apply(len) desc_length
= data.groupby('type')['description_length'].mean().reset_index()



# Plot
plt.figure(figsize=(8, 5)) sns.barplot(data=desc_length, x='type',
y='description_length', palette='coolwarm') plt.title("Average Description Length by
Content Type", fontsize=14) plt.xlabel("Type of Content") plt.ylabel("Average
Description Length") plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



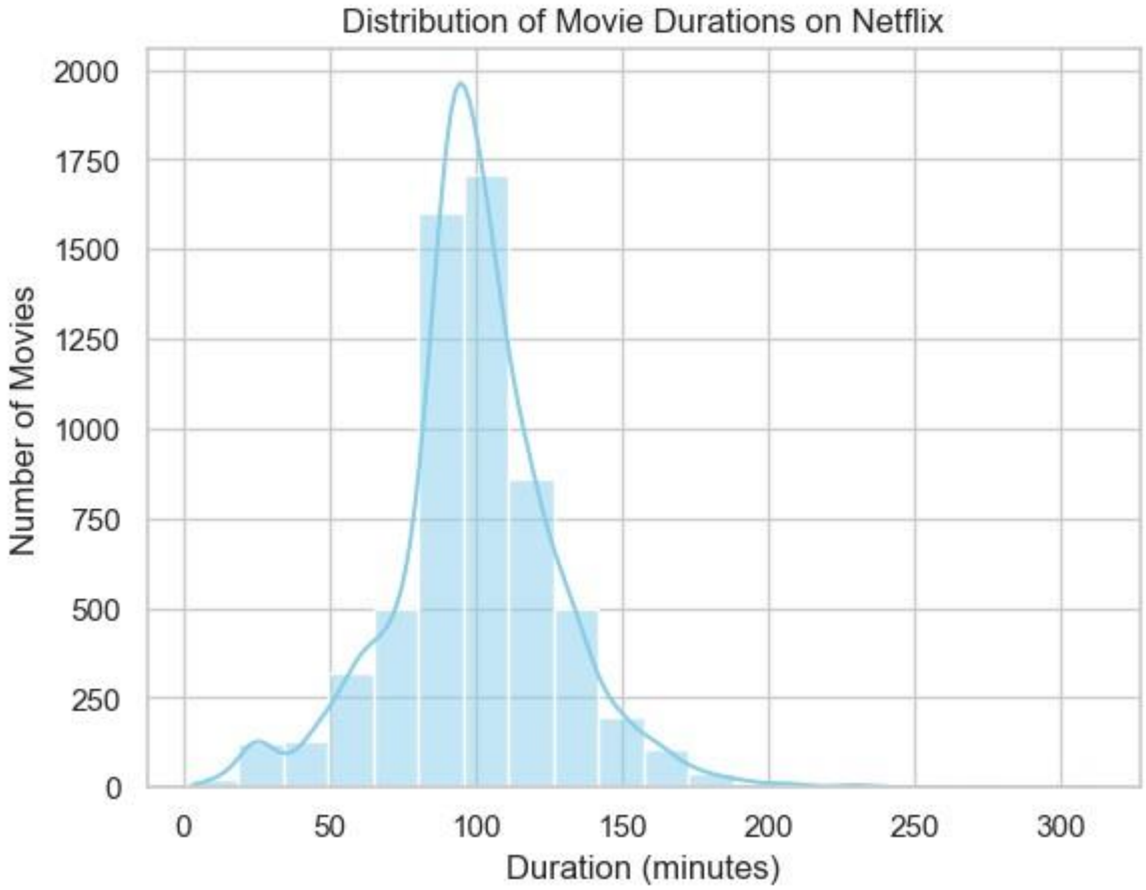
```
#insights:
#TV Shows usually have longer descriptions on average.
#This happens because they often describe multiple characters, storylines, and seasons.
#Movies tend to have shorter and more concise descriptions since they focus on a single story arc.
#Insight helps metadata writers and marketers understand how much information users expect when browsing each content type.
```

```
<Figure size 640x480 with 0 Axes> In
[ ]:
```

In [32]:

```
#DURATION AND ENGAGEMENT
# 15-AVERAGE MOVIE DURATION
movies_duration = data[data['type'] == 'Movie'].copy()
movies_duration['duration'] = movies_duration['duration'].str.replace(' min', '').astype(float)

avg_duration = movies_duration['duration'].mean()
print(f"Average Duration of Movies: {avg_duration:.2f} minutes")

sns.histplot(movies_duration['duration'], bins=20, kde=True, color='skyblue')
plt.title('Distribution of Movie Durations on Netflix') plt.xlabel('Duration
(minutes)') plt.ylabel('Number of Movies') plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```

```
Average
               Duration of Movies: 99.58 minutes
```



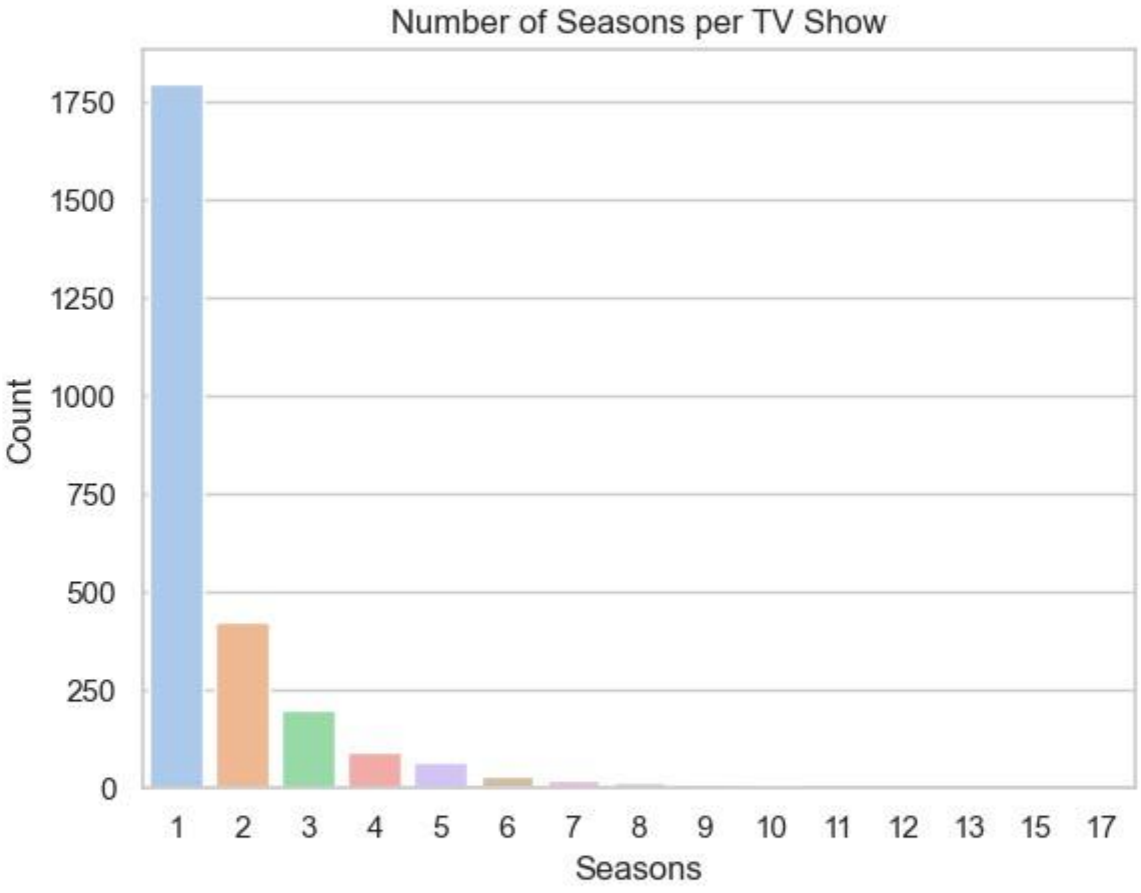
```
<Figure size 640x480 with 0 Axes> In
```

[ ]:

```
# Insight:
#- Shows typical movie length on Netflix.
#- Observation: Average movie duration is around 90-110 minutes.
#- Significance: Helps determine optimal content length for production or acquisition.
#- Recommendation: Produce movies within this duration range to align with viewer preferences and attention spans.
```

In [33]:

```
#16- MOST COMMON NUMBER OF SEASONS FOR TV SHOWS tv_shows
= data[data['type'] == 'TV Show'].copy()
tv_shows['seasons'] = tv_shows['duration'].str.replace(' Season', '').str.replace('s', '').astype(int) season_count =
tv_shows['seasons'].value_counts().sort_index()

sns.barplot(x=season_count.index, y=season_count.values, palette='pastel')
plt.title('Number of Seasons per TV Show') plt.xlabel('Seasons')
plt.ylabel('Count') plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



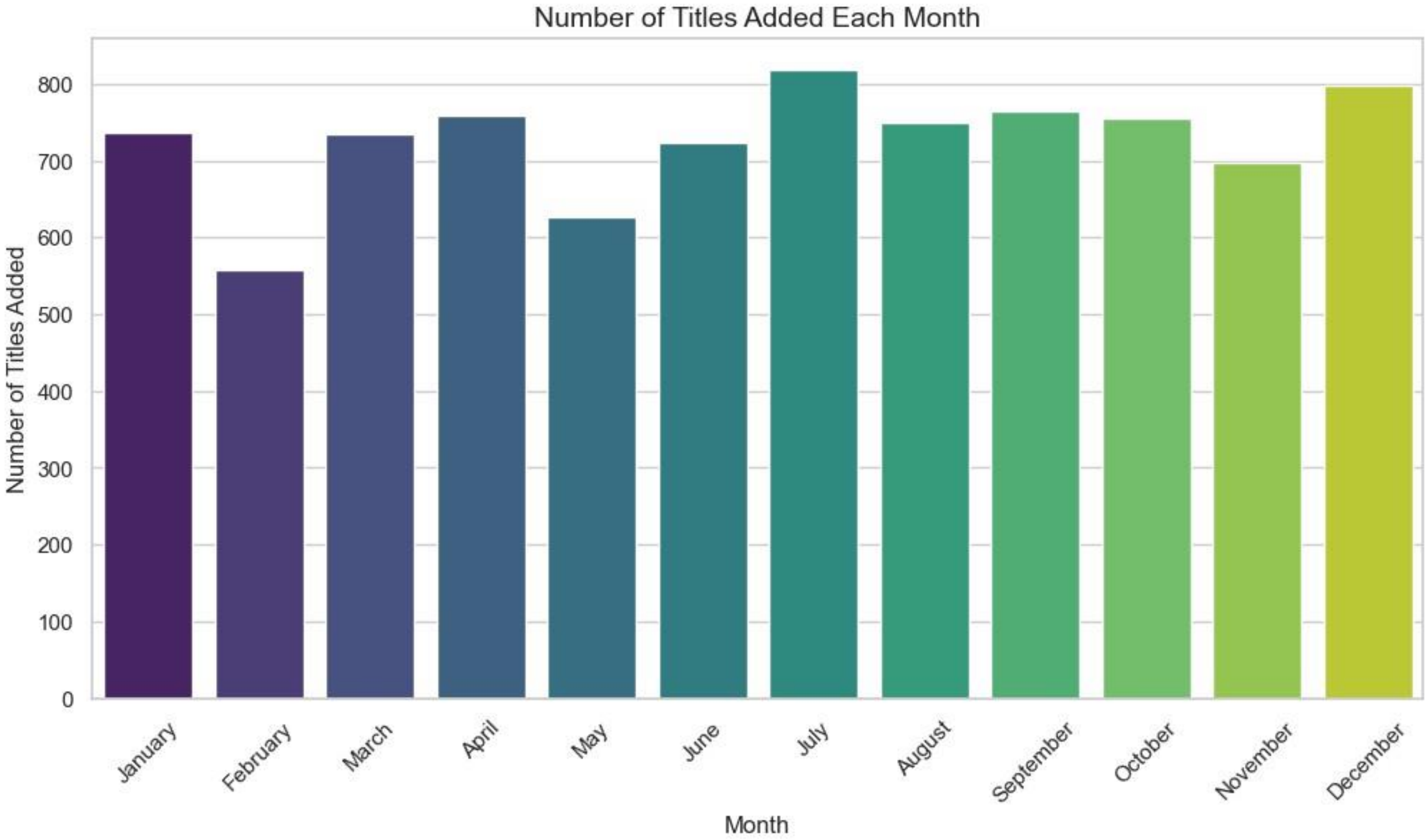
```
<Figure size 640x480 with 0 Axes> In
```

[ ]:

```
# Insight: #- Identifies the typical life cycle of
TV Shows.

#- Observation: Most TV Shows have 1 or 2 seasons.
#- Significance: Helps understand viewer engagement patterns and show renewal expectations. #- Recommendation:
Plan content production accordingly; consider mini-series or limited-run shows.
```

In [34]:

```
#17-Does the addition of new content follow any monthly or seasonal pattern
data['date_added'] = pd.to_datetime(data['date_added'], errors='coerce')
data['month_added'] = data['date_added'].dt.month_name() data['year_added']
= data['date_added'].dt.year

# Group by month
monthly_content = data['month_added'].value_counts().reindex([
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
])
# Plot
plt.figure(figsize=(12, 6)) sns.barplot(x=monthly_content.index,
y=monthly_content.values, palette='viridis') plt.xticks(rotation=45)
plt.title("Number of Titles Added Each Month", fontsize=14) plt.xlabel("Month")
plt.ylabel("Number of Titles Added") plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



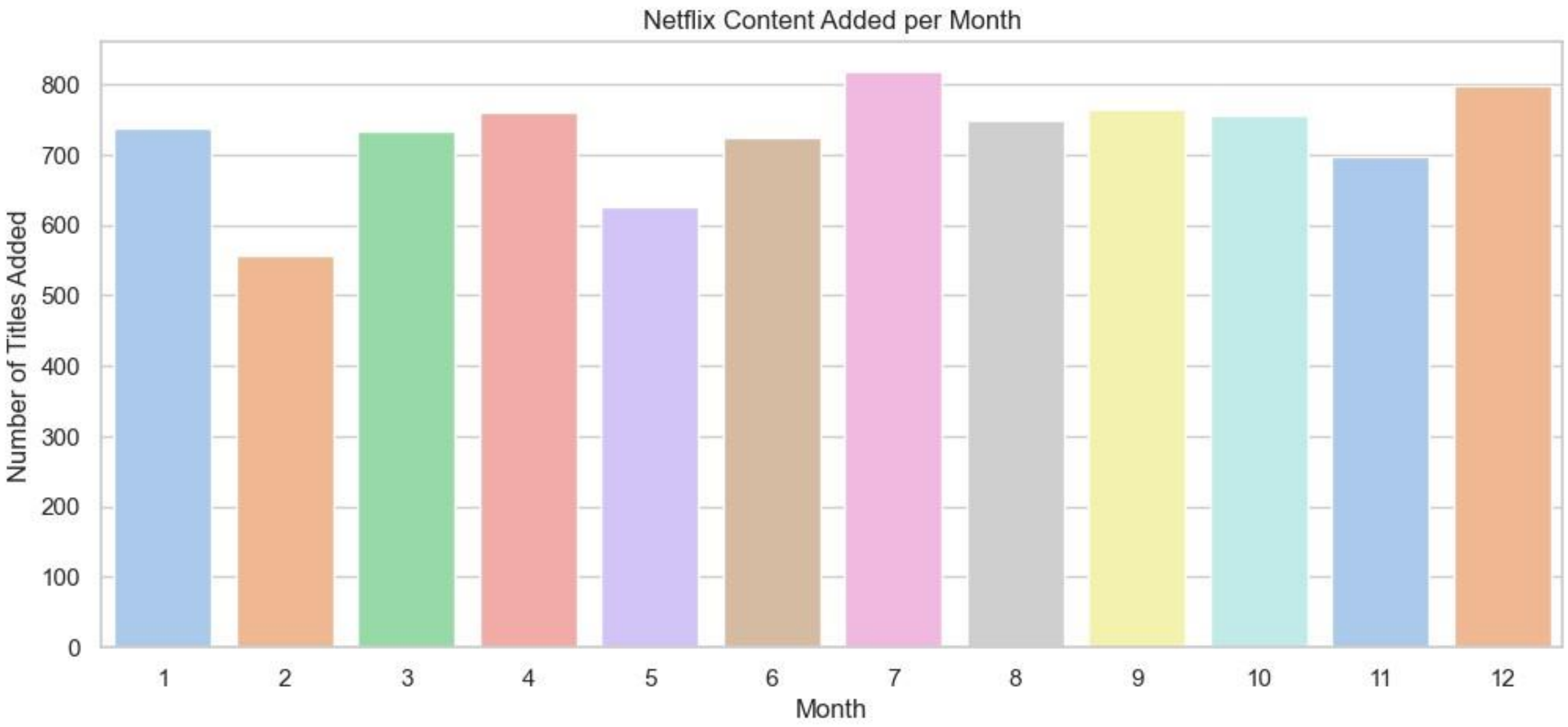
```
<Figure size 640x480 with 0 Axes>
```

In [ ]:

```
#insights:
#Netflix tends to release more titles between October and December, aligning with holiday and winter seasons when viewership s
#Early months (Jan–March) usually have moderate additions — possibly to balance content throughout the year.
#Such patterns help marketing and operations teams plan trailer releases and recommendations for maximum engagement.
```

In [35]:

```
#CONTENT LAUNCH STRATEGY
#18- MONTH-WISE CONTENT ADDITION
# Extract month
data_cleaned['month_added'] = data_cleaned['date_added'].dt.month

# Count titles added per month
month_count = data_cleaned['month_added'].value_counts().sort_index()

plt.figure(figsize=(12,5)) sns.barplot(x=month_count.index,
y=month_count.values, palette='pastel') plt.title('Netflix Content Added
per Month') plt.xlabel('Month') plt.ylabel('Number of Titles Added')
plt.show()
# Save chart to output folder
plt.savefig("../output/type_vs_count.png", bbox_inches='tight') plt.show()
```



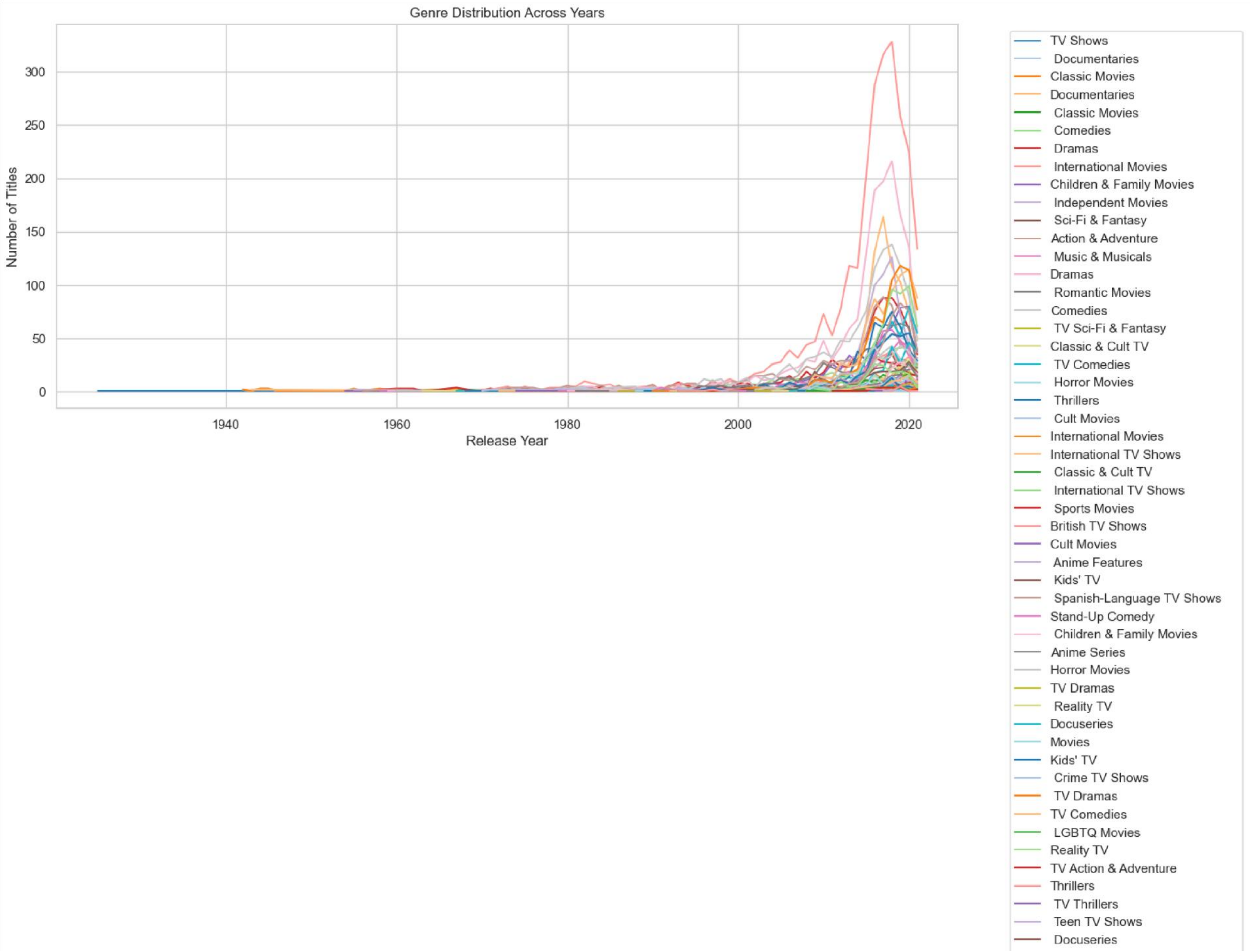
```
<Figure size 640x480 with 0 Axes>
```

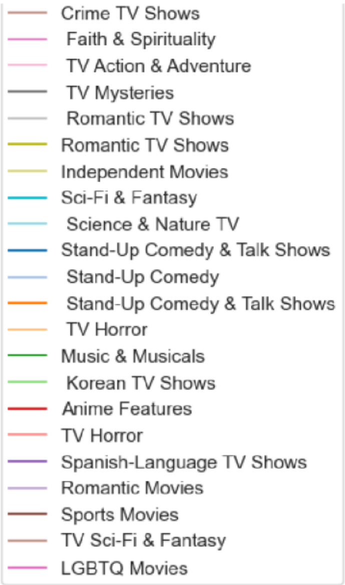
In [ ]:

```
# Insight: #- Shows when Netflix adds most content throughout
the year.
#- Observation: Peaks occur around Q1 and Q4 (January, October-December), likely for holidays or subscription drives.
#- Significance: Understanding seasonality helps optimize launch timing and promotions.
#- Recommendation: Schedule new releases in high-addition months to maximize viewer engagement.
```

In [37]:

```
# 19- Genre distribution across years
# Prepare exploded dataset data_exploded = data.copy()
data_exploded['listed_in'] = data_exploded['listed_in'].str.split(',')
data_exploded = data_exploded.explode('listed_in')

# Group by release year and genre


genre_year = data_exploded.groupby(['release_year', 'listed_in']).size().reset_index(name='count')

# Plot
plt.figure(figsize=(14,6)) sns.lineplot(data=genre_year, x='release_year', y='count', hue='listed_in',
palette='tab20', legend='full') plt.title("Genre Distribution Across Years") plt.xlabel("Release Year")
plt.ylabel("Number of Titles") plt.legend(bbox_to_anchor=(1.05,1), loc='upper left') plt.tight_layout()

# Save chart to output folder
plt.savefig("../output/genre_distribution_year.png", bbox_inches='tight') plt.show()
```


Genre Distribution Across Years
Number of Titles
0
50
100
150
200
250
300
1940
1960
1980
2000
2020
Release Year
TV Shows
Documentaries
Classic Movies
Documentaries
Classic Movies
Comedies
Dramas
International Movies
Children & Family Movies
Independent Movies
Sci-Fi & Fantasy
Action & Adventure
Music & Musicals
Dramas
Romantic Movies
Comedies
TV Sci-Fi & Fantasy
Classic & Cult TV
TV Comedies
Horror Movies
Thrillers
Cult Movies
International Movies
International TV Shows
Classic & Cult TV
International TV Shows
Sports Movies
British TV Shows
Cult Movies
Anime Features
Kids' TV
Spanish-Language TV Shows
Stand-Up Comedy
Children & Family Movies
Anime Series
Horror Movies
TV Dramas
Reality TV
Docuseries
Movies
Kids' TV
Crime TV Shows
TV Dramas
TV Comedies
LGBTQ Movies
Reality TV
TV Action & Adventure
Thrillers
TV Thrillers
Teen TV Shows
Docuseries



Out[37]:

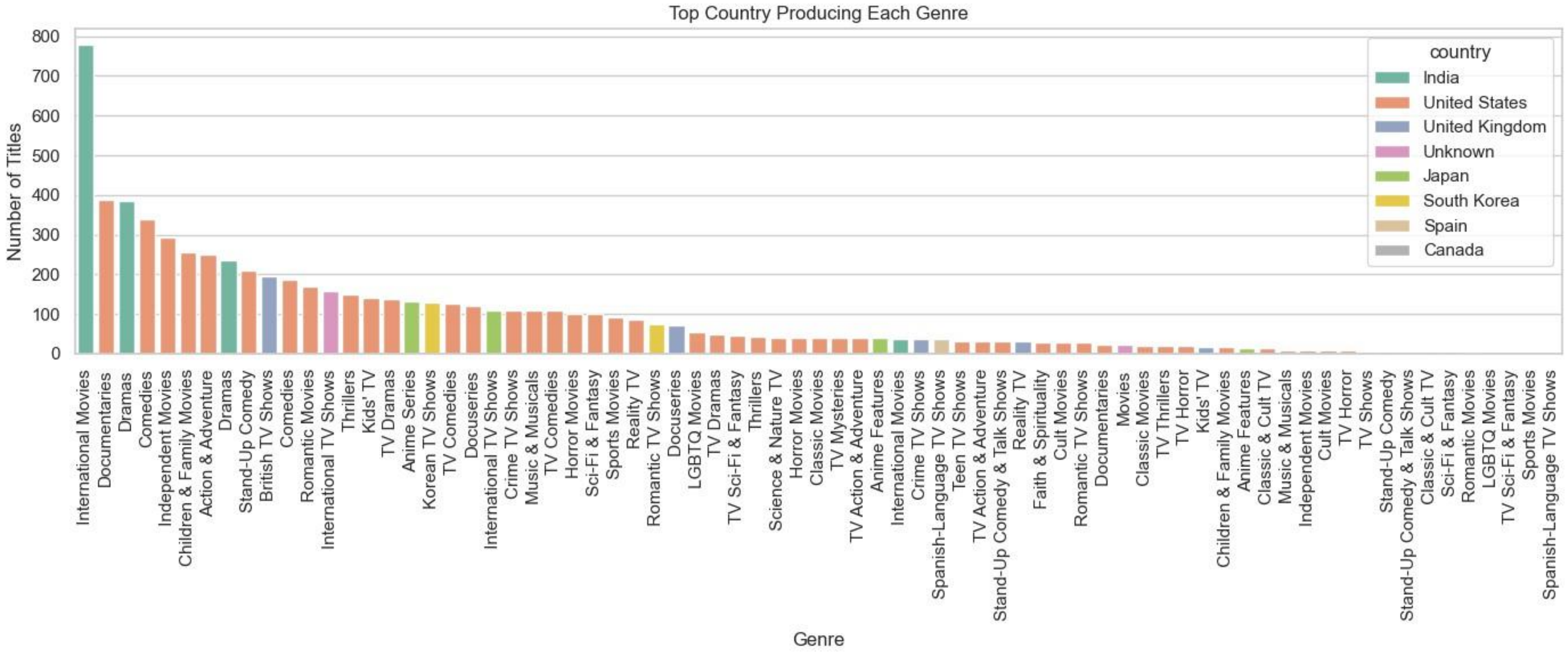
```
'\n- Drama, Comedy, and Documentary dominate consistently across years.\n- Recent years show rising Horror, Reality, and Thri ller
content.\n- Netflix can invest more in growing genres to attract younger audiences.\n- Shows a clear diversification tre nd over
time.\n'
```

In [ ]:

```
# Insight:
#- Tracks how the popularity of genres has evolved over the years.
#- Observation: Drama, Comedy, and Documentary consistently dominate; newer genres occasionally spike.
#- Significance: Helps understand shifting audience interests and Netflix’s strategic genre focus.
#- Recommendation: Monitor rising genres and produce content accordingly to stay relevant.
```

In [38]:

```
#20- Top country producing each genre
country_genre = data_exploded.groupby(['country', 'listed_in']).size().reset_index(name='count')

# Get top country per genre
top_country_genre = country_genre.sort_values('count', ascending=False).groupby('listed_in').head(1)

# Plot
plt.figure(figsize=(14,6)) sns.barplot(data=top_country_genre, x='listed_in', y='count', hue='country',
dodge=False, palette='Set2') plt.xticks(rotation=90) plt.title("Top Country Producing Each Genre")
plt.xlabel("Genre")


plt.ylabel("Number of Titles") plt.tight_layout()

# Save chart to output folder
plt.savefig("../output/top_country_genre.png", bbox_inches='tight') plt.show()
```



In [ ]:

```
# Insight:
#- Shows which genres dominate in the top producing countries.
#- Observation: USA leads in all top genres; India produces mostly Drama and Comedy.
#- Significance: Highlights cultural preferences and potential market opportunities.
#- Recommendation: Use this data to localize content, strengthen licensing, and target country-specific campaigns.
```