

```
In [4]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud

df = pd.read_csv("C:\\Users\\HP\\Downloads\\netflix1 (1).csv")
print(df.info())

# Remove duplicates
df.drop_duplicates(inplace=True)

# Convert date_added to datetime
df['date_added'] = pd.to_datetime(df['date_added'])

# Extract year, month, day
df['year'] = df['date_added'].dt.year
df['month'] = df['date_added'].dt.month
df['day'] = df['date_added'].dt.day

# Overview of cleaned data
df.info()
```

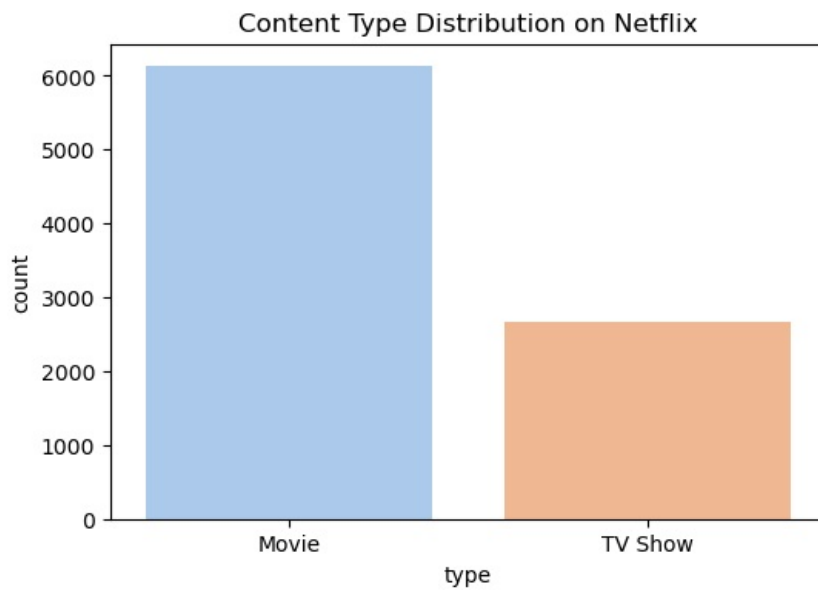
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  -
0   show_id         8790 non-null   object
1   type            8790 non-null   object
2   title           8790 non-null   object
3   director        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
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 13 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   datetime64[ns]
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
10  year            8790 non-null   int32
11  month           8790 non-null   int32
12  day             8790 non-null   int32
dtypes: datetime64[ns](1), int32(3), int64(1), object(8)
memory usage: 789.9+ KB
```

```
In [5]: # Content Type Distribution (Movies vs TV Shows)
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='type', palette='pastel')
plt.title("Content Type Distribution on Netflix")
plt.show()
```

C:\Users\HP\AppData\Local\Temp\ipykernel_4952\277233955.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df, x='type', palette='pastel')
```

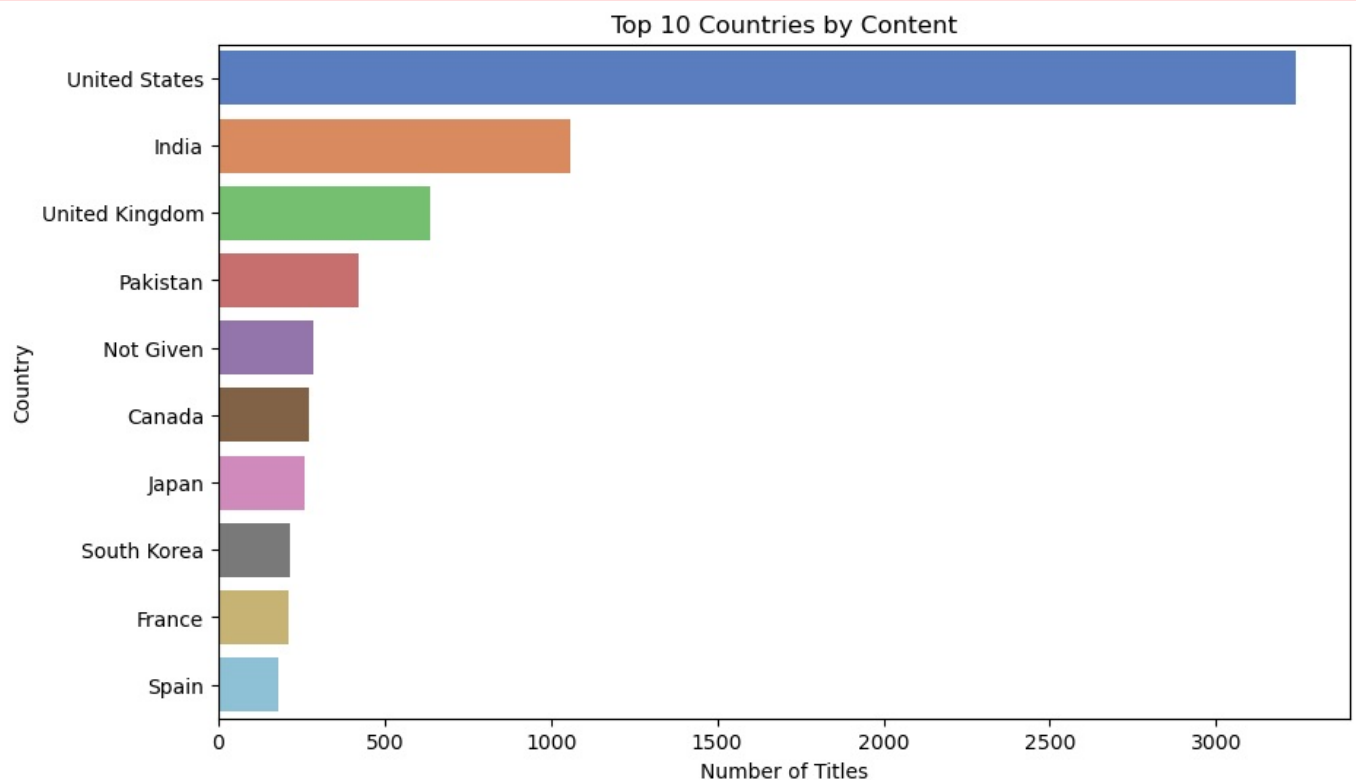


```
In [6]: #Top 10 Countries by Content
top_countries = df['country'].value_counts().head(10)
plt.figure(figsize=(10,6))
sns.barplot(x=top_countries.values, y=top_countries.index, palette="muted")
plt.title("Top 10 Countries by Content")
plt.xlabel("Number of Titles")
plt.ylabel("Country")
plt.show()
```

C:\Users\HP\AppData\Local\Temp\ipykernel_4952\3163303443.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=top_countries.values, y=top_countries.index, palette="muted")
```



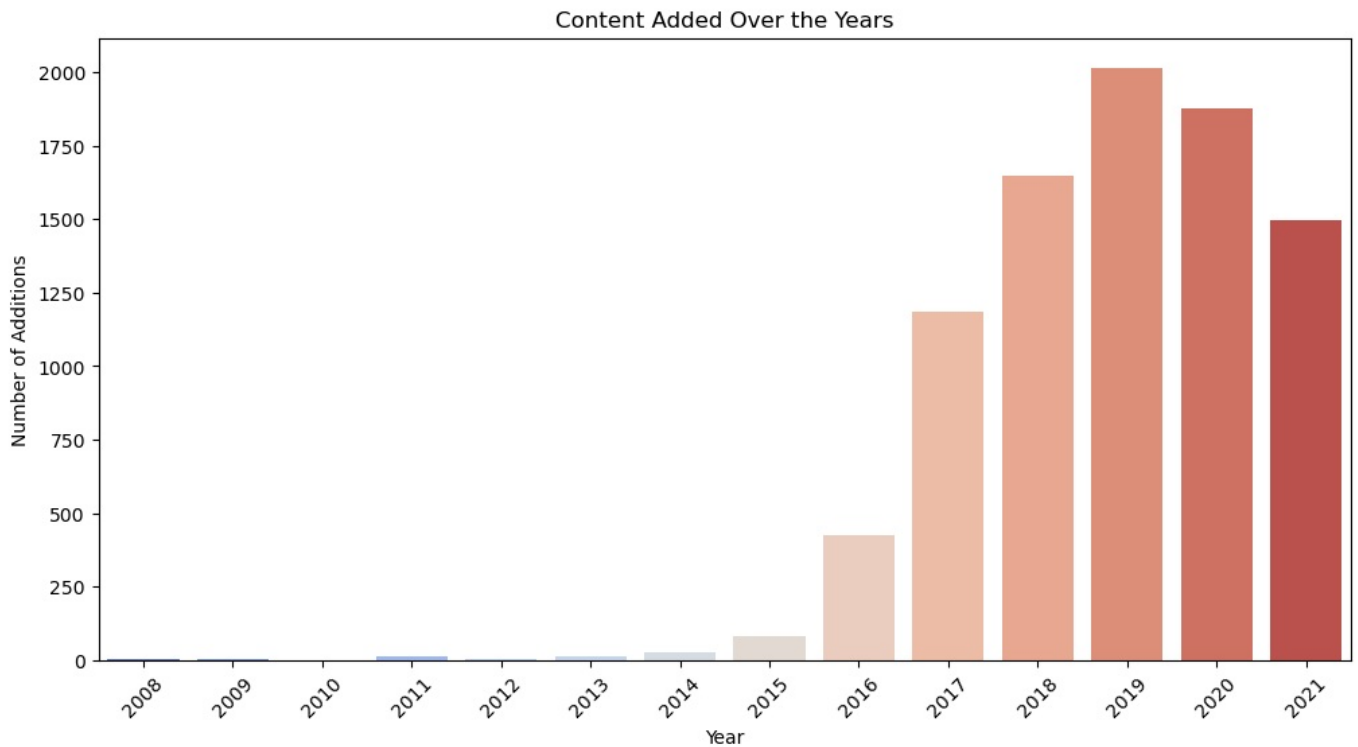
```
In [7]: #Content Added Over Time (Yearly)
plt.figure(figsize=(12,6))
sns.countplot(x='year', data=df, palette='coolwarm')
plt.title('Content Added Over the Years')
```

```
plt.xticks(rotation=45)
plt.xlabel("Year")
plt.ylabel("Number of Additions")
plt.show()
```

C:\Users\HP\AppData\Local\Temp\ipykernel_4952\3460935283.py:3: FutureWarning:

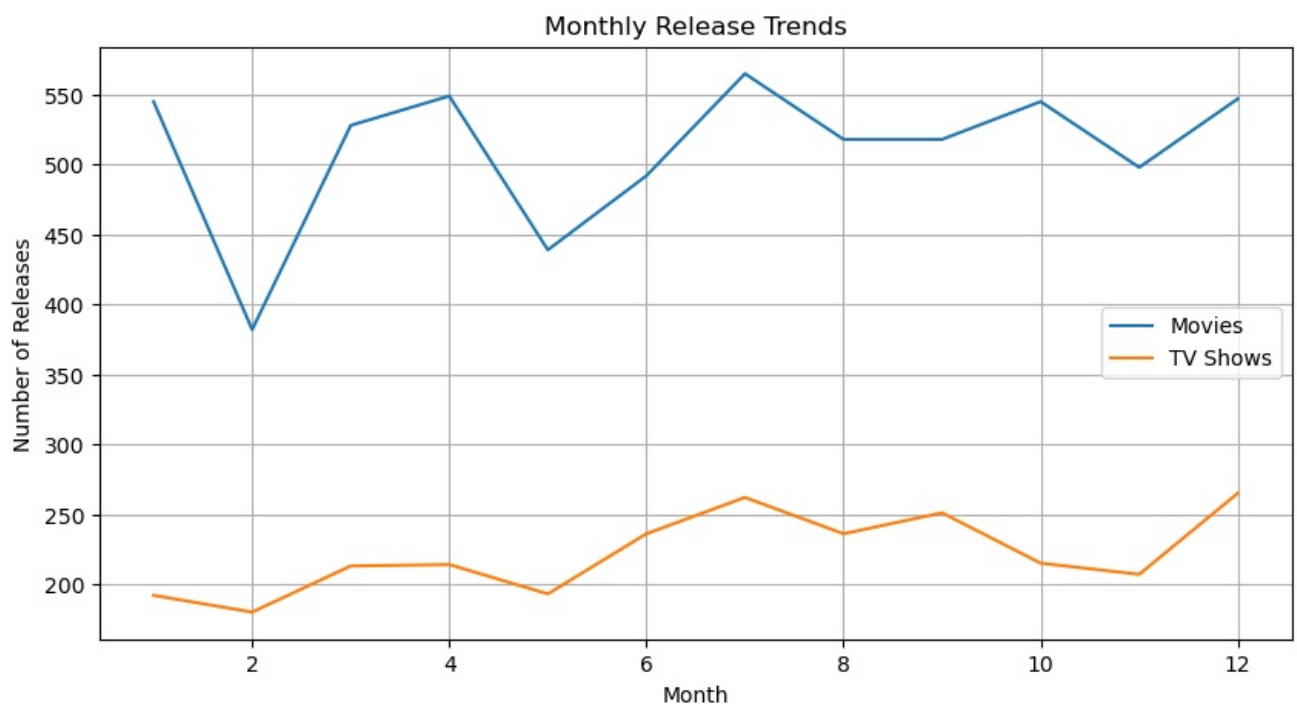
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(x='year', data=df, palette='coolwarm')
```



```
In [8]: # Monthly Release Trends
monthly_movies = df[df['type']=='Movie']['month'].value_counts().sort_index()
monthly_tv = df[df['type']=='TV Show']['month'].value_counts().sort_index()

plt.figure(figsize=(10,5))
plt.plot(monthly_movies.index, monthly_movies.values, label='Movies')
plt.plot(monthly_tv.index, monthly_tv.values, label='TV Shows')
plt.legend()
plt.title("Monthly Release Trends")
plt.xlabel("Month")
plt.ylabel("Number of Releases")
plt.grid(True)
plt.show()
```



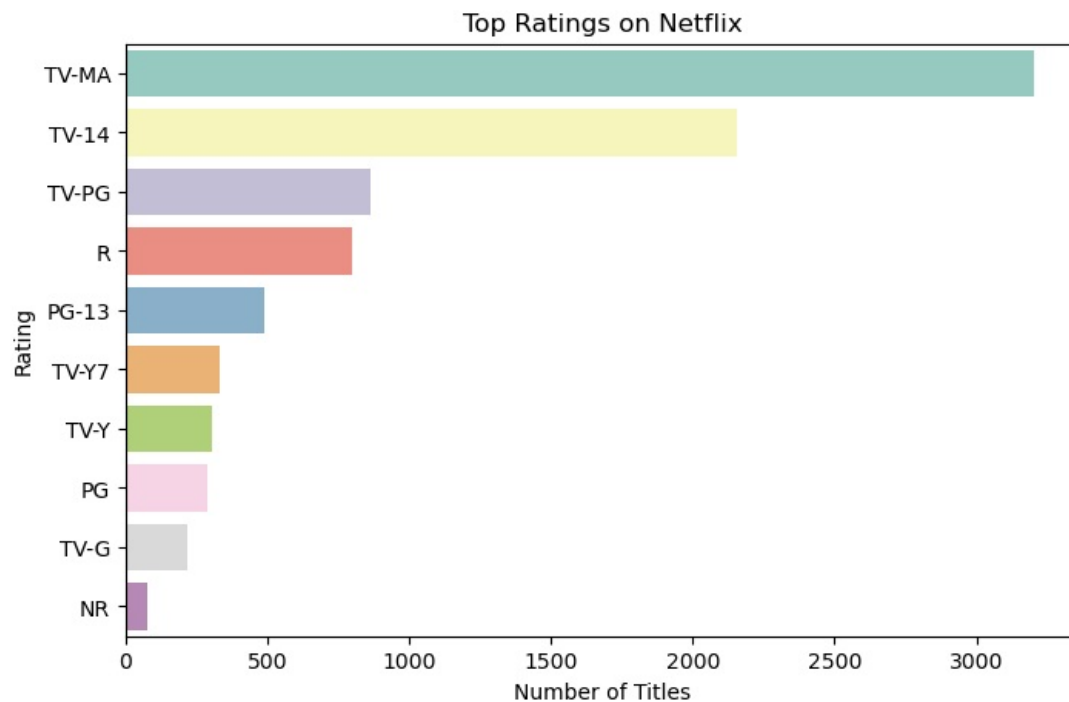
In [9]:

```
# Rating Distribution
rating_counts = df['rating'].value_counts().head(10)
plt.figure(figsize=(8,5))
sns.barplot(x=rating_counts.values, y=rating_counts.index, palette='Set3')
plt.title("Top Ratings on Netflix")
plt.xlabel("Number of Titles")
plt.ylabel("Rating")
plt.show()
```

C:\Users\HP\AppData\Local\Temp\ipykernel_4952\3749839957.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=rating_counts.values, y=rating_counts.index, palette='Set3')
```



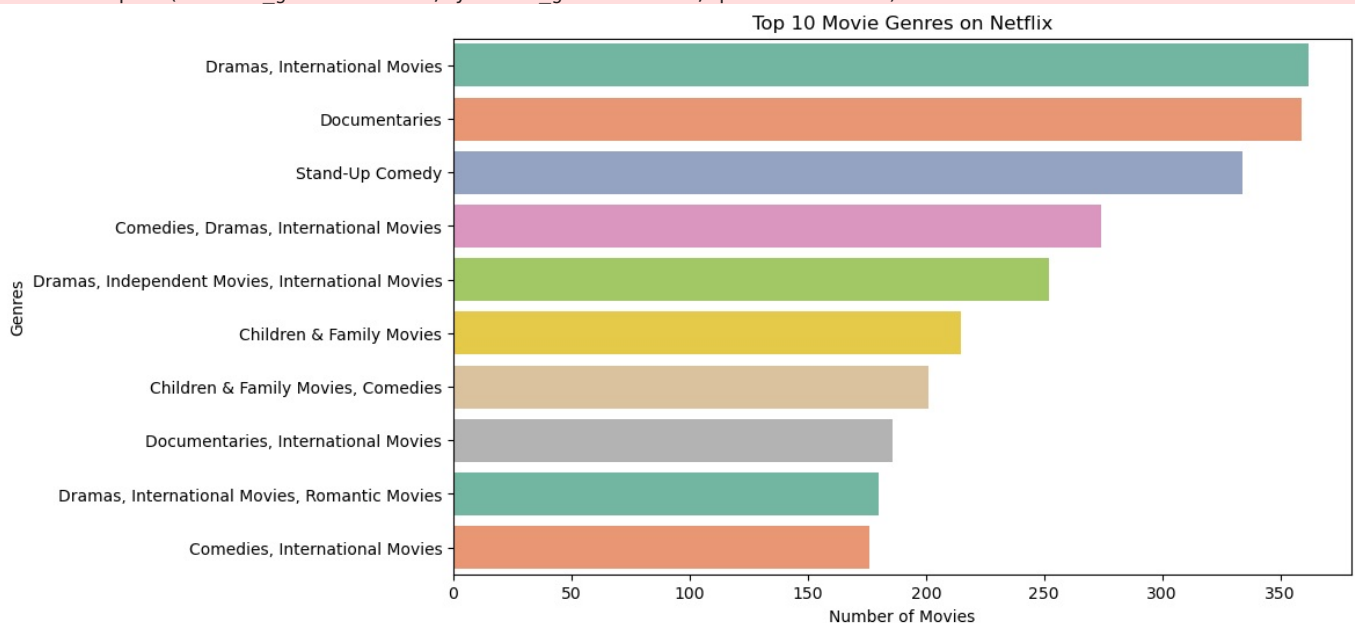
In [10]:

```
# Top 10 Movie Genres
movie_genres = df[df['type']=='Movie']['listed_in'].value_counts().head(10)
plt.figure(figsize=(10,6))
sns.barplot(x=movie_genres.values, y=movie_genres.index, palette='Set2')
plt.title("Top 10 Movie Genres on Netflix")
plt.xlabel("Number of Movies")
plt.ylabel("Genres")
plt.show()
```

C:\Users\HP\AppData\Local\Temp\ipykernel_4952\60832601.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=movie_genres.values, y=movie_genres.index, palette='Set2')
```

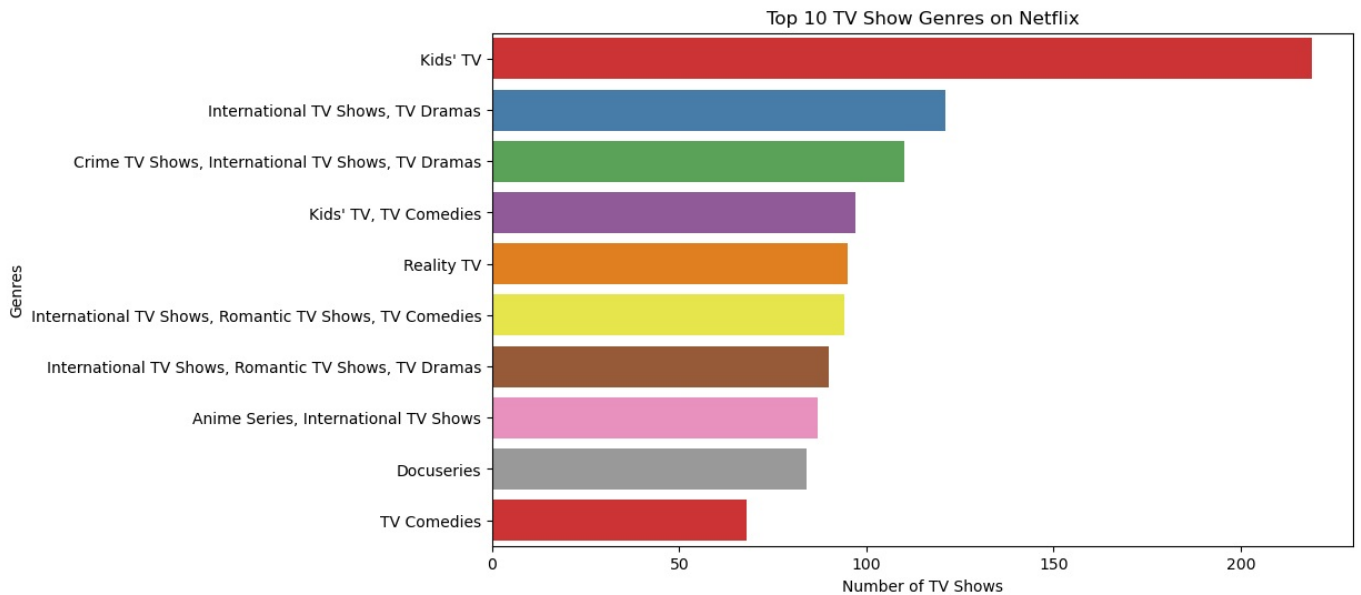


```
In [11]: # Top 10 TV Show Genres
tv_genres = df[df['type']=='TV Show']['listed_in'].value_counts().head(10)
plt.figure(figsize=(10,6))
sns.barplot(x=tv_genres.values, y=tv_genres.index, palette='Set1')
plt.title("Top 10 TV Show Genres on Netflix")
plt.xlabel("Number of TV Shows")
plt.ylabel("Genres")
plt.show()
```

C:\Users\HP\AppData\Local\Temp\ipykernel_4952\2089549016.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=tv_genres.values, y=tv_genres.index, palette='Set1')
```

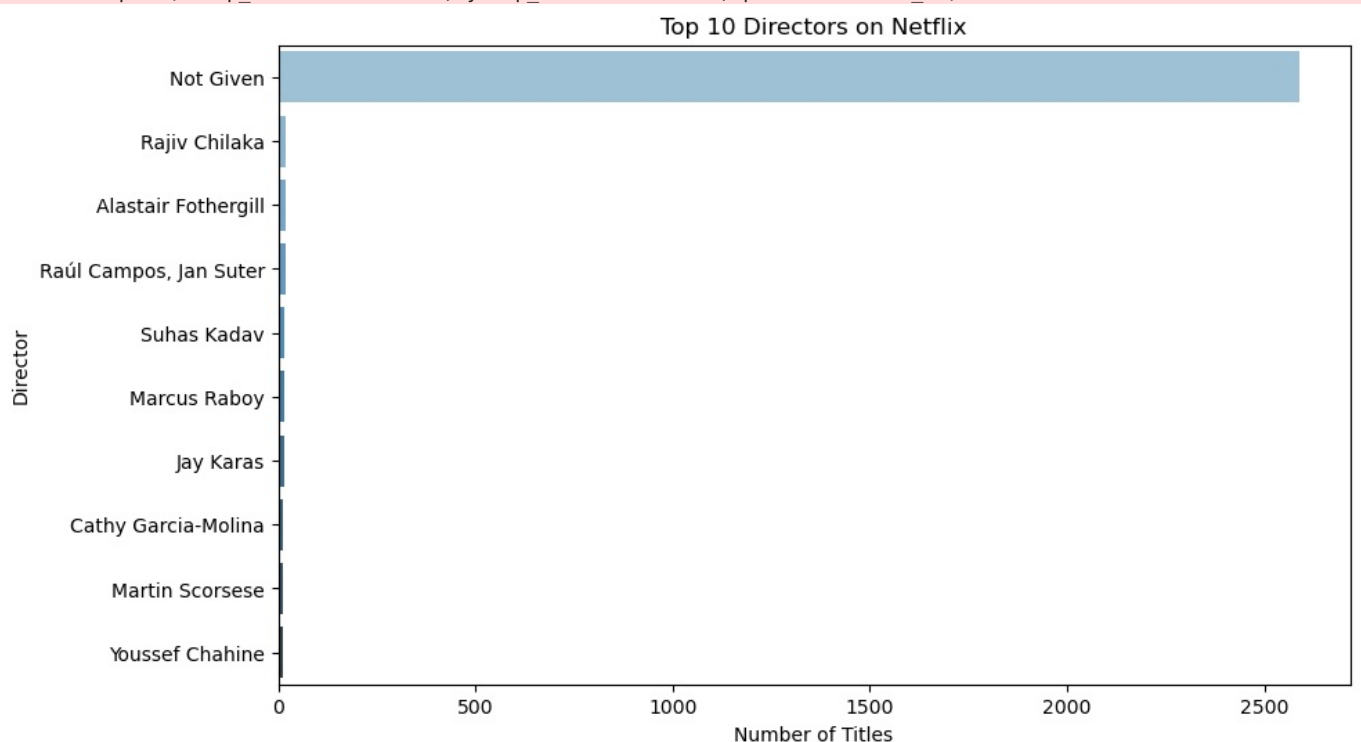


```
In [12]: #Top 10 Directors with Most Content
top_directors = df['director'].value_counts().dropna().head(10)
plt.figure(figsize=(10,6))
sns.barplot(x=top_directors.values, y=top_directors.index, palette="Blues_d")
plt.title("Top 10 Directors on Netflix")
plt.xlabel("Number of Titles")
plt.ylabel("Director")
plt.show()
```

C:\Users\HP\AppData\Local\Temp\ipykernel_4952\4293768552.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=top_directors.values, y=top_directors.index, palette="Blues_d")
```



```
In [13]: #Word Cloud of Movie Titles
titles = " ".join(df[df['type'] == "Movie"]['title'].dropna())
wordcloud = WordCloud(width=800, height=400, background_color='black').generate(titles)

plt.figure(figsize=(10,6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Word Cloud of Movie Titles")
plt.show()
```

Word Cloud of Movie Titles



In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js