In [14]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [15]:
```
# Set a consistent style for the plots
sns.set_style("whitegrid")
```

In [16]:
```
# Load the data
df = pd.read_csv("netflix_titles (1).csv")

# Task 1: Content Growth Over Time (by Year Added)

# Convert 'date_added' to datetime and extract the year.
df_added = df.dropna(subset=['date_added']).copy()
df_added['date_added'] = pd.to_datetime(df_added['date_added'])
df_added['year_added'] = df_added['date_added'].dt.year
```

In [17]:
```
#Initial Inspection

print("First 5 rows of the dataset:")
print(df.head().to_markdown(index=False, numalign="left", stralign="left"))
print("\nGeneral Information about the dataset:")
df.info()
```

First 5 rows of the dataset:

| show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| s1 | Movie | Dick Johnson Is Dead | Kirsten Johnson | nan | United States | September 25, 2021 | 2020 | PG-13 | 90 min | Documentaries | As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inventive and comical ways to help them both face the inevitable. |
| s2 | TV Show | Blood & Water | nan | Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmore Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake Ka-Ncube, Odwa Gwanya, Mekaila Mathys, Sandi Schultz, Duane Williams, Shamilla Miller, Patrick Mofokeng | 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 teen sets out to prove whether a private-school swimming star is her sister who was abducted at birth. |
| s3 | TV Show | Ganglands | Julien Leclercq | Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabiha Akkari, Sofia Lesaffre, Salim Kechiouche, Noureddine Farihi, Geert Van Rampelberg, Bakary Diombera | nan | September 24, 2021 | 2021 | TV-MA | 1 Season | Crime TV Shows, International TV Shows, TV Action & Adventure | To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team of robbers are pulled into a violent and deadly turf war. |
| 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 among the incarcerated women at the Orleans Justice Center in New Orleans on this gritty reality series. |
| s5 | TV Show | Kota Factory | nan | Mayur More, Jitendra Kumar, Ranjan Raj, Alam Khan, Ahsaas Channa, Revathi Pillai, Urvi Singh, Arun Kumar | India | September 24, 2021 | 2021 | TV-MA | 2 Seasons | International TV Shows, Romantic TV Shows, TV Comedies | In a city of coaching centers known to train India’s finest collegiate minds, an earnest but unexceptional student and his friends navigate campus life. |

General Information about the dataset:

```
<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
```

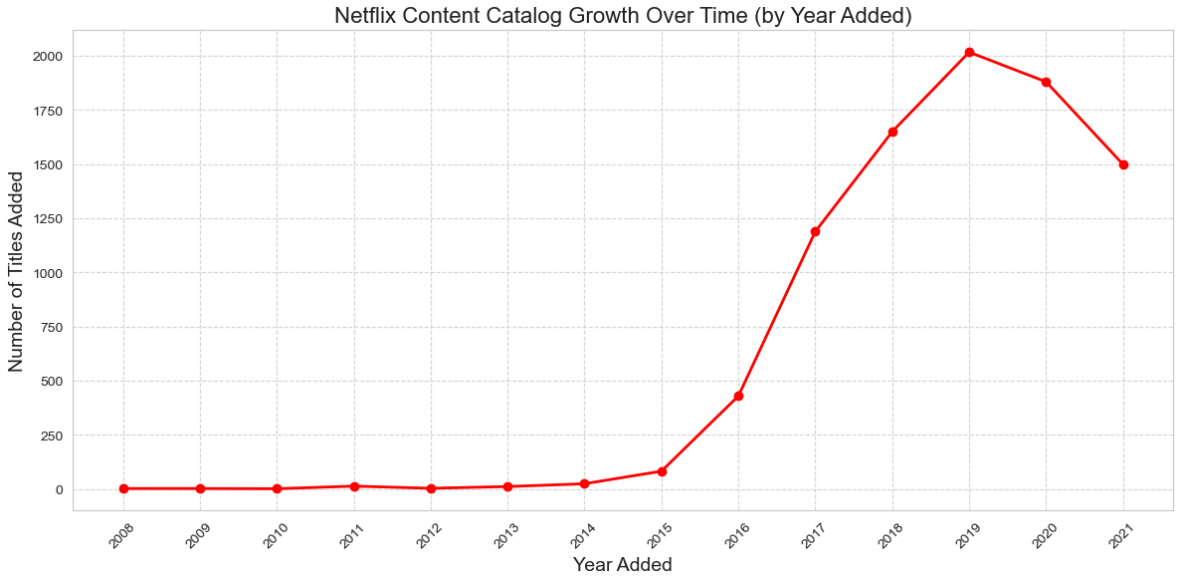
In [18]:
```
#  Data Cleaning - Missing Values check

print("\nMissing values in each column:")
print(df.isnull().sum().sort_values(ascending=False).to_markdown(numalign="lef
```

```
Missing values in each column:
|              | 0    |
|:-------------|:-----|
| director     | 2634 |
| country      | 831  |
| cast         | 825  |
| date_added   | 10   |
| rating       | 4    |
| duration     | 3    |
| show_id      | 0    |
| type         | 0    |
| title        | 0    |
| release_year | 0    |
| listed_in    | 0    |
| description  | 0    |
```

In [19]:
```
# Task-1 Count titles added per year and filter for recent years (post-2007)

content_added_yearly = df_added['year_added'].value_counts().sort_index()
content_added_yearly = content_added_yearly[content_added_yearly.index >= 2008
```

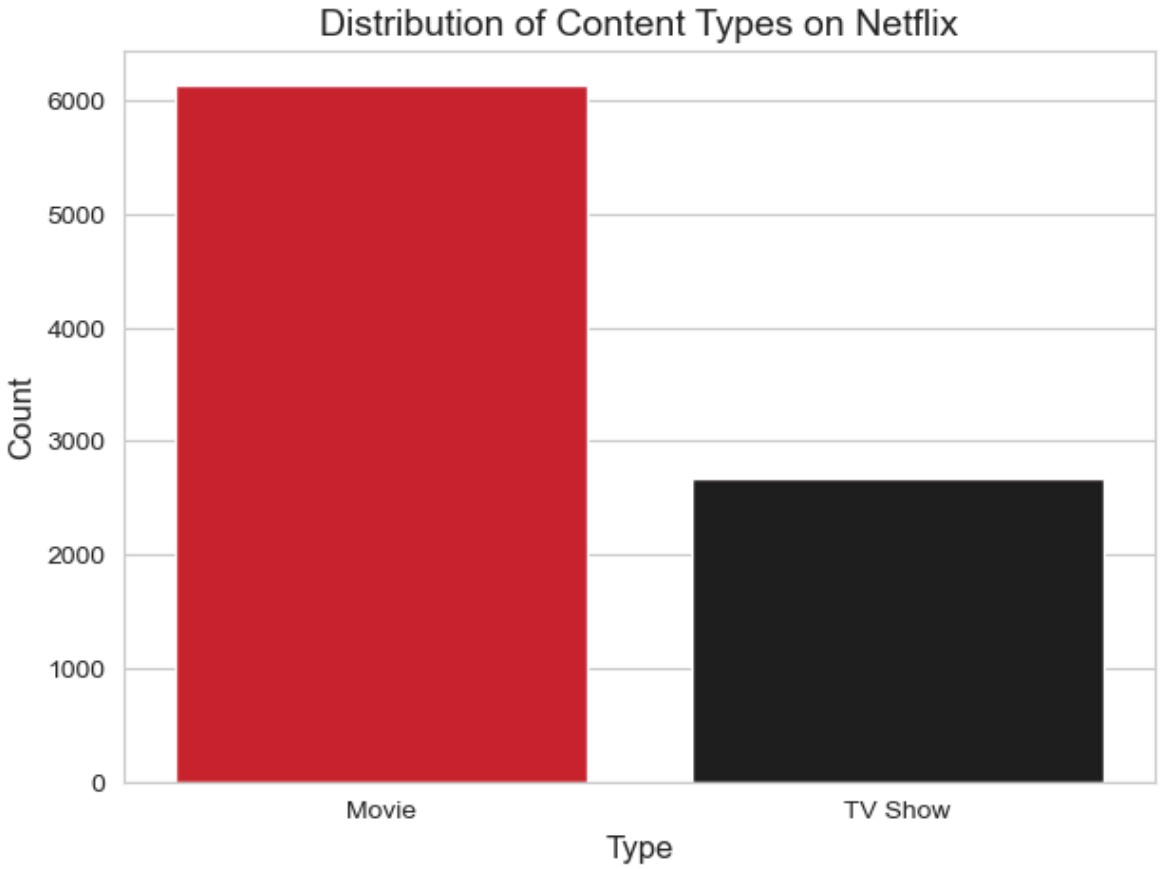
In [20]:

```
plt.figure(figsize=(12, 6))
content_added_yearly.plot(kind='line', marker='o', color='red', linewidth=2)
plt.title('Netflix Content Catalog Growth Over Time (by Year Added)', fontsize
plt.xlabel('Year Added', fontsize=14)
plt.ylabel('Number of Titles Added', fontsize=14)
plt.xticks(content_added_yearly.index, rotation=45)
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.savefig('content_growth_over_time.png')
plt.show()
```

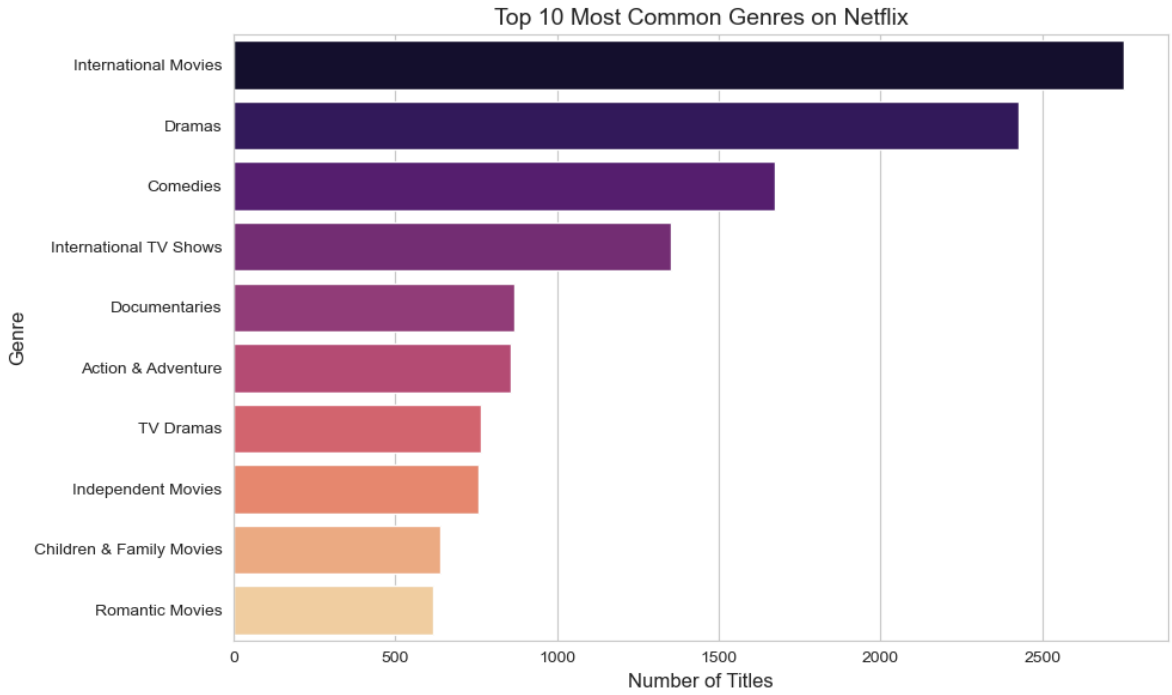


In [21]:

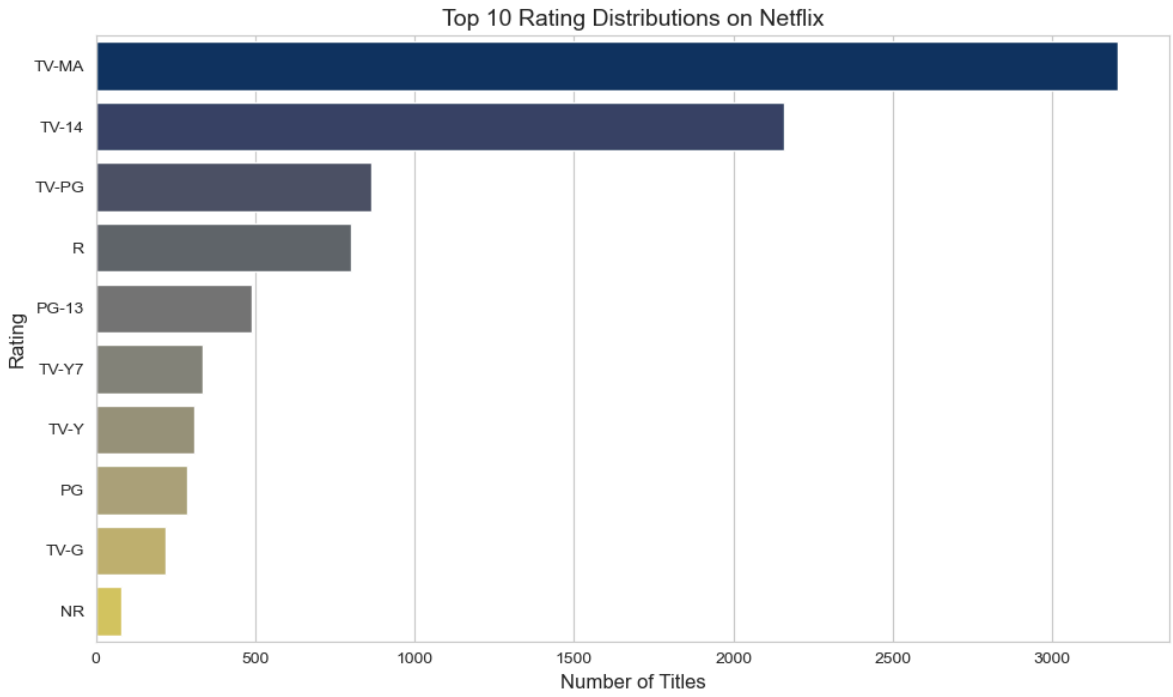
```
# Task 2: Distribution Analysis (Content Type, Genres, Ratings)

#  A. Content Type Distribution
type_counts = df['type'].value_counts()
plt.figure(figsize=(7, 5))
sns.barplot(x=type_counts.index, y=type_counts.values, palette=['#E50914', '#2
plt.title('Distribution of Content Types on Netflix', fontsize=14)
plt.xlabel('Type', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.savefig('content_type_distribution.png')
plt.show()
```



In [22]:
```
#  B. Top 10 Genres

# Split and count multiple genres listed per title
genre_df = df['listed_in'].str.split(',', expand=True).stack()
genre_df = genre_df.str.strip()
genre_counts = genre_df.value_counts().head(10)

plt.figure(figsize=(10, 6))
sns.barplot(x=genre_counts.values, y=genre_counts.index, palette='magma')
plt.title('Top 10 Most Common Genres on Netflix', fontsize=14)
plt.xlabel('Number of Titles', fontsize=12)
plt.ylabel('Genre', fontsize=12)
plt.tight_layout()
plt.savefig('top_10_genres.png')
plt.show()
```

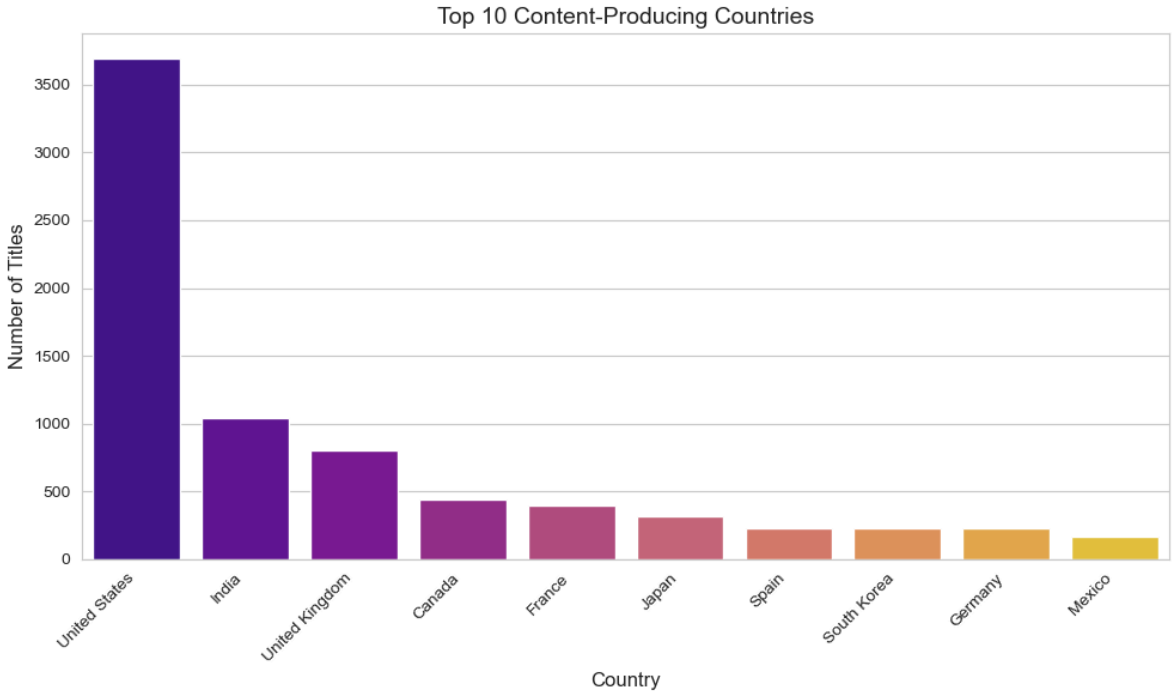


In [23]:
```
#  C. Top 10 Rating Distributions

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

plt.figure(figsize=(10, 6))
sns.barplot(x=rating_counts.values, y=rating_counts.index, palette='cividis')
plt.title('Top 10 Rating Distributions on Netflix', fontsize=14)
plt.xlabel('Number of Titles', fontsize=12)
plt.ylabel('Rating', fontsize=12)
plt.tight_layout()
plt.savefig('rating_distribution.png')
plt.show()
```




In [24]:
```
# Task 3: Country-Level Analysis (Top 10 Contributors)

# Temporarily fill 'country' NaN for splitting, then filter out 'Missing'

df_country = df.copy()
df_country['country'] = df_country['country'].fillna('Missing')
country_df = df_country['country'].str.split(',', expand=True).stack()
country_df = country_df.str.strip()
```

In [25]:
```
# Exclude 'Missing' and count

country_counts = country_df[country_df != 'Missing'].value_counts().head(10)

plt.figure(figsize=(10, 6))
sns.barplot(x=country_counts.index, y=country_counts.values, palette='plasma')
plt.title('Top 10 Content-Producing Countries', fontsize=14)
plt.xlabel('Country', fontsize=12)
plt.ylabel('Number of Titles', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.savefig('top_10_countries.png')
plt.show()
```




In [ ]: