Netflix Data Analysis

Importing Libraries

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt

%matplotlib inline
```

Loading the dataset

In [3]:

```
df=pd.read_csv(r'C:\Users\shweta\Downloads\archive (19)\netflix_titles.csv')
```

Data overview

In [4]:

```
df.head(2)
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA

Shape of the dataset

In [5]:

```
df.shape
```

Out[5]:

```
(8807, 12)
```

In [6]:

```
df.size
```

Out[6]:

```
105684
```

Columns of the dataset

In [7]:

```
df.columns
```

Out[7]:

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_a
dded',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

In [8]:

```
df.dtypes
```

Out[8]:

```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

Information of the dataset

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
show_id         8807 non-null object
type            8807 non-null object
title           8807 non-null object
director        6173 non-null object
cast            7982 non-null object
country         7976 non-null object
date_added      8797 non-null object
release_year    8807 non-null int64
rating          8803 non-null object
duration        8804 non-null object
listed_in       8807 non-null object
description     8807 non-null object
dtypes: int64(1), object(11)
memory usage: 825.7+ KB
```

Duplicates values

In [10]:

```
df[df.duplicated()]
```

Out[10]:

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	li

Number of null values per coloumn

In [11]:

```
df.isnull().sum()
```

Out[11]:

```
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
```

Data Cleansing

Fill in missing values

In [12]:

```
df.director.fillna('None',inplace=True)
df.cast.fillna('None',inplace=True)
df.country.fillna('None',inplace=True)
```

Dropping missing values

In [13]:

```
df.dropna(subset=['date_added','rating'],inplace=True)
```

In [14]:

```
df.isnull().sum()
```

Out[14]:

```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        3
listed_in       0
description     0
dtype: int64
```

Converting data type

In [15]:

```
df['release_date']= pd.to_datetime(df['date_added'])
```

In [16]:

```
df.dtypes
```

Out[16]:

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

In [17]:

```
df['release_date'].dt.year.value_counts()
```

Out[17]:

```
2019    2016
2020    1879
2018    1648
2021    1498
2017    1186
2016     428
2015      82
2014      24
2011      13
2013      11
2012       3
2009       2
2008       2
2010       1
Name: release_date, dtype: int64
```

Exploratory Data Analysis and Visualization

Movies & TV shows Ratings

In [52]:

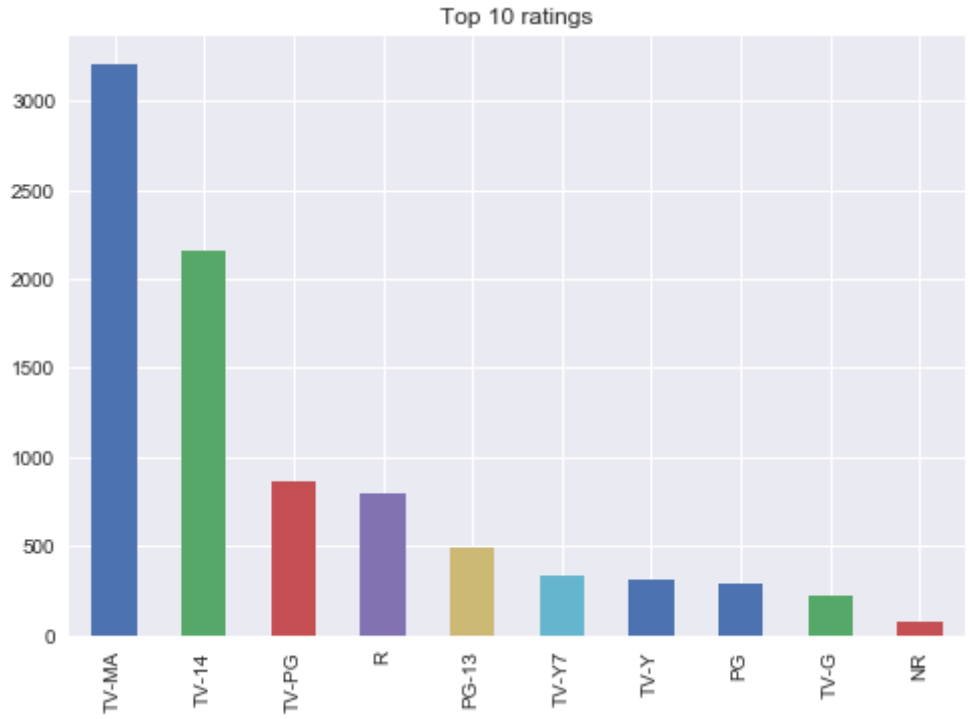
```
plt.style.use('seaborn')
```

In [53]:

```
df.rating.value_counts().head(10).plot(kind='bar',title='Top 10 ratings')
```

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a442e056a0>
```



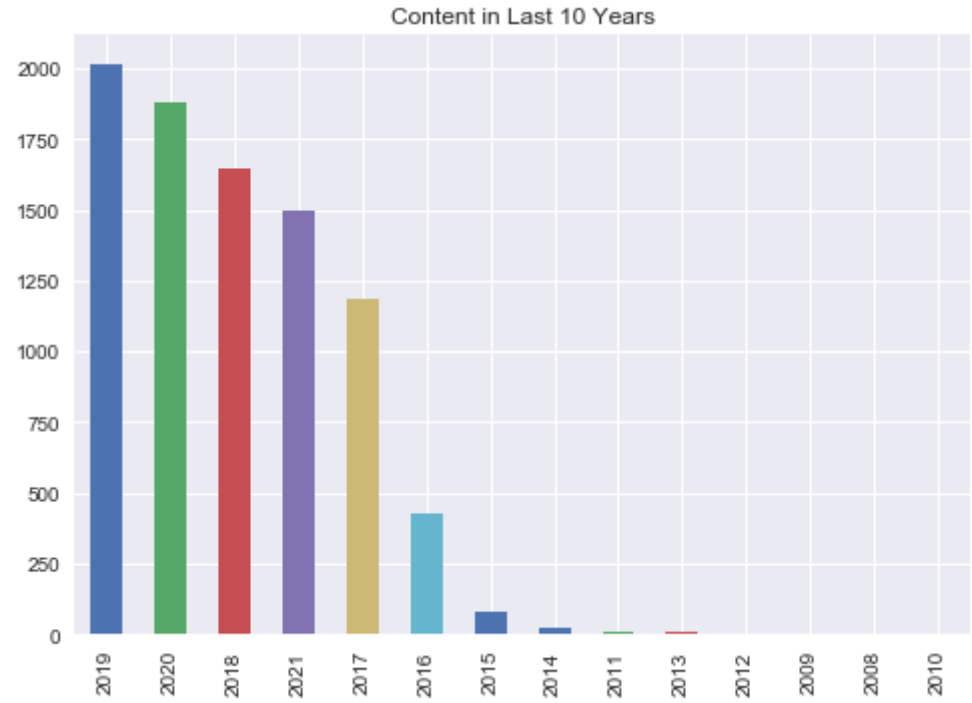
Number of Content in last 10 Years

In [54]:

```
df['release_date'].dt.year.value_counts().plot(kind='bar',title='Content in Last 10 Year
```

Out[54]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a442d2eac8>
```



Number of Movies & TV Shows

In [20]:

```
df.groupby('type').type.count()
```

Out[20]:

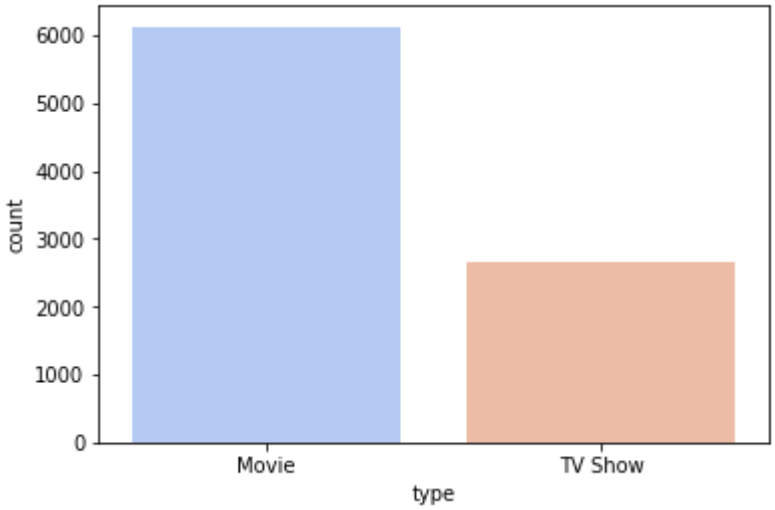
```
type
Movie      6129
TV Show    2664
Name: type, dtype: int64
```

In [30]:

```
sns.countplot(df['type'],palette='coolwarm')
```

Out[30]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a440e1de80>
```



In [ ]:

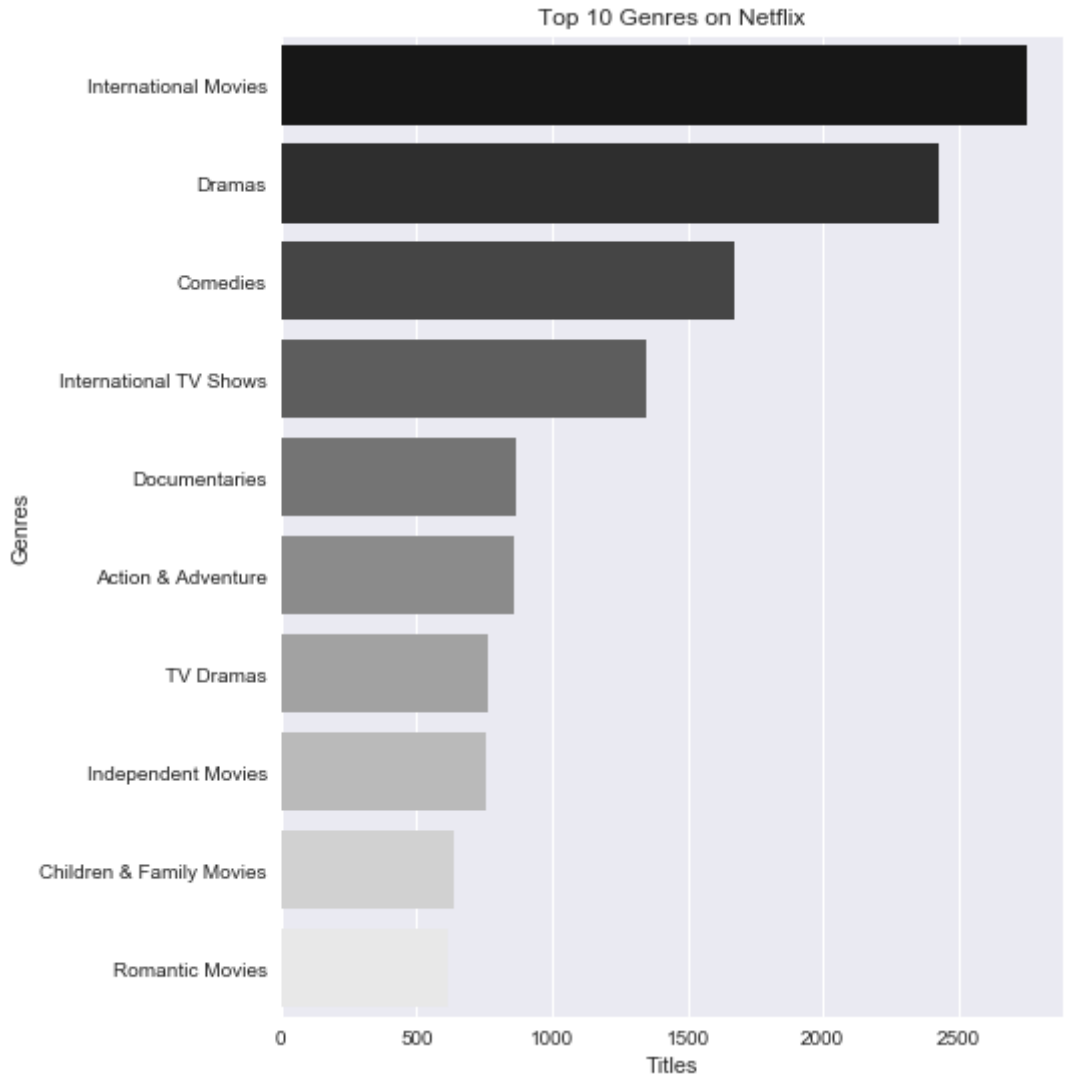
```
df['year'] = df['release_date'].dt.year
```

Top 10 Genres by number of title

In [55]:

```
filtered_genres = df.set_index('title').listed_in.str.split(', ', expand=True).stack();

plt.figure(figsize=(7,9))
g = sns.countplot(y = filtered_genres, order=filtered_genres.value_counts().index[:10],p
plt.title('Top 10 Genres on Netflix')
plt.xlabel('Titles')
plt.ylabel('Genres')
plt.show()
```

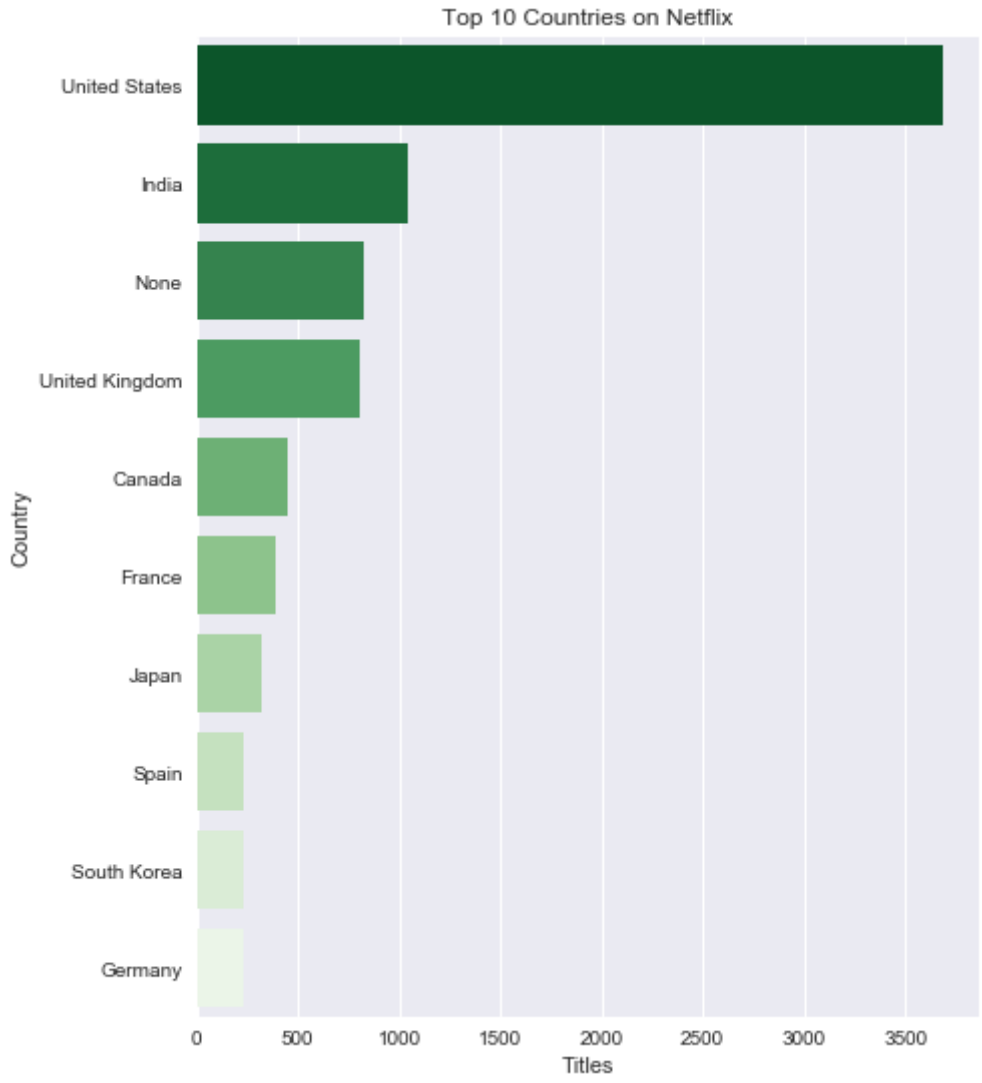


Top 10 countries by title

In [56]:

```
filtered_countries = df.set_index('title').country.str.split(', ', expand=True).stack()


plt.figure(figsize=(7,9))
g = sns.countplot(y = filtered_countries, order=filtered_countries.value_counts().index
plt.title('Top 10 Countries on Netflix')
plt.xlabel('Titles')
plt.ylabel('Country')
plt.show()
```

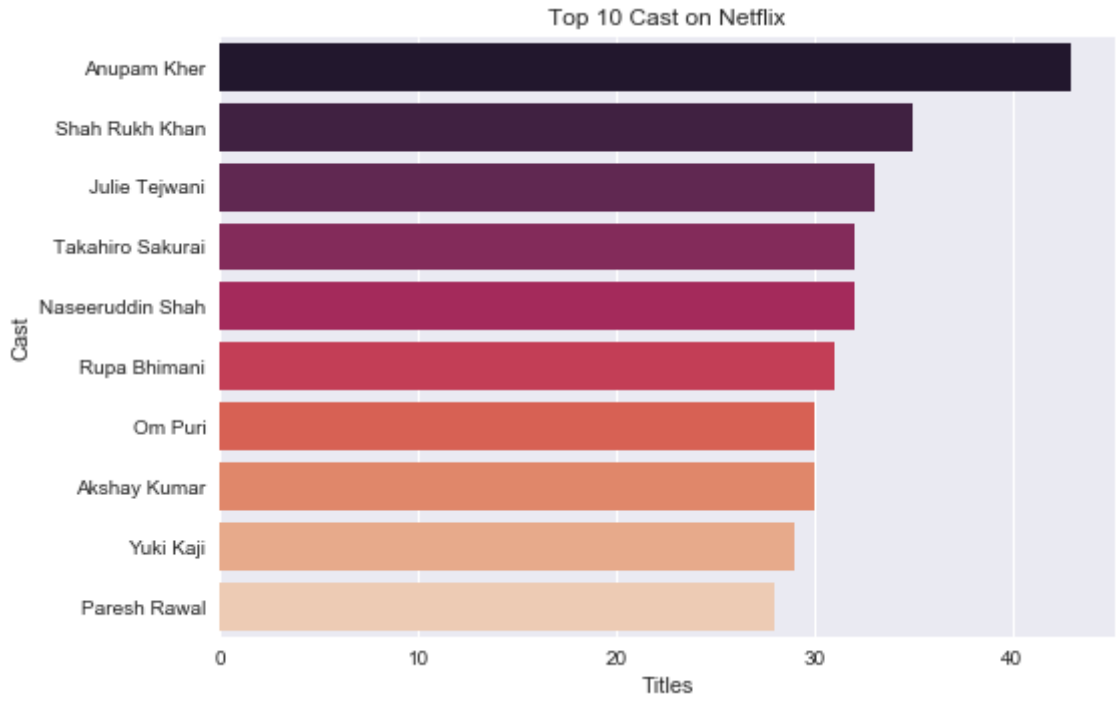


Top 10 cast by number of titles

In [57]:

```
filtered_cast = df[df.cast!= 'None'].set_index('title').cast.str.split(', ', expand=True
sns.countplot(y = filtered_cast, order=filtered_cast.value_counts().index[:10], palette=
plt.title('Top 10 Cast on Netflix')
plt.xlabel('Titles')
plt.ylabel('Cast')


plt.show()
```



Top 10 directors by number of title

In [58]:

```
filtered_director = df.set_index('title').director.str.split(', ', expand=True).stack()
filtered_director = filtered_director[filtered_director != 'None']


plt.figure(figsize=(5,7))
g = sns.countplot(y =filtered_director , order=filtered_director.value_counts().index[:1
plt.title('Top 10 directors on Netflix')
plt.xlabel('')
plt.ylabel('')
plt.show()
```