NETFLIX DATA: CLEANING, ANALYSIS, VISUALIZATION

IMPORTING LIBRARIES

In [49]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

IMPORTING DATASET

In [51]:
```
data = pd.read_csv('C:\\Users\\Gowthami Galla\\Desktop\\netflix1.csv')
```

In [53]:
```
data.head(5)
```

Out[53]:

	show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	9/25/2021	2020	PG-13	90 min	Documentaries
1	s3	TV Show	Ganglands	Julien Leclercq	France	9/24/2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...
2	s6	TV Show	Midnight Mass	Mike Flanagan	United States	9/24/2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries
3	s14	Movie	Confessions of an Invisible Girl	Bruno Garotti	Brazil	9/22/2021	2021	TV-PG	91 min	Children & Family Movies, Comedies
4	s8	Movie	Sankofa	Haile Gerima	United States	9/24/2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies

ABOUT DATASET

In [56]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8790 non-null   object
 1   type          8790 non-null   object
 2   title         8790 non-null   object
 3   director      8790 non-null   object
 4   country       8790 non-null   object
 5   date_added    8790 non-null   object
 6   release_year  8790 non-null   int64 
 7   rating        8790 non-null   object
 8   duration      8790 non-null   object
 9   listed_in     8790 non-null   object
dtypes: int64(1), object(9)
memory usage: 686.8+ KB
```

DATA PREPROCESSING : Cleaning Missing Values, Removing Duplicates, and Standardizing Formats

In [65]:
```
required_columns = ['director', 'cast', 'country']
available_columns = [col for col in required_columns if col in data.columns]

if available_columns:
    data.dropna(subset=available_columns, inplace=True)
else:
    print("Some critical columns are missing and cannot be processed.")

data.drop_duplicates(inplace=True)
data['date_added'] = pd.to_datetime(data['date_added'], errors='coerce')
print(data.dtypes)
```

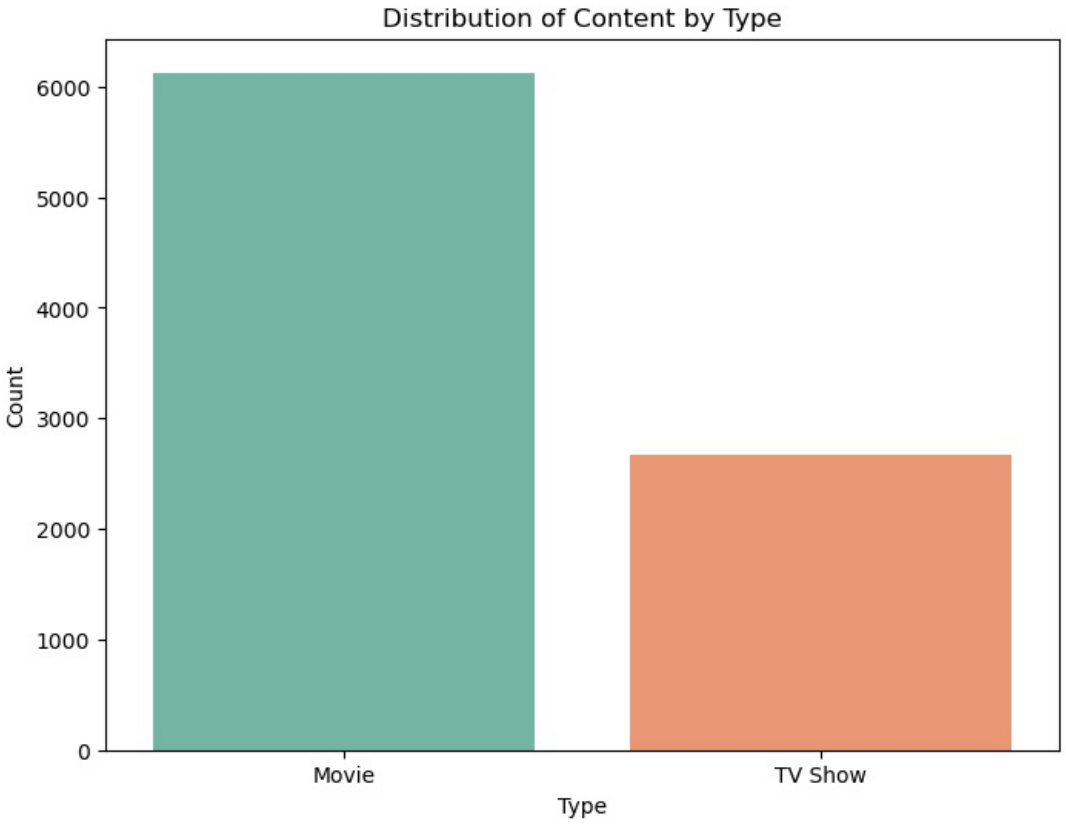
```
show_id                 object
type                    object
title                   object
director                object
country                 object
date_added      datetime64[ns]
release_year             int64
rating                  object
duration                object
listed_in               object
dtype: object
```

In [67]:
```
data.drop(columns=['unnecessary_column'], inplace=True, errors='ignore')
```

EXPLORATORY DATA ANALYSIS(EDA)

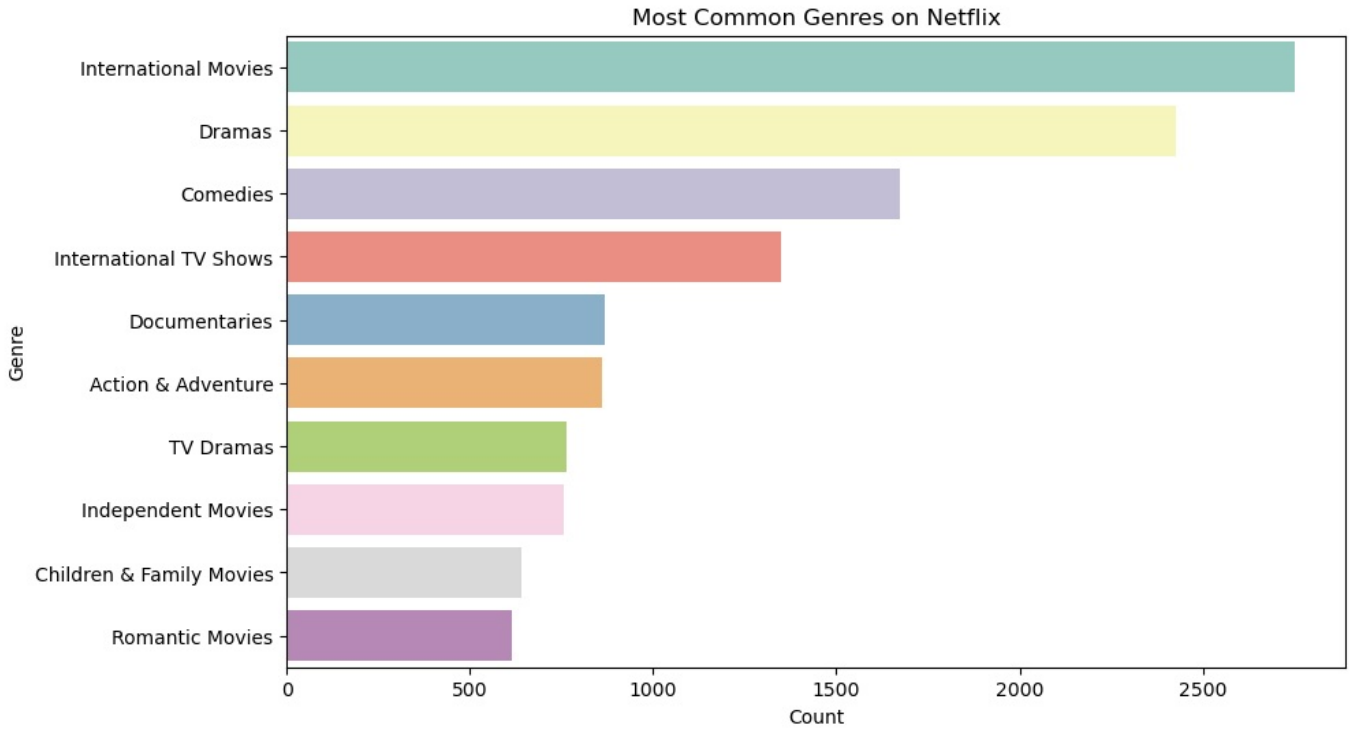
CONTENT TYPE DISTRIBUTION

In [78]:
```
type_counts = data['type'].value_counts()

plt.figure(figsize=(8, 6))
sns.barplot(x=type_counts.index, y=type_counts.values, hue=type_counts.index, palette='Set2', legend=False)
plt.title('Distribution of Content by Type')
plt.xlabel('Type')
plt.ylabel('Count')
plt.show()
```



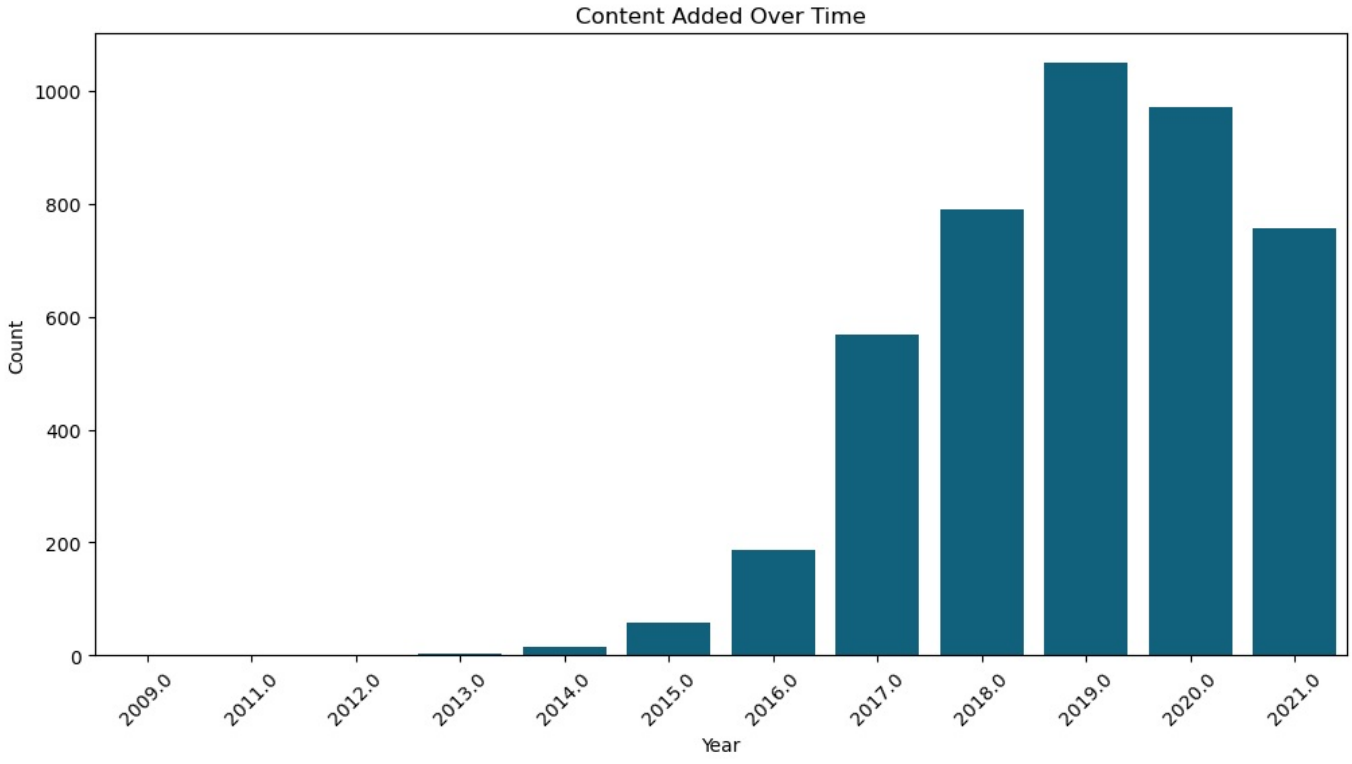
TOP CONTENT GENRES ON NETFLIX

In [84]:
```
data['genres'] = data['listed_in'].apply(lambda x: x.split(', '))

all_genres = sum(data['genres'], [])
genre_counts = pd.Series(all_genres).value_counts().head(10)

plt.figure(figsize=(10, 6))
sns.barplot(x=genre_counts.values, y=genre_counts.index, hue=genre_counts.index, palette='Set3', legend=False)
plt.title('Most Common Genres on Netflix')
plt.xlabel('Count')
plt.ylabel('Genre')
plt.show()
```



CONTENT GROWTH OVERTIME

In [90]:
```
data['year_added'] = data['date_added'].dt.year
data['month_added'] = data['date_added'].dt.month

plt.figure(figsize=(12, 6))
sns.countplot(x='year_added', data=data, color='#006A8E')
plt.title('Content Added Over Time')
plt.xlabel('Year')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



TOP 10 DIRECTORS WITH MOST TITLES

In [102…
```
sns.barplot(x=top_directors.values, y=top_directors.index, color='#90EE90')  # Light green color
plt.title('Top 10 Directors with the Most Titles')
plt.xlabel('Number of Titles')
plt.ylabel('Director')
plt.show()
```