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


url = "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv"
df = pd.read_csv(url)

df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More,						International	In a city of

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

checking shape, attribute and type of each row for dataframe

```
print (df.shape)
print(df.info())
print(df.describe())
```

```
(8807, 12)
<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
None
       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
```

*null values and converting type to datetime *

```
df.fillna('Unknown', inplace=True)
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
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      8807 non-null   object
 4   cast          8807 non-null   object
 5   country       8807 non-null   object
 6   date_added    0 non-null      datetime64[ns]
 7   release_year  8807 non-null   object
 8   rating        8807 non-null   object
 9   duration      8807 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: datetime64[ns](1), object(11)
memory usage: 825.8+ KB
<ipython-input-208-022d00fc5576>:1: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise an error in a
  df.fillna('Unknown', inplace=True)
<ipython-input-208-022d00fc5576>:2: UserWarning: Could not infer format, so each element will be parsed individually, falling back t
  df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
```

*Director names having highest content *

```
df.groupby('director')['title'].count().sort_values(ascending=False).head(10)
```

director	title
Unknown	2634
Rajiv Chilaka	19
Raúl Campos, Jan Suter	18
Marcus Raboy	16
Suhas Kadav	16
Jay Karas	14
Cathy Garcia-Molina	13
Martin Scorsese	12
Jay Chapman	12
Youssef Chahine	12

dtype: int64

converting list of countires to segregated format

```
country_df = df[["title", "country"]]
country_df["unnested_country"] = country_df ["country"].apply(lambda x: str(x).split(", "))
country_df = country_df.explode("unnested_country")
country_df.head(10)
```

```
<ipython-input-194-2552acf7a06d>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  country_df["unnested_country"] = country_df ["country"].apply(lambda x: str(x).split(", "))
```

	title	country	unnested_country
0	Dick Johnson Is Dead	United States	United States
1	Blood & Water	South Africa	South Africa
2	Ganglands	Unknown	Unknown
3	Jailbirds New Orleans	Unknown	Unknown
4	Kota Factory	India	India
5	Midnight Mass	Unknown	Unknown
6	My Little Pony: A New Generation	Unknown	Unknown
7	Sankofa	United States, Ghana, Burkina Faso, United Kin...	United States
7	Sankofa	United States, Ghana, Burkina Faso, United Kin...	Ghana
7	Sankofa	United States, Ghana, Burkina Faso, United Kin...	Burkina Faso

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

Converting cast in row wise segregated format

Converting country in rowwise

```
cast_df = df[["title", "cast"]]
cast_df["unnested_cast"] = cast_df["cast"].apply(lambda x: str(x).split(", "))
cast_df = cast_df.explode("unnested_cast")
cast_df.head(10)
```

```
<ipython-input-195-b18fdf5d30d7>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  cast_df["unnested_cast"] = cast_df["cast"].apply(lambda x: str(x).split(", "))
```

	title	cast	unnested_cast
0	Dick Johnson Is Dead	Unknown	Unknown
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Ama Qamata
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Khosi Ngema
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Gail Mabalane
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Thabang Molaba
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Dillon Windvogel
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Natasha Thahane
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Arno Greeff
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Xolile Tshabalala
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Getmore Sithole

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

Types of content and **counts**

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

type	count
Movie	6131
TV Show	2676

dtype: int64

*country having maxium TV shows and movies *

```
country_df['unnested_country'].value_counts().head(6) # top countires for which content created
```

unnested_country	count
United States	3689
India	1046
Unknown	831
United Kingdom	804
Canada	445
France	393

dtype: int64

Actor done most of content

```
cast_df['unnested_cast'].value_counts().head(10)   # actor who worked most time in TV shows and movies
```

unnested_cast	count
Unknown	825
Anupam Kher	43
Shah Rukh Khan	35
Julie Tejwani	33
Naseeruddin Shah	32
Takahiro Sakurai	32
Rupa Bhimani	31
Akshay Kumar	30
Om Puri	30
Yuki Kaji	29

dtype: int64

Types of geners

```
print(df['listed_in'].value_counts().head(5))
```

```
listed_in
Dramas, International Movies                        362
Documentaries                                       359
Stand-Up Comedy                                     334
Comedies, Dramas, International Movies              274
Dramas, Independent Movies, International Movies    252
Name: count, dtype: int64
```

```
df_country = df.assign(Country=df['country'].str.split(', ')).explode('country')
df_cast = df.assign(Cast=df['cast'].str.split(', ')).explode('cast')

df_country.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	C
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	
1	s2	TV Show	Blood & Water	Unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	[U
3	s4	TV Show	Jailbirds New Orleans	Unknown	Unknown	Unknown	2021-09-24	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...	[U
4	s5	TV Show	Kota Factory	Unknown	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...	

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

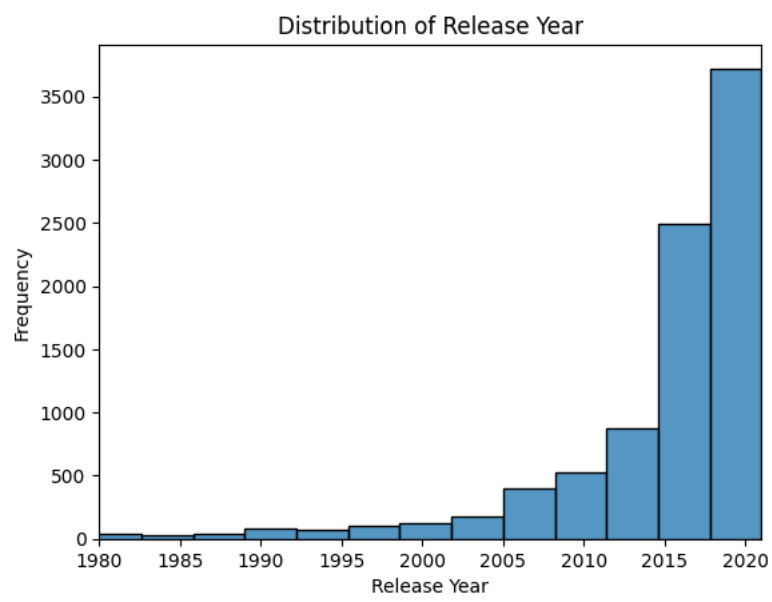
```
print(df['listed_in'].value_counts().head(5))
```

```
listed_in
Dramas, International Movies                        362
Documentaries                                       359
Stand-Up Comedy                                     334
Comedies, Dramas, International Movies              274
Dramas, Independent Movies, International Movies    252
Name: count, dtype: int64
```

Histogram

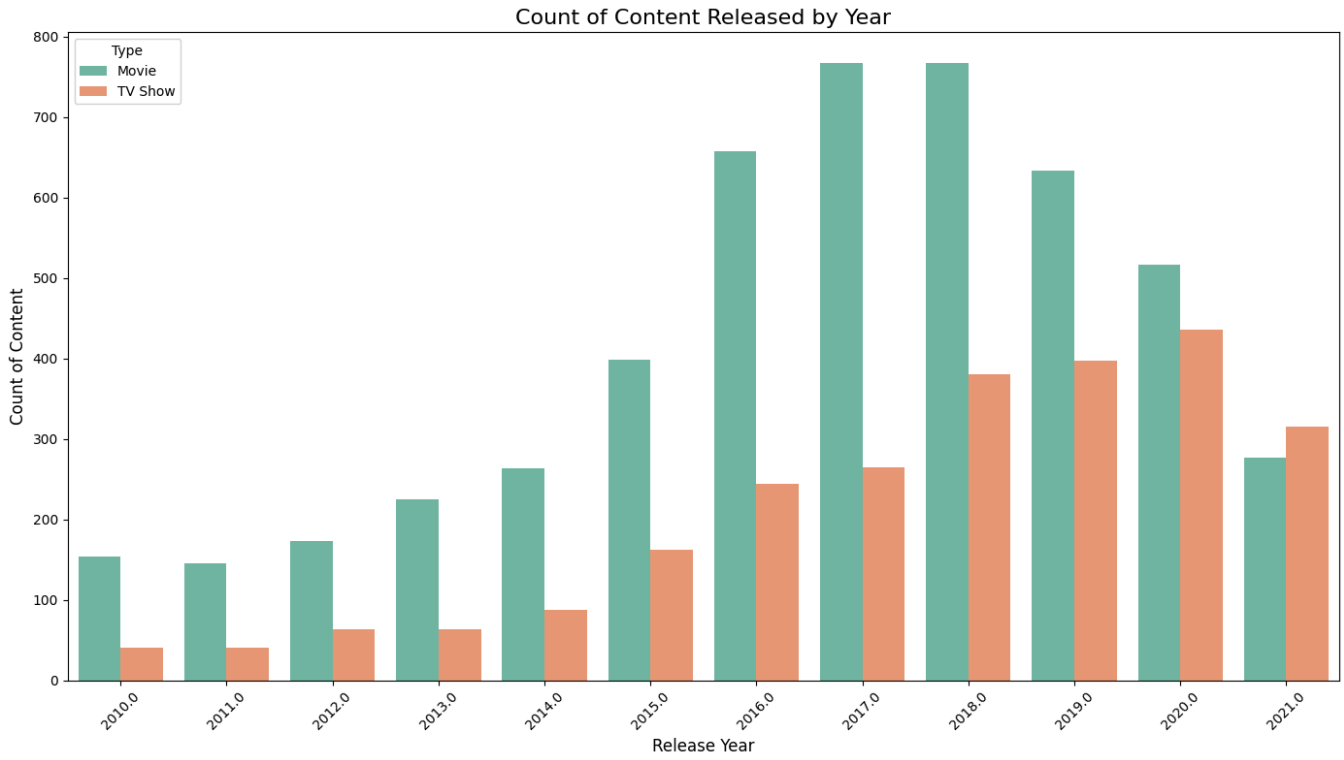
```
sns.histplot(df['release_year'], bins=30)

plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.title('Distribution of Release Year')
plt.xlim(1980,2021 )
plt.show()
```



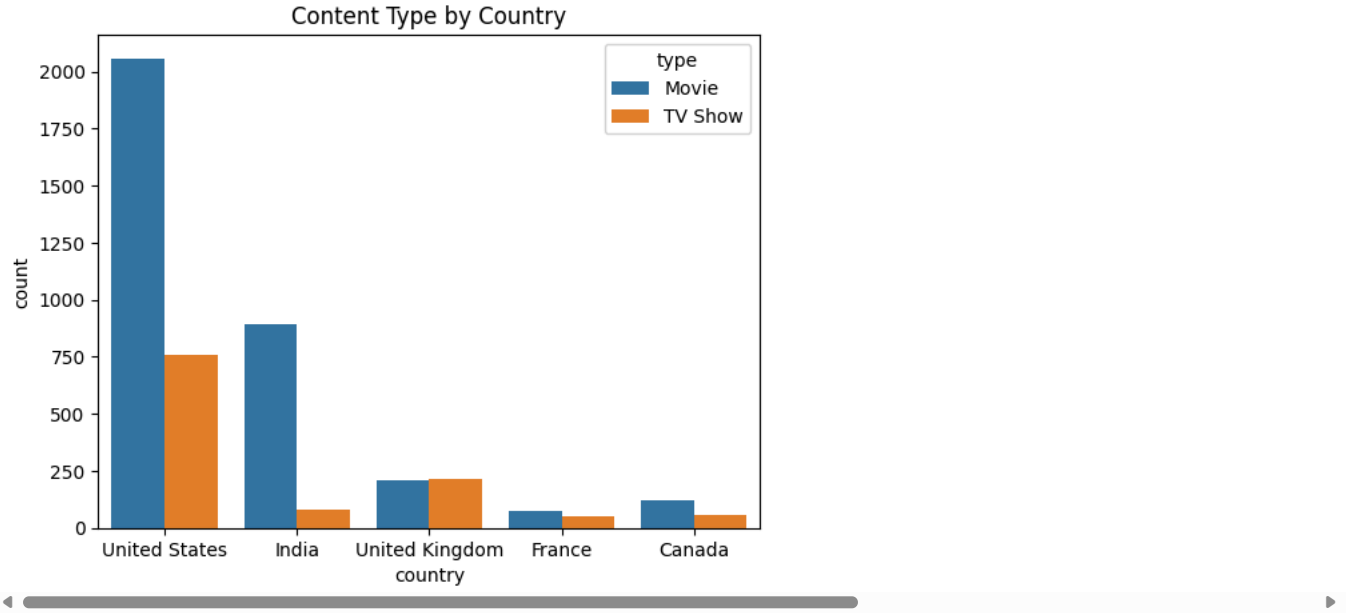
Countplot

```
plt.figure(figsize=(14, 8))
df1=df
df1['release_year']=df1[df1['release_year']>2009]['release_year']
sns.countplot(data=df1, x='release_year', hue='type', palette='Set2', order=sorted(df['release_year'].dropna().unique()))
plt.title('Count of Content Released by Year', fontsize=16)
plt.xlabel('Release Year', fontsize=12)
plt.ylabel('Count of Content', fontsize=12)
plt.legend(title='Type', loc='upper left')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```



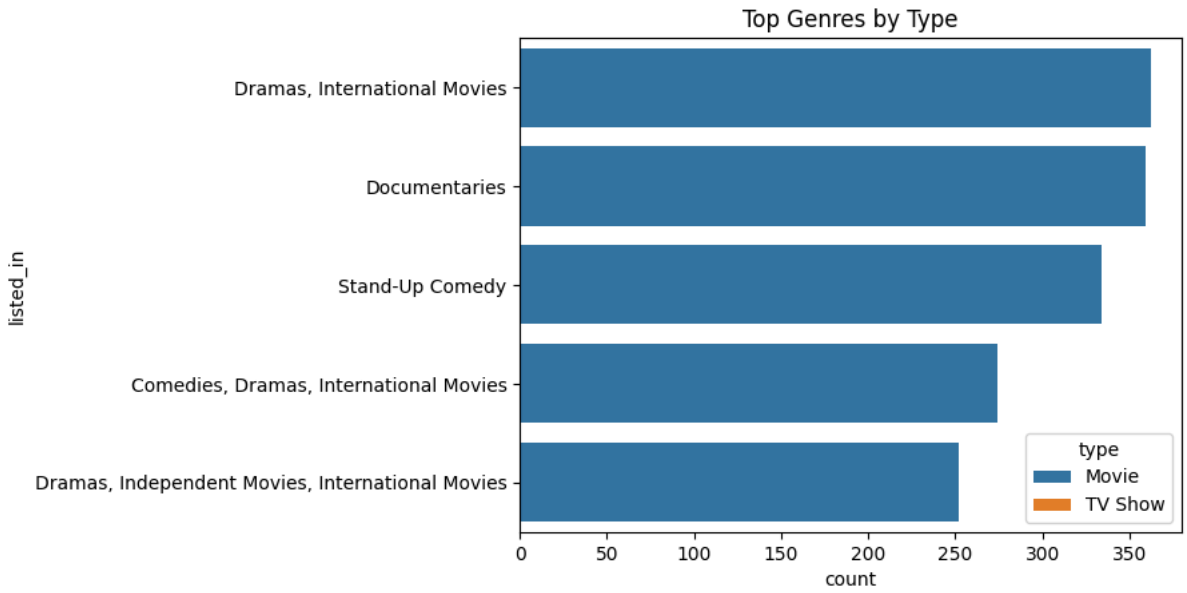
count plot aganist country and counts of type

```
sns.countplot(data=df_country[df_country['country'].isin(['United States', 'India', 'United Kingdom','Canada','France'])], x='country',
plt.title('Content Type by Country')
plt.show()
```



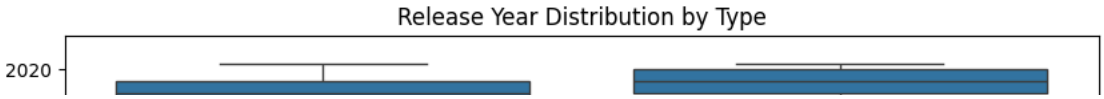
countplot for genres aganist count of types

```
sns.countplot(data=df, y='listed_in', hue='type', order=df['listed_in'].value_counts().head(5).index)
plt.title('Top Genres by Type')
plt.show()
```



Boxplot

```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='type', y='release_year')
plt.title('Release Year Distribution by Type')
plt.xlabel('Type')
plt.ylabel('Release Year')
plt.show()
```


Release Year Distribution by Type
2020