```
#importing the libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from google.colab import files
uploaded =files.upload()
df = pd.read_csv("netflix.csv")
```

Choose Files netflix.csv

- **netflix.csv**(text/csv) - 3399671 bytes, last modified: 1/25/2025 - 100% done

```
Saving netflix.csv to netflix.csv
```

```
df.head(10)
```

	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...
					Mayur More,						International	In a city of

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

```
# Checking the shape of the dataset
df_shape= df.shape
df_shape
```

```
(8807, 12)
```

```
# Checking data types of the attributes
df_type= df.dtypes
df_type
```

	0
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

```
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
```

```
# Displaying basic metrics using the describe() method for numerical columns
df_numerical= df.describe()
df_numerical
```

	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

```
# Displaying basic metrics for categorical columns like 'Type', 'Country', and 'Rating'
df_categorical= df[['type','country','rating']].describe(include=['object'])
df_categorical
```

	type	country	rating
count	8807	7976	8803
unique	2	748	17
top	Movie	United States	TV-MA
freq	6131	2818	3207

```
# Convert categorical attributes to 'category' data type if required
categorical_columns = ['type', 'country', 'rating']
df[categorical_columns] = df[categorical_columns].astype('category')

converted_df = df.dtypes
converted_df
```

	0
show_id	object
type	category
title	object
director	object
cast	object
country	category
date_added	object
release_year	int64
rating	category
duration	object
listed_in	object
description	object

```
#missing values
missing_values=df.isnull().sum()
missing_values
```

	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

```
# Non-Graphical Analysis: Value counts for key attributes
vc_type=df['type'].value_counts()
vc_country=df['country'].value_counts().head(10)
vc_rating=df['rating'].value_counts()
vc_release_year=df['release_year'].value_counts().head(10)
vc_director=df['director'].value_counts().head(10)
vc_type, vc_country, vc_rating, vc_release_year,vc_director
```

```
(type
 Movie    3
 Name: count, dtype: int64,
 country
 United States    3
 Name: count, dtype: int64,
 rating
 74 min    1
 84 min    1
 66 min    1
 Name: count, dtype: int64,
 release_year
 2017    1
 2010    1
 2015    1
 Name: count, dtype: int64,
 director
 Louis C.K.    3
 Name: count, dtype: int64)
```

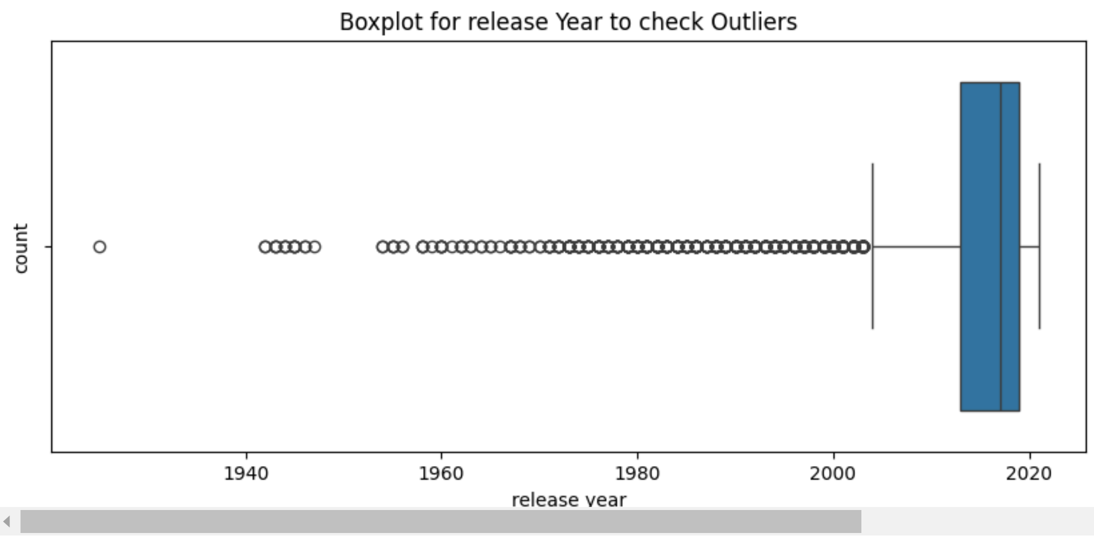
```
# Unique attributes for key columns
u_type=df['type'].unique()
u_country=df['country'].unique()
u_rating= df['rating'].unique()
u_release_year=df['release_year'].unique()
u_type,u_country,u_rating,u_release_year
```

```
(['Movie', 'TV Show']
 Categories (2, object): ['Movie', 'TV Show'],
 ['United States', 'South Africa', NaN, 'India', 'United States, Ghana, Burkina Faso, United Ki..., ..., 'Russia, Spain', 'Croatia, Slovenia, Serbia, Montenegro', 'Japan, Canada', 'United States, France, South Korea, Indonesia', 'United Arab Emirates, Jordan']
 Length: 749
 Categories (748, object): [', France, Algeria', ', South Korea', 'Argentina',
                            'Argentina, Brazil, France, Poland, Germany, D..., ..., 'Venezuela, Colombia', 'Vietnam', 'West Germany',
                            'Zimbabwe'],
 ['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', ..., '66 min', 'NR', NaN, 'TV-Y7-FV', 'UR']
 Length: 18
 Categories (17, object): ['66 min', '74 min', '84 min', 'G', ..., 'TV-Y', 'TV-Y7', 'TV-Y7-FV', 'UR'],
 array([2020, 2021, 1993, 2018, 1996, 1998, 1997, 2010, 2013, 2017, 1975,
        1978, 1983, 1987, 2012, 2001, 2014, 2002, 2003, 2004, 2011, 2008,
        2009, 2007, 2005, 2006, 1994, 2015, 2019, 2016, 1982, 1989, 1990,
        1991, 1999, 1986, 1992, 1984, 1980, 1961, 2000, 1995, 1985, 1976,
        1959, 1988, 1981, 1972, 1964, 1945, 1954, 1979, 1958, 1956, 1963,
        1970, 1973, 1925, 1974, 1960, 1966, 1971, 1962, 1969, 1977, 1967,
        1968, 1965, 1946, 1942, 1955, 1944, 1947, 1943]))
```

```
#duplicates
df.duplicated().value_counts()
```

	count
False	8807

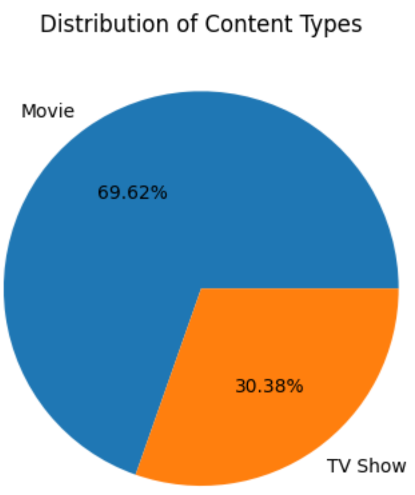
```
#check outliers in numerical value
plt.figure(figsize=(10,4))
sns.boxplot(x=df['release_year'])
plt.title('Boxplot for release Year to check Outliers')
plt.xlabel('release year')
plt.ylabel('count')
plt.show()
```



```
#unnesting the columns with multiple data using split
unnested_director=df.assign(director=df['director'].str.split(',')).explode('director')
unnested_cast= df.assign(cast=df['cast'].str.split(',')).explode('cast')
unnested_country=df.assign(country=df['country'].str.split(',')).explode('country')
```

```
#univariate analysis - categorical data
#distribution of content type using piechart

content_type=df['type'].value_counts()
plt.pie(content_type,labels=content_type.index, autopct='%.2f%%')
plt.title('Distribution of Content Types')
plt.show()
```



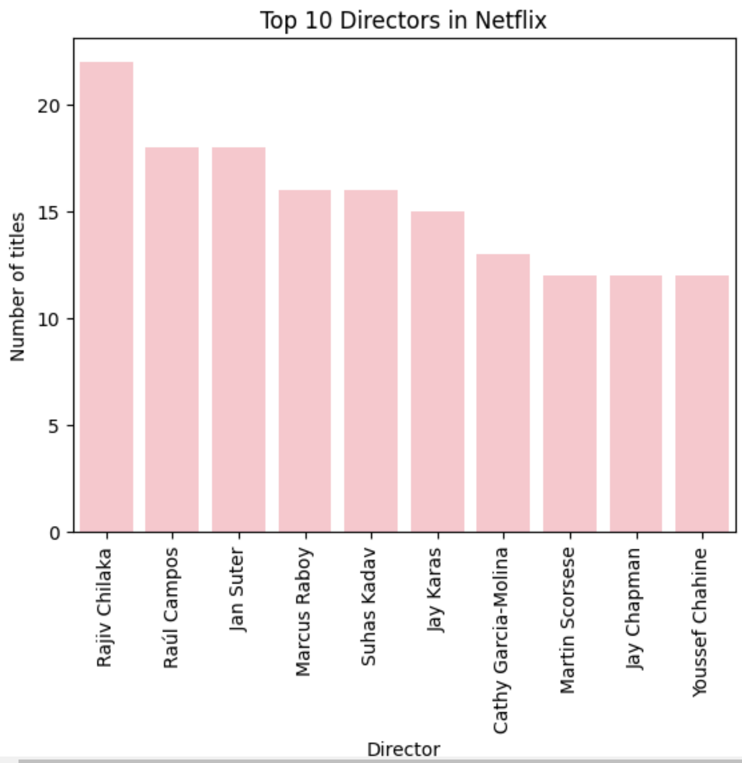
Observation:

The count of movies is significantly higher than TV shows which shows that, they are more interested in launching movies rather than TV shows.`

```
# finding top 10 10 directorss using non graphical analysis
top_directors= unnested_director['director'].value_counts().head(10)
top_directors
```

director	count
Rajiv Chilaka	22
Raúl Campos	18
Jan Suter	18
Marcus Raboy	16
Suhas Kadav	16
Jay Karas	15
Cathy Garcia-Molina	13
Martin Scorsese	12
Jay Chapman	12
Youssef Chahine	12

```
# finding top directors by barplot
top_directors= unnested_director['director'].value_counts().head(10)
sns.barplot(x=top_directors.index, y=top_directors.values, color='pink')
plt.title('Top 10 Directors in Netflix')
plt.xlabel('Director')
plt.xticks(rotation=90)
plt.ylabel('Number of titles')
plt.show()
```



Observations:

The chart shows that , netflix released highest number of movies directed by Rajiv Chilaka with the count 22 and least number of movies directed by Martin Scorsese, Jay Chapman, Youssef Chahine with the count 12.

```
# top ratings using non graphical analysis
top_rating=df['rating'].value_counts()
top_rating.drop(['74 min','84 min','66 min'], inplace=True)
top_rating
```

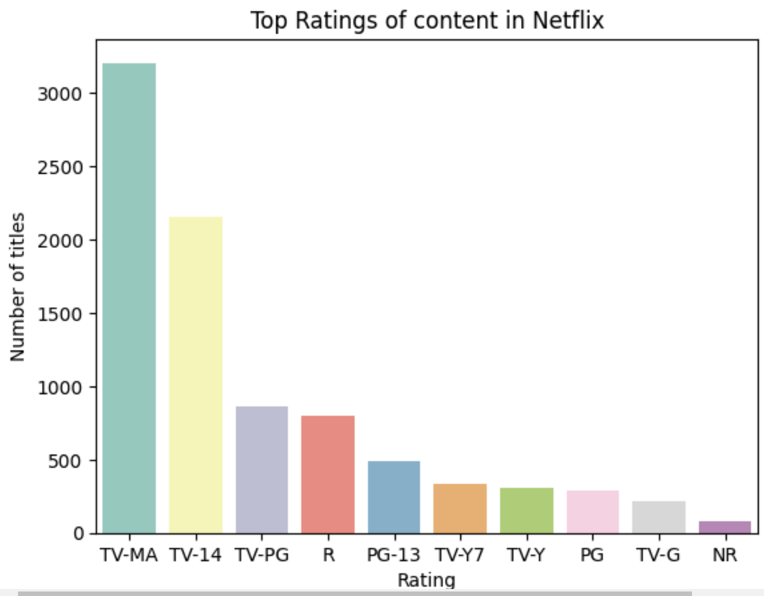
rating	count
TV-MA	3207
TV-14	2160
TV-PG	863
R	799
PG-13	490
TV-Y7	334
TV-Y	307
PG	287
TV-G	220
NR	80
G	41
TV-Y7-FV	6
NC-17	3
UR	3

```
# find top 10 rating using countplot
top_rating=df['rating'].value_counts().head(10)
sns.countplot(x='rating',data=df,order=top_rating.index, palette='Set3')
plt.title('Top Ratings of content in Netflix')
plt.xlabel('Rating')
plt.ylabel('Number of titles')
plt.show()
```

```
<ipython-input-15-55e8c214b35a>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.countplot(x='rating',data=df,order=top_rating.index, palette='Set3')
```



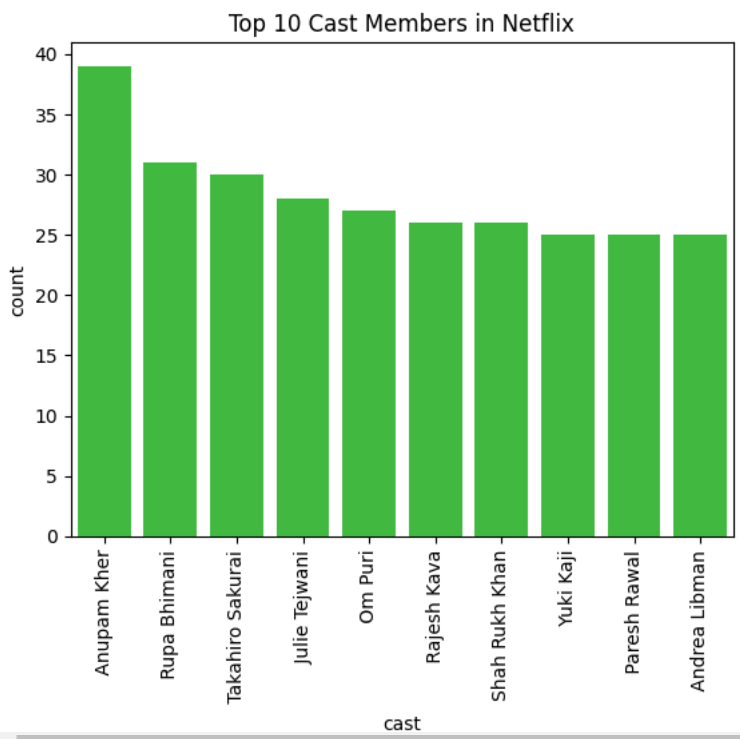
Observations:

The majority of the content is rated 'TV-MA', followed by 'TV-14' which are watched mainly by mature adults and teenagers.

```
# top 10 cast members
top_cast=unnested_cast['cast'].value_counts().head(10)
top_cast
```

cast	count
Anupam Kher	39
Rupa Bhimani	31
Takahiro Sakurai	30
Julie Tejwani	28
Om Puri	27
Rajesh Kava	26
Shah Rukh Khan	26
Yuki Kaji	25
Paresh Rawal	25
Andrea Libman	25

```
#bar plot
sns.barplot(x=top_cast.index, y=top_cast.values, color="limegreen")
plt.title('Top 10 Cast Members in Netflix')
plt.xticks(rotation=90)
plt.xlabel('cast')
plt.ylabel('count')
plt.show()
```



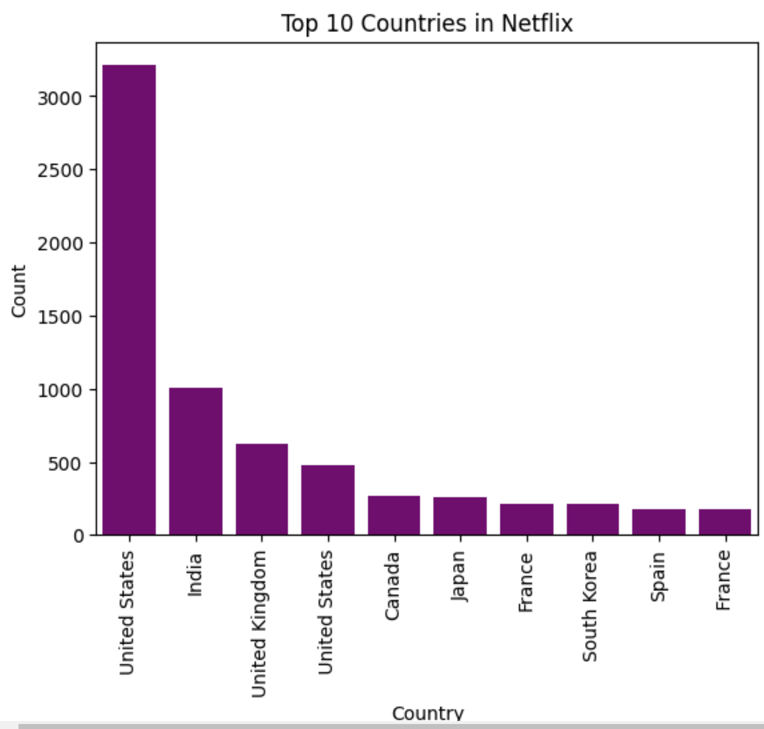
Observation:

The graph shows "Anupam Kher" is the most popular artist on our platform with 39 movies that he has acted in.

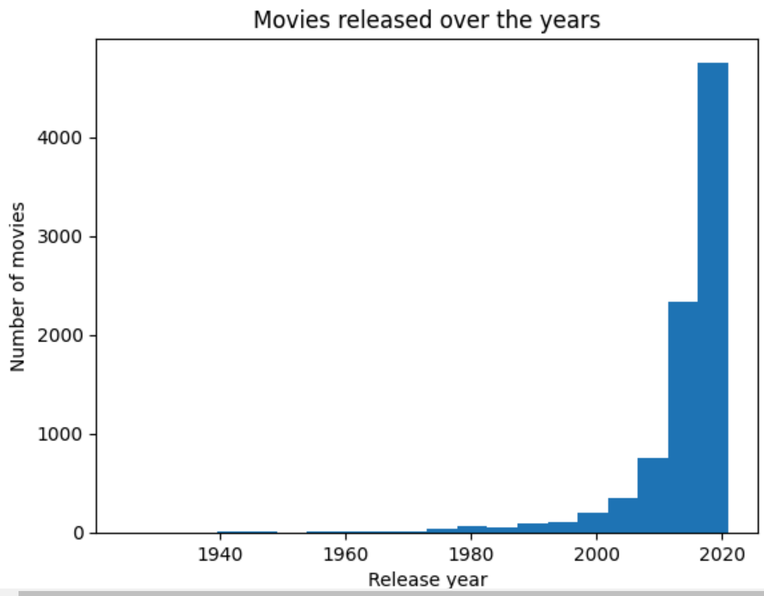
```
# top 10 countries
top_countries=unnested_country['country'].value_counts().head(10)
top_countries
```

	count
country	
United States	3211
India	1008
United Kingdom	628
United States	479
Canada	271
Japan	259
France	212
South Korea	211
Spain	181
France	181

```
sns.barplot(x=top_countries.index, y=top_countries.values, color='purple')
plt.title('Top 10 Countries in Netflix')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



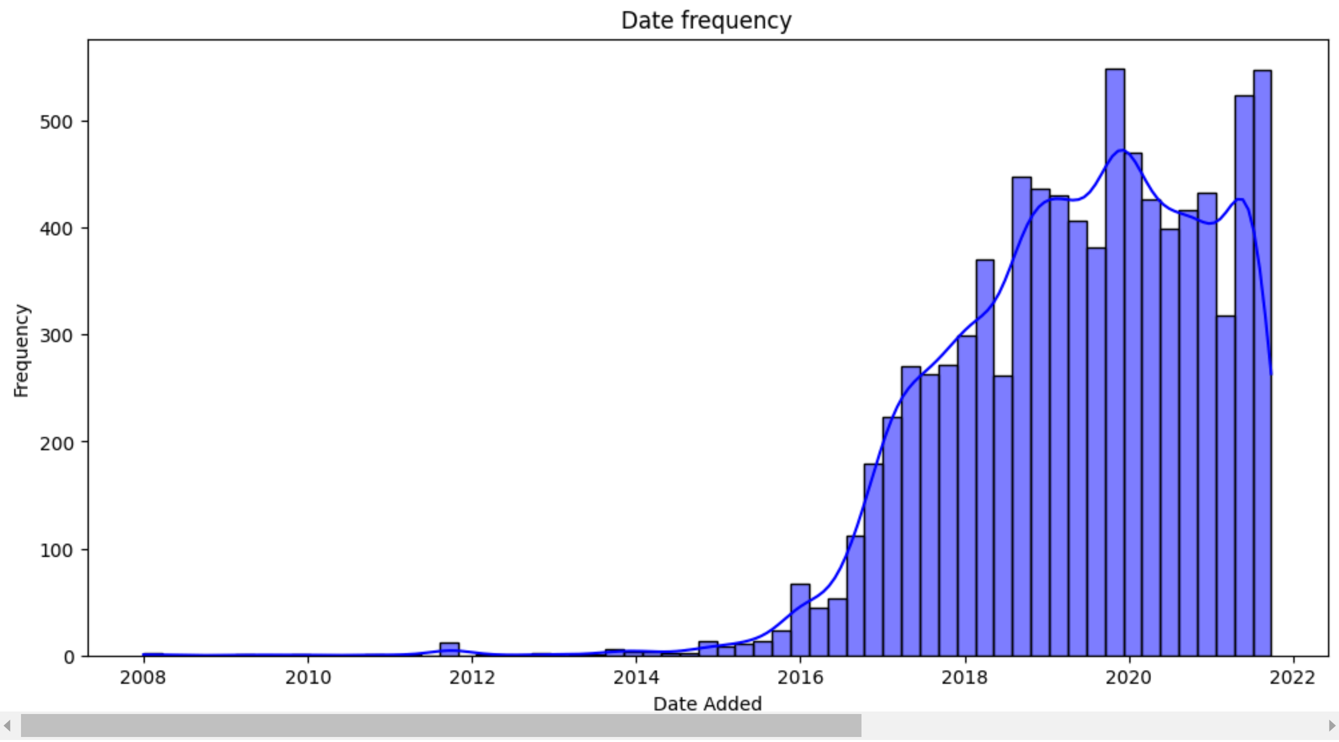
```
#univariate analysis - numerical data
#histogram for release year
plt.hist(df['release_year'], bins=20,ec='black')
plt.title('Movies released over the years')
plt.xlabel('Release year')
plt.ylabel('Frequency')
plt.show()
```



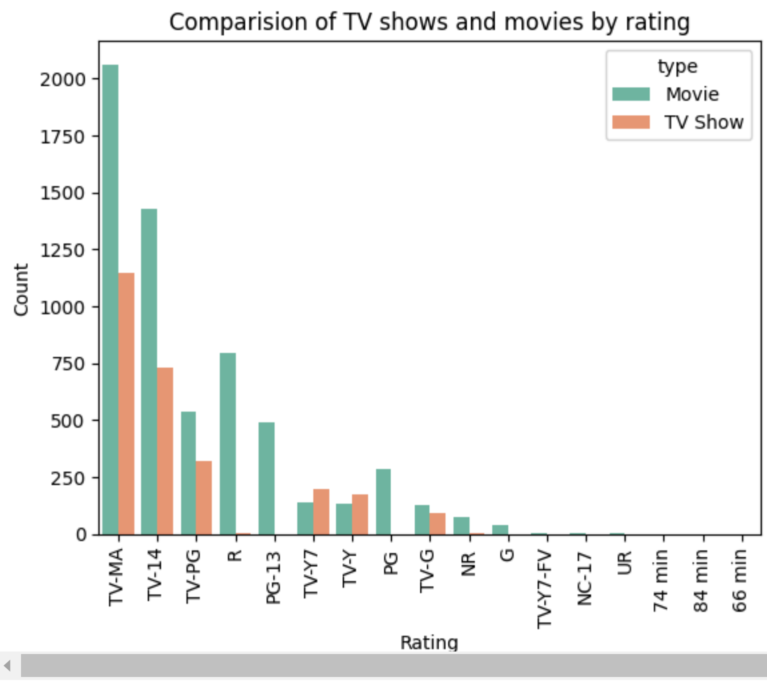
Observation:

The graph is right skewed indicating that, netflix relaeased most of the content directed in recent times with siginifcant release of movies in last decade.

```
# convert date time column into correct format
df['date_added']=pd.to_datetime(df['date_added'], errors='coerce')
# create histogram with kde for date_added
plt.figure(figsize=(12,6))
sns.histplot(data=df, x='date_added',kde=True, color='blue')
plt.title('Date frequency')
plt.xlabel('Date Added')
plt.ylabel('Frequency')
plt.show()
```



```
# comparision of type vs rating using countplot
sns.countplot(x='rating', hue='type', data=df, order=df['rating'].value_counts().index,palette='Set2')
plt.title("Comparision of TV shows and movies by rating")
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



observation:

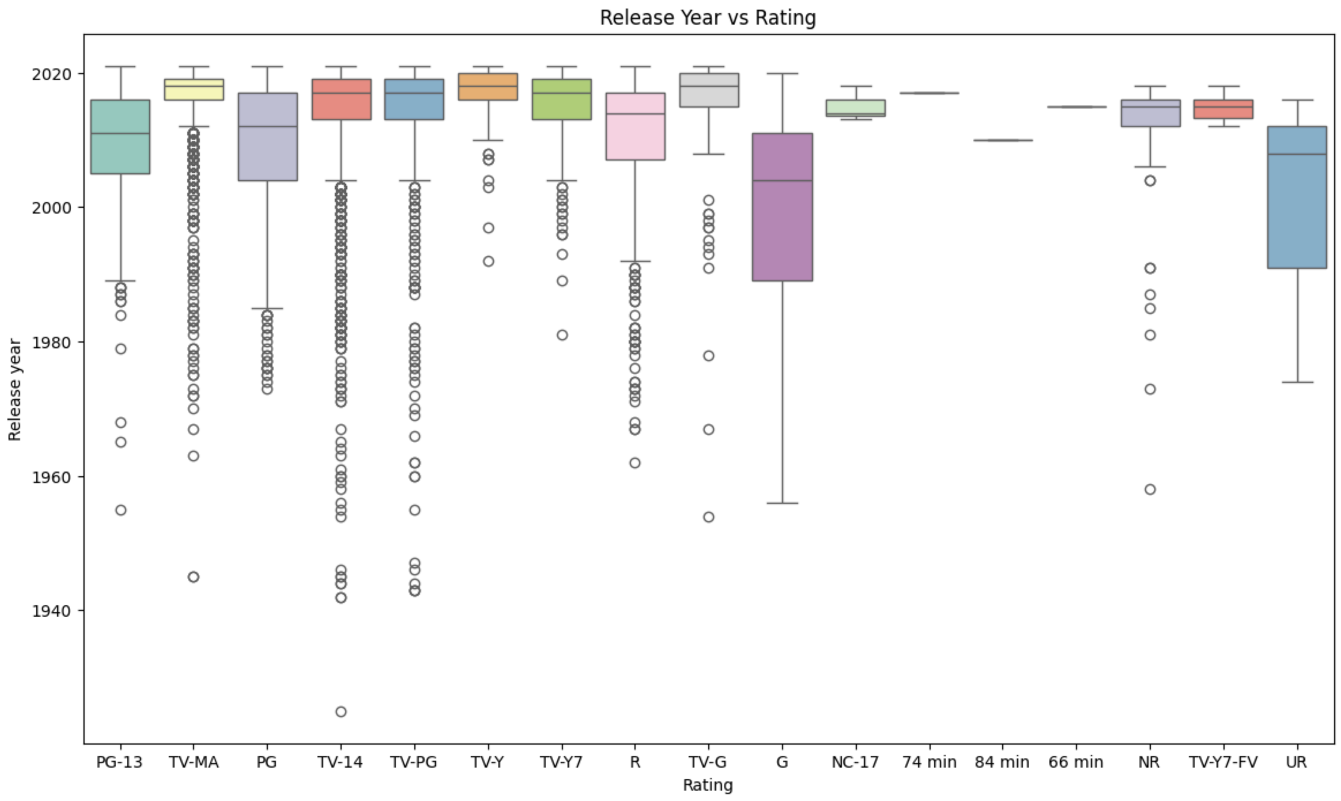
- TV-MA is the Most Common Rating – Both TV shows and movies have the highest count under the TV-MA rating, indicating a significant number of mature content titles.
- TV-14 and TV-PG are also Prominent – These ratings have high counts, suggesting a large number of teen and family-friendly content.
- Movies are More Frequent Across Most Ratings – In almost all rating categories, movies seem to have a higher count than TV shows.

```
# boxplot for release year vs rating
plt.figure(figsize=(14,8))
sns.boxplot(x='rating', y='release_year', data=df,palette="Set3")
plt.title('Release Year vs Rating')
plt.xlabel('Rating')
plt.ylabel('Release year')
plt.show()
```

```
<ipython-input-67-c18280e67ce6>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.boxplot(x='rating', y='release_year', data=df,palette="Set3")
```



Observation:

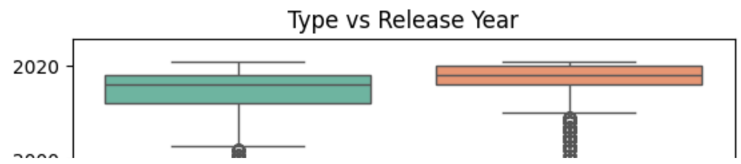
- Most content is from the 2000s and later – The majority of ratings have medians around 2000 or later, with fewer older releases.
- TV-MA, TV-14, and TV-PG are dominant – These ratings show a large number of releases spanning multiple decades.
- G and PG-rated content have older distributions – Many G and PG-rated titles extend back several decades, indicating a long history of family-friendly content. q1`
- Outliers exist across all ratings – Some very old movies and TV shows appear as outliers, especially in ratings like TV-Y, PG, and G.

```
#boxplot for type vs release year
sns.boxplot(x='type', y='release_year',data=df,palette='Set2')
plt.title('Type vs Release Year')
plt.xlabel('type')
plt.ylabel('Release year')
plt.show()
```

```
<ipython-input-73-b8cea6ad9964>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.boxplot(x='type', y='release_year',data=df,palette='Set2')
```



Observation:

- Most content is recent (2000s onward) – Both movies and TV shows have a median release year close to 2020.
- TV Shows are more concentrated in recent years – The interquartile range (IQR) for TV shows is narrower, suggesting more consistent recent releases. Movies have a wider spread – Older movies (pre-2000) appear more frequently compared to TV shows.