Importing necessary libraries required to perform analysis on the Netflix dataset.

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

Downloading and Reading the dataset.

```
! gdown 1ZKlD0XIwkcFuEXjAL1ctYB-oCQ-j240a
```

```
Downloading...
From: https://drive.google.com/uc?id=1ZKlD0XIwkcFuEXjAL1ctYB-oCQ-j240a
To: /content/Netflix_project.csv
100% 3.40M/3.40M [00:00<00:00, 65.1MB/s]
```

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

	show_id	type	title	director	cast	country	date_added	release_year	rating	durati
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 n
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	Seasc
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Seas
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Seas
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	Seasc
...	...	...	...	...	...	...	...	...	...	

SOME BASIC EDA QUESTIONS (before data cleaning) :

1.How has the number of movies released per year changed over the last 20-30 years?

```
df_movies = df.loc[df['type']=='Movie']
df_movies
```

```
movies_per_year = df_movies.groupby('release_year')[['release_year','title']].aggregate(title_count = ('title','count')).reset_index()
movies_per_year
```

	release_year	title_count
0	1942	2
1	1943	3
2	1944	3
3	1945	3
4	1946	1
...	...	...
68	2017	767
69	2018	767
70	2019	633
71	2020	517
72	2021	277

73 rows × 2 columns

BASIC EDA QUESTIONS

2.Comparison of tv shows vs. movies.

```
a=df.type.value_counts()
a
```

```
Movie      6131
TV Show    2676
Name: type, dtype: int64
```

BASIC EDA QUESTIONS

3. Understanding what content is available in different countries

```
df_tv = df.loc[df['type']=='TV Show']
df_tv
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	durat
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	Seas
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Sea
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Sea
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	Seas
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	September 24, 2021	2021	TV-MA	1 Sea
...	...	...	...	...	...	...	...	...	...	
8795	s8796	TV Show	Yu-Gi-Oh! Arc-V	NaN	Mike Liscio, Emily Bauer, Billy Bob Thompson	Japan, Canada	May 1, 2018	2015	TV-Y7	Seas

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

```
United States                              2818
India                                       972
United Kingdom                              419
Japan                                       245
South Korea                                 199
                                           ...
Romania, Bulgaria, Hungary                    1
Uruguay, Guatemala                            1
France, Senegal, Belgium                      1
Mexico, United States, Spain, Colombia        1
United Arab Emirates, Jordan                  1
Name: country, Length: 748, dtype: int64
```

**NETFLIX PROJECT **

Problem Statement : **Helping Netflix in deciding which type of shows/movies to produce and how they can grow the business in different countries**

Basic Metrics and Non-graphical Analysis.

Analysis on the dataset.

- Shape
- Various Attributes
- Data Types of the Attributes
- Statistical Summary of the data

```
df.shape
```

```
(8807, 12)
```

```
df.columns
```

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

```
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

DATA CLEANING

- After performing basic analysis on the dataset we can see some of the attributes contain various amount of null values.
- Firstly what I have done is calculated the number of nulls in each attribute.
- As you can see below since 'rating' and 'duration' have hust 3-4 null values those rows have been dropped.
- For the 'director' and 'cast' attributes the null values have been replaced with 'unknown'

```
nulls = df.isnull().sum()
nulls
```

```
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.dropna(subset=['rating','duration'],inplace=True)
```

```
df.director.fillna('Director unknown',inplace=True)
df.cast.fillna('cast unknown',inplace=True)
```

Unnesting and Correction of the attributes

- For the 'country' attribute there are 2 issues that had to be dealt with : 1) The null values 2) Multiple countries being present in the same row
- To deal with the null values what was done is we replaed the nulls with the 'mode' of the country column.
- For the unnesting of the multiple countries 'apply' and 'split' was used to get only the first country mentioned in the dataset.
- Similar cleaning was done on the 'listed_in' attribute as well which would give us the Genres of the titles.

```
df.country.value_counts()
```

```
United States                              2815
India                                       972
United Kingdom                              419
Japan                                       244
South Korea                                 199
                                           ...
Romania, Bulgaria, Hungary                    1
Uruguay, Guatemala                            1
France, Senegal, Belgium                      1
Mexico, United States, Spain, Colombia        1
United Arab Emirates, Jordan                  1
Name: country, Length: 748, dtype: int64
```

```
df.country=df.country.fillna(df.country.mode()[0])
```

```
df.country.value_counts()
```

```
United States                              3645
India                                       972
United Kingdom                              419
Japan                                       244
South Korea                                 199
                                           ...
Romania, Bulgaria, Hungary                    1
Uruguay, Guatemala                            1
France, Senegal, Belgium                      1
Mexico, United States, Spain, Colombia        1
United Arab Emirates, Jordan                  1
Name: country, Length: 748, dtype: int64
```

```
df.country=df.country.apply(lambda x: x.split(", ")[0])
```

```
df.country.value_counts().head(10)
```

```
United States     4037
India             1008
United Kingdom     626
Canada             271
Japan              258
```

```
France             212
South Korea        211
Spain              181
Mexico             134
Australia          116
Name: country, dtype: int64
```

```
df.listed_in=df.listed_in.apply(lambda x: x.split(", ")[0])
df.listed_in.value_counts().head(10)
```

```
Dramas                      1599
Comedies                    1210
Action & Adventure           859
Documentaries                829
International TV Shows       774
Children & Family Movies     605
Crime TV Shows               399
Kids' TV                     387
Stand-Up Comedy              334
Horror Movies                275
Name: listed_in, dtype: int64
```

Comparison of TV-Shows and Movies

```
df.type.value_counts()
```

```
Movie      6126
TV Show    2674
Name: type, dtype: int64
```

Top 10 years with highest content produced

```
df.release_year.value_counts().head(10)
```

```
2018    1147
2017    1030
2019    1030
2020     953
2016     902
2021     592
2015     557
2014     352
2013     287
2012     237
Name: release_year, dtype: int64
```

No. of contents based on ratings.

```
df.rating.value_counts()
```

```
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
Name: rating, dtype: int64
```

Statistical Summary post Cleaning

```
df.describe()
```

	release_year
count	8800.000000
mean	2014.179886
std	8.822583
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

Creating a seperate Movie dataframe

```
df_movies = df.loc[df['type']=='Movie']
df_movies
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	dur
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	cast unknown	United States	September 25, 2021	2020	PG-13	9
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	United States	September 24, 2021	2021	PG	9
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States	September 24, 2021	1993	TV-MA	12
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, Luna	United States	September 24, 2021	2021	PG-13	10

Creating a seperate TV-show dataframe

```
df_tv = df.loc[df['type']=='TV Show']
df_tv
```

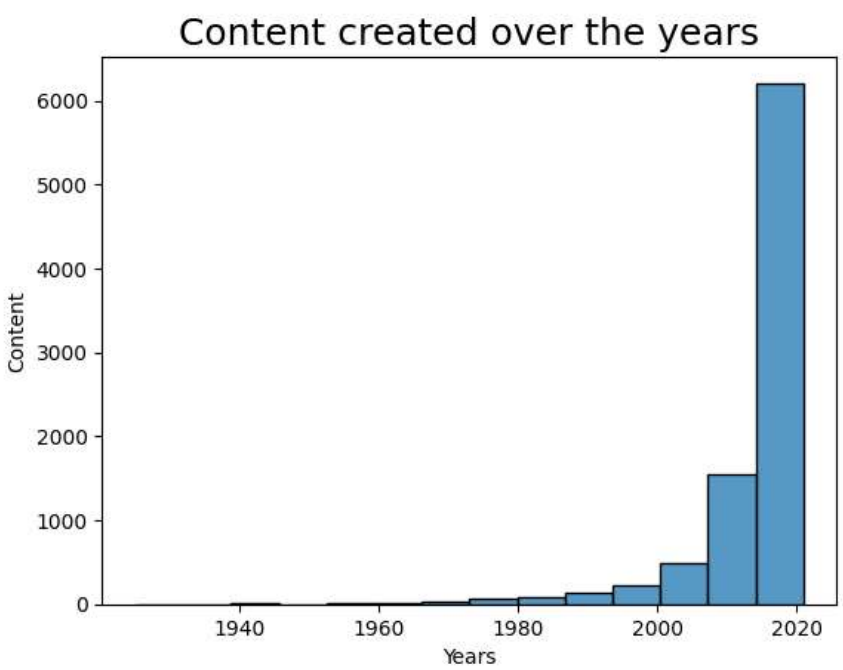
	show_id	type	title	director	cast	country	date_added	release_year	rating	durati
1	s2	TV Show	Blood & Water	Director unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	Seaso
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	September 24, 2021	2021	TV-MA	1 Seas
3	s4	TV Show	Jailbirds New Orleans	Director unknown	cast unknown	United States	September 24, 2021	2021	TV-MA	1 Seas
4	s5	TV Show	Kota Factory	Director unknown	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	Seaso
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	United States	September 24, 2021	2021	TV-MA	1 Seas

VISUAL ANALYSIS

Content through the years

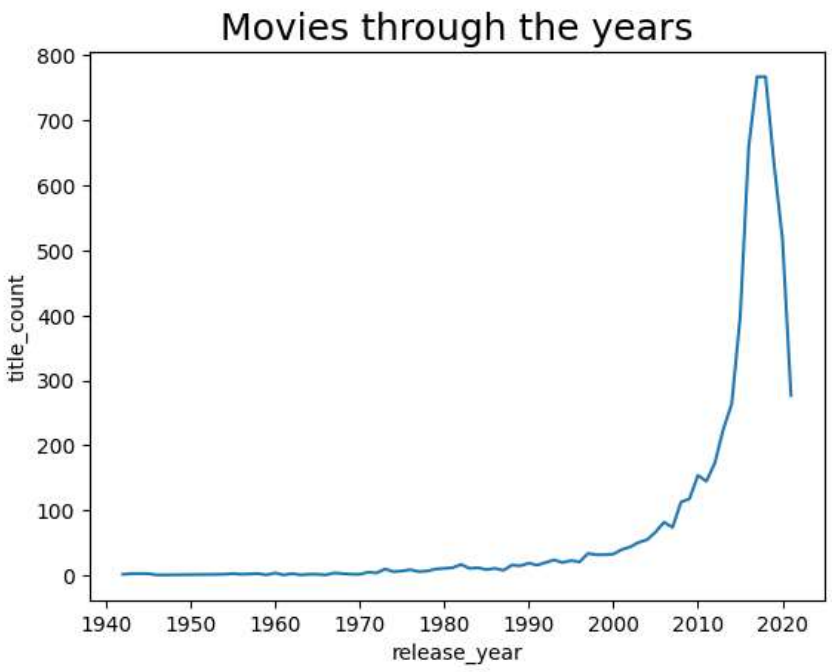
- Plotted a histogram showing the content created over the years.

```
sns.histplot(df.release_year,bins=14)
plt.xlabel('Years')
plt.ylabel('Content')
plt.title('Content created over the years',fontsize=18)
plt.show()
```



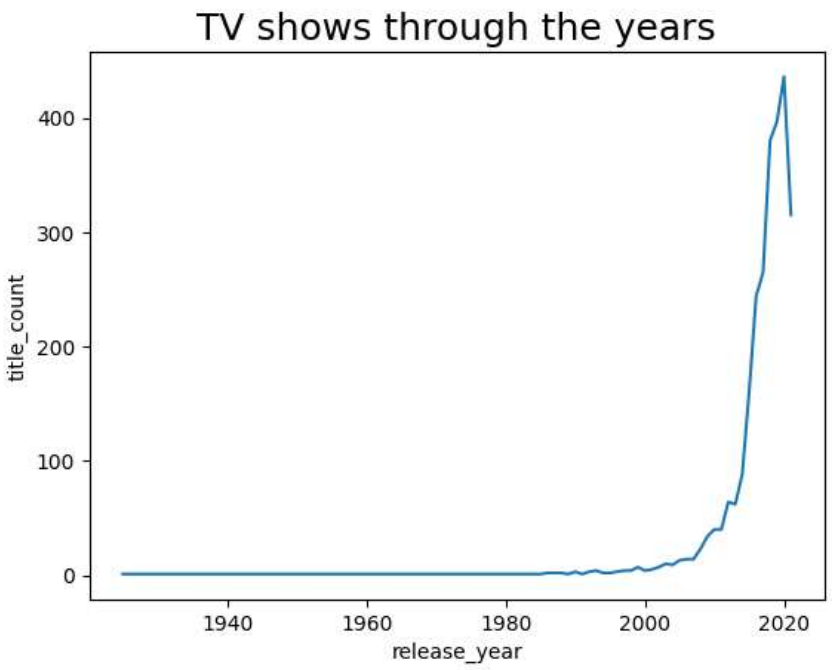
Movies throughout the years

```
sns.lineplot(data=movies_per_year,x='release_year',y='title_count')
plt.title('Movies through the years',fontsize=18)
plt.show()
```



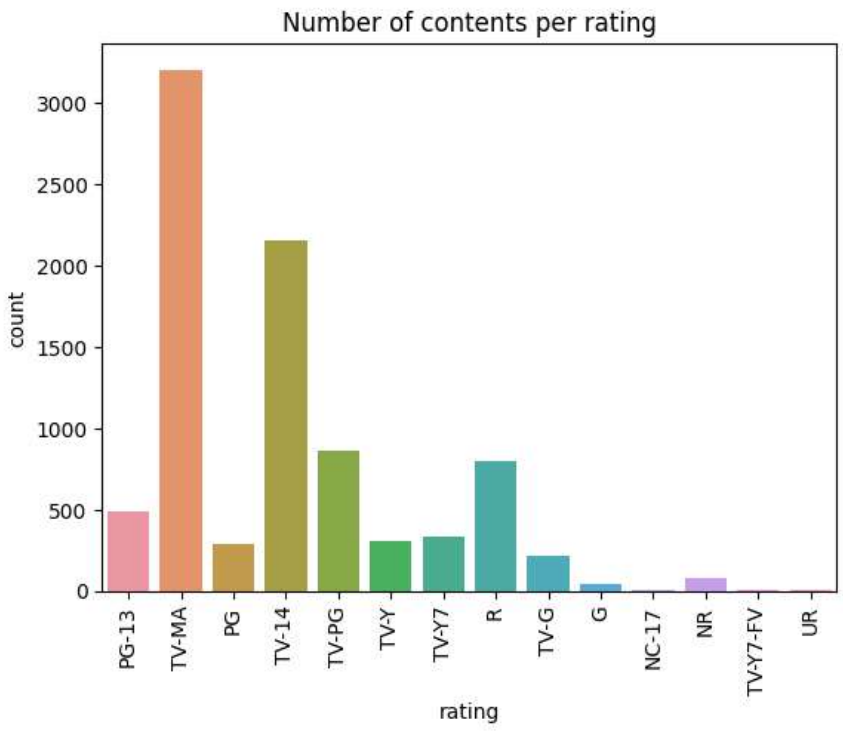
TV shows throughout the years

```
shows_per_year = df_tv.groupby('release_year')[['release_year','title']].aggregate(title_count = ('title','count')).reset_index()
shows_per_year
sns.lineplot(data=shows_per_year,x='release_year',y='title_count')
plt.title('TV shows through the years',fontsize=18)
plt.show()
```



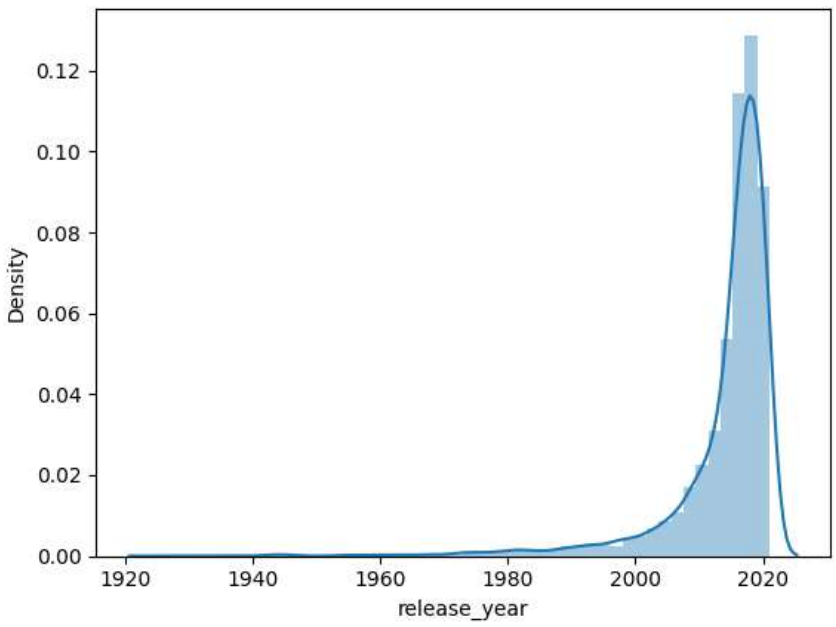
Content per rating

```
sns.countplot(x=df.rating)
plt.title("Number of contents per rating")
plt.xticks(rotation=90)
plt.show()
```



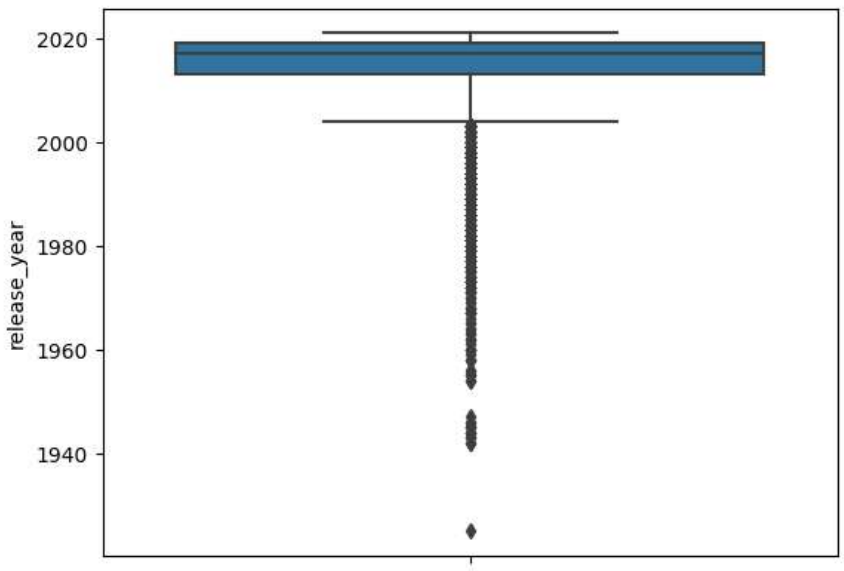
```
sns.distplot(df.release_year)
plt.show()
```

```
<ipython-input-54-5d499bfea5f5>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.release_year)
```



From the Box-plot plotted below we can see that the graph is slightly negatively skewed and all the outliers as well can be seen.

```
sns.boxplot(y=df['release_year'])
plt.ylabel=('Years')
plt.show()
```



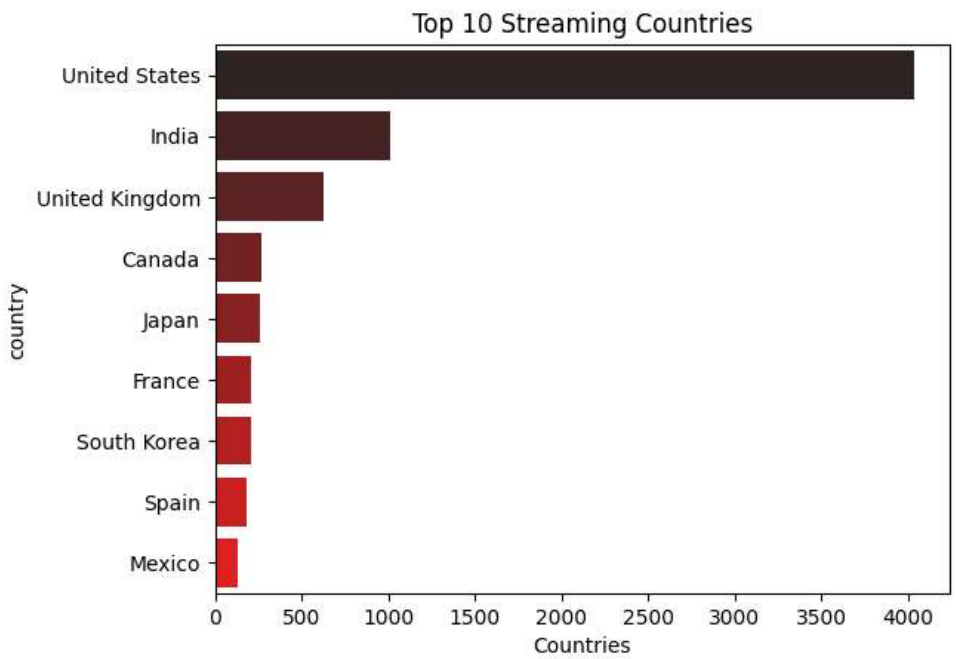
Top 10 countries with highest content streamed

```
top_10_countries = df['country'].value_counts()[:9].reset_index()
top_10_countries.columns = ['country','title_count']
top_10_countries
```

	country	title_count
0	United States	4037
1	India	1008
2	United Kingdom	626
3	Canada	271
4	Japan	258
5	France	212
6	South Korea	211
7	Spain	181
8	Mexico	134

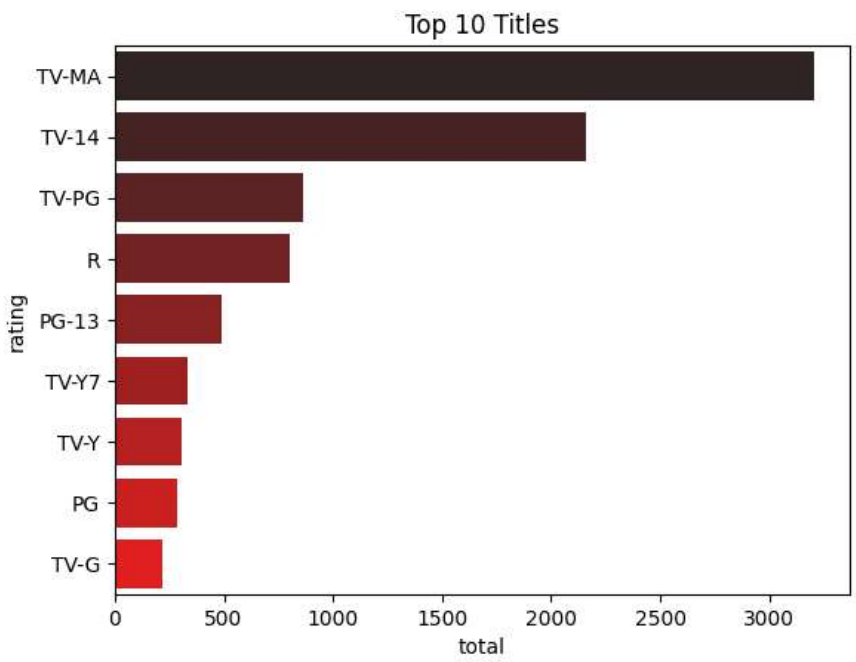
```
top_10_c = sns.countplot(y=df.country,order = df.country.value_counts().index[:9],palette='dark:red')
```

```
plt.title('Top 10 Streaming Countries')
plt.xlabel("Countries")
plt.show()
```



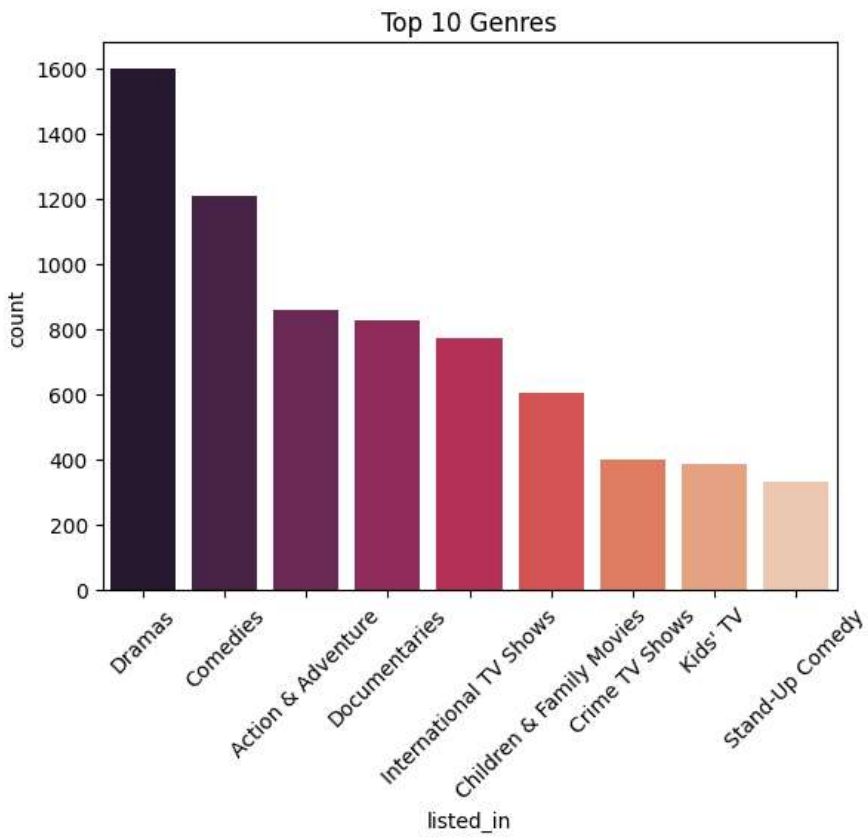
Top 10 ratings with highest number of titles

```
top_10_c = sns.countplot(y=df.rating,order = df.rating.value_counts().index[:9],palette='dark:red')
plt.title('Top 10 Titles')
plt.xlabel("total")
plt.show()
```



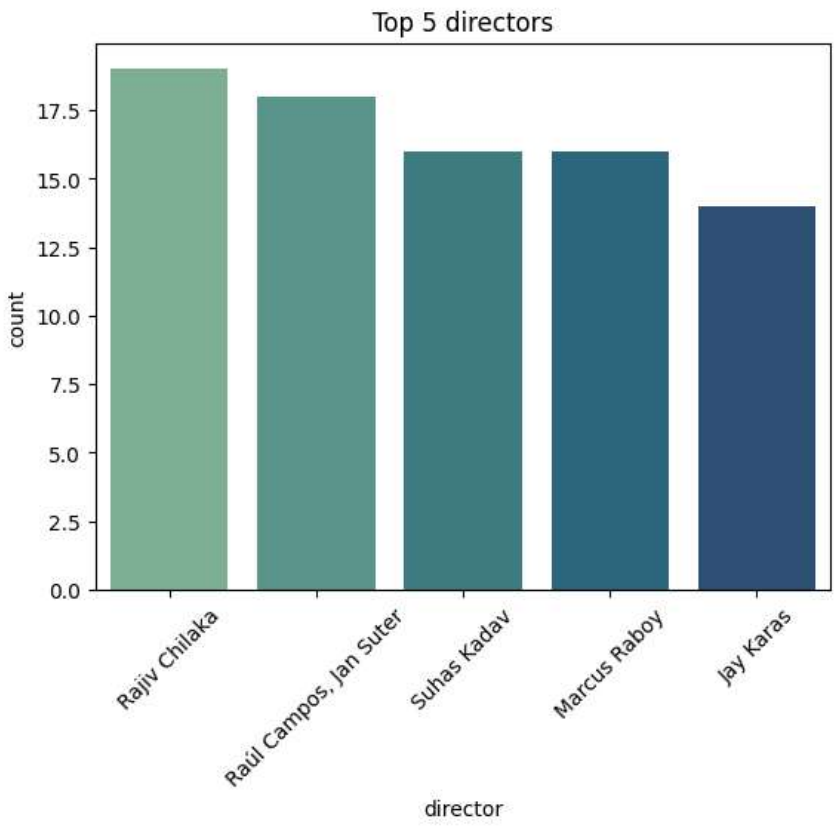
Top 10 Genres

```
top_10_c = sns.countplot(x=df.listed_in,order = df.listed_in.value_counts().index[:9],palette='rocket')
plt.xticks(rotation = 45)
plt.title('Top 10 Genres')
plt.show()
```



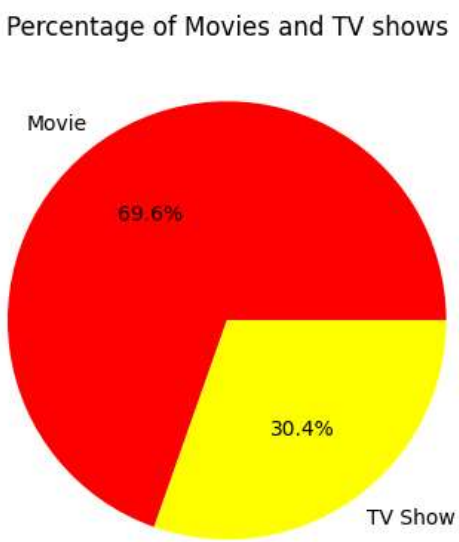
Top 5 directors

```
top_10_c = sns.countplot(x=df.director,order = df.director.value_counts().index[1:6],palette='crest')
plt.xticks(rotation = 45)
plt.title('Top 5 directors')
plt.show()
```



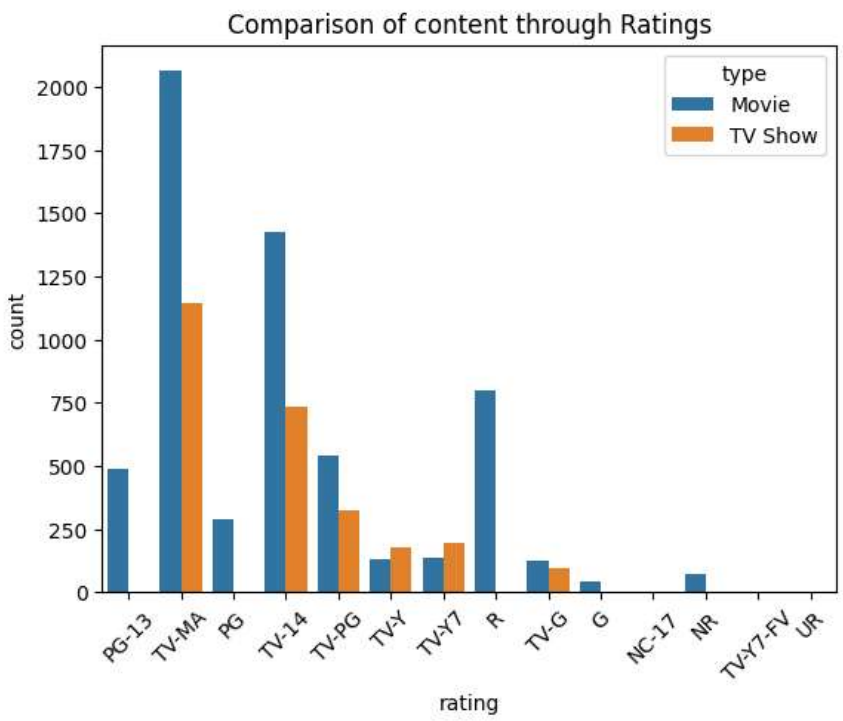
Comparison between Movies and TV shows

```
comparison = plt.pie(df.type.value_counts(),labels=df.type.value_counts().index, colors =['red','yellow'],autopct='%1.1f%%')
plt.title("Percentage of Movies and TV shows")
plt.show()
```



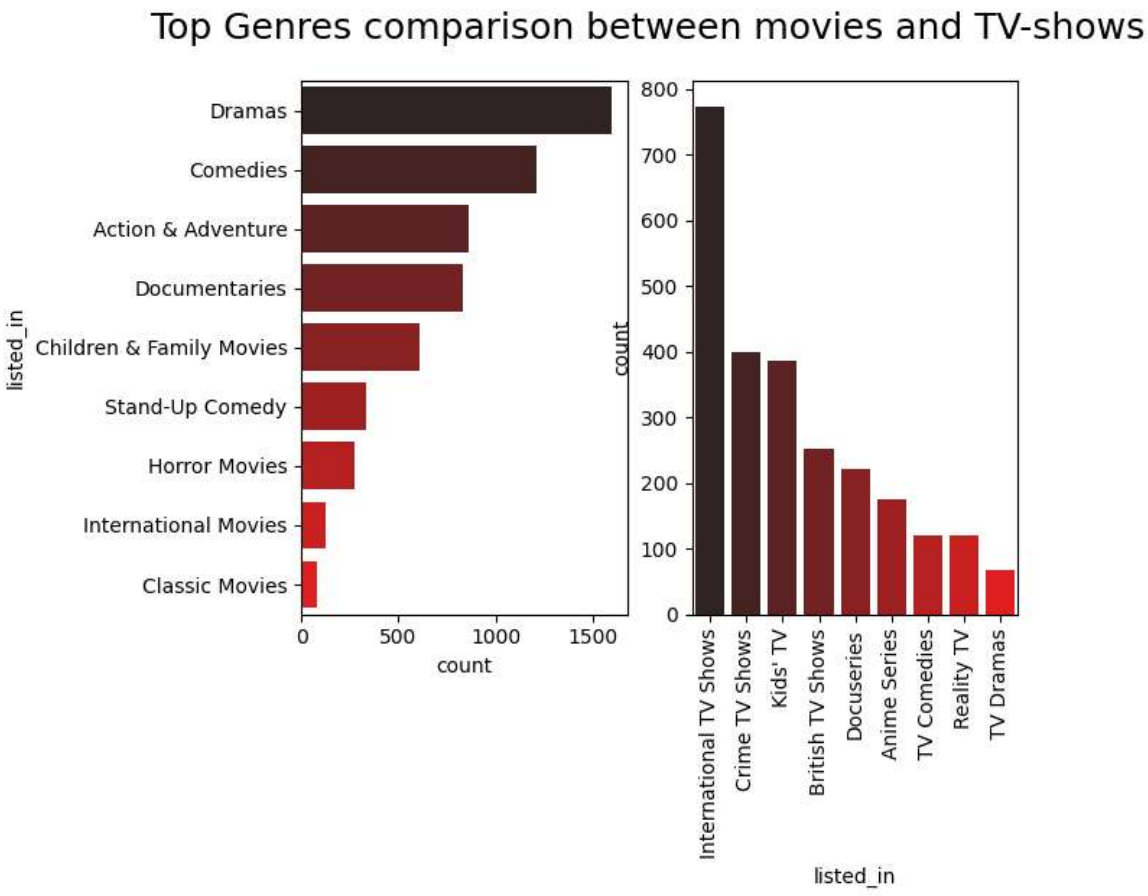
Bi-Variate Analysis showing the comparison of Movies and TV shows based on the ratings

```
compare_ratings = sns.countplot(x=df.rating,hue = df.type)
plt.xticks(rotation=45)
plt.title("Comparison of content through Ratings")
plt.show()
```



Sub-plot showing the Top Genres comparison between Movies and Tv-shows

```
fig , ax = plt.subplots(1,2,)

sns.countplot(y=df_movies.listed_in,order=df_movies.listed_in.value_counts().index[:9], ax=ax[0],palette = 'dark:red')
plt.xticks(rotation=90)
sns.countplot(x=df_tv.listed_in,order=df_tv.listed_in.value_counts().index[:9],ax=ax[1],palette = 'dark:red')
plt.suptitle("Top Genres comparison between movies and TV-shows",fontsize = 18)
plt.show()
```

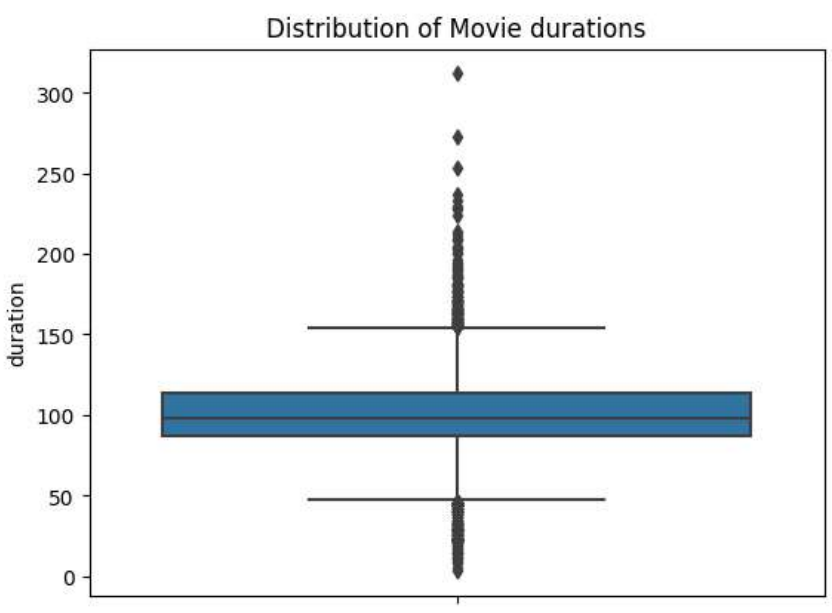


For duration analysis for movies the attribute had to be cleaned first

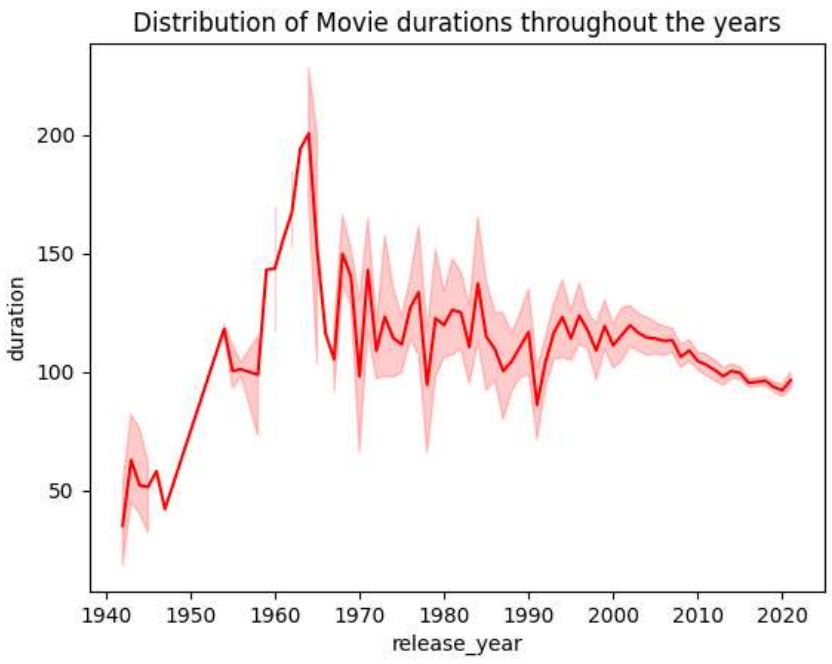
- Firstly the numeric part of the 'duration' column was fetched.
- Datatype of the column was converted.

```
df_movies.duration = df_movies.duration.apply(lambda x: x.split(" ")[0])
df_movies.duration=df_movies.duration.astype("int")
df_movies.duration.info()
```

```
sns.boxplot(y=df_movies.duration)
plt.title("Distribution of Movie durations")
plt.show()
```

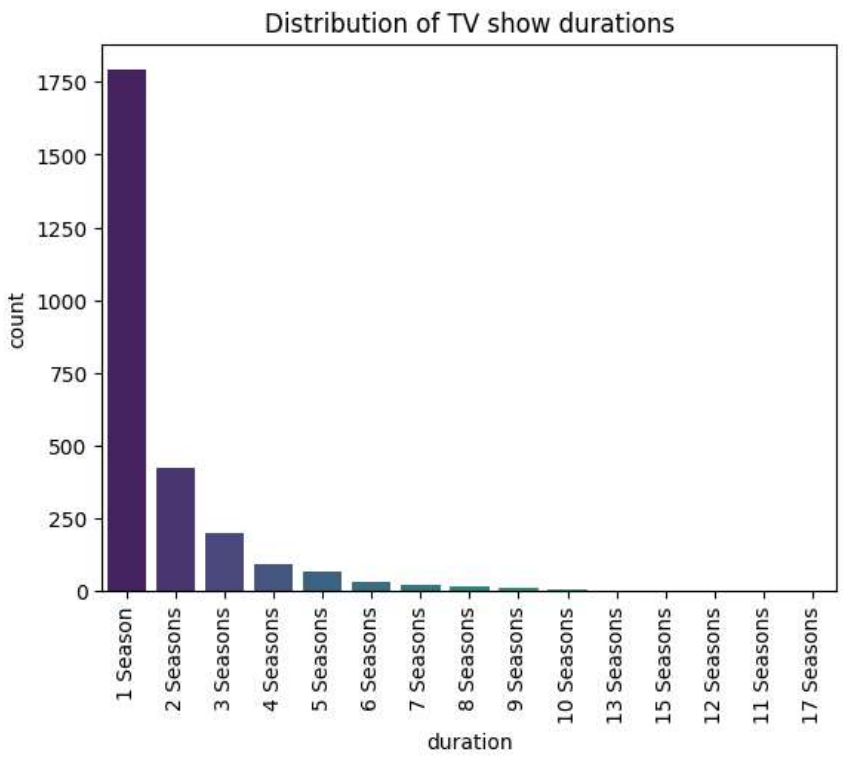


```
sns.lineplot(data=df_movies,x='release_year',y='duration',color='red')
plt.title("Distribution of Movie durations throughout the years")
plt.show()
```



TV-show duration ditribution

```
sns.countplot(x=df_tv.duration,order=df_tv.duration.value_counts().index,palette='viridis')
plt.title('Distribution of TV show durations')
plt.xticks(rotation=90)
plt.show()
```



✓ 0s completed at 1:39 PM