```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv -O netflix_data.csv
```

```
--2023-08-06 04:56:31--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 99.84.178.132, 99.84.178.93, 99.84.178.172, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|99.84.178.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3399671 (3.2M) [text/plain]
Saving to: ‘netflix_data.csv’

netflix_data.csv    100%[===================>]   3.24M  --.-KB/s    in 0.05s

2023-08-06 04:56:31 (63.7 MB/s) - ‘netflix_data.csv’ saved [3399671/3399671]
```

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

## Analysing basic metrics

```
data = pd.read_csv('netflix_data.csv')
data.shape
```

```
(8807, 12)
```

```
data.head()
```

| | show_id | type | title | director | cast | country | date_added | release_year |
|---|---|---|---|---|---|---|---|---|
| **0** | s1 | Movie | Dick Johnson Is Dead | Kirsten Johnson | NaN | United States | September 25, 2021 | 2020 |
| **1** | s2 | TV Show | Blood & Water | NaN | Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban... | South Africa | September 24, 2021 | 2021 |
| **2** | s3 | TV Show | Ganglands | Julien Leclercq | Sami Bouajila, Tracy Gotoas, | NaN | September 24, 2021 | 2021 |

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   type          8807 non-null   object
 1   title         8807 non-null   object
 2   director      8807 non-null   object
 3   cast          8807 non-null   object
 4   country       8807 non-null   object
 5   release_year  8807 non-null   int64 
 6   rating        8807 non-null   object
 7   duration      8807 non-null   object
 8   listed_in     8807 non-null   object
 9   month         8807 non-null   object
 10  year          8807 non-null   object
dtypes: int64(1), object(10)
memory usage: 757.0+ KB
```

```
data.describe()
```

| | release_year |
|---|---|
| count | 8807.000000 |
| mean | 2014.180198 |
| std | 8.819312 |
| min | 1925.000000 |

```
data.columns
```

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

## Data Cleaning

```
data.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
```

Director has 2634, cast has 825 and country has 831 null values.

```
data.nunique()
```

```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1767
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
```

```
data.duplicated().sum()
```

```
0
```

Fill missing values with 'Unknown': director, cast. Fill missing values with mode value: Country, date_added, rating. For 'rating', 'UR' stands for unrated, with the same meaning as 'NR' not rated, set only one form of expression. Drop rows with null values: duration. There are only three missing values in duration and I decide to drop these 3 rows.

```
data['country'] = data['country'].fillna(data['country'].mode()[0])
data['country'] = data['country'].astype(str)
data['country'] = data['country'].apply(lambda x : x.split(', ')[0])
```

```
data['rating'] = data['rating'].replace({'74 min' : np.nan, '84 min' : np.nan, '66 min': np.nan, 'TV-Y7-FV' : 'TV-Y7','NR': 'Unrated','UR
data['rating'].unique()
```

```
array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
       'TV-G', 'G', 'NC-17', nan, 'Unrated'], dtype=object)
```

```
data = data.dropna(subset = ['duration'])
```

```
data['director'].isnull().sum()
```

```
2634
```

```
data['director'].fillna('Unknown', inplace= True)
data['cast'].fillna('Unknown', inplace= True)
```

```
mode_im = ['date_added','rating','duration']
for i in mode_im:
    data[i] = data[i].fillna(data[i].mode()[0])
```

Two new columns are created for month and year.

```
data['month'] = data['date_added'].apply(lambda x : x.lstrip().split(' ')[0])
data['year'] = data['date_added'].apply(lambda x : x.split(', ')[-1])
```

```
#drop useless columns
data.drop(['show_id','date_added','description'],axis=1, inplace= True)
```

```
type            0
title           0
director        0
cast            0
country         0
release_year    0
rating          0
duration        0
listed_in       0
month           0
year            0
dtype: int64
```

```
data.isna().sum()
```

```
type            0
title           0
director        0
cast            0
country         0
release_year    0
rating          0
duration        0
listed_in       0
month           0
year            0
dtype: int64
```

```
data.head()
```

| | type | title | director | cast | country | release_year | rating | duration | |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Movie | Dick Johnson Is Dead | Kirsten Johnson | Unkown | United States | 2020 | PG-13 | 90 min | Doc |
| **1** | TV Show | Blood & Water | Unkown | Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban... | South Africa | 2021 | TV-MA | 2 Seasons | I TV I |
| | | | | Sami Bouajila, | | | | | |

## Conclusion

I separated the countries in country column. There are wrong values in rating column so i replace it with right values. There are a lot of null values in director we impute the null values with unkown. I used mode() to impute null values with most frequent values in 'country','date_added','rating','duration'. I added month and year columns to the data set. Finally, the dataset now has no null values.

## EDA

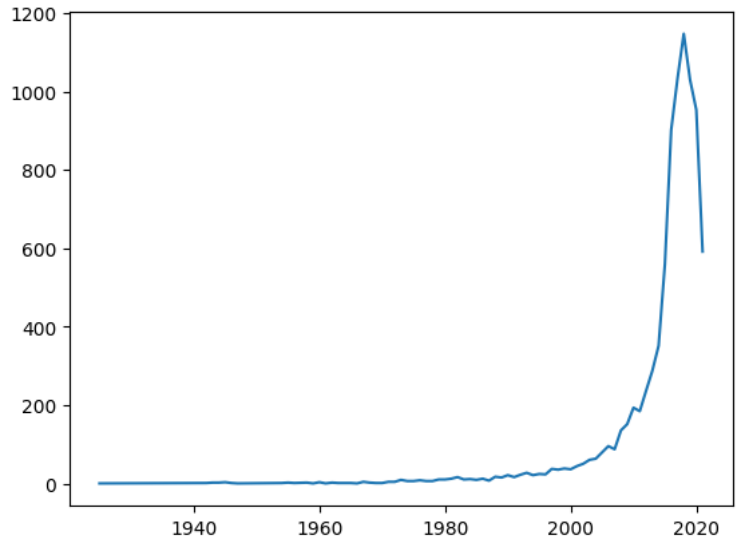
```
data['type'].value_counts()
```

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

```
Year_count = data['release_year'].value_counts()
Year_count
```

```
2018    1147
2017    1032
2019    1030
2020     953
2016     902
        ...
1959       1
1925       1
1961       1
1947       1
1966       1
Name: release_year, Length: 74, dtype: int64
```

```
sns.lineplot(data = Year_count, x = Year_count.index, y = Year_count.values)
plt.show()
```



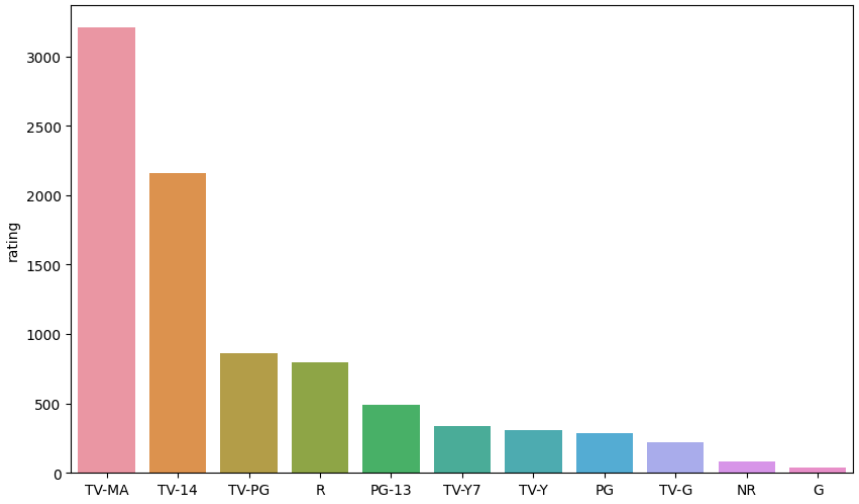

```
data['listed_in'].value_counts().head(20)
```

```
Dramas, International Movies                          362
Documentaries                                         359
Stand-Up Comedy                                       334
Comedies, Dramas, International Movies                274
Dramas, Independent Movies, International Movies      252
Kids' TV                                              220
Children & Family Movies                              215
Children & Family Movies, Comedies                    201
Documentaries, International Movies                   186
Dramas, International Movies, Romantic Movies         180
Comedies, International Movies                        176
Comedies, International Movies, Romantic Movies       152
Dramas                                                138
Dramas, International Movies, Thrillers               134
Action & Adventure, Dramas, International Movies      132
Action & Adventure                                    128
International TV Shows, TV Dramas                     121
Comedies, Dramas, Independent Movies                  116
Crime TV Shows, International TV Shows, TV Dramas     110
Comedies                                              110
Name: listed_in, dtype: int64
```

```
Rating_count = data['rating'].value_counts().head(11)
Rating_count
```

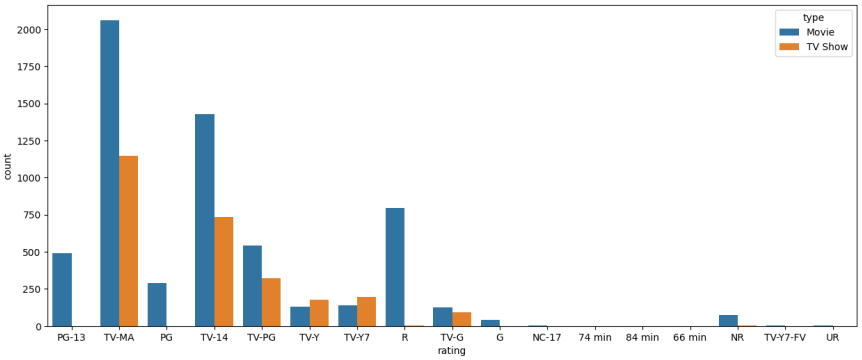
```
TV-MA    3207
TV-14    2160
TV-PG     863
R         799
PG-13     490
TV-Y7     334
TV-Y      307
PG        287
TV-G      220
NR         80
G          41
Name: rating, dtype: int64
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x=Rating_count.index,y=Rating_count)
plt.show()
```




TV-MA, TV-14, TV-PG, R are top ratings in Netflix contents.

```
plt.figure(figsize=(15, 6))
sns.countplot(data = data,x='rating',hue = 'type')
plt.show()
```




```
data['country'].nunique()
```

```
748
```

```
data['country'].value_counts().head(20)
```

```
United States                2818
India                         972
United Kingdom                419
Japan                         245
South Korea                   199
Canada                        181
Spain                         145
France                        124
```

```
Mexico                          110
Egypt                           106
Turkey                          105
Nigeria                          95
Australia                        87
Taiwan                           81
Indonesia                        79
Brazil                           77
Philippines                      75
United Kingdom, United States    75
United States, Canada            73
Germany                          67
Name: country, dtype: int64
```

```
top_10_country = data['country'].value_counts().index[:10]
top_10_country
```

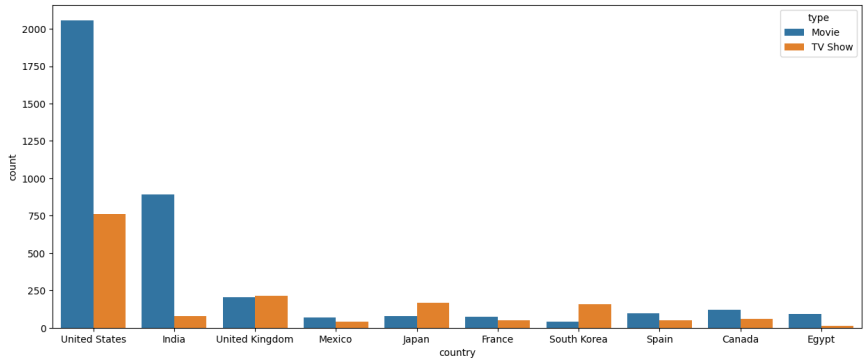
```
Index(['United States', 'India', 'United Kingdom', 'Japan', 'South Korea',
       'Canada', 'Spain', 'France', 'Mexico', 'Egypt'],
      dtype='object')
```

```
top_10 = data.loc[(data['country'].isin(top_10_country))]
top_10.shape
```

```
(5319, 12)
```

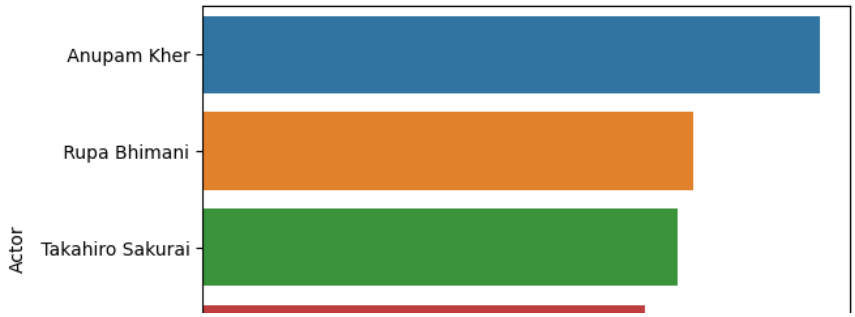
```
plt.figure(figsize=(15, 6))
sns.countplot(data = top_10, x='country',hue='type')
```

```
<Axes: xlabel='country', ylabel='count'>
```



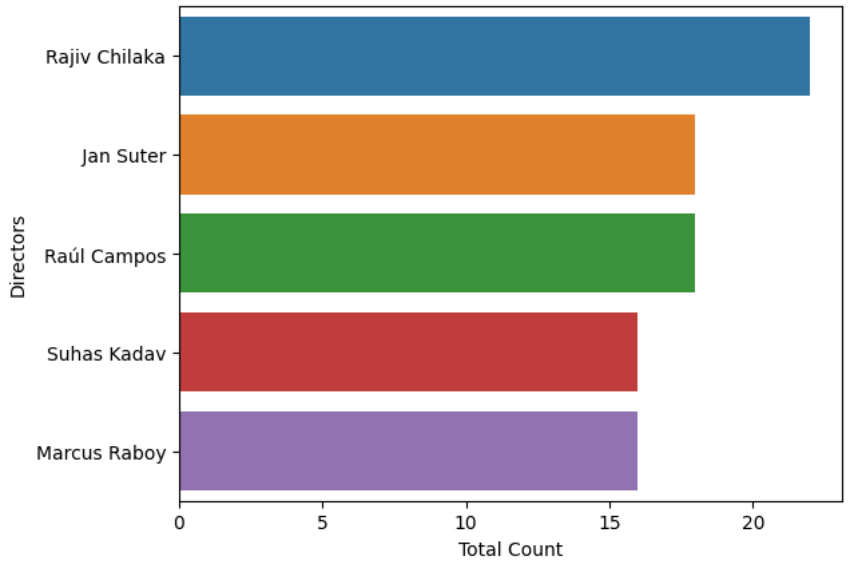

```
cast_df = pd.DataFrame()
cast_df = data['cast'].str.split(',',expand=True).stack()
cast_df = cast_df.to_frame()
cast_df.columns = ['Actor']
actors = cast_df.groupby(['Actor']).size().reset_index(name = 'Total Count')
actors = actors[actors.Actor != 'Unknown']
actors = actors.sort_values(by=['Total Count'], ascending=False)
top5Actors = actors.head()

barChart2 = sns.barplot(top5Actors, x='Total Count', y='Actor')
```




```
cast_df = pd.DataFrame()
cast_df = data['director'].str.split(',',expand=True).stack()
cast_df = cast_df.to_frame()
cast_df.columns = ['Directors']
directors = cast_df.groupby(['Directors']).size().reset_index(name = 'Total Count')
directors = directors[directors.Directors != 'Unknown']
directors = directors.sort_values(by=['Total Count'], ascending=False)
top5directors = directors.head()

barChart2 = sns.barplot(top5directors, x='Total Count', y='Directors')
```




1. Top 5 Popular Cast: Anupam Kher, Rupa Bhimani, Takahiro Sakurai, Julie Tejwani,Om Puri
2. Top 5 Popular Directors: Rajiv Chilaka, Jan Suter, Raúl Campos, Suhas Kadav, Marcus Raboy

```
df1 = data[['type', 'release_year']]
df1 = df1.rename(columns = {"release_year":"Release_Year","type":"Type"})
df2 = df1.groupby(['Release_Year','Type']).size().reset_index(name = 'Total_count')
df2
```

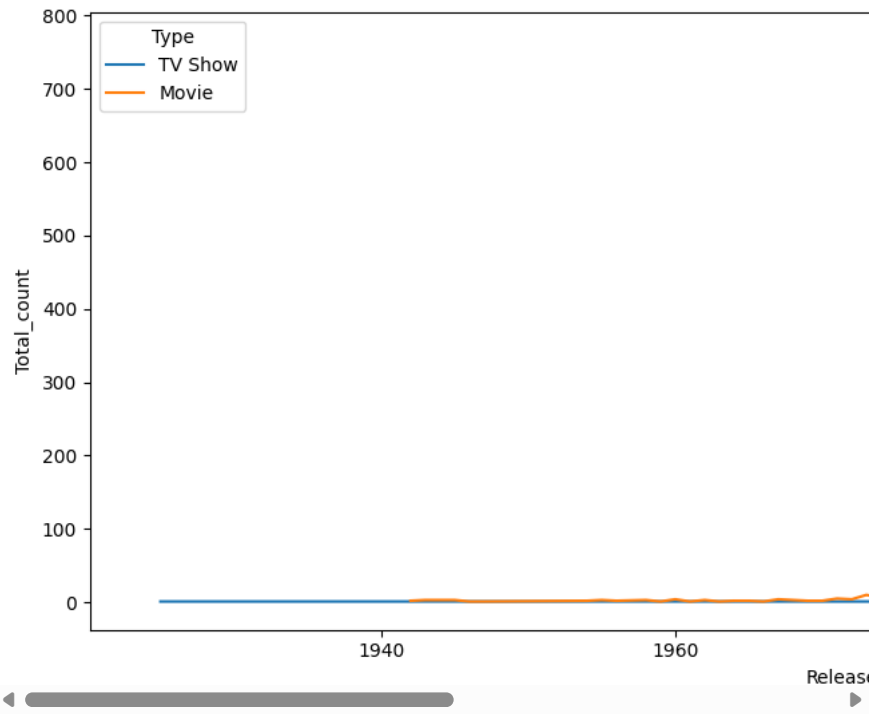
| | Release_Year | Type | Total_count |
|---|---|---|---|
| **0** | 1925 | TV Show | 1 |
| **1** | 1942 | Movie | 2 |
| **2** | 1943 | Movie | 3 |
| **3** | 1944 | Movie | 3 |
| **4** | 1945 | Movie | 3 |
| **...** | ... | ... | ... |
| **114** | 2019 | TV Show | 397 |
| **115** | 2020 | Movie | 517 |
| **116** | 2020 | TV Show | 436 |
| **117** | 2021 | Movie | 277 |
| **118** | 2021 | TV Show | 315 |

119 rows × 3 columns

+ Code + Text

```
plt.figure(figsize=(15, 6))
```

```
plt.figure(figsize=(15, 6))
graph = sns.lineplot(df2, x = "Release_Year", y="Total_count", hue = "Type")
```




```
data['duration']
```

```
0          90 min
1       2 Seasons
2        1 Season
3        1 Season
4       2 Seasons
          ...
8802     158 min
8803    2 Seasons
8804       88 min
8805       88 min
8806      111 min
Name: duration, Length: 8804, dtype: object
```

Create two new columns 'duration_min' and 'duration_seasons' to store minutes for movies and seasons for TV shows.

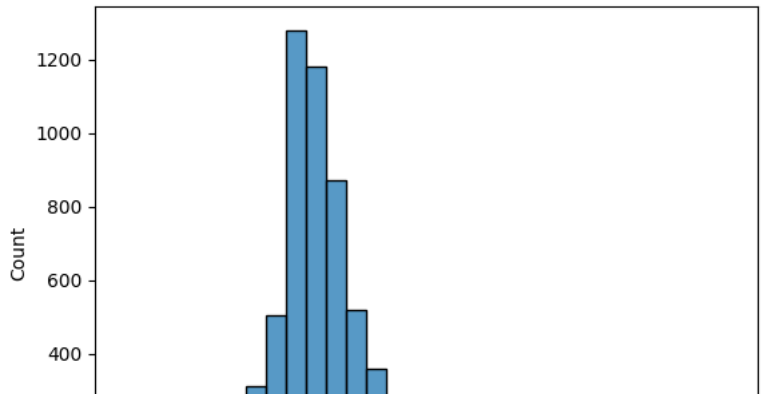
```
data['duration_min'] = data[data['type'] == 'Movie']['duration'].str.extract('(\d+)').astype(float)
data['duration_seasons'] = data[data['type'] == 'TV Show']['duration'].str.extract('(\d+)').astype(float)
# fill NaN values in the new columns with 0
data[['duration_min', 'duration_seasons']] = data[['duration_min', 'duration_seasons']].fillna(0)
data = data.drop('duration', axis = 1)
```

```
data['duration_min']
data['duration_seasons']
```

```
0       0.0
1       2.0
2       1.0
3       1.0
4       2.0
       ...
8802    0.0
8803    2.0
8804    0.0
8805    0.0
8806    0.0
Name: duration_seasons, Length: 8804, dtype: float64
```

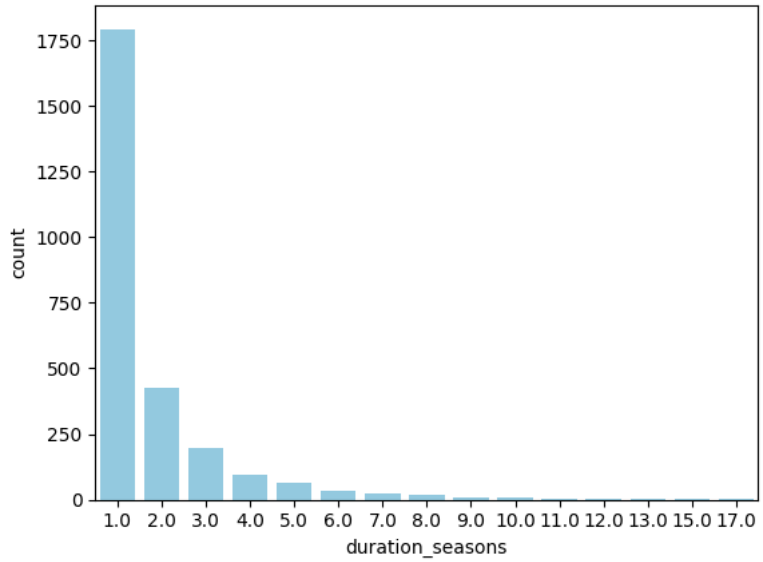
```
sns.histplot(data = data[data['type'] == 'Movie'], x = 'duration_min', bins = 30)
```

```
<Axes: xlabel='duration_min', ylabel='Count'>
```




```
sns.countplot(data = data[data['type'] == 'TV Show'], x = 'duration_seasons',color='skyblue')
```

```
<Axes: xlabel='duration_seasons', ylabel='count'>
```



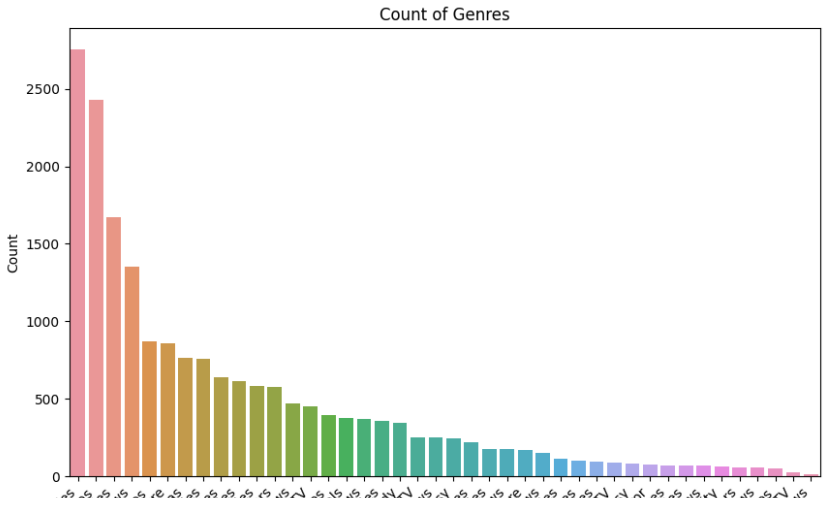

Netflix present more movies than TV shows. Nearly 70% of all productions are movies.

Most movies last for 90 - 120 minutes. Most TV series are new with only 1 or 2 seasons.

There was an increase since 2015 in both TV shows and movies. The number of movies was greater than TV shows before the decrease. Netflix content experienced a sharp decrease in 2019, and the number of TV shows exceeded movies for the first time.

```
genre_counts = data['listed_in'].str.split(', ').explode().value_counts()

# Plot the count of genres using a bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x=genre_counts.index, y=genre_counts.values)
plt.xlabel('Genre')
plt.ylabel('Count')
plt.title('Count of Genres')
plt.xticks(rotation=45, ha='right')
plt.show()
```




international movies, dramas, comedies, international TV shows are very popular genres.

## Business Insights

1. Netflix present more movies than TV shows. Most movies last for 90 - 120 minutes. Most TV series are new with only 1 or 2 seasons. There was an increase since 2015 in both TV shows and movies. The number of movies was greater than TV shows before the decrease. Netflix content experienced a sharp decrease in 2019, and the number of TV shows exceeded movies for the first time.
2. Anupam Kher, Rupa Bhimani, Takahiro Sakurai, Julie Tejwani,Om Puri are
   popular casts and Rajiv Chilaka, Jan Suter, Raúl Campos, Suhas Kadav, Marcus Raboy are popular directors.
3. United States, India and United Kingdom are top countries for movies and TV shows.
4. TV-MA, TV-14, TV-PG, R are top ratings in Netflix contents.
5. International movies, Dramas, Comedies, International TV shows are very popular genres.

## Recommendations

1. According to above analysis,it is advisable for Netflix to prioritize future collaborations with renowned directors, casts, genres and contents.
2. Rajiv Chilaka, Jan Suter are considered as best directors according to
   more number of Movie releases i.e 22, 21 respectively. So these directors movies could help in grow bussiness for netflix
3. International Movies, Dramas, Comedies, are mostly viewed genre in Netflix, So movies with these genre has more demand.
4. United States , India , United Kingdom these countries have the highest releases. So their is no risk of amount burn either in marketing or sales because its already done.
5. Movies having duration between 90-120 mins are recommended.