```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e

# Input data files are available in the read-only "../
# For example, running this (by clicking run or pressi

import os
for dirname, _, filenames in os.walk('/collab/input'):
    for filename in filenames:
        print(os.path.join(sample_data, netflix_titles


# Step 1: Upload the Dataset
from google.colab import files
uploaded = files.upload()

# Step 2: Read the Dataset
import pandas as pd
df = pd.read_csv("netflix_titles.csv", encoding='latin

# Step 3: Data Visualization
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go
import plotly.offline as py

# Initialize Plotly
py.init_notebook_mode(connected=True)

# Example Visualization: Distribution of Release Years
plt.figure(figsize=(10, 6))
sns.histplot(df['release_year'], bins=30, kde=True)
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.title('Distribution of Release Years')
plt.show()

# Example Visualization: Distribution of Content Types
content_types = df['type'].value_counts()
labels = content_types.index
values = content_types.values

fig = go.Figure(data=[go.Pie(labels=labels, values=val
fig.update_layout(title='Distribution of Content Types
py.iplot(fig)
```

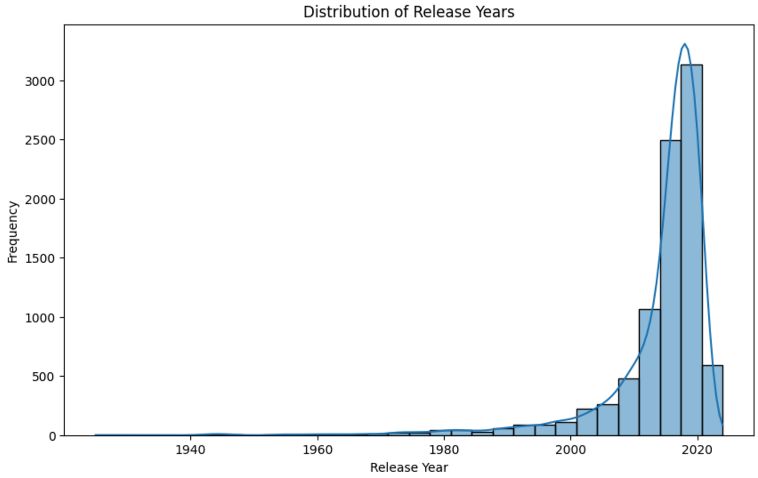
netflix netflix_titles.csv •••

1 to 10 of 8809 entries Filter

Choose Files netflix_titles.csv

- **netflix_titles.csv**(text/csv) - 3532881 bytes, last modified: 4/10/2024 - 100% done

```
Saving netflix_titles.csv to netflix_titles (1).cs
```



show_id	type	title	directo
s1	Movie	Dick Johnson Is Dead	Kirsten Johnson
s2	TV Show	Blood & Water	
s3	TV Show	Ganglands	Julien Leclercq

s4	TV Show	Jailbirds New Orleans	
s5	TV Show	Kota Factory	
s6	TV Show	Midnight Mass	Mike Flanagar

s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Lui Ucha
s8	Movie	Sankofa	Haile Gerima
s9	TV Show	The Great British Baking Show	Andy Devonsh
s10	Movie	The Starling	Theodor Melfi

Show 10 per page

1 2 10 100 800

880 881

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, AffinityPropagatio
```

Double-click (or enter) to edit

```
import pandas as pd

# Read the CSV file
df = pd.read_csv('netflix_titles.csv', encoding='latin

# Display the first few rows of the DataFrame
df.head()
```

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

5 rows × 26 columns

```
df = df[df.columns[:12]]
df.head()
```

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

Next steps: View recommended plots

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8809 entries, 0 to 8808
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8809 non-null   object
 1   type          8809 non-null   object
 2   title         8809 non-null   object
 3   director      6175 non-null   object
 4   cast          7984 non-null   object
 5   country       7978 non-null   object
 6   date_added    8799 non-null   object
 7   release_year  8809 non-null   int64 
 8   rating        8805 non-null   object
 9   duration      8806 non-null   object
 10  listed_in     8809 non-null   object
 11  description   8809 non-null   object
dtypes: int64(1), object(11)
memory usage: 826.0+ KB
```

```
# Check for missing values
df.isnull().sum()
```

```
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
```

```
# Replacments

df['country'] = df['country'].fillna(df['country'].mod


df['cast'].replace(np.nan, 'No Data',inplace  = True)
df['director'].replace(np.nan, 'No Data',inplace  = Tr

# Drops

df.dropna(inplace=True)

# Drop Duplicates

df.drop_duplicates(inplace= True)
```

```
# We need to use the strip module first because some v
df["date_added"] = df["date_added"].str.strip()

# convert dtype to datetime
df["date_added"] = pd.to_datetime(df['date_added'])

# extract month and year
df['month_added']=df['date_added'].dt.month_name()
df['year_added'] = df['date_added'].dt.year
```

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

release_year	rating	duration	listed_in	desc
2020	PG-13	90 min	Documentaries	As h r e life
2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	p party
2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To pr fami c
2021	TV-MA	1 Season	Docuseries, Reality TV	flirtat toile dow
2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In (I

Next steps: View recommended plots

```
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
listed_in       0
description     0
month_added     0
```

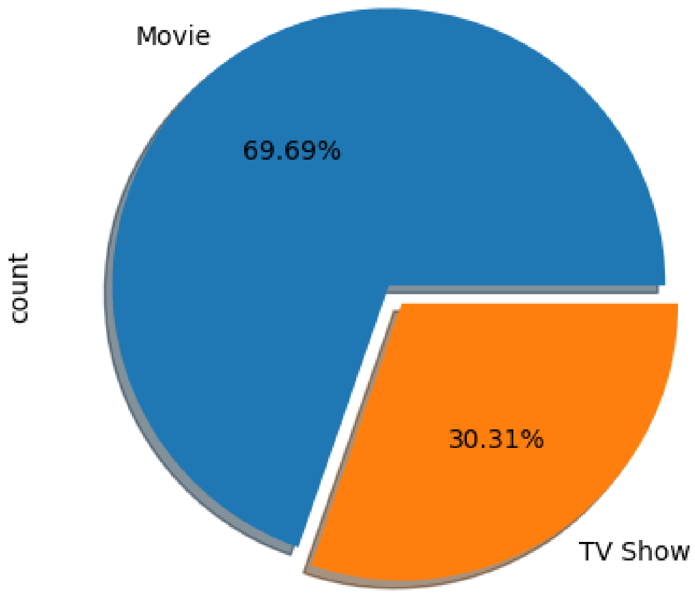
```
year_added      0
dtype: int64
```

```
df.info()
```

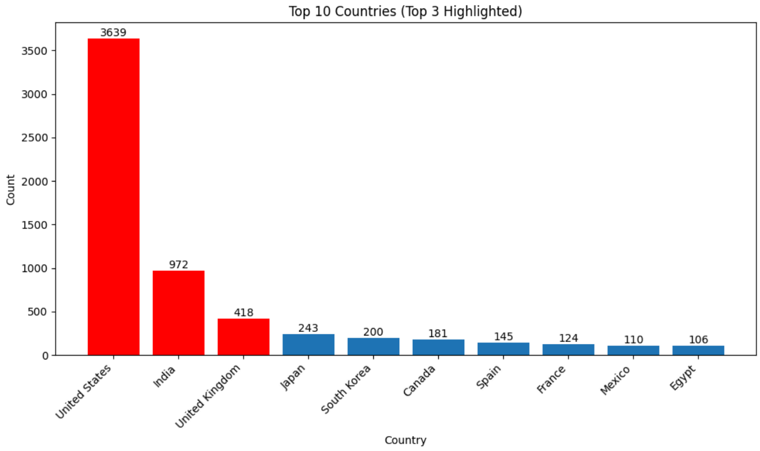
```
<class 'pandas.core.frame.DataFrame'>
Index: 8792 entries, 0 to 8808
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8792 non-null   object
 1   type          8792 non-null   object
 2   title         8792 non-null   object
 3   director      8792 non-null   object
 4   cast          8792 non-null   object
 5   country       8792 non-null   object
 6   date_added    8792 non-null   datetime64[ns]
 7   release_year  8792 non-null   int64
 8   rating        8792 non-null   object
 9   duration      8792 non-null   object
 10  listed_in     8792 non-null   object
 11  description   8792 non-null   object
 12  month_added   8792 non-null   object
 13  year_added    8792 non-null   int32
dtypes: datetime64[ns](1), int32(1), int64(1), obj
memory usage: 996.0+ KB
```

```
#
# Create our pie chart with labels

df["type"].value_counts().plot.pie(autopct='%1.2f%%',ex
```

```
<Axes: ylabel='count'>
```



```
country_counts = df['country'].value_counts().head(10)

# Create the bar chart
plt.figure(figsize=(10, 6))  # Adjust figure size as des
bars = plt.bar(country_counts.index, country_counts.valu

# Add count values on top of bars
for bar, count in zip(bars, country_counts.values):
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, yval + (

# Highlight top 3 countries
plt.bar(country_counts.index[:3], country_counts.values

# Customize the plot
plt.xlabel('Country')
plt.ylabel('Count')
plt.title('Top 10 Countries (Top 3 Highlighted)')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis lab
plt.tight_layout()

plt.show()
```



```
# Count movies and TV shows per country
movie_counts_country = df[df['type'] == 'Movie']['coun
tv_show_counts_country = df[df['type'] == 'TV Show']['

# Combine counts into a single DataFrame with total (u
df_counts = pd.DataFrame({'Movie': movie_counts_countr
df_counts['total_by_country'] = df_counts.sum(axis=1)

# Sort by total count in descending order and select t
top_10_counts = df_counts.sort_values(by='total_by_cou

# Print the top 10 countries with movie, TV show, and
print(top_10_counts)
```

```
                 Movie  TV Show  total_by_country
country
United States   2495.0   1144.0            3639.0
India            893.0     79.0             972.0
United Kingdom   206.0    212.0             418.0
Japan             76.0    167.0             243.0
South Korea       41.0    159.0             200.0
Canada           122.0     59.0             181.0
Spain             97.0     48.0             145.0
France            75.0     49.0             124.0
Mexico            70.0     40.0             110.0
Egypt             92.0     14.0             106.0
```

```
# Next, we will compare between Movie and TV Show for
rows, cols = 2, 5
fig, axes = plt.subplots(rows, cols, figsize=(16, 6))

# Counter to keep track of subplot position
counter = 0

# Loop through each row (country) in the DataFrame
for country, row in top_10_counts.iterrows():
  # Extract movie, tv show, and total counts
  movie_count = row['Movie']
  tv_show_count = row['TV Show']
  total_count = row['total_by_country']

  # Create labels for pie chart slices
  labels = ['Movie', 'TV Show']

  # Create pie chart slice sizes
  sizes = [movie_count, tv_show_count]

  # Select the current subplot based on counter
  ax = axes[counter // cols, counter % cols]

  # Create a pie chart on the selected subplot
  ax.pie(sizes, labels=labels, autopct="%1.1f%%", expl
  ax.set_title(country)

  # Increase counter for next subplot position
  counter += 1

fig.text(-0.28, 0.93, 'Insight', fontsize=15, fontweig

fig.text(-0.28, 0.44, '''
Interestingly, Netflix in India
is made up nearly entirely of Movies.

Bollywood is big business, and perhaps
the main focus of this industry is Movies
and not TV Shows.

South Korean Netflix on the other hand is
almost entirely TV Shows.

The underlying resons for the difference
in content must be due to market research
conducted by Netflix.
'''
         , fontsize=12, fontweight='light', fontfamily


# Adjust layout to prevent overlapping elements
plt.tight_layout()
plt.show()
```

```
# Next, We will check rating of content.¶
# Count movies and TV shows per country
movie_counts_rating = df[df['type'] == 'Movie']['rating
```