



```
In [1]: # This Python 3 environment comes with many helpful analytics libraries instal
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/c
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will lis

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

# You can write up to 20GB to the current directory (/kaggle/working/) that ge
# You can also write temporary files to /kaggle/temp/, but they won't be saved

In [2]: df=pd.read_csv('netflix_titles.csv.zip')
```

Data assessing

- Overview ---> head(),sample(),data card,shape,Validity issues,duplicacy check
- Seeking information ---> completeness
- Seeking description ---> accuracy issues,validity issues,completeness
- Distribution ---> numerical columns,categorical(imbalancing) --> feature engg,feature selection

Data visualisation

- Import major libraries
- Univariate analysis
- Bivariate analysis
- Multivariate analysis
- Information
- Insights

----> Report

----> Dashboarding

Data Assessing

About Company

Netflix, Inc. is a global streaming service and production company that provides on-demand movies, TV shows, documentaries, and original content over the internet. Founded in 1997 by Reed Hastings and Marc Randolph in the United States, Netflix started as a DVD rental-by-mail service but transitioned to online streaming in 2007.

Today, Netflix is known for its wide range of content, including popular original series like Stranger Things, The Crown, and Money Heist. It operates in over 190 countries and has millions of subscribers worldwide.

Netflix has also expanded into content production, creating films, documentaries, and series under its Netflix Originals banner, making it both a platform and a studio.

```
In [3]: # overview data  
df.head()
```

Out[3]:

	show_id	type	title	director	cast	country	date_added	release_
--	---------	------	-------	----------	------	---------	------------	----------

0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	:
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	:
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	:
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	:
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	:

In [4]: `# shape`
`df.shape`

Out[4]: (8807, 12)

In [5]: `df.columns`

Out[5]: Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
 'release_year', 'rating', 'duration', 'listed_in', 'description'],
 dtype='object')

Data Card: Netflix Movies & TV Shows Dataset (Sample)

Dataset Name

Netflix Movies and TV Shows – Sample Content Data

Dataset Description

This dataset is a **sample of movies and TV shows available on Netflix**. It contains structured metadata about Netflix content, including **title details, content type, release year, ratings, duration, country of origin, genres, and descriptions**.

The dataset is suitable for **content analysis, exploratory data analysis (EDA), and recommendation-based insights**.

Number of Records

- 5 Netflix titles
-

Number of Features

- 12 attributes
-

Feature Description

Column Name	Description
show_id	Unique identifier assigned to each Netflix title
type	Indicates whether the content is a Movie or a TV Show
title	Name of the movie or TV show
director	Director(s) of the content (may contain missing values)
cast	Main actors and actresses featured in the title
country	Country or countries where the content was produced
date_added	Date when the title was added to Netflix
release_year	Original year of release
rating	Content rating based on audience suitability

Column Name	Description
duration	Runtime in minutes for movies or number of seasons for TV shows
listed_in	Genre(s) or category classification
description	Short summary describing the storyline or theme

```
In [6]: # seeking info
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
```

```
In [7]: # Seeking description
df.describe()
```

```
Out[7]:
```

	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

Accuracy Issues Identified in the Dataset

- Mismatch between `release_year` and `year_added`

Several records show a large gap between the content's release year and the year it was added to the platform. While some lag is expected, extreme differences (e.g., very old release years added recently) should be validated to ensure correctness.

- **Very old `release_year` values**

The minimum `release_year` goes as far back as **1925**, which, although possible, may include data entry errors or misclassified content that needs verification.

- **Inconsistent date precision**

The `date_added` column includes full timestamps, whereas `release_year` and `year_added` are stored only as years. This inconsistency can introduce inaccuracies during time-based comparisons or trend analysis.

- **Potential mis-typed or incorrect year values**

Outlier years far from the interquartile range (e.g., much earlier than 2013 or later than expected) may indicate manual entry mistakes.

- **Missing variability reporting in `date_added`**

The standard deviation for `date_added` is shown as `NaN`, which can hide dispersion-related accuracy insights and suggests the need to recheck data type handling.

- **Possible logical inconsistencies**

In some cases, `year_added` may appear earlier than the actual `release_year`, which is logically incorrect and points to accuracy issues in year-related fields.

Data Quality Observations

- The dataset contains **null (missing) values**.
- The dataset consists of **1 numerical features** and **11 categorical (object) features**.

```
In [8]: # Completeness
df.isnull().sum().sum()
# Percentage
df.isnull().mean()*100
```

```
Out[8]: show_id      0.000000
        type        0.000000
        title       0.000000
        director    29.908028
        cast        9.367549
        country     9.435676
        date_added  0.113546
        release_year 0.000000
        rating      0.045418
        duration    0.034064
        listed_in   0.000000
        description 0.000000
        dtype: float64
```

```
In [9]: df.isnull().sum()
```

```
Out[9]: 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
```

```
In [10]: df['director'] = df['director'].fillna('Not Available')
```

```
In [11]: df['cast'] = df['cast'].fillna('Not Available')
```

```
In [12]: df['country'] = df['country'].fillna('Not Available')
```

```
In [13]: df['rating'] = df['rating'].fillna(df['rating'].mode()[0])
        df['duration'] = df['duration'].fillna(df['duration'].mode()[0])
```

```
In [14]: df.isnull().sum()
```

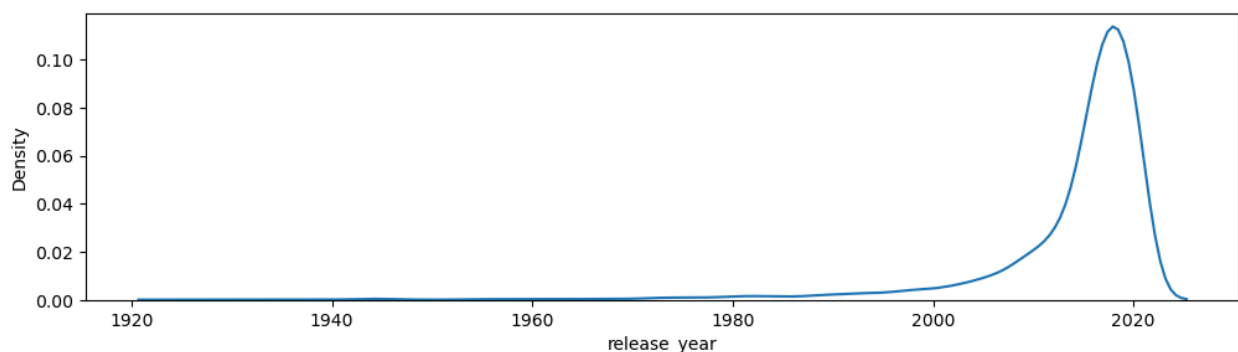
```
Out[14]: show_id      0
         type         0
         title        0
         director     0
         cast         0
         country      0
         date_added   10
         release_year  0
         rating       0
         duration     0
         listed_in    0
         description  0
         dtype: int64
```

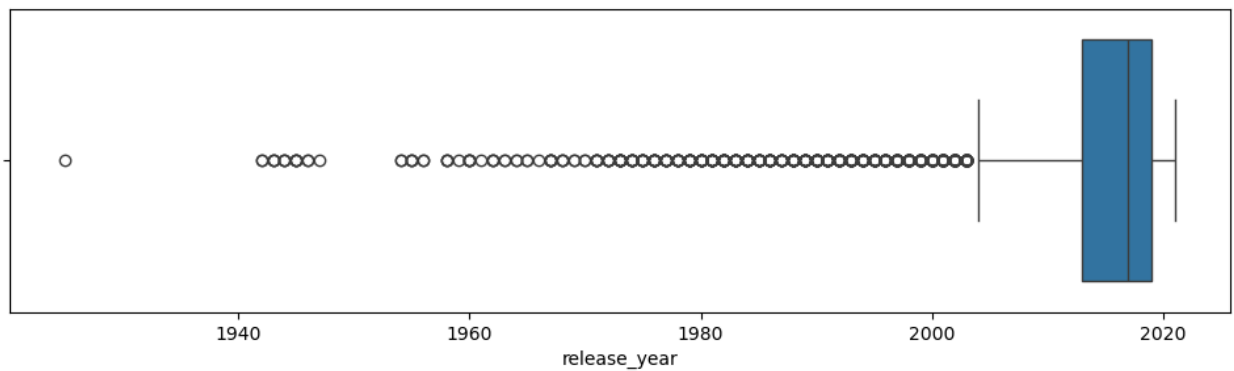
```
In [15]: # importing major libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# additional libraries
import warnings
warnings.filterwarnings('ignore')
```

```
In [16]: df.release_year.describe()
# Visualisation
plt.figure(figsize=(12,3))
sns.kdeplot(data=df,x='release_year')
plt.show()
plt.figure(figsize=(12,3))
sns.boxplot(data=df,x='release_year')
plt.show()

print('skewness',df.release_year.skew())
```

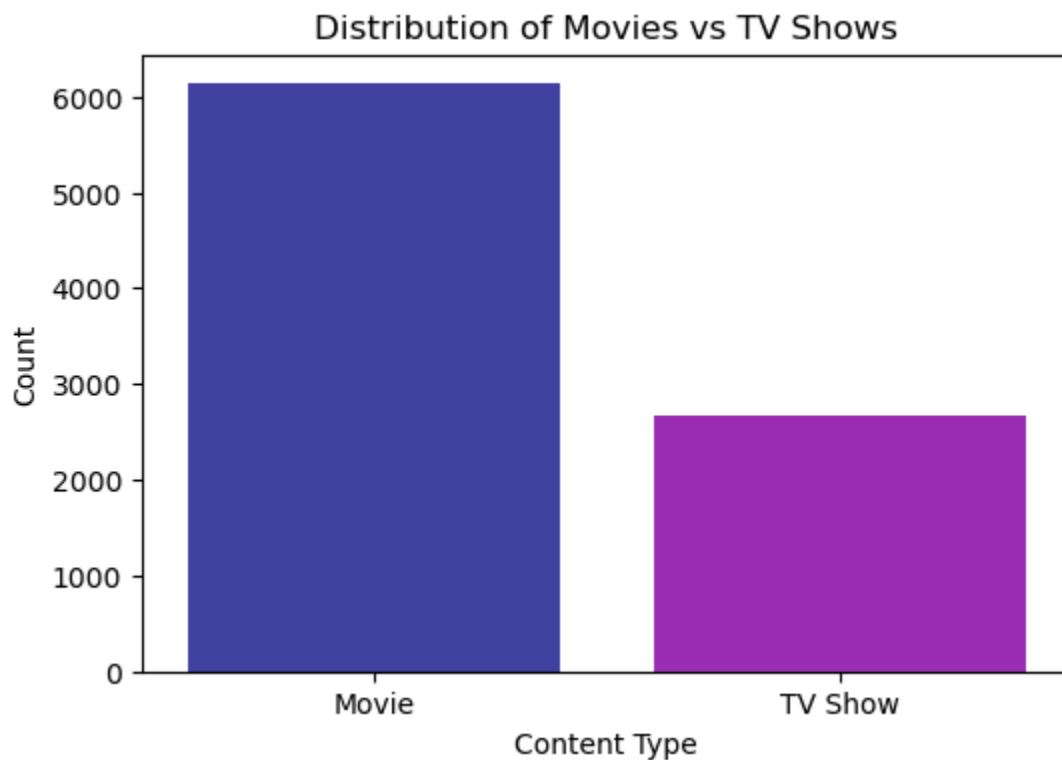




skewness -3.4465650403316013

Univariate Analysis

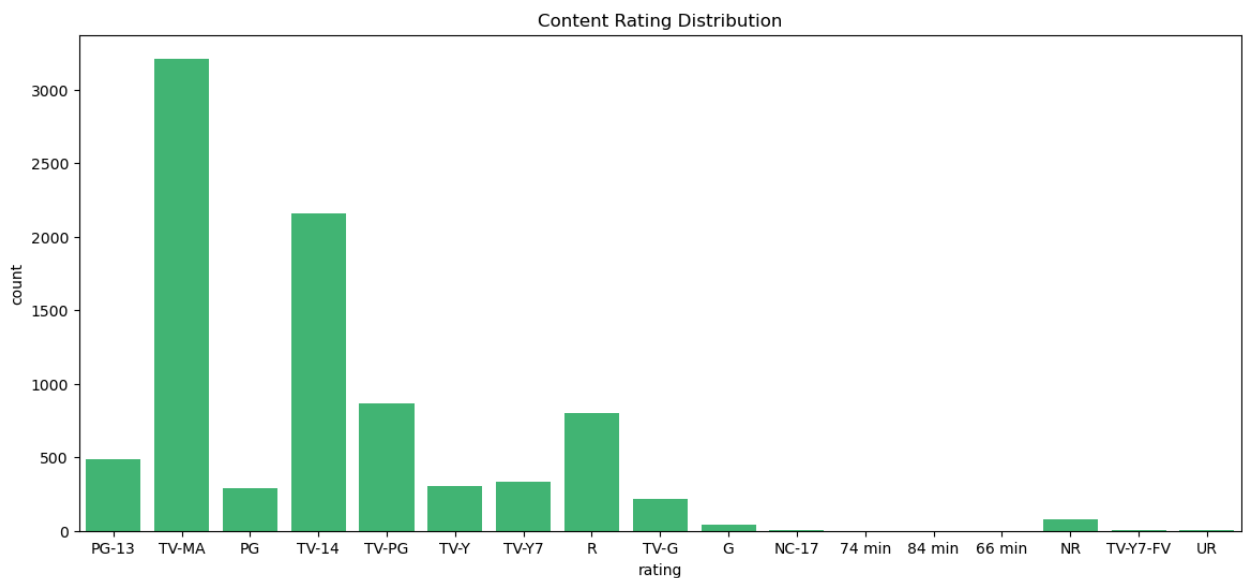
```
In [17]: # Movies vs TV Shows Count
colors=["#3333B0", "#A91BCC"]
plt.figure(figsize=(6,4))
sns.countplot(x='type', hue='type', data=df, palette=colors, legend=False)
plt.title('Distribution of Movies vs TV Shows')
plt.xlabel('Content Type')
plt.ylabel('Count')
plt.show()
```



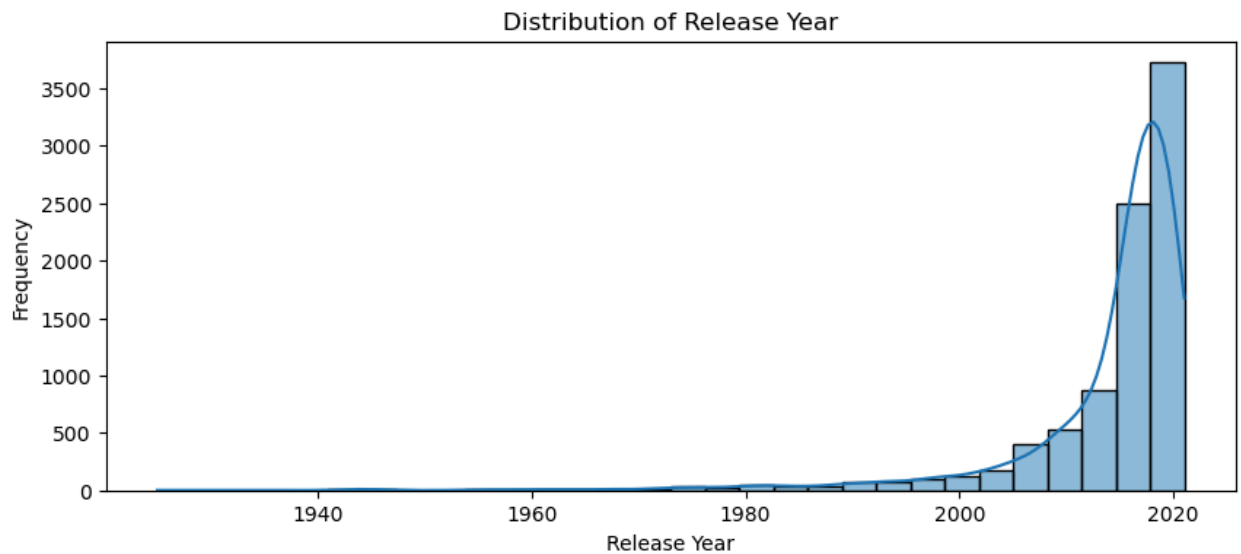
Insight: Movies vs TV Shows

- Movies are available in **much higher numbers** than TV Shows on Netflix.
- This shows a clear preference for **movie-based content** in the platform's catalog.
- TV Shows are fewer, suggesting they are **more selective and production-intensive** compared to movies.
- The distribution reflects Netflix's strategy of offering **wide variety through movies** while maintaining a limited set of series.

```
In [18]: # Rating Distribution
colors=["#2ECC71"]
plt.figure(figsize=(14,6))
sns.countplot(x='rating',hue="rating", data=df,palette=colors,legend=False)
plt.title('Content Rating Distribution')
plt.show()
```



```
In [19]: #Release Year Distribution
plt.figure(figsize=(10,4))
sns.histplot(df['release_year'], bins=30, kde=True)
plt.title('Distribution of Release Year')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.show()
```



```
In [20]: #Pie Chart Visualization
temp = df['rating'].value_counts().reset_index()
temp.columns = ['Rating', 'Count']
temp

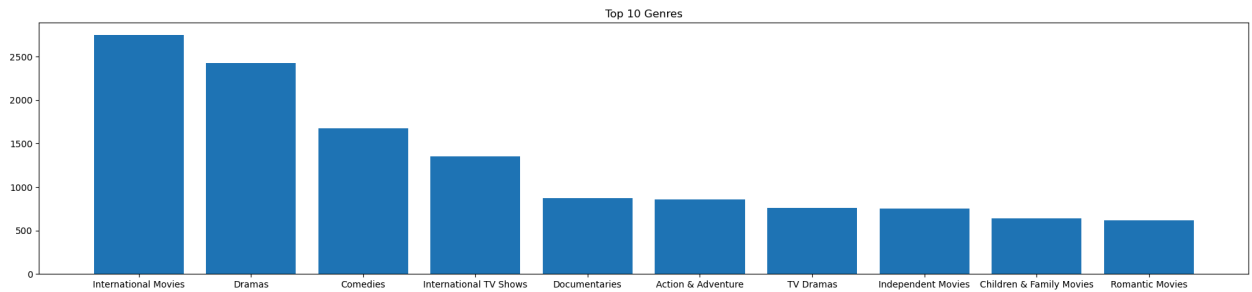
px.pie(temp, names='Rating', values='Count',
        color_discrete_sequence=px.colors.sequential.RdBu,
        height=500,
        title='Distribution of Netflix Titles by Rating')
```

Key Insights

- The majority of Netflix titles have ratings like TV-MA, TV-14, or PG-13, which suggests a large portion of content is targeted toward teens and adults.
- Fewer titles have ratings like NC-17 or G, indicating limited extreme content or children-only content.
- Count plots highlight the absolute number of titles per rating, while pie charts show the relative proportions clearly.
- Imbalances in rating distribution could influence recommendation strategies or content targeting.

```
In [21]: #Top 10 Genres
genres = df['listed_in'].str.split(', ').explode().value_counts().head(10)
plt.figure(figsize=(24,5))
plt.bar(genres.index, genres.values)
```

```
plt.title('Top 10 Genres')
plt.show()
```



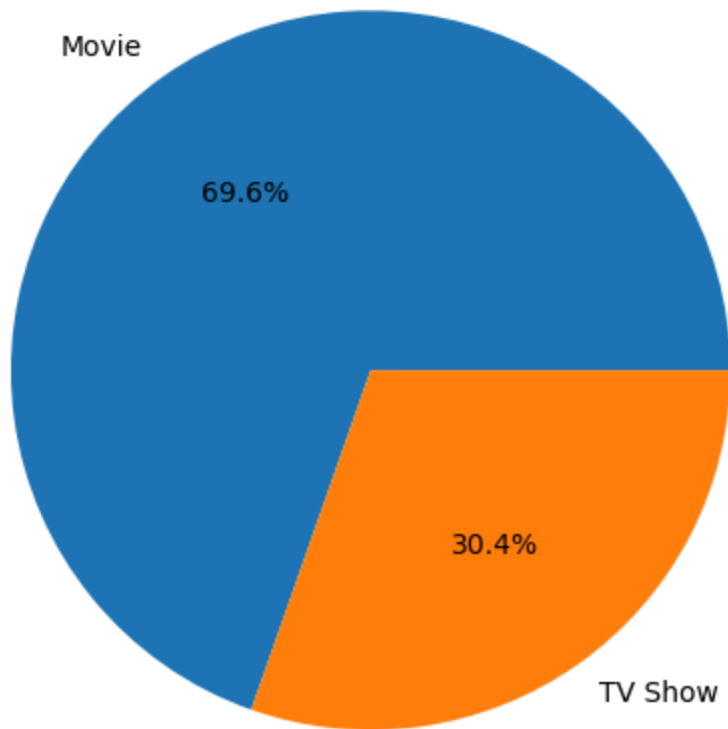
Top 10 Genres – Key Insights

- **International Movies and Dramas dominate** the Netflix content library, showing a strong focus on global and story-driven content.
- **Comedies and Action & Adventure** remain consistently popular, indicating high demand for mass-appeal entertainment.
- **International TV Shows** rank high, highlighting Netflix's expansion into regional and non-English markets.
- The genre distribution is **highly imbalanced**, with a few genres contributing to the majority of available content.

```
In [22]: #Movies vs TV Shows (Pie Chart)
type_counts = df['type'].value_counts()

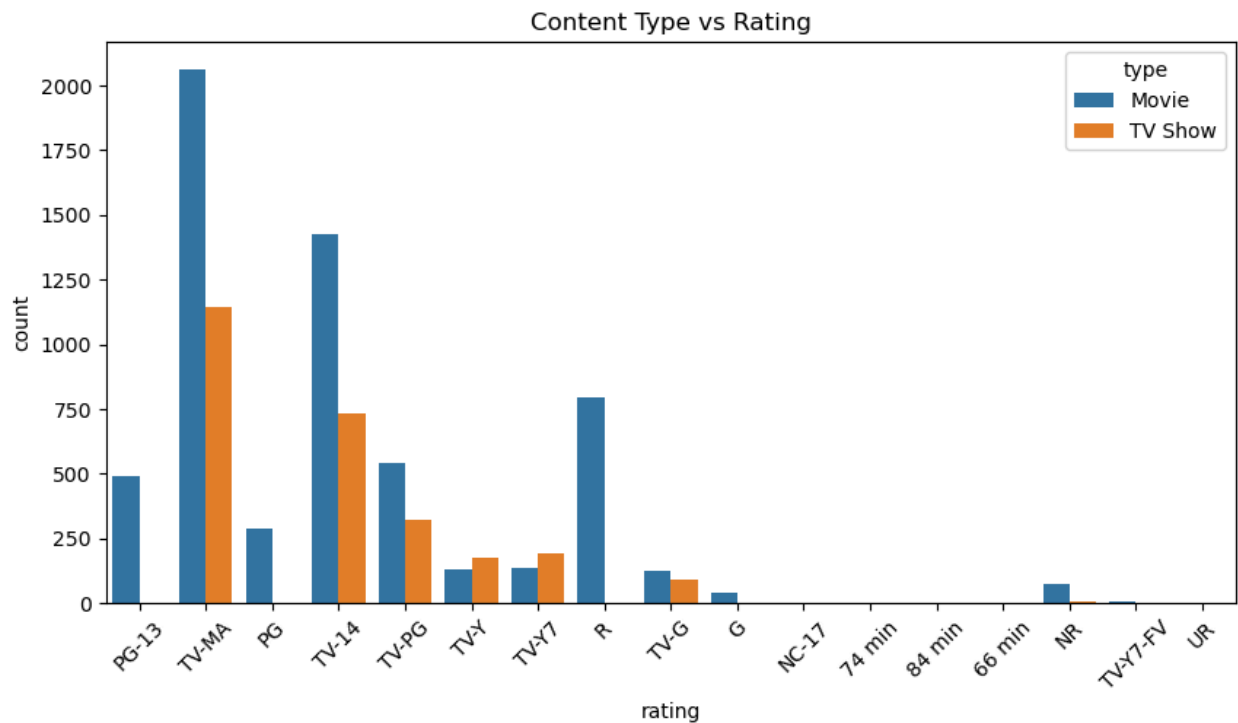
plt.figure(figsize=(5,5))
plt.pie(type_counts.values, labels=type_counts.index, autopct='%1.1f%%')
plt.title('Movies vs TV Shows Share')
plt.tight_layout()
plt.show()
```

Movies vs TV Shows Share



Bivariate Analysis

```
In [23]: #Content type Vs Rating
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='rating', hue='type')
plt.title('Content Type vs Rating')
plt.xticks(rotation=45)
plt.show()
```



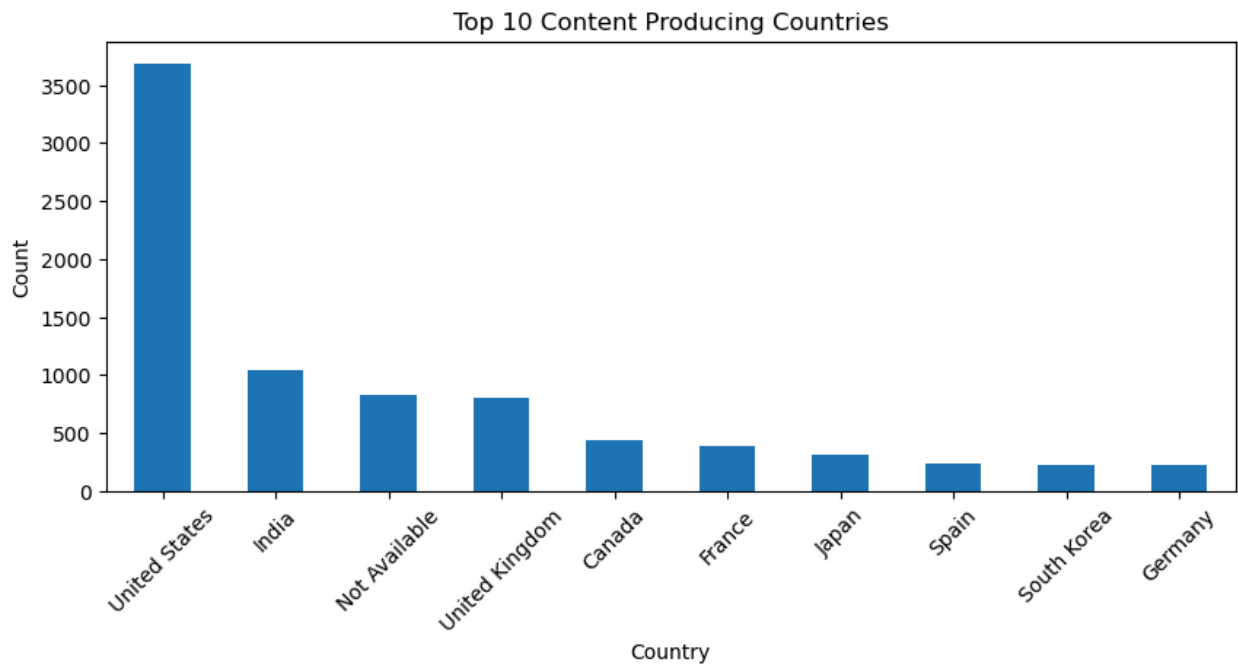
In [24]: `df.info()`

```
<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      8797 non-null   object
7   release_year    8807 non-null   int64
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: int64(1), object(11)
memory usage: 825.8+ KB
```

In [25]: `#plotly`
`# Release year vs duration`
`px.scatter(df, x='release_year', y='duration', color='type',`
 `hover_data=['title','country','rating'], height=500)`

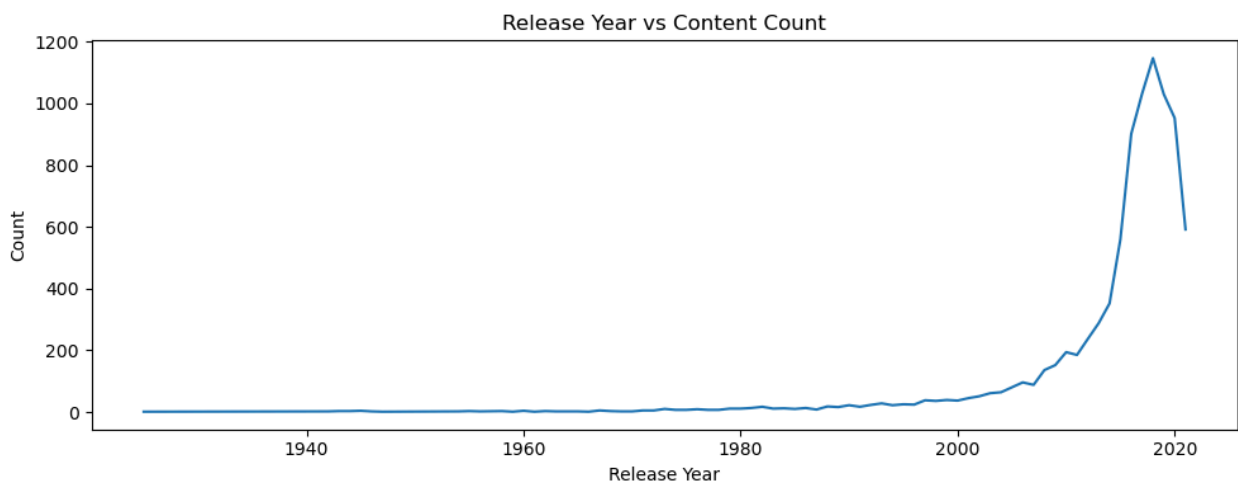
```
In [26]: #Top 10 Countries by content count
top_countries = df['country'].str.split(', ').explode().value_counts().head(10)

plt.figure(figsize=(10,4))
top_countries.plot(kind='bar')
plt.title('Top 10 Content Producing Countries')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

```
In [27]: #Release Year vs Content Count
year_counts = df['release_year'].value_counts().sort_index()

plt.figure(figsize=(10,4))
plt.plot(year_counts.index, year_counts.values)
plt.title('Release Year vs Content Count')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.tight_layout()
plt.show()
```

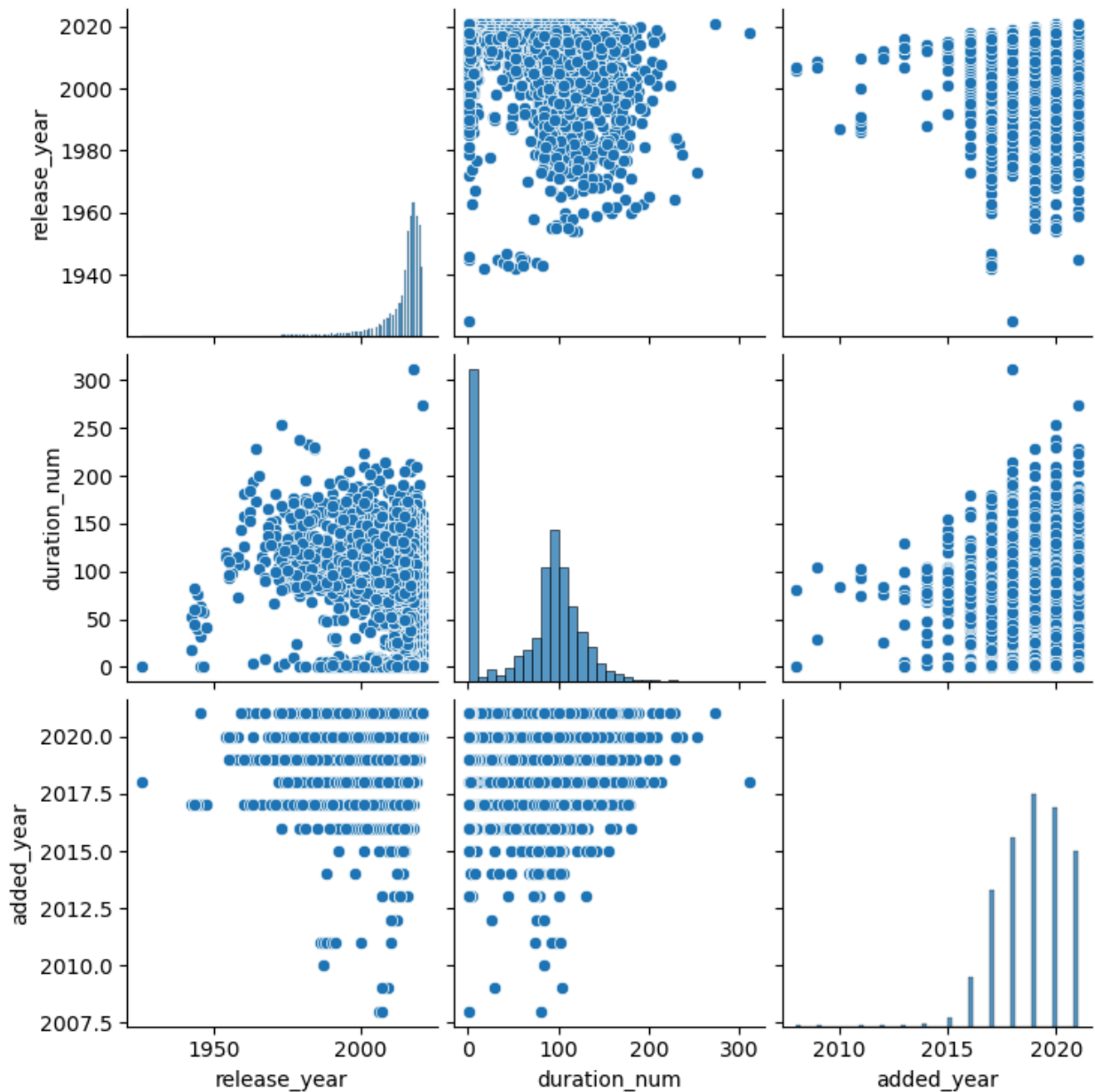


Multivariate Analysis

```
In [28]: # multivariate analysis
df['duration_num'] = df['duration'].str.extract('(\d+)').astype(float)
```

```
df['added_year'] = pd.to_datetime(df['date_added'], errors='coerce').dt.year

sns.pairplot(df)
plt.show()
```



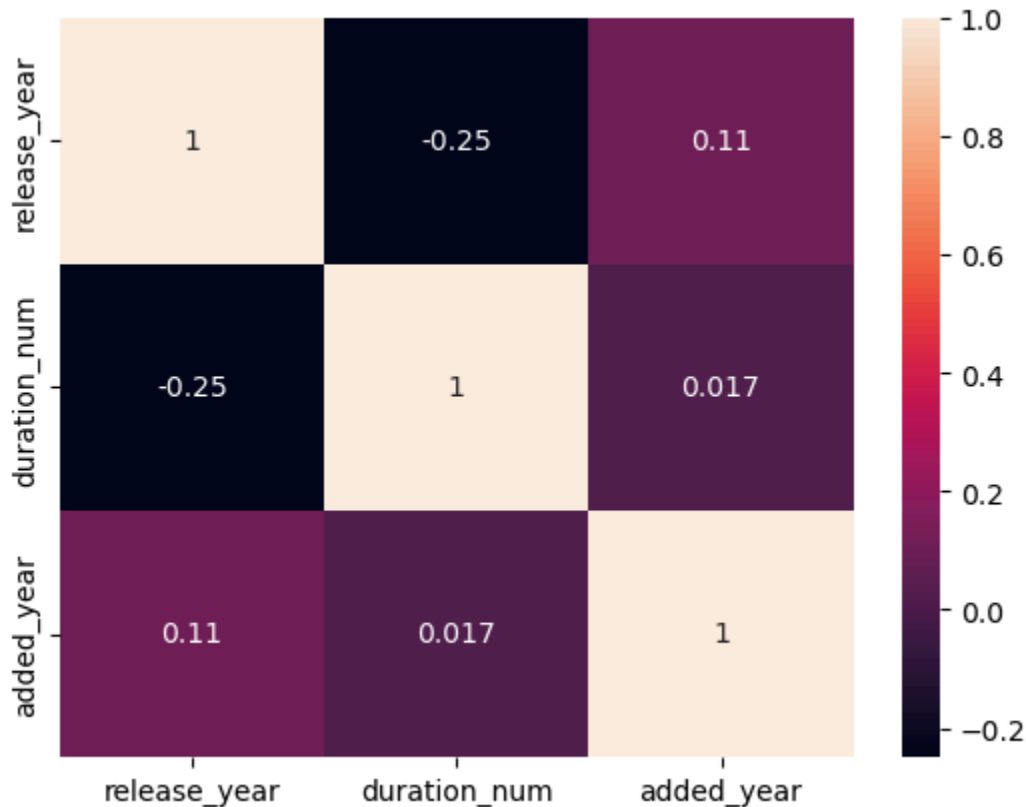
Key Insights

- **release_year vs year_added:** Many older shows/movies were added recently, which shows a lag between production and platform addition.
- **duration_numeric vs release_year:** Movie durations don't vary much with release year, but TV shows (number of seasons) can show trends

over time.

- Correlation checks: Pairplots help spot correlations and clusters visually, which can inform content analysis or recommendation strategies.

```
In [29]: # corr
# heat map
sns.heatmap(df[['release_year', 'duration_num', 'added_year']].corr(), annot=True,
plt.show())
```



Key Insights from the Correlation Heatmap

release_year vs year_added :

- Slight positive correlation (if any) indicates that older content was often added much later.
- Many classic titles were added recently, so the correlation is not very strong.

release_year vs duration_numeric :

- Very low or near-zero correlation, meaning movie length doesn't significantly change with release year.

year_added vs duration_numeric :

- Also low correlation, showing that Netflix adds content of varying durations regardless of the year it was added.

Overall:

- Netflix content addition is more driven by platform strategy than by production year or content length.
- No strong correlations exist between numeric features, which is expected because most of the dataset is categorical.

```
In [30]: temp = df['rating'].value_counts().reset_index()
temp.columns = ['rating', 'count']

px.pie(temp, names='rating', values='count',
        title='Distribution of Ratings on Netflix',
        hole=0.4,
        color_discrete_sequence=px.colors.sequential.Plasma, height=500).show()

temp2 = df['rating'].value_counts().reset_index()
temp2.columns = ['rating', 'count']

px.bar(temp2, x='count', y='rating',
        title='Number of Netflix Titles by Rating',
        color_discrete_sequence=px.colors.sequential.RdBu).show()
```


Insights from Netflix Rating Distribution

Pie Chart: Distribution of Ratings

- Most Netflix titles are rated **TV-MA**, **TV-14**, or **PG-13**, indicating a focus on content for teens and adults.
- Fewer titles are rated **G** or **NC-17**, showing limited content for children or extreme adult content.
- The pie chart clearly shows the proportion of each rating category.

Bar Chart: Number of Titles by Rating

- **TV-MA** has the highest number of titles, reflecting Netflix's emphasis on mature content.
- Ratings like **NC-17** and **G** have very few titles, highlighting content imbalance.
- This can inform content recommendations and production strategy.

Optional: Median Release Year by Rating

- Older ratings like **G** tend to have earlier release years.
- Most adult or teen-rated content (**TV-MA, TV-14, PG-13**) were produced in recent years.
- Shows Netflix's strategy of updating the library with modern content for the main audience.

Conclusion

The analysis of Netflix ratings shows that the platform primarily focuses on **teen and adult audiences**, with the majority of content rated **TV-MA, TV-14, or PG-13**. There is a clear **imbalance in rating distribution**, as very few titles are aimed at children (G) or extreme adult content (NC-17).

From a temporal perspective, most popular ratings correspond to **recently produced content**, while older ratings like G appear mostly in classic titles. This indicates that Netflix's content strategy prioritizes **modern, mature, and teen-oriented content**, likely to match viewer demand and engagement trends.

Overall, Netflix's library is curated more by **audience targeting and content relevance** than by maintaining a balanced representation across all rating categories.

In []: