```
from google.colab import files
uploaded=files.upload()
```

Choose files netflix_titles.csv

- **netflix_titles.csv**(text/csv) - 3399671 bytes, last modified: 21/09/2024 - 100% done

Saving netflix_titles.csv to netflix_titles (1).csv

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


df=pd.read_csv('netflix_titles.csv')
df
```

	show_id	type	title	director	cast	country	date_added	release_yea
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	202
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	202
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	202
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	202
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam	India	September 24, 2021	202

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.head(5)
```

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

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.tail(5)
```

	show_id	type	title	director	cast	country	date_added	release_yea
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	200
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	201

```
df1=df.dropna()
df1
```

	show_id	type	title	director	cast	country	date_added	release_
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	
8	s9	TV Show	The Great British Baking Show	Andy Devonshire	Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho...	United Kingdom	September 24, 2021	
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	
12	s13	Movie	Je Suis Karl	Christian Schwochow	Luna Wedler, Jannis Niewöhner, Milan Peschel, ...	Germany, Czech Republic	September 23, 2021	
					Prashanth, Aishwarya Rai		Septembe	

Next steps: Generate code with df1 | View recommended plots | New interactive sheet

```
df.info
```

```
pandas.core.frame.DataFrame.info
def info(verbose: bool | None=None, buf: WriteBuffer[str] | None=None, max_cols:
int | None=None, memory_usage: bool | str | None=None, show_counts: bool |
None=None) -> None

/usr/local/lib/python3.10/dist-packages/pandas/core/frame.py
Print a concise summary of a DataFrame.

This method prints information about a DataFrame including
the index dtype and columns, non-null values and memory usage.
```

```
df.describe
```

pandas.core.generic.NDFrame.describe
```
def describe(percentiles=None, include=None, exclude=None) -> Self
```

/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py
Generate descriptive statistics.

Descriptive statistics include those that summarize the central
tendency, dispersion and shape of a
dataset's distribution, excluding ``NaN`` values.

```
df.columns
```

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

```
df.shape
```

```
(8807, 12)
```

```
df["type"].value_counts()
```

type	count
Movie	6131
TV Show	2676

dtype: int64

```
df["release_year"].value_counts()
```

release_year	count
2018	1147
2017	1032
2019	1030
2020	953
2016	902
...	...
1959	1
1925	1
1961	1
1947	1
1966	1

74 rows × 1 columns

dtype: int64

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

	0
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

```
df['rating']=df['rating'].fillna('TV-MA')
df['rating']
```

	rating
0	PG-13
1	TV-MA
2	TV-MA
3	TV-MA
4	TV-MA
...	...
8802	R
8803	TV-Y7
8804	R
8805	PG
8806	TV-14

8807 rows × 1 columns

dtype: object

```
df['country']=df['country'].fillna('United States')
df['country']
```

	country
0	United States
1	South Africa
2	United States
3	United States
4	India
...	...
8802	United States
8803	United States
8804	United States
8805	United States
8806	India

8807 rows × 1 columns

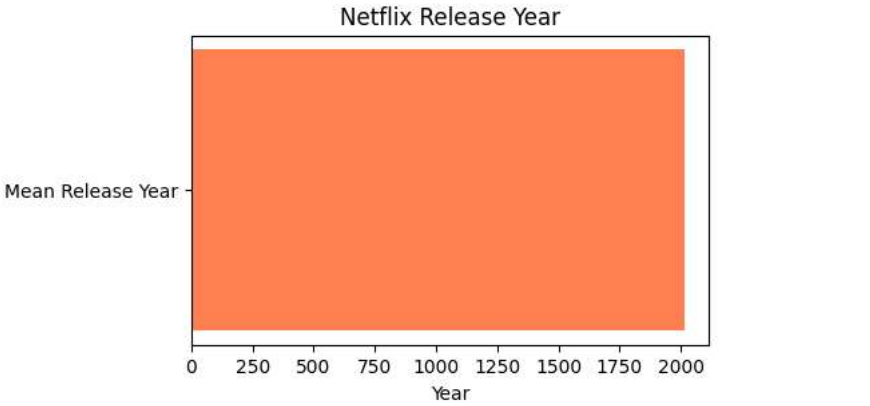
dtype: object

```
df2=df.rename(columns={"listed_in":"genre"})
df2
```

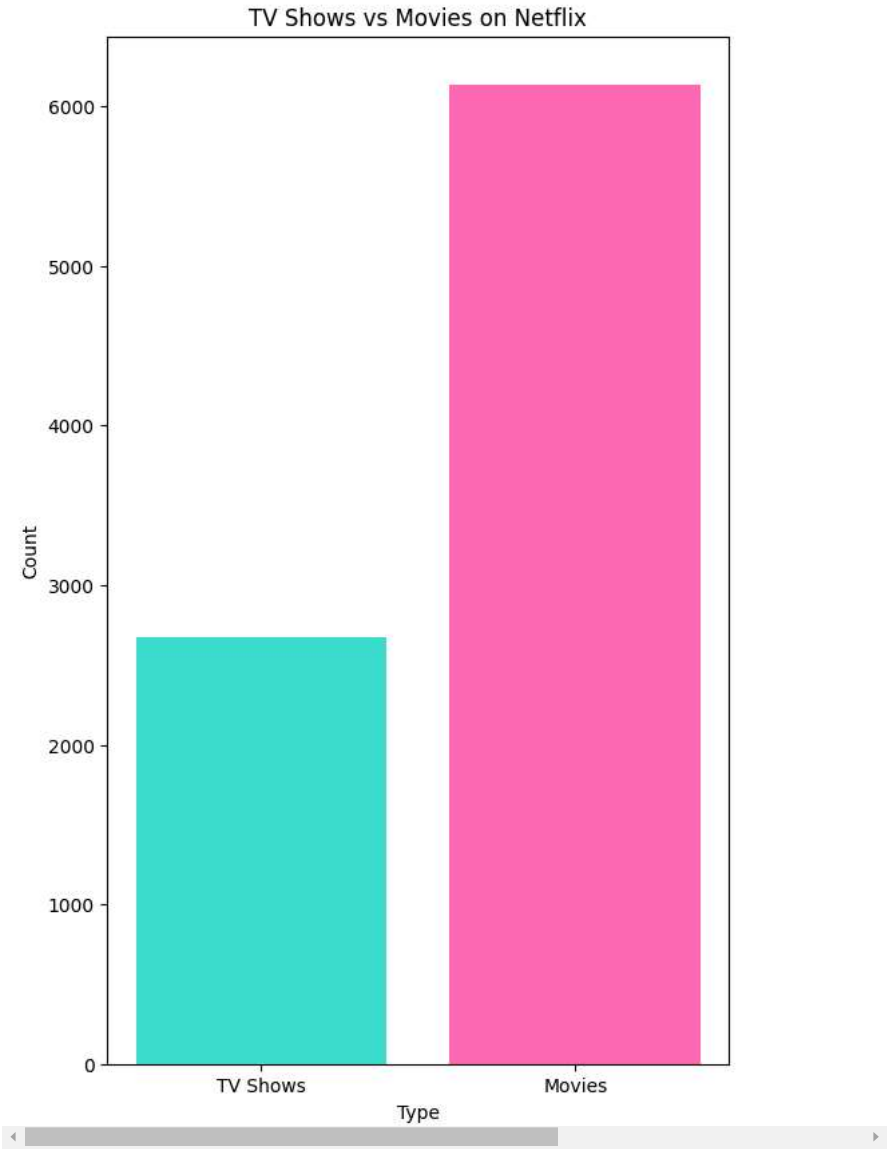
	show_id	type	title	director	cast	country	date_added	release_yea
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	202
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	202
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	202
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	202
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam	India	September 24, 2021	202

Next steps: Generate code with df2 | View recommended plots | New interactive sheet

```
mean_release_year=df['release_year'].mean()

plt.figure(figsize=(5,3))
plt.barh(['Mean Release Year'],[mean_release_year],color='coral')
plt.xlabel('Year')
plt.title('Netflix Release Year')
plt.show()
```




```
df = pd.read_csv('netflix_titles.csv')
tv_shows_count = len(df[df['type'] == 'TV Show'])
movies_count = len(df[df['type'] == 'Movie'])

plt.figure(figsize=(6, 10))
plt.bar(['TV Shows', 'Movies'], [tv_shows_count, movies_count], color=['turquoise', 'hotpi
plt.xlabel('Type')
plt.ylabel('Count')
plt.title('TV Shows vs Movies on Netflix')
plt.show()
```

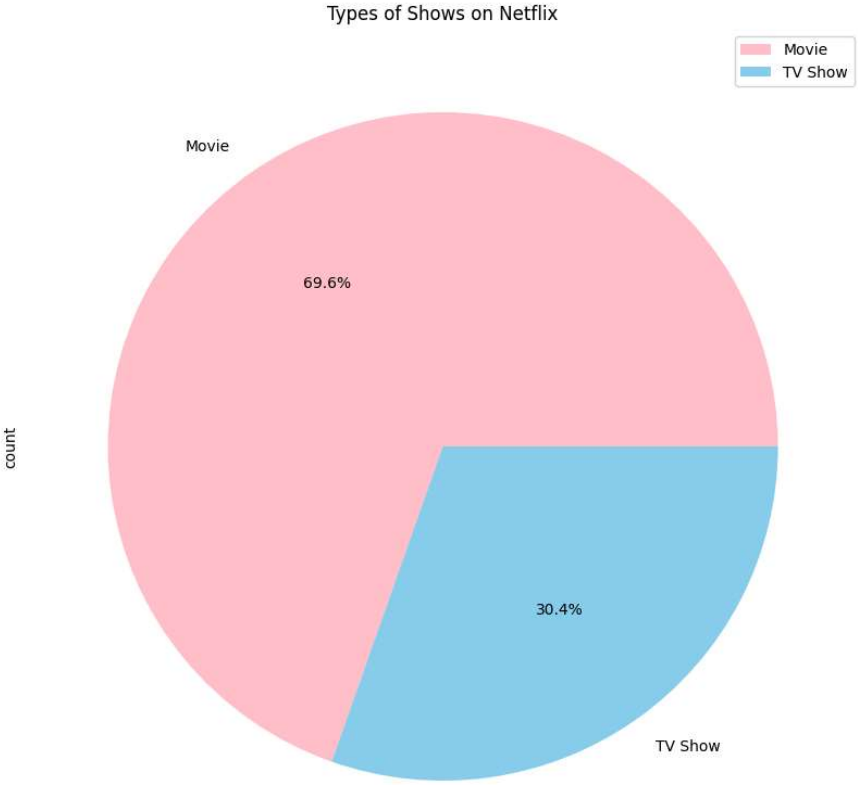



```
plt.title("Types of Shows on Netflix")
df["type"].value_counts().plot.pie(autopct='%1.1f%%',figsize=(10,10),colors=['pink','skybl
plt.legend()
plt.show
```

matplotlib.pyplot.show
def show(*args, **kwargs)

/usr/local/lib/python3.10/dist-packages/matplotlib/pyplot.py
Display all open figures.

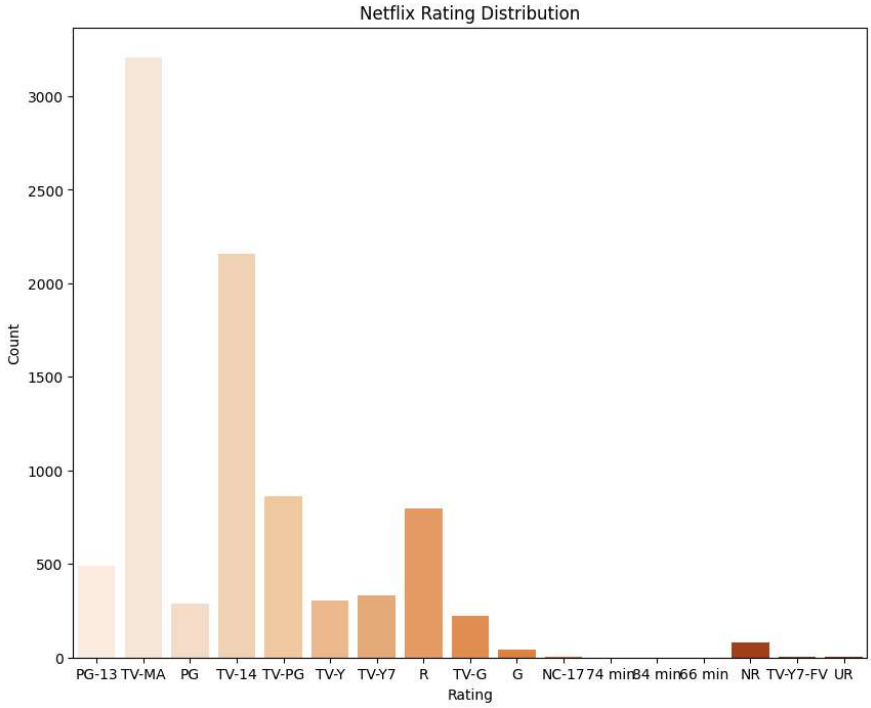
Parameters
----
block : bool, optional



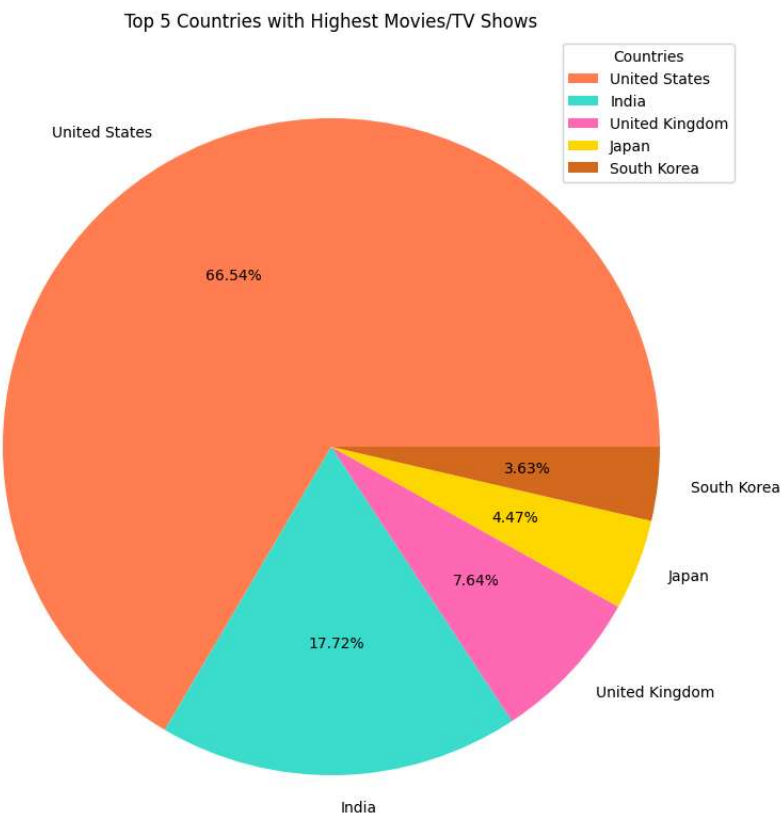

```
df = pd.read_csv('netflix_titles.csv')
plt.figure(figsize=(10, 8))
sns.countplot(x='rating', data=df,palette='Oranges')
```

```
plt.title('Netflix Rating Distribution')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.show()
```

```
<ipython-input-13-288342266359>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0

  sns.countplot(x='rating', data=df,palette='Oranges')
```

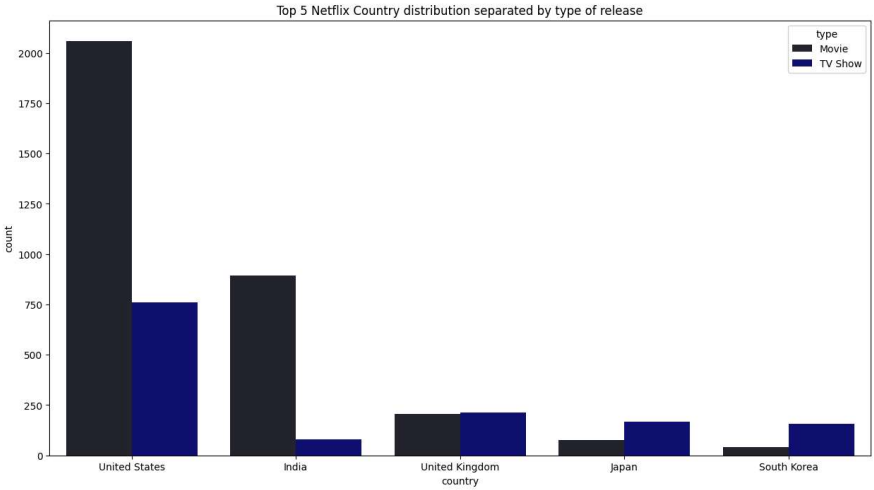



```
country_counts = df['country'].value_counts().head(5)

plt.figure(figsize=(10, 10))
plt.pie(country_counts.values, labels=country_counts.index, autopct='%1.2f%%',colors=['cora
plt.title('Top 5 Countries with Highest Movies/TV Shows')
plt.legend(title="Countries")
plt.show()
```




```
plt.figure(figsize=(15,8))
plt.title('Top 5 Netflix Country distribution separated by type of release')
sns.countplot(x='country',data=df,hue='type',order=df.country.value_counts().iloc[:5].inde
plt.show()
```

```
<ipython-input-68-f6ea7f06c667>:3: FutureWarning:

Setting a gradient palette using color= is deprecated and will be removed in v0.14.0.

  sns.countplot(x='country',data=df,hue='type',order=df.country.value_counts().iloc[:5
```




```
df['genre'] = df2['genre'].str.split(',')
df = df.explode('genre')

movies_df = df2[df2['type'] == 'Movie']
tv_shows_df = df2[df2['type'] == 'TV Show']

top_10_movie_genres = movies_df['genre'].value_counts().head(10)

top_10_tv_genres = tv_shows_df['genre'].value_counts().head(10)

fig, axes = plt.subplots(2, 1, figsize=(10,10))
```

```
top_10_movie_genres.plot(kind='bar', ax=axes[0], color='chartreuse')
axes[0].set_title('Top 10 Movie Genres on Netflix')
axes[0].set_xlabel('Genre')
axes[0].set_ylabel('Count')
axes[0].set_xticklabels(top_10_movie_genres.index, rotation=45)

top_10_tv_genres.plot(kind='bar', ax=axes[1], color='chocolate')
axes[1].set_title('Top 10 TV Show Genres on Netflix')
axes[1].set_xlabel('Genre')
axes[1].set_ylabel('Count')
axes[1].set_xticklabels(top_10_tv_genres.index, rotation=45)

plt.tight_layout()
plt.show()
```