

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

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

netflix_titles.csv

netflix_titles.csv(text/csv) - 3399671 bytes, last modified: 12/02/2026 - 100% done
Saving netflix_titles.csv to netflix_titles.csv

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

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  -
0   show_id         8807 non-null   object
1   type            8807 non-null   object
2   title           8807 non-null   object
3   director        6173 non-null   object
4   cast            7982 non-null   object
5   country         7976 non-null   object
6   date_added      8797 non-null   object
7   release_year    8807 non-null   int64
8   rating          8803 non-null   object
9   duration        8804 non-null   object
10  listed_in       8807 non-null   object
11  description     8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
## null values check
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
```

```
len(df)
```

```
8807
```

```
df.isnull().sum() * 100 / len(df)
```

```

0
show_id    0.000000
type       0.000000
title      0.000000
director   29.908028
cast       9.367549
country    9.435676
date_added 0.113546
release_year 0.000000
rating     0.045418
duration   0.034064
listed_in  0.000000
description 0.000000

```

dtype: float64

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
df.describe()
```

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

```

## fill the empty rows
df['director'].fillna('Not Available', inplace=True)
df['cast'].fillna('Not Available', inplace=True)
df['country'].fillna('Unknown', inplace=True)
df['rating'].fillna(df['rating'].mode()[0], inplace=True)
df['duration'].fillna(df['duration'].mode()[0], inplace=True)

```

/tmp/ipython-input-383530487.py:5: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]

```
df['duration'].fillna(df['duration'].mode()[0], inplace=True)
```

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

	0
show_id	0
type	0
title	0
director	0
cast	0
country	0
date_added	10
release_year	0
rating	0
duration	0
listed_in	0
description	0

dtype: int64

```
Total_Titles = len(df)
print("Total_Titles:",Total_Titles)
```

Total_Titles: 8807

```
Total_Movies = len(df[df['type']=='Movie'])
print ("Total_Movies:",Total_Movies)
```

Total_Movies: 6131

```
Total_TV_Shows = len(df[df['type']=='TV Show'])
print ("Total_TV_Shows:",Total_TV_Shows)
```

Total_TV_Shows: 2676

```
df['country'].value_counts().idxmax()
```

'United States'

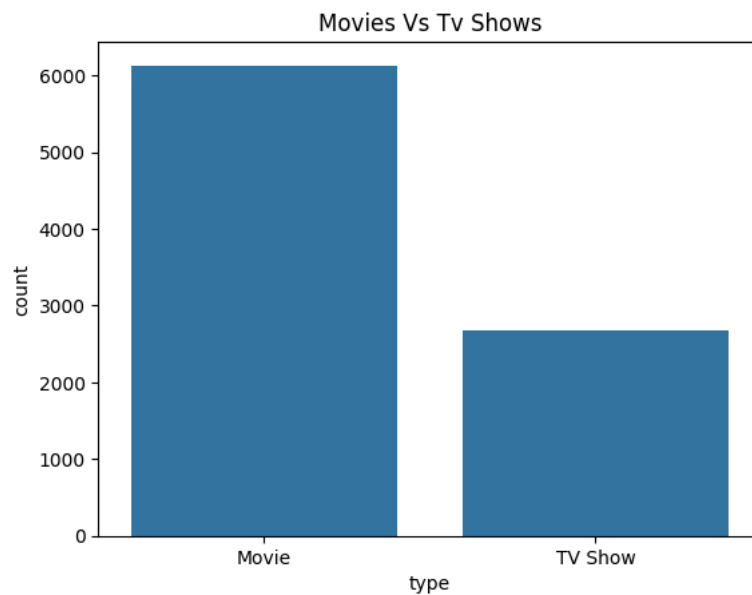
```
df['date_added'] = pd.to_datetime(df['date_added'], format='mixed')
df['year_added'] = df['date_added'].dt.year
```

```
### movies vs shows
df['type'].value_counts()
```

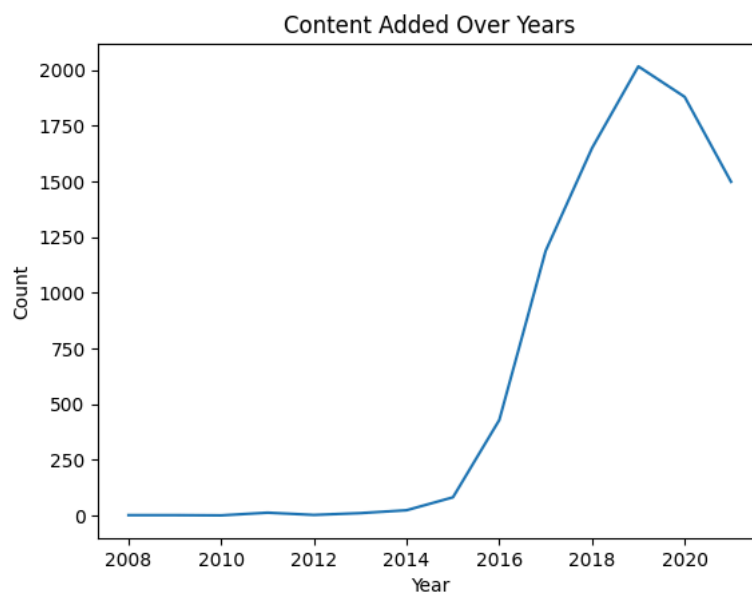
	count
type	
Movie	6131
TV Show	2676

dtype: int64

```
sns.countplot(x='type', data=df)
plt.title ("Movies Vs Tv Shows")
plt.show()
```



```
### content added per year
df['year_added'].value_counts().sort_index().plot(kind='line')
plt.title("Content Added Over Years")
plt.xlabel("Year")
plt.ylabel("Count")
plt.show()
```

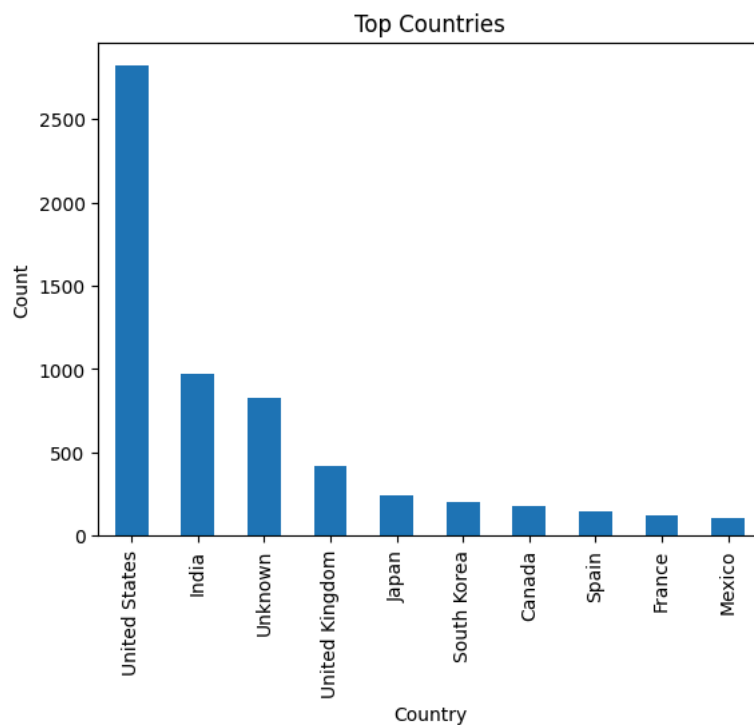


```
## top countries
top_countries = df['country'].value_counts().head(10)
print(top_countries)
```

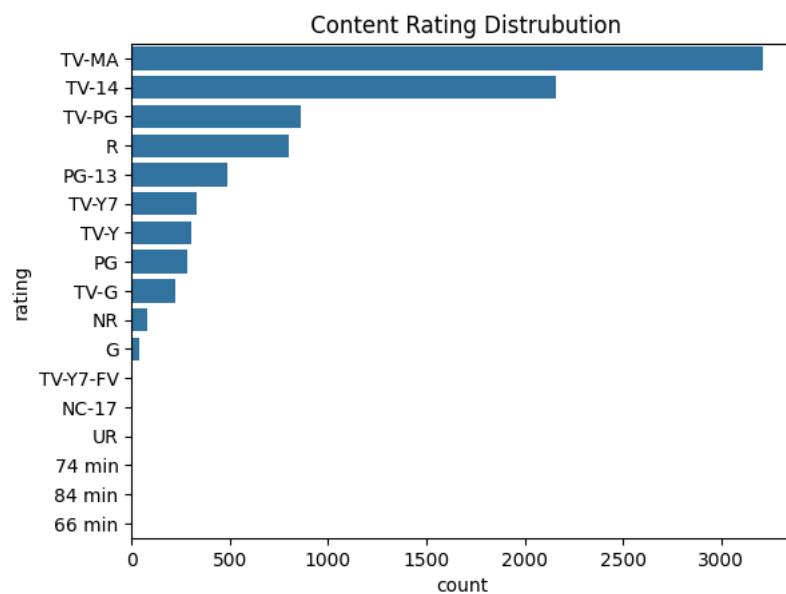
```
country
United States    2818
India            972
Unknown         831
United Kingdom  419
Japan           245
South Korea     199
Canada          181
Spain           145
France          124
Mexico          110
Name: count, dtype: int64
```

```
## top countries visualization
top_countries.plot(kind='bar')
plt.title("Top Countries")
plt.xlabel("Country")
plt.ylabel("Count")
```

Text(0, 0.5, 'Count')



```
## most common Rating
sns.countplot(y='rating',data=df, order=df['rating'].value_counts().index)
plt.title(" Content Rating Distrubution")
plt.show()
```



```
## top genres
df['listed_in'].value_counts().head(10)
```

	count
listed_in	
Dramas, International Movies	362

df['director'].value_counts().head(10)

Stand-Up Comedy	334
Comedies, Dramas, International Movies	274
Dramas, Independent Movies, International Movies	252
Not Available	2634
Rajiv Chilaka	19
Children & Family Movies	215
Raúl Campos, Jan Suter	18
Comedies	201
Suhas Kadav	16
Documentaries, International Movies	186
Marcus Raboy	16
omantic Movies	180
Jay Karas	14
Cathy Garcia-Molina	13
Martin Scorsese	12
Youssef Chahine	12
Jay Chapman	12

dtype: int64

```
movies = df[df['type'] == 'Movie'].copy()

# Filter out rows where 'duration' does not contain ' min' (e.g., '1 Season')
movies_filtered = movies[movies['duration'].str.contains(' min', na=False)].copy()

movies_filtered.loc[:, 'duration'] = movies_filtered['duration'].str.replace(' min', '')
movies_filtered.loc[:, 'duration'] = movies_filtered['duration'].astype(float)

movies_filtered['duration'].hist()
plt.title("Movie Duration Distribution")
plt.show()
```

