In [110…

```
import pandas as pd

df=pd.read_csv("Netflix.csv")
print(df.head())
```

```
  show_id     type                                      title        director
\
0      s1  TV Show                                         3%
NaN
1     s10    Movie                                       1920    Vikram
Bhatt
2    s100    Movie                                 3 Heroines  Iman
Brotoseno
3   s1000    Movie  Blue Mountain State: The Rise of Thadland    Lev L.
Spiro
4   s1001  TV Show                             Blue Planet II
NaN

                                                cast         country  \
0  João Miguel, Bianca Comparato, Michel Gomes, R...          Brazil
1  Rajneesh Duggal, Adah Sharma, Indraneil Sengup...           India
2  Reza Rahadian, Bunga Citra Lestari, Tara Basro...       Indonesia
3  Alan Ritchson, Darin Brooks, James Cade, Rob R...   United States
4                                 David Attenborough  United Kingdom

  date_added  release_year rating  duration  \
0  14-Aug-20          2020  TV-MA         4
1  15-Dec-17          2008  TV-MA       143
2   5-Jan-19          2016  TV-PG       124
3   1-Mar-16          2016      R        90
4   3-Dec-18          2017   TV-G         1

                                              genres  \
0  International TV Shows, TV Dramas, TV Sci-Fi &...
1     Horror Movies, International Movies, Thrillers
2        Dramas, International Movies, Sports Movies
3                                           Comedies
4  British TV Shows, Docuseries, Science & Nature TV

                                         description
0  In a future where the elite inhabit an island ...
1  An architect and his wife move into a castle t...
2  Three Indonesian women break records by becomi...
3  New NFL star Thad buys his old teammates' belo...
4  This sequel to the award-winning nature series...
```

In [112…

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7787 entries, 0 to 7786
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       7787 non-null   object
 1   type          7787 non-null   object
 2   title         7787 non-null   object
 3   director      5398 non-null   object
 4   cast          7069 non-null   object
 5   country       7280 non-null   object
 6   date_added    7777 non-null   object
 7   release_year  7787 non-null   int64 
 8   rating        7780 non-null   object
 9   duration      7787 non-null   int64 
 10  genres        7787 non-null   object
 11  description   7787 non-null   object
dtypes: int64(2), object(10)
memory usage: 730.2+ KB
None
```

```
In [114… print(df.isnull().sum())
```

```
show_id            0
type               0
title              0
director        2389
cast             718
country          507
date_added        10
release_year       0
rating             7
duration           0
genres             0
description        0
dtype: int64
```

There are missing values in this datasets, so we using dropna() and Fillna() In this step

```
In [92]: df.fillna({'director':'Unknown'}, inplace=True)          #Filling
         Missing Vales as Unknown
         df.fillna({'cast':'Unknown'}, inplace=True)              #Filling
         Missing Vales as Unknown
         df.fillna({'country':'Global'}, inplace=True)            #Filling
         Missing Vales as Global
         df.fillna({'rating':'Not Rated'}, inplace=True)          #Filling
         Missing Vales as 8
         df['date_added']=pd.to_datetime(df['date_added'])
         mode_date=df['date_added'].mode()[0]
         #calculate mode first
         df.loc[df['date_added'].isna(),'date_added'] = mode_date #Filling
         Missing with most common date
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_108\3420880857.py:5: UserWarning:
Could not infer format, so each element will be parsed individually, falling
back to `dateutil`. To ensure parsing is consistent and as-expected, please
specify a format.
  df['date_added']=pd.to_datetime(df['date_added'])
```

```
In [94]: 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
genres          0
description     0
dtype: int64
```

Remove Duplicates

```
In [96]: df.drop_duplicates(inplace=True)
```

Convert Data types

```
In [98]: #convert rating into categorical
         df['rating']=df['rating'].astype('category')

         #Ensure 'release_year' is integer
         df['release_year']=df['release_year'].astype(int)
```

Understand Data Distribution

```
In [100… print(df.info())
         print(df.describe())
         print(df.nunique())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7787 entries, 0 to 7786
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       7787 non-null   object        
 1   type          7787 non-null   object        
 2   title         7787 non-null   object        
 3   director      7787 non-null   object        
 4   cast          7787 non-null   object        
 5   country       7787 non-null   object        
 6   date_added    7787 non-null   datetime64[ns]
 7   release_year  7787 non-null   int32         
 8   rating        7787 non-null   category      
 9   duration      7787 non-null   int64         
 10  genres        7787 non-null   object        
 11  description   7787 non-null   object        
dtypes: category(1), datetime64[ns](1), int32(1), int64(1), object(8)
memory usage: 647.2+ KB
None
                          date_added  release_year     duration
count                           7787   7787.000000  7787.000000
mean   2019-01-03 06:32:35.566970624   2013.932580    69.122769
min              2008-01-01 00:00:00   1925.000000     1.000000
25%              2018-02-01 00:00:00   2013.000000     2.000000
50%              2019-03-08 00:00:00   2017.000000    88.000000
75%              2020-01-17 12:00:00   2018.000000   106.000000
max              2021-01-16 00:00:00   2021.000000   312.000000
std                              NaN      8.757395    50.950743
show_id         7787
type               2
title           7787
director        4050
cast            6832
country          682
date_added      1512
release_year      73
rating            15
duration         206
genres           492
description     7769
dtype: int64
```

```
In [105…  print(f'Duplicate Rows:{df.duplicated().sum()}')   #Checking Duplicates entries
```

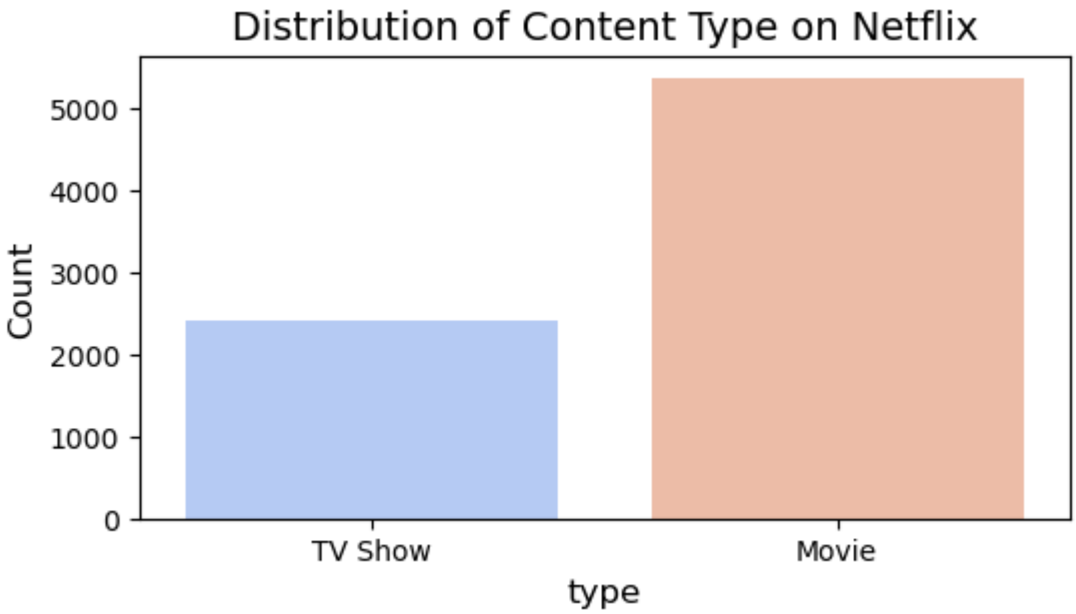
```
Duplicate Rows:0
```

Distribution of Content Types

In [106…

```
import matplotlib.pyplot as plt
import seaborn as sns

#Count plot  for types of content ( Movie vs TV Show )
%matplotlib inline

plt.figure(figsize=(6,3))
sns.countplot(x=df['type'], palette='coolwarm')
plt.title('Distribution of Content Type on Netflix', fontsize=14)
plt.xlabel('type', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.show()
```

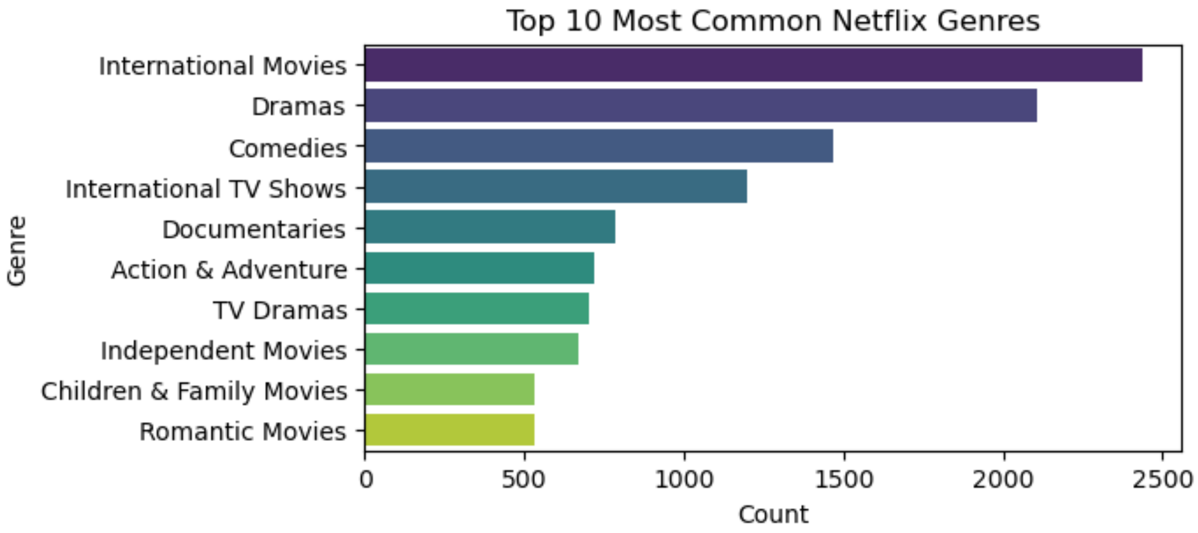
```
C:\Users\Dell\AppData\Local\Temp\ipykernel_108\2919100073.py:8:
FutureWarning:

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=df['type'], palette='coolwarm')
```



Top 10 Most Common Genres

```
In [116…  from collections import Counter

          # Drop missing values in 'genres' column and split multiple genres
          all_genres = ', '.join(df['genres'].dropna()).split(', ')

          # Count occurrences of each genre
          genre_count = Counter(all_genres)

          # Convert to DataFrame
          genre_df = pd.DataFrame(genre_count.items(), columns=['Genre', 'Co
          unt']).sort_values(by='Count', ascending=False)

          # Plot Top 10 Genres
          plt.figure(figsize=(6, 3))
          sns.barplot(y=genre_df['Genre'][:10], x=genre_df['Count'][:10], pa
          lette="viridis")

          plt.xlabel("Count")
          plt.ylabel("Genre")
          plt.title("Top 10 Most Common Netflix Genres")

          plt.show()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_108\3851561423.py:14:
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(y=genre_df['Genre'][:10], x=genre_df['Count'][:10],
palette="viridis")
```

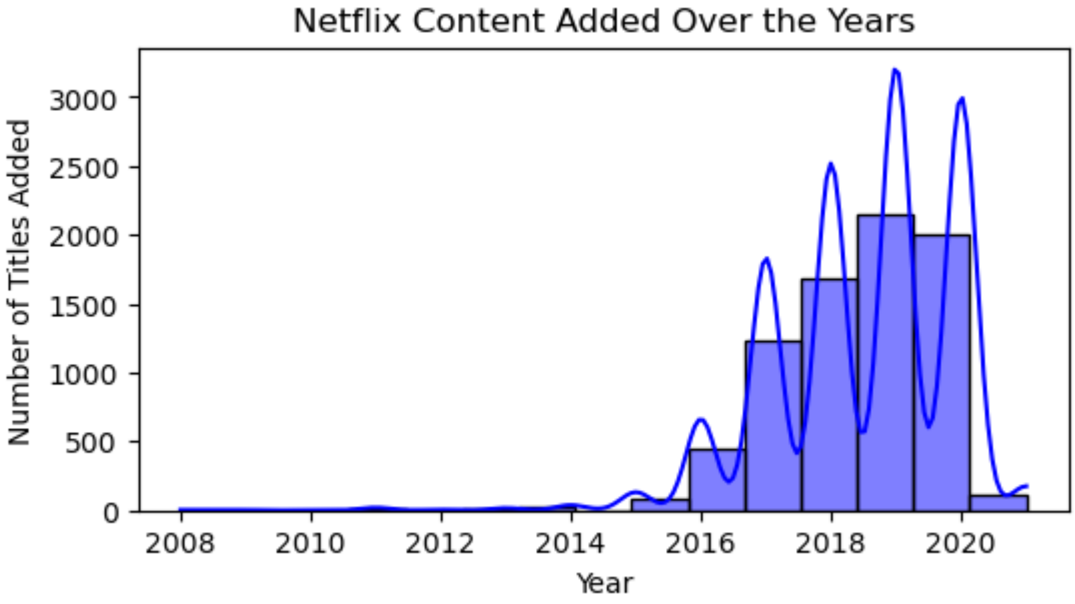


Trend of Content Added Over the Years

In [118…

```
df['date_added'] = pd.to_datetime(df['date_added'],
errors='coerce')

df['year_added'] = df['date_added'].dt.year  # Extract year

plt.figure(figsize=(6,3))
sns.histplot(df['year_added'], bins=15, kde=True, color="blue")
plt.title("Netflix Content Added Over the Years")
plt.xlabel("Year")
plt.ylabel("Number of Titles Added")
plt.show()
```

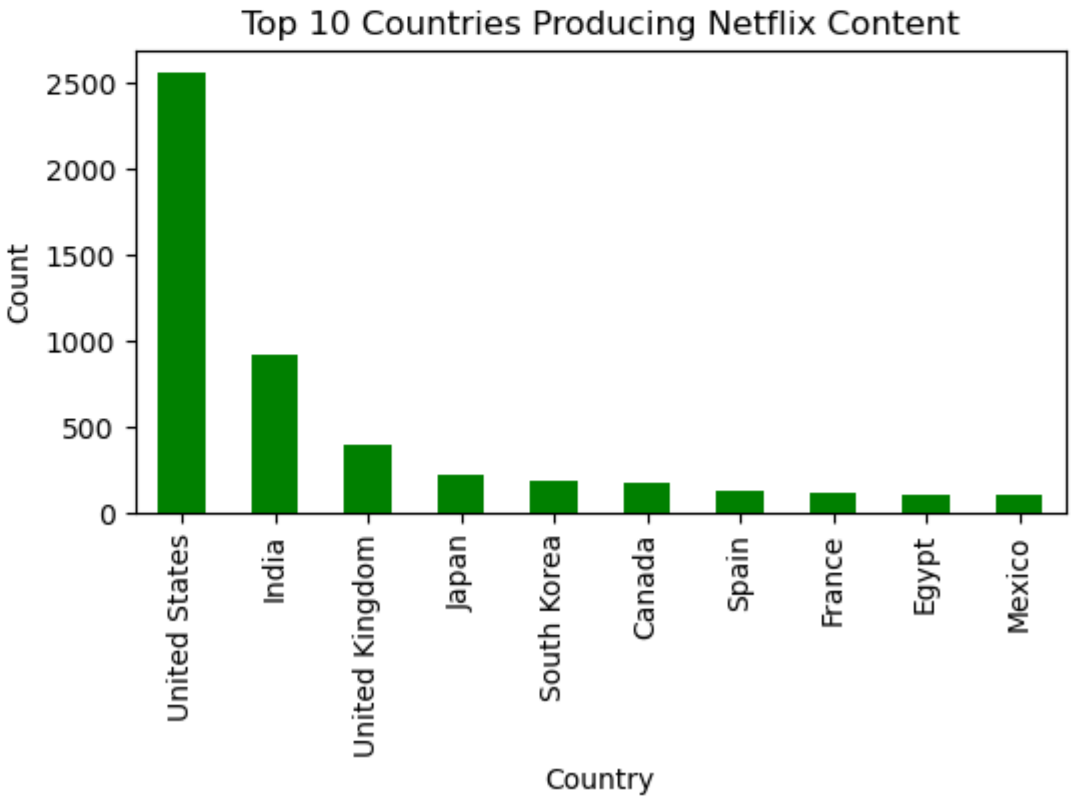
```
C:\Users\Dell\AppData\Local\Temp\ipykernel_108\3877449300.py:1: UserWarning:
Could not infer format, so each element will be parsed individually, falling
back to `dateutil`. To ensure parsing is consistent and as-expected, please
specify a format.
  df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
```



Top 10 Content-Producing Countries

In [120…

```
plt.figure(figsize=(6,3))
df['country'].value_counts().head(10).plot(kind='bar', color='green
')
plt.title("Top 10 Countries Producing Netflix Content")
plt.xlabel("Country")
plt.ylabel("Count")
plt.show()
```

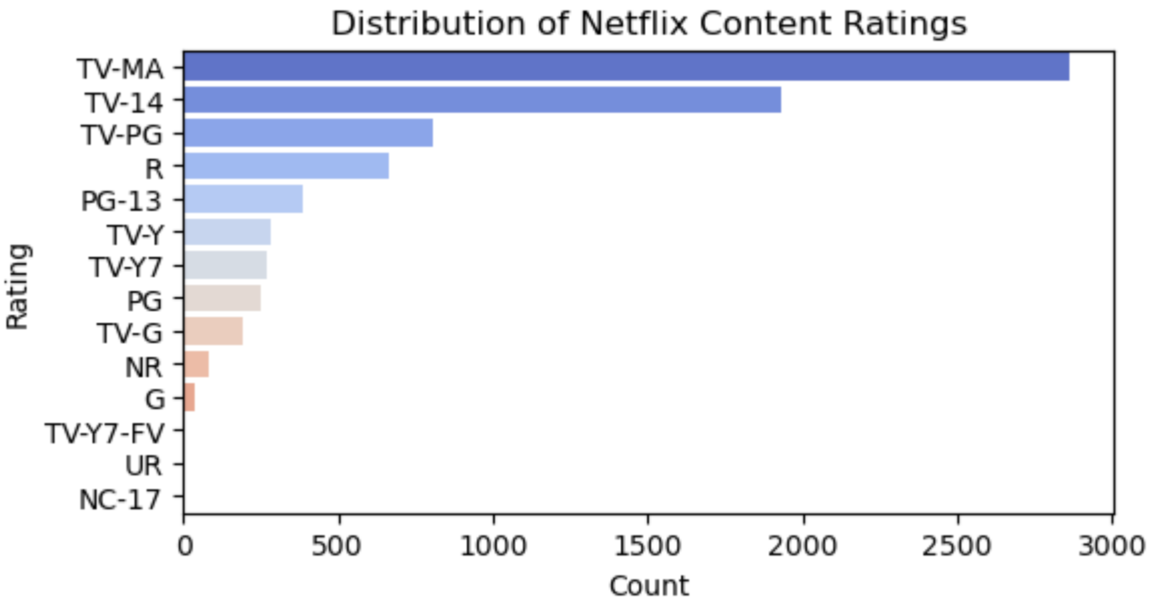


Rating Distribution Analysis

In [122…

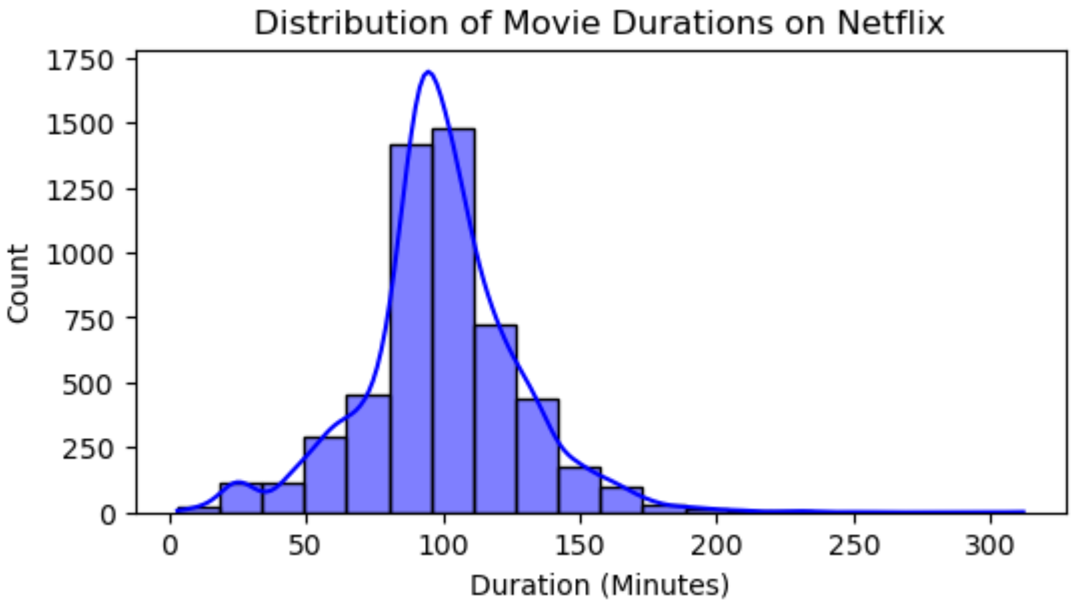
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(6, 3))
sns.countplot(y=df['rating'], order=df['rating'].value_counts().index, palette="coolwarm")

plt.title("Distribution of Netflix Content Ratings")
plt.xlabel("Count")
plt.ylabel("Rating")
plt.show()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_108\3227217918.py:5:
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.countplot(y=df['rating'], order=df['rating'].value_counts().index,
palette="coolwarm")
```



Duration Analysis for Movies

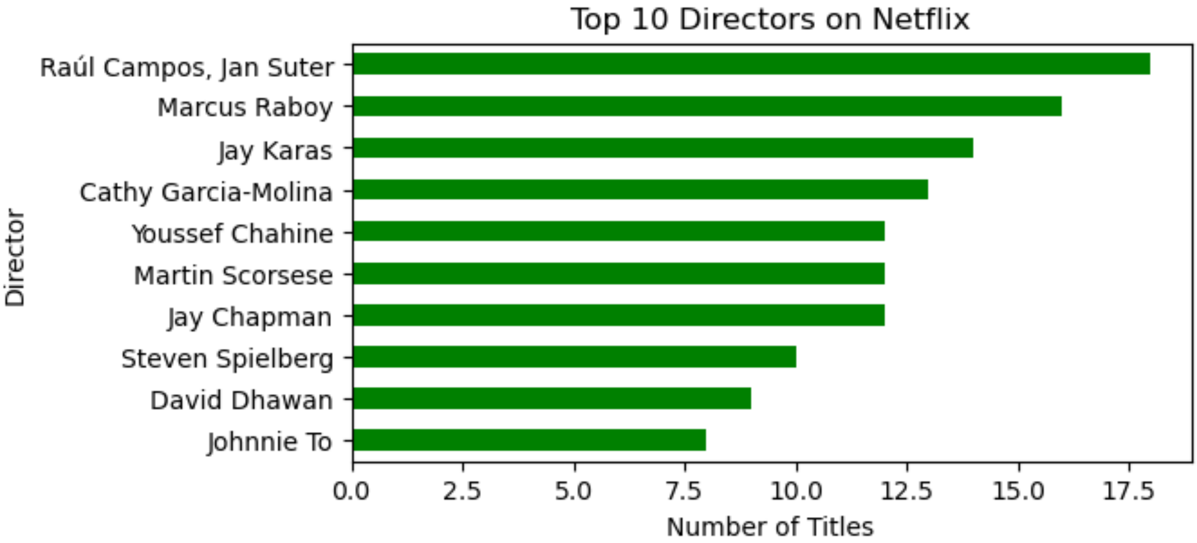
```
In [124… df_movies = df[df['type'] == 'Movie'].copy()  # Filter only movies
         df_movies['duration'] = df_movies['duration'].astype(str).str.repl
         ace(" min", "").astype(float)  # Convert to numeric

         plt.figure(figsize=(6, 3))
         sns.histplot(df_movies['duration'], bins=20, kde=True,
         color='blue')

         plt.title("Distribution of Movie Durations on Netflix")
         plt.xlabel("Duration (Minutes)")
         plt.ylabel("Count")
         plt.show()
```



Top 10 Directors with Most Content on Netflix

In [126…

```
plt.figure(figsize=(6, 3))
df['director'].value_counts().head(10).plot(kind='barh', color='green')

plt.title("Top 10 Directors on Netflix")
plt.xlabel("Number of Titles")
plt.ylabel("Director")
plt.gca().invert_yaxis()  # Invert y-axis for better readability
plt.show()
```


Top 10 Directors on Netflix
Director
Raúl Campos, Jan Suter
Marcus Raboy
Jay Karas
Cathy Garcia-Molina
Youssef Chahine
Martin Scorsese
Jay Chapman
Steven Spielberg
David Dhawan
Johnnie To
Number of Titles