Business Case: Netflix - Data Exploration and Visualisation

1. Defining Problem Statement and identifying basic metrics to analyze:

Problem statement: Netflix is one of the most popular media and video streaming platforms. They have over 10000 movies or tv shows available on their platform, as of mid-2021, they have over 222M Subscribers globally. Conduct a comprehensive analysis of the Netflix dataset to provide actionable insights for content production and business growth. Address key questions related to genre popularity, regional content preferences, content consumption trends, content duration's impact on viewership, user ratings and recommendations, demographic targeting, keyword analysis, release month influence, actor/director association, and growth opportunities through TV shows vs. movies comparison.

Components of the Problem:

i. **Genre Popularity and Regional Preferences:**

- Identify popular genres globally and regionally to guide content production strategy.

ii. **Content Contribution by Countries:**

- Determine the countries contributing the most content to Netflix for targeted regional focus.

iii. **Content Consumption Trends over Time:**

- Explore trends in TV show and movie consumption over the past decades to anticipate future viewer behavior.

iv. **Impact of Content Duration on Viewership:**

- Investigate the relationship between content duration and viewership, considering its impact on user ratings

v. **Ratings and its impact:**

- Analyze ratings to provide data-driven insights on the types of shows or movies to launch.

vi. **Demographic Analysis for Targeted Content:**

- Understand how user demographics influence content preferences for targeted and personalized production.

vii. **Keyword Analysis:**

- Identify common keywords or themes in show titles and descriptions to inform content themes and trends.

viii. **Release Month Influence:**

- Investigate if there is a relationship between the release month and show popularity, guiding strategic release timing.

ix. **Actor/Director Association with Highly Rated Shows:**

- Explore which actors or directors are consistently associated with highly-rated content.

x. **Uncovering Growth opportunities:**

- Assess growth opportunities by comparing TV shows vs. movies to determine focus areas for expansion.

Identifying Basic Metrics to Analyze:

i. Genre Popularity and Regional Preferences:

- Genre distribution globally and by region.

ii. Content Contribution by Countries:

- Distribution of content by countries and their contribution percentages.

iii. Content Consumption Trends over Time:

- Trends in TV show and movie consumption over the last 20-30 years.

iv. Impact of Content Duration on Viewership:

- Correlation between content duration and viewership, along with average duration metrics.

v. Ratings and it's impact:

- The distribution of ratings for Netflix content and the trend of various ratings over the last 40 years

vi. Demographic Analysis for Targeted Content:

- Relationship between user demographics (age, gender) and content preferences.

vii. Keyword Analysis:

- Commonly occurring keywords or themes in show titles and descriptions.

viii. **Release Month Influence:**

- Seasonal viewership patterns and optimal release timing analysis.

ix. **Actor/Director Association with Highly Rated Shows:**

- Metrics on actors or directors consistently associated with highly-rated shows.

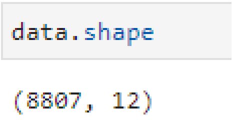
x. **Uncovering Growth opportunities:**

- Comparative analysis of TV shows vs. movies in terms of user ratings, viewership, and growth potential.

2. Observations on the Data:

I have done observations on the shape of the data, data types of all attributes, conversion of categorical attributes to 'category' (if required), missing value detection, and statistical summary.

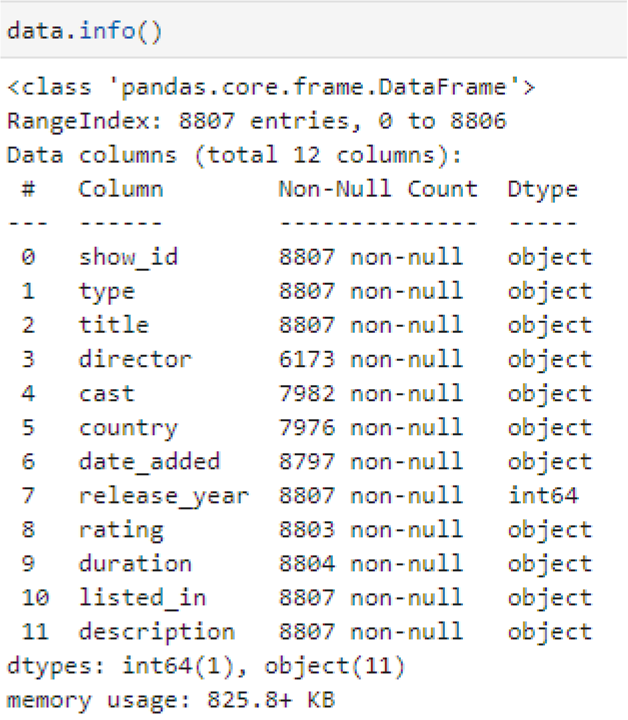
a. Shape of data: The shape of data is :



```
data.shape
```

```
(8807, 12)
```

b. **Information about the dataset:** The basic information about the dataset like the columns, data types, null values is taken.

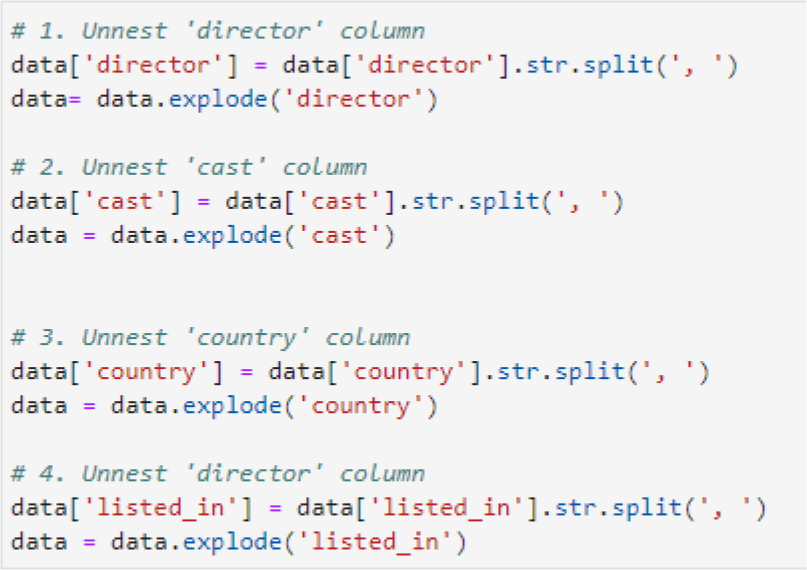


```
data.info()
```

```
<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
```

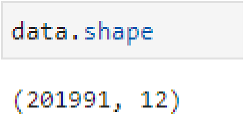
c. Conversion of categorical attributes to ‘category(if required) : For now, no explicit need for converting attributes to 'category' since most are text-based. However, if any categorical attributes are identified, they can be converted for more efficient memory usage later on.

d. **Preprocessing of data:** It involves unnesting of the data in columns like cast , director , listed_in . Followed by handling of missing values .



```
# 1. Unnest 'director' column
data['director'] = data['director'].str.split(', ')
data= data.explode('director')

# 2. Unnest 'cast' column
data['cast'] = data['cast'].str.split(', ')
data = data.explode('cast')


# 3. Unnest 'country' column
data['country'] = data['country'].str.split(', ')
data = data.explode('country')

# 4. Unnest 'director' column
data['listed_in'] = data['listed_in'].str.split(', ')
data = data.explode('listed_in')
```

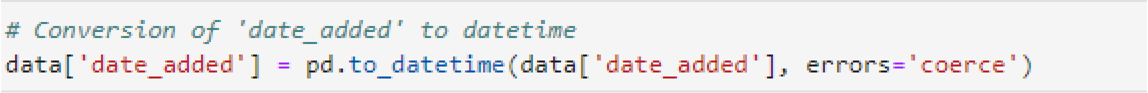
After unnesting the shape of data has changed to



```
data.shape
```

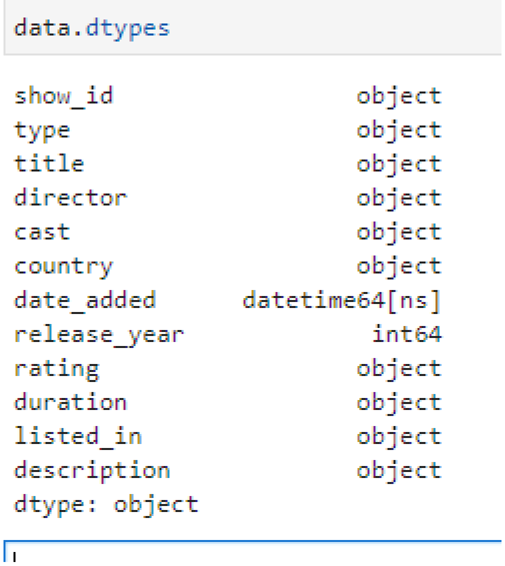
```
(201991, 12)
```

Changed the data type of date_added column



```
# Conversion of 'date_added' to datetime
data['date_added'] = pd.to_datetime(data['date_added'], errors='coerce')
```

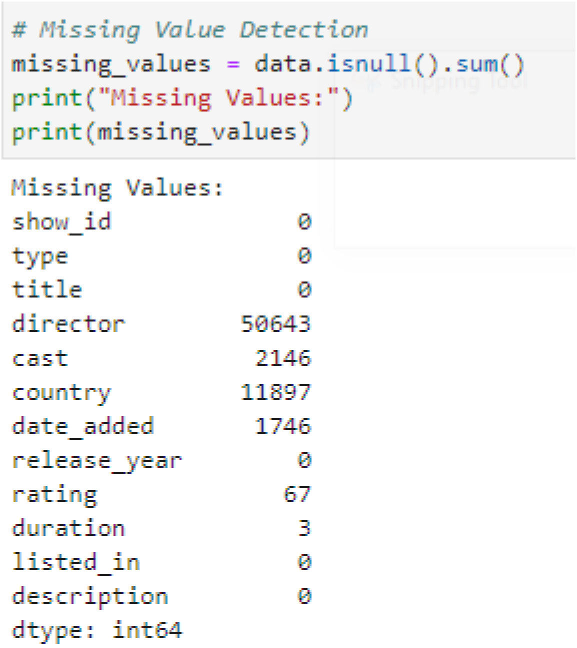
After changing the data type of date column we have checked the columns of dataset and their datatypes:



```
data.dtypes
```

```
show_id                 object
type                    object
title                   object
director                object
cast                    object
country                 object
date_added      datetime64[ns]
release_year             int64
rating                  object
duration                object
listed_in               object
description             object
dtype: object
```

e. Handling Missing values: First checked the dataset for the missing values.

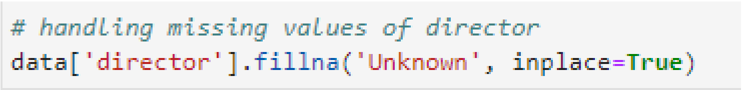


```
# Missing Value Detection
missing_values = data.isnull().sum()
print("Missing Values:")
print(missing_values)
```

```
Missing Values:
show_id             0
type                0
title               0
director        50643
cast             2146
country         11897
date_added       1746
release_year        0
rating             67
duration            3
listed_in           0
description         0
dtype: int64
```

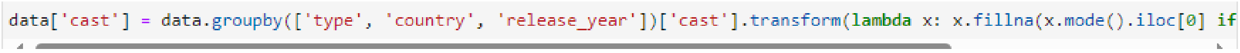
We can see that for the columns of director, cast, country, date_added, rating and duration we have missing values in the dataset. Now one by one all these missing values will be taken care off.

- Director column: First we dealt with director column. Here filled the missing values with the string ‘unknown’.



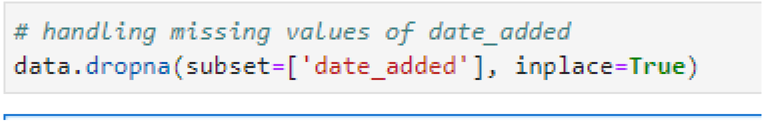
```
# handling missing values of director
data['director'].fillna('Unknown', inplace=True)
```

- Cast Column: Now I have used the explicit imputation for filling in the missing values of cast. This computation was done the basis of type, country and release year to make more relevant while filling the missing values.



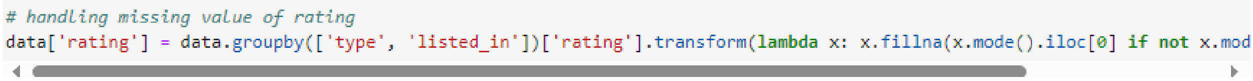
```
data['cast'] = data.groupby(['type', 'country', 'release_year'])['cast'].transform(lambda x: x.fillna(x.mode().iloc[0] if
```

- Date_added column: For this column I have dropped the missing values as they were less in number also making a judge on date will not be that appropriate.



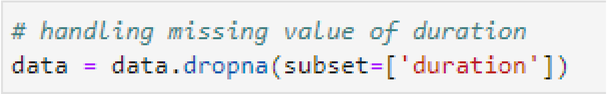
```
# handling missing values of date_added
data.dropna(subset=['date_added'], inplace=True)
```

- Rating column: For rating column too I have used the explicit imputation for filling in the missing values. This computation was done the basis of type and listed_in to make more relevant while filling the missing values.



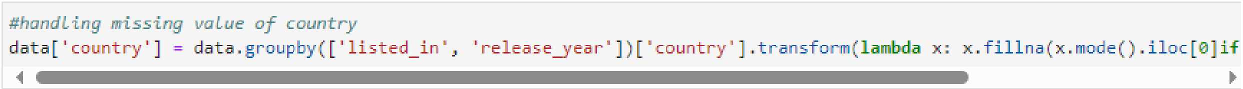
```
# handling missing value of rating
data['rating'] = data.groupby(['type', 'listed_in'])['rating'].transform(lambda x: x.fillna(x.mode().iloc[0] if not x.mod
```

- Duration column: For duration column I have dropped the missing values as there were only 3 values.



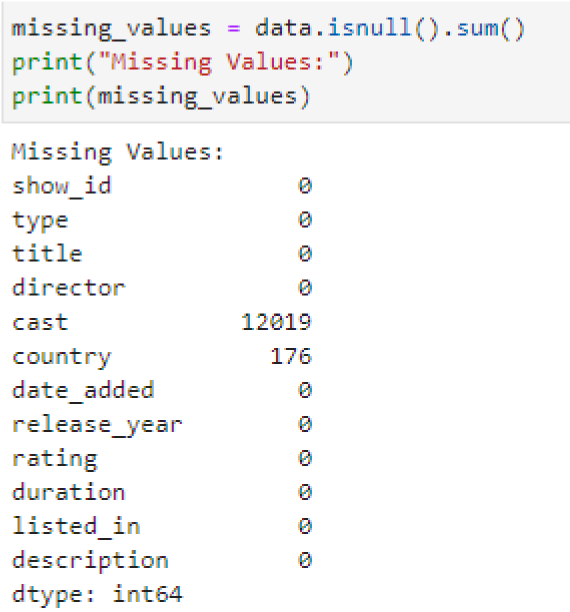
```
# handling missing value of duration
data = data.dropna(subset=['duration'])
```

- Country Column: For country column also I have used computation of listed_in and release year.



```
#handling missing value of country
data['country'] = data.groupby(['listed_in', 'release_year'])['country'].transform(lambda x: x.fillna(x.mode().iloc[0]if
```

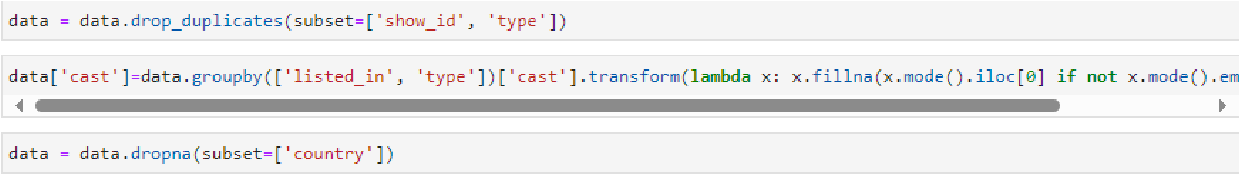
Since we have used imputations while handling missing values there are some duplicates and some new missing values introduced sonow going to handle them.



```
missing_values = data.isnull().sum()
print("Missing Values:")
print(missing_values)
```

```
Missing Values:
show_id             0
type                0
title               0
director            0
cast            12019
country           176
date_added          0
release_year        0
rating              0
duration            0
listed_in           0
description         0
dtype: int64
```

Deleting duplicates and handling new missing values introduced:

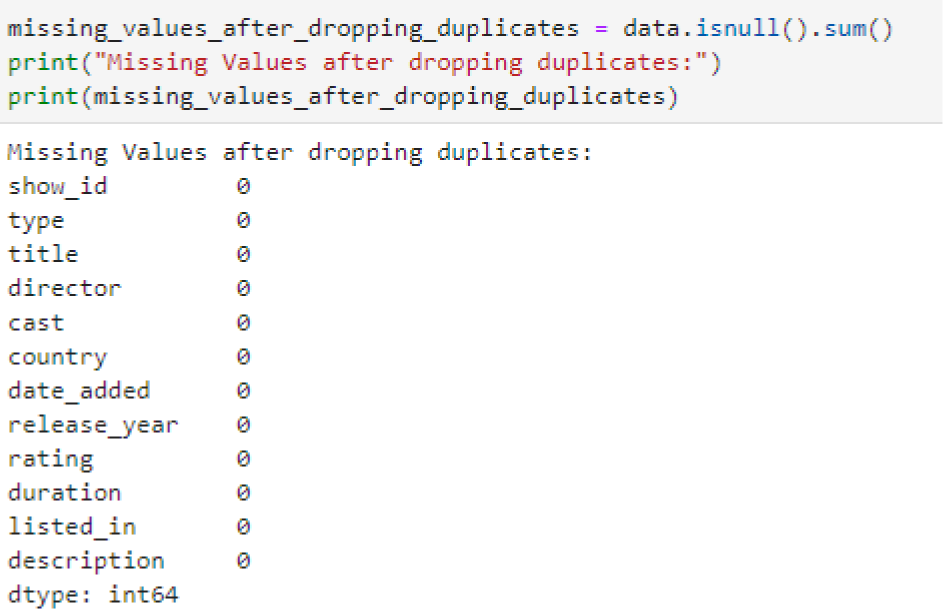


```
data = data.drop_duplicates(subset=['show_id', 'type'])
```

```
data['cast']=data.groupby(['listed_in', 'type'])['cast'].transform(lambda x: x.fillna(x.mode().iloc[0] if not x.mode().em
```

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

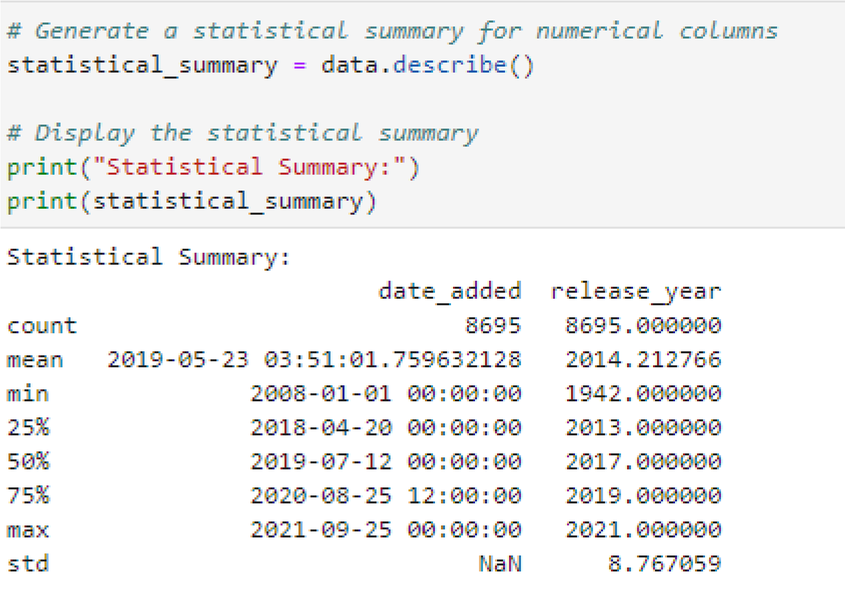
Now there are no missing values.



```
missing_values_after_dropping_duplicates = data.isnull().sum()
print("Missing Values after dropping duplicates:")
print(missing_values_after_dropping_duplicates)
```

```
Missing Values after dropping duplicates:
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

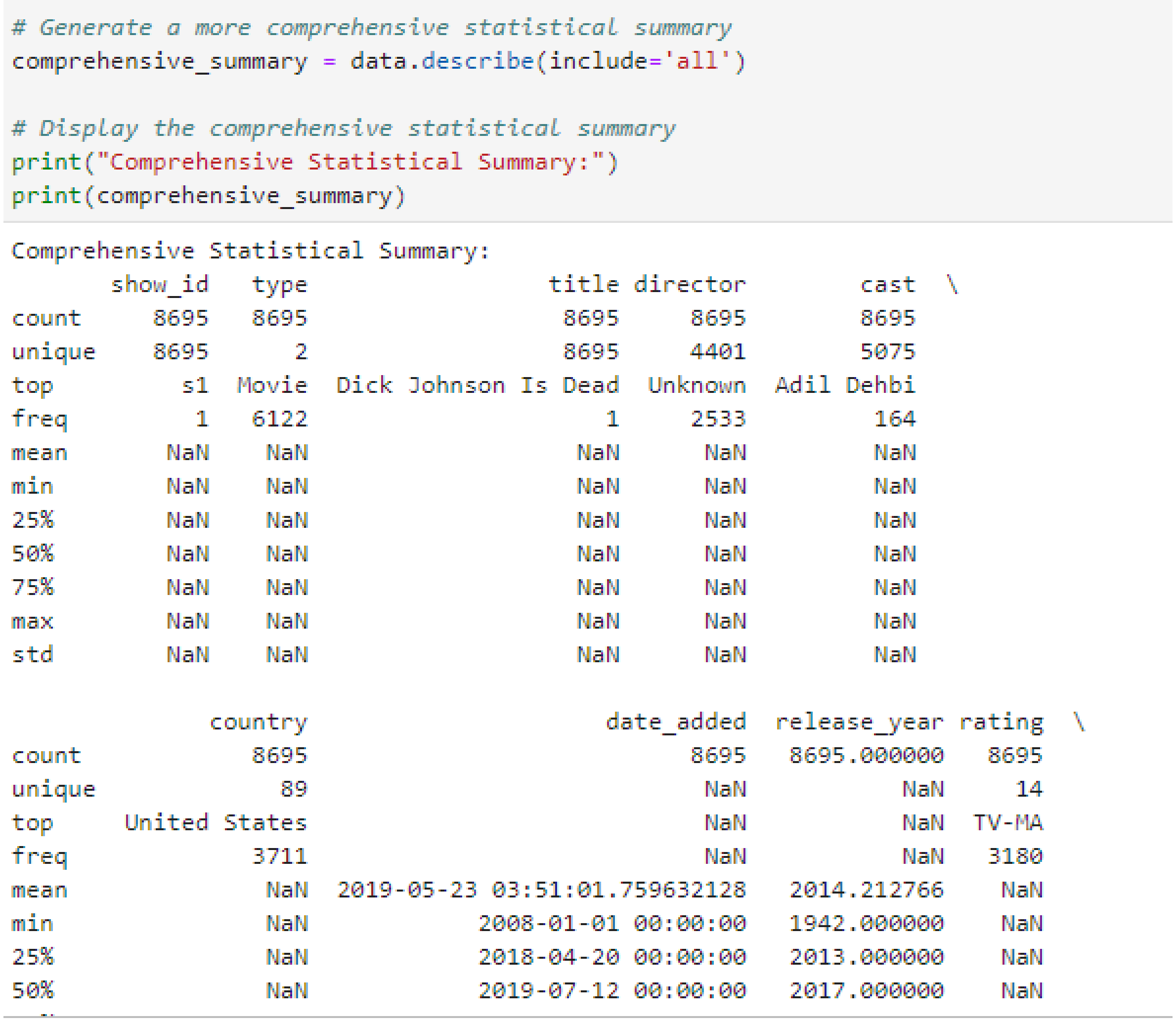
f. Statistical Summary: Since we have only two numerical columns so there is nothing much to show in statistical summary. Those columns are also not relevant for this summary.



```python
# Generate a statistical summary for numerical columns
statistical_summary = data.describe()

# Display the statistical summary
print("Statistical Summary:")
print(statistical_summary)
```

```
Statistical Summary:
                          date_added  release_year
count                           8695   8695.000000
mean   2019-05-23 03:51:01.759632128   2014.212766
min              2008-01-01 00:00:00   1942.000000
25%              2018-04-20 00:00:00   2013.000000
50%              2019-07-12 00:00:00   2017.000000
75%              2020-08-25 12:00:00   2019.000000
max              2021-09-25 00:00:00   2021.000000
std                              NaN      8.767059
```

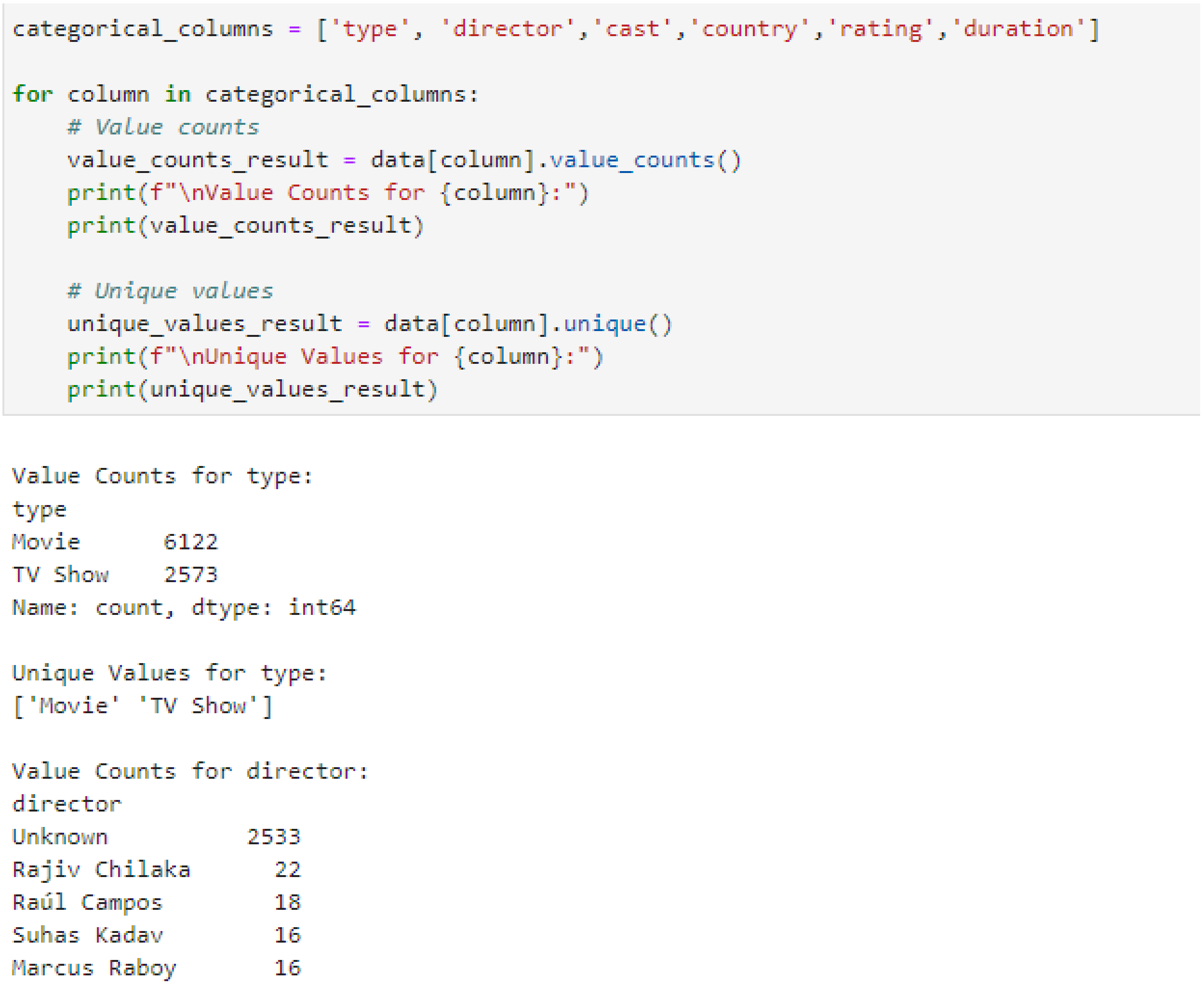
So computed comprehensive statistics.



```python
# Generate a more comprehensive statistical summary
comprehensive_summary = data.describe(include='all')

# Display the comprehensive statistical summary
print("Comprehensive Statistical Summary:")
print(comprehensive_summary)
```

```
Comprehensive Statistical Summary:
       show_id   type                 title director        cast  \
count     8695   8695                  8695     8695        8695
unique    8695      2                  8695     4401        5075
top         s1  Movie  Dick Johnson Is Dead  Unknown  Adil Dehbi
freq         1   6122                     1     2533         164
mean       NaN    NaN                   NaN      NaN         NaN
min        NaN    NaN                   NaN      NaN         NaN
25%        NaN    NaN                   NaN      NaN         NaN
50%        NaN    NaN                   NaN      NaN         NaN
75%        NaN    NaN                   NaN      NaN         NaN
max        NaN    NaN                   NaN      NaN         NaN
std        NaN    NaN                   NaN      NaN         NaN

              country                     date_added  release_year rating  \
count            8695                           8695   8695.000000   8695
unique             89                            NaN           NaN     14
top     United States                            NaN           NaN  TV-MA
freq             3711                            NaN           NaN   3180
mean              NaN  2019-05-23 03:51:01.759632128   2014.212766    NaN
min               NaN            2008-01-01 00:00:00   1942.000000    NaN
25%               NaN            2018-04-20 00:00:00   2013.000000    NaN
50%               NaN            2019-07-12 00:00:00   2017.000000    NaN
```

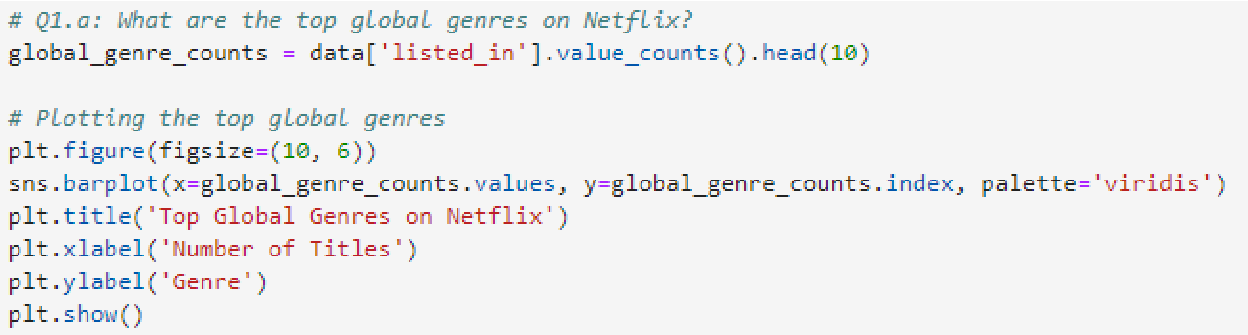
3. **Non- Graphical Analysis:** In the next step done the non graphical analysis which includes getting the unique data of all columns as well as getting the count of them.



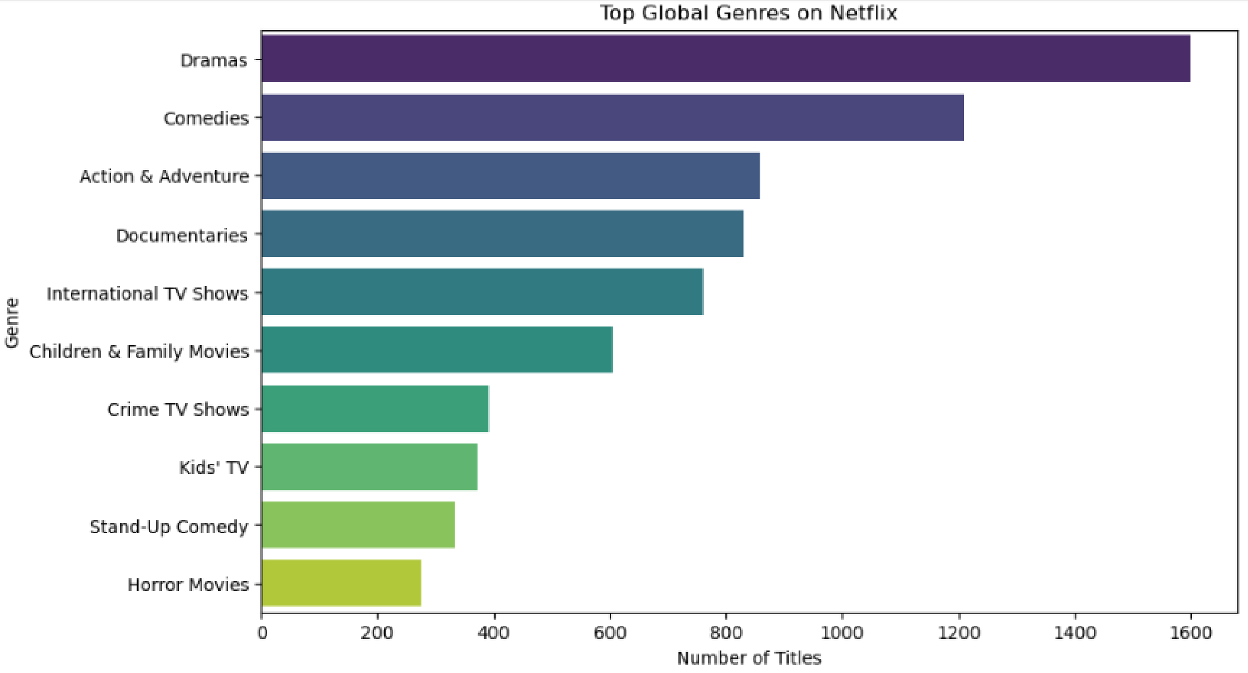
```python
categorical_columns = ['type', 'director','cast','country','rating','duration']

for column in categorical_columns:
    # Value counts
    value_counts_result = data[column].value_counts()
    print(f"\nValue Counts for {column}:")
    print(value_counts_result)

    # Unique values
    unique_values_result = data[column].unique()
    print(f"\nUnique Values for {column}:")
    print(unique_values_result)
```

```
Value Counts for type:
type
Movie      6122
TV Show    2573
Name: count, dtype: int64

Unique Values for type:
['Movie' 'TV Show']

Value Counts for director:
director
Unknown           2533
Rajiv Chilaka       22
Raúl Campos         18
Suhas Kadav         16
Marcus Raboy        16
```

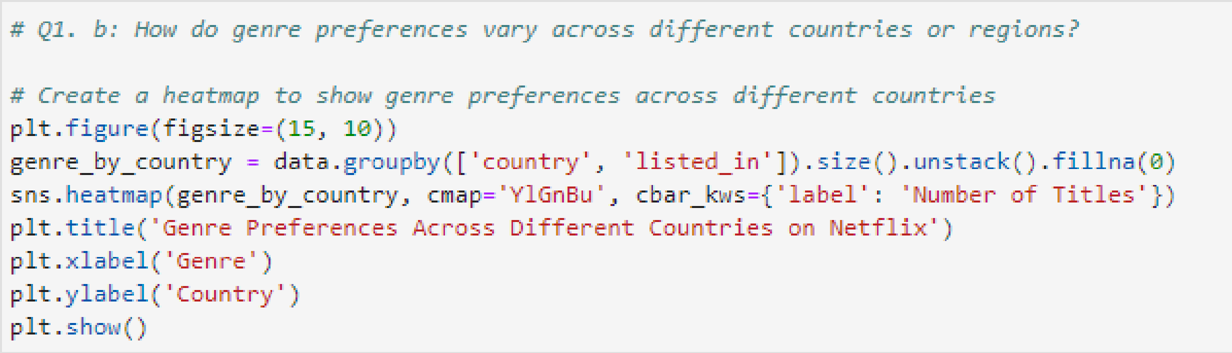
4. **Outlier Treatment:** For outlier treatment we plot bloxplots and then accordingly deal with the outliers using the IQR. But for boxplot we need numerical data. Here only two column has numerical data and for that no outlier treatment is required.
5. **Visual Aanalysis and their insights:** Now I am going to take all those ten questions I have portrayed in the starting and along with the visual will be sharing the insights gained.
   i. **Genre Popularity and Regional Preferences:** Identify popular genres globally and regionally to guide content production strategy. Questions: What are the top global genres on Netflix? How do genre preferences vary across different countries or regions?



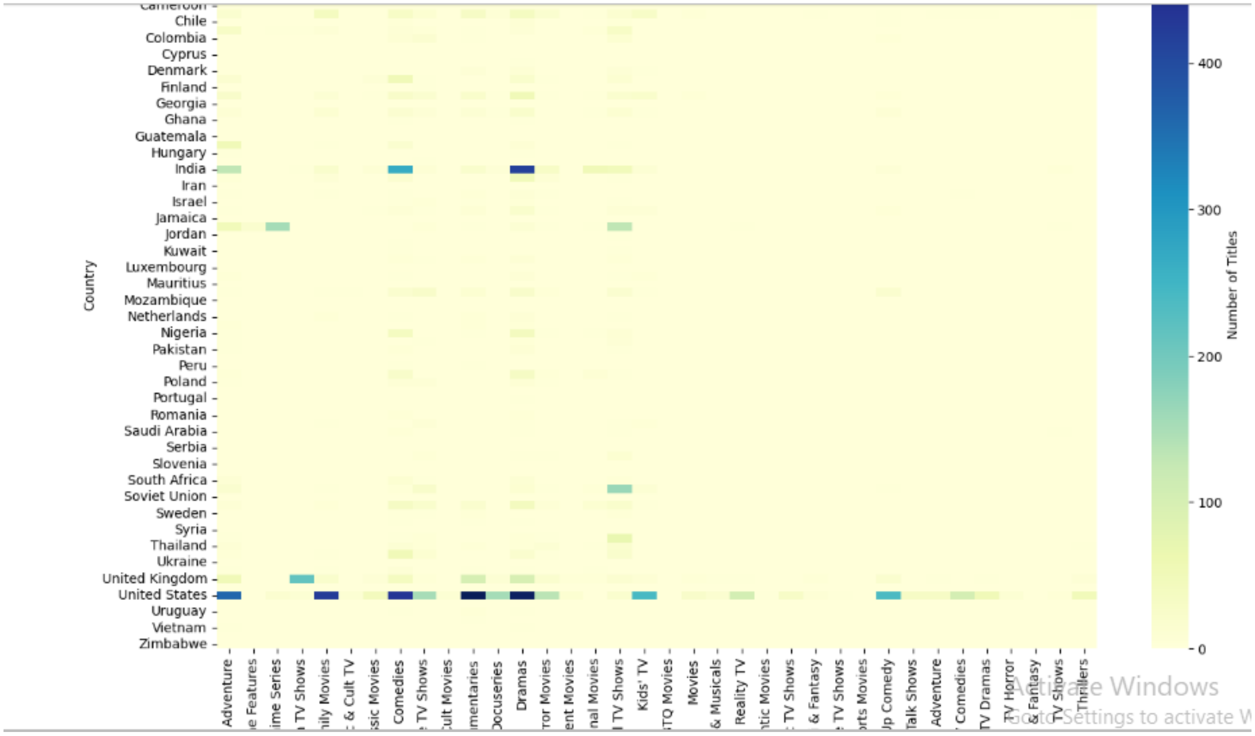
```
# Q1.a: What are the top global genres on Netflix?
global_genre_counts = data['listed_in'].value_counts().head(10)

# Plotting the top global genres
plt.figure(figsize=(10, 6))
sns.barplot(x=global_genre_counts.values, y=global_genre_counts.index, palette='viridis')
plt.title('Top Global Genres on Netflix')
plt.xlabel('Number of Titles')
plt.ylabel('Genre')
plt.show()
```



<u>Insights:</u> The most popular genre is the Dramas. Around 1600 movies all over the globe came under Dramas. Second most liked genre is comedies followed by action and adventure. So to expand the business Netflix should release more of these top three genre based movies/tv shows. The most focus should be on these genre.

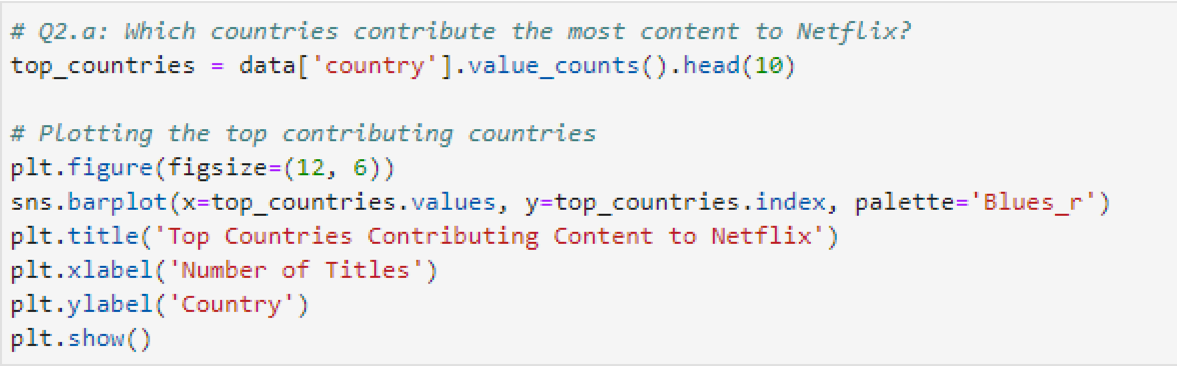


```
# Q1. b: How do genre preferences vary across different countries or regions?

# Create a heatmap to show genre preferences across different countries
plt.figure(figsize=(15, 10))
genre_by_country = data.groupby(['country', 'listed_in']).size().unstack().fillna(0)
sns.heatmap(genre_by_country, cmap='YlGnBu', cbar_kws={'label': 'Number of Titles'})
plt.title('Genre Preferences Across Different Countries on Netflix')
plt.xlabel('Genre')
plt.ylabel('Country')
plt.show()
```

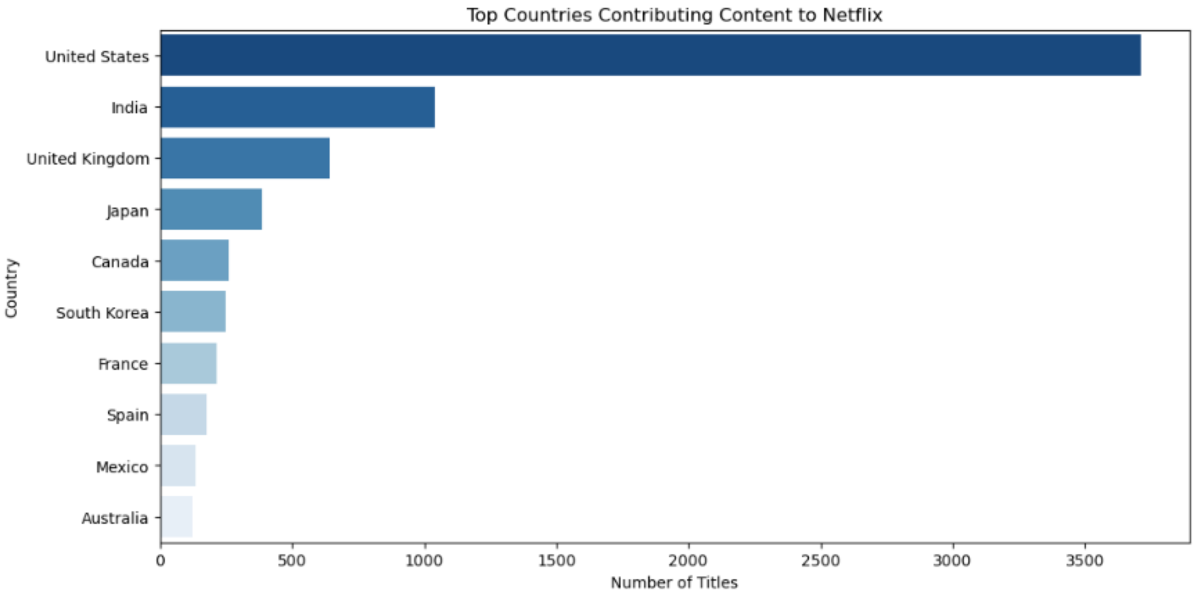


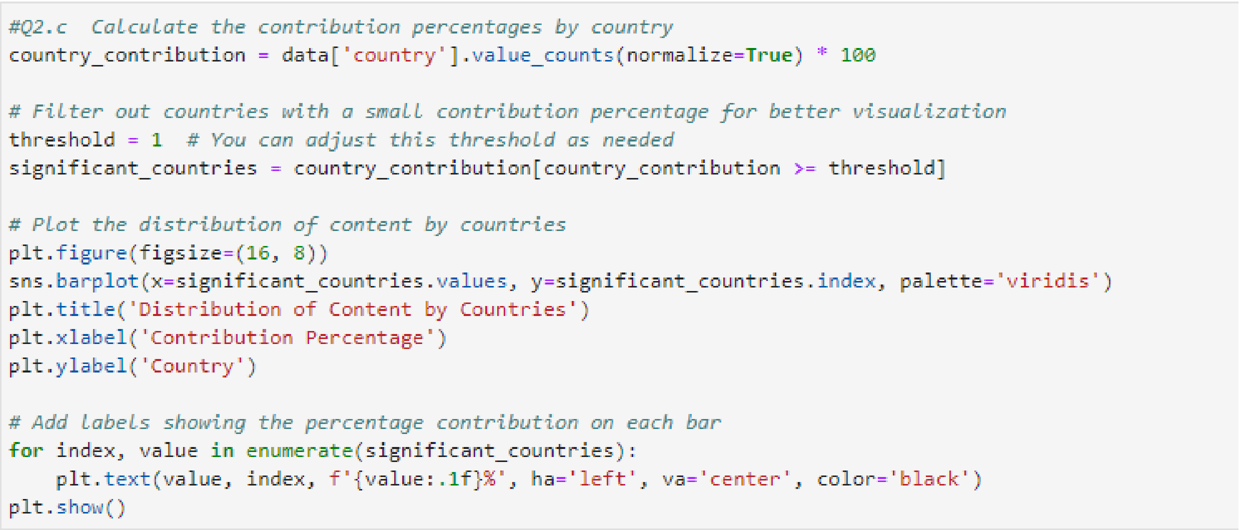
<u>Insights: </u>Plotted the heatmap to show in which country which genre is popular. We can see that the USA in the country where different genres are preferred by the people and it's the country which is the one the best market for Netflix. So it should focus more on the USA customers and their preferences. From here also we can see Dramas is the most popular genre. The next marketplace is India and in India also Dramas is the most popular genre.

ii. **Content Contribution by Countries:** Determine the countries contributing the most content to Netflix for targeted regional focus. Questions: Which countries contribute the most content to Netflix? Can we identify content trends specific to certain regions?

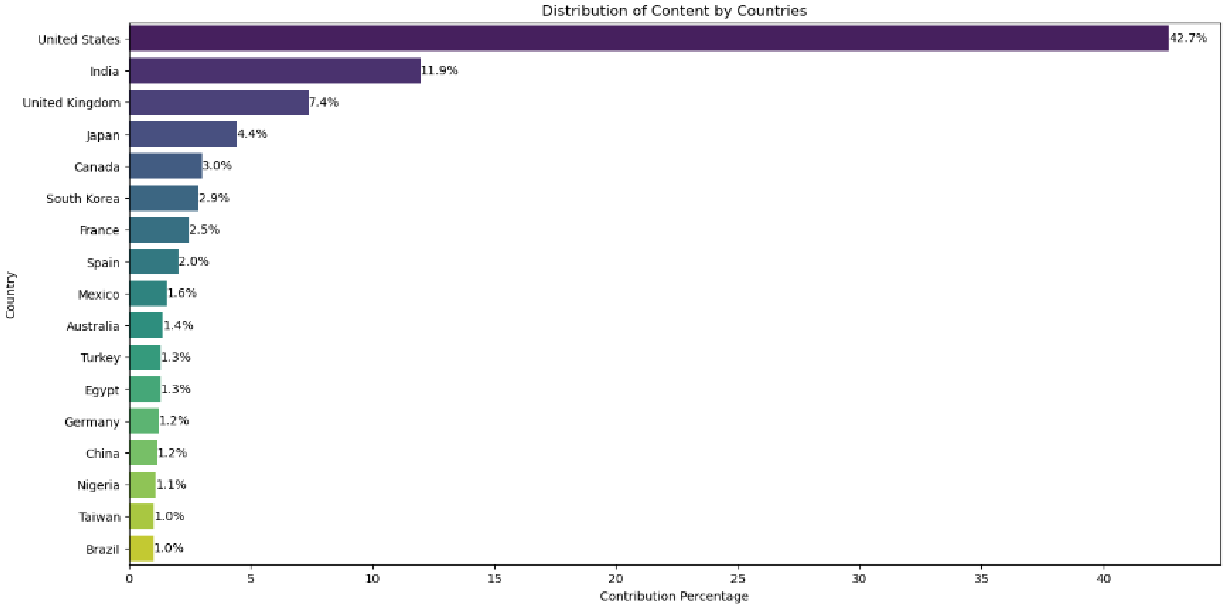


```
# Q2.a: Which countries contribute the most content to Netflix?
top_countries = data['country'].value_counts().head(10)

# Plotting the top contributing countries
plt.figure(figsize=(12, 6))
sns.barplot(x=top_countries.values, y=top_countries.index, palette='Blues_r')
plt.title('Top Countries Contributing Content to Netflix')
plt.xlabel('Number of Titles')
plt.ylabel('Country')
plt.show()
```

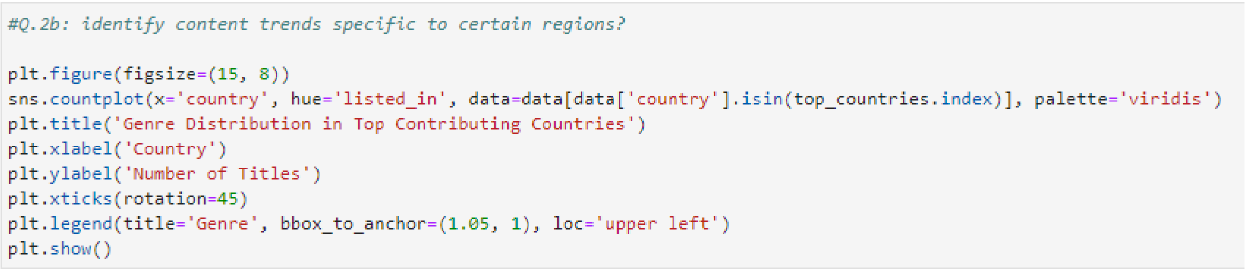




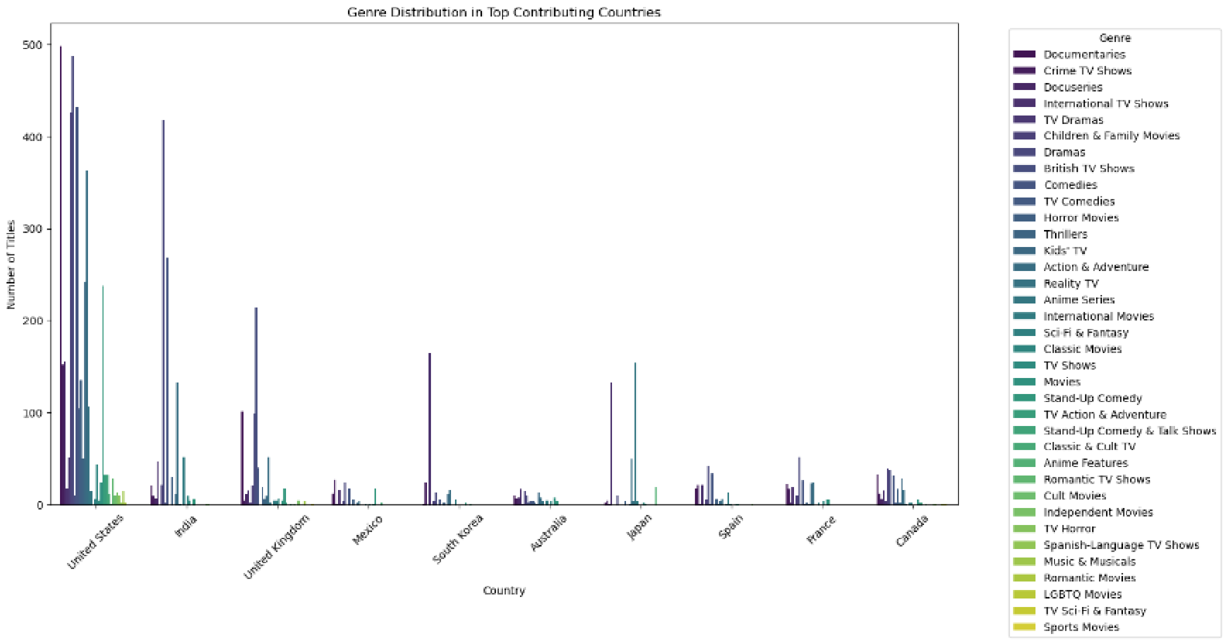
```
#Q2.c  Calculate the contribution percentages by country
country_contribution = data['country'].value_counts(normalize=True) * 100

# Filter out countries with a small contribution percentage for better visualization
threshold = 1  # You can adjust this threshold as needed
significant_countries = country_contribution[country_contribution >= threshold]

# Plot the distribution of content by countries
plt.figure(figsize=(16, 8))
sns.barplot(x=significant_countries.values, y=significant_countries.index, palette='viridis')
plt.title('Distribution of Content by Countries')
plt.xlabel('Contribution Percentage')
plt.ylabel('Country')

# Add labels showing the percentage contribution on each bar
for index, value in enumerate(significant_countries):
    plt.text(value, index, f'{value:.1f}%', ha='left', va='center', color='black')
plt.show()
```



Insights: From the above visual we can see that the USA is the top most contributing to the maximum number (i.e approx. 3700) i.e 42.7% of movies/tv shows launched. Second top most country is India with contribution of around 1200 movies/tv shows i.e 11.9% and United Kingdom is the third top most country contributing to around 700 movies/tv shows i.e 7.4%. So according to this these three countries must be focused more for future publishing.

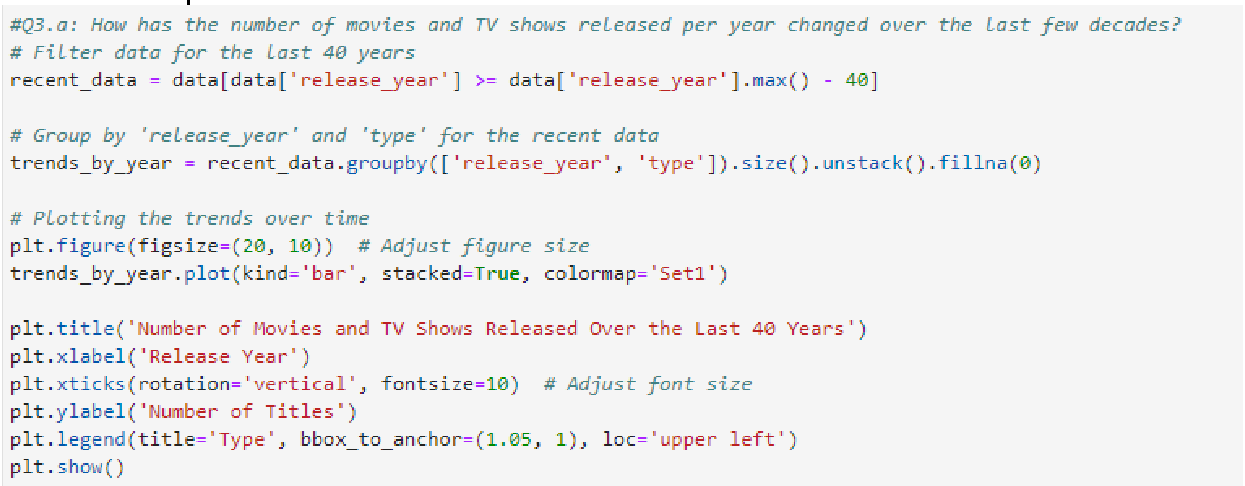


```
#Q.2b: identify content trends specific to certain regions?

plt.figure(figsize=(15, 8))
sns.countplot(x='country', hue='listed_in', data=data[data['country'].isin(top_countries.index)], palette='viridis')
plt.title('Genre Distribution in Top Contributing Countries')
plt.xlabel('Country')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.legend(title='Genre', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

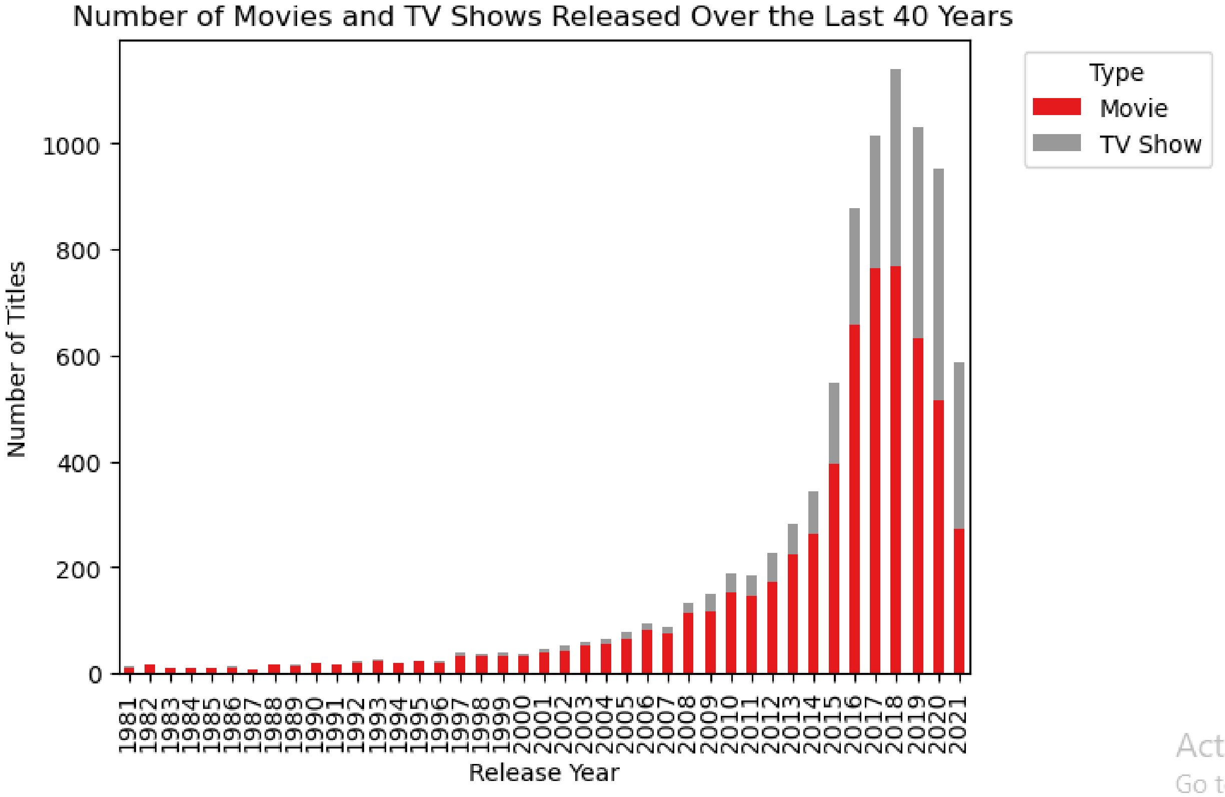


Insights: The above visual gives us the insight that though the USA is the top most country watching the max number of movies/tv shows but its also seen that people there like to watch many genres. Maximum number of genres watched is in the USA followed by UK and India. Although India is on second position if we see the number of movies/tv shows watched but the variety of genre watched is greater in UK then India that's considering Uk on second position. Canada, Australia and France were also found watching multiple genre. But the number is less so for launching new movies or tv shows along with the top three countries these countries can also be considered as they watch variety so may if they launch according to their taste then the business at these countries may also expand.

iii. **Content Consumption Trends over Time:** Explore trends in TV show and movie consumption over the past decades to anticipate future viewer behavior. Questions: How has the number of movies and TV shows released per year changed over the last few decades? Are there discernible patterns in viewer behavior over time?

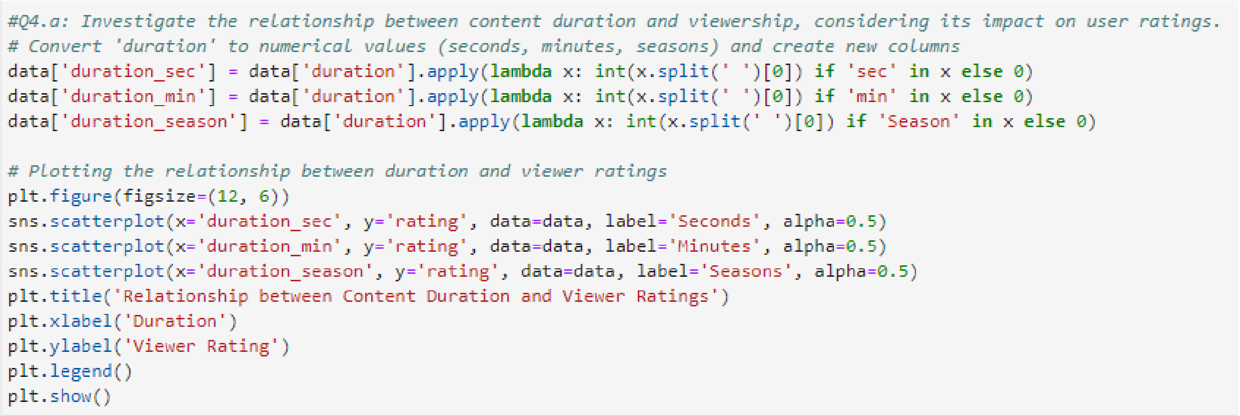


```
#Q3.a: How has the number of movies and TV shows released per year changed over the last few decades?
# Filter data for the last 40 years
recent_data = data[data['release_year'] >= data['release_year'].max() - 40]

# Group by 'release_year' and 'type' for the recent data
trends_by_year = recent_data.groupby(['release_year', 'type']).size().unstack().fillna(0)

# Plotting the trends over time
plt.figure(figsize=(20, 10))  # Adjust figure size
trends_by_year.plot(kind='bar', stacked=True, colormap='Set1')

plt.title('Number of Movies and TV Shows Released Over the Last 40 Years')
plt.xlabel('Release Year')
plt.xticks(rotation='vertical', fontsize=10)  # Adjust font size
plt.ylabel('Number of Titles')
plt.legend(title='Type', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

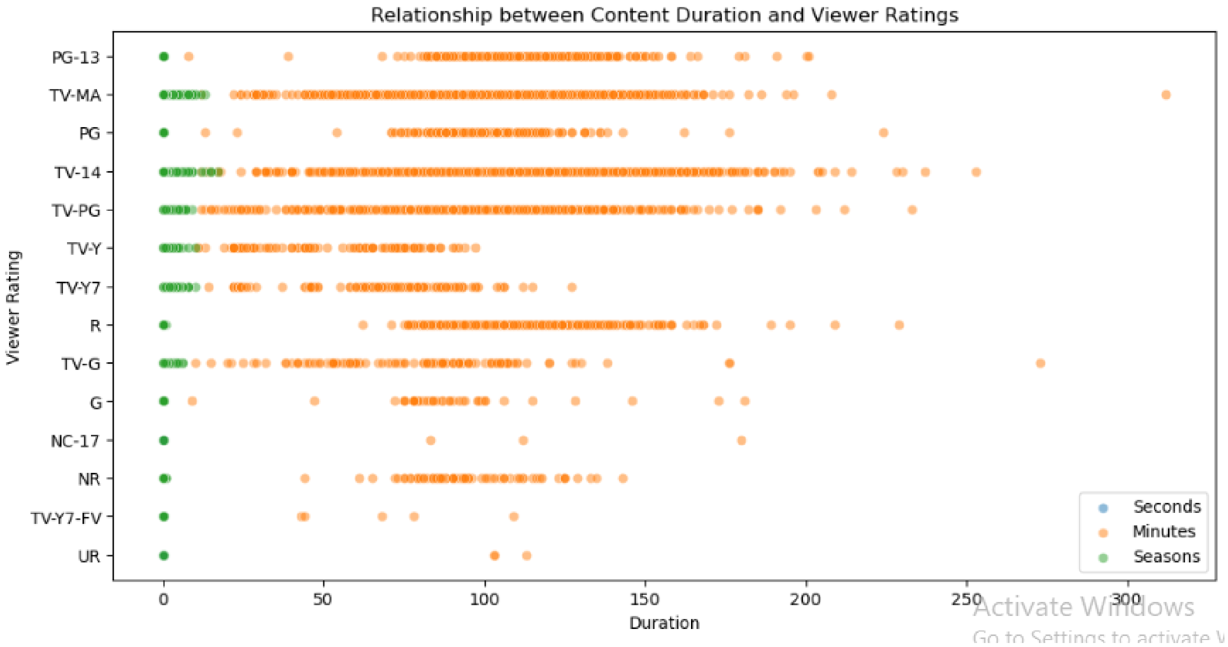


Insights: We can see the trend from 1981-2021 (ie for the last 40 year). We can see I rising trend till 2018 and then there is decreasing trend till 2021. Also we can see that viewers like tv shows more than the movies.

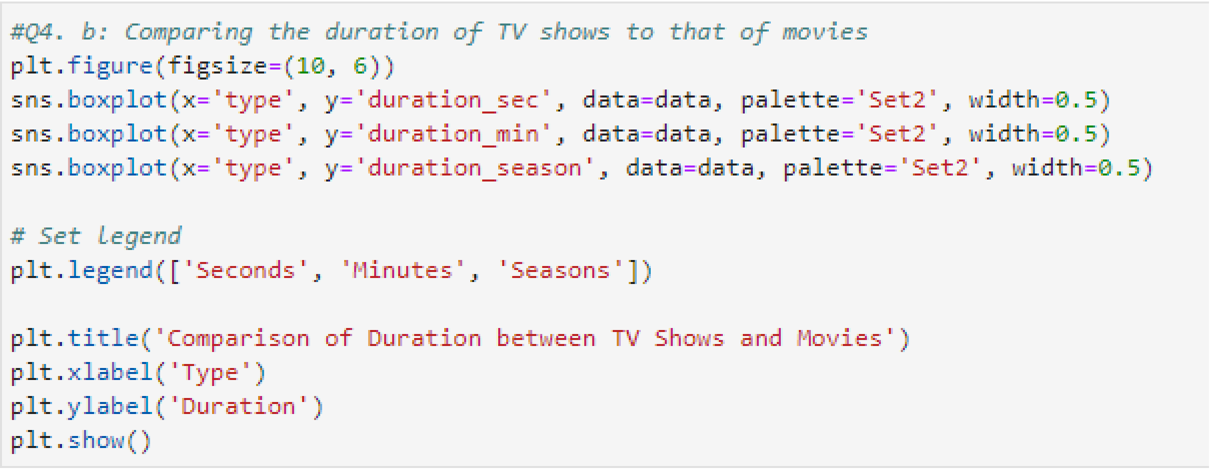
iv. **Impact of Content Duration on Viewership:** Investigate the relationship between content duration and viewership, considering its impact on user ratings. Questions: Is there a correlation between the duration of content and viewer ratings? How does the duration of TV shows compare to that of movies?



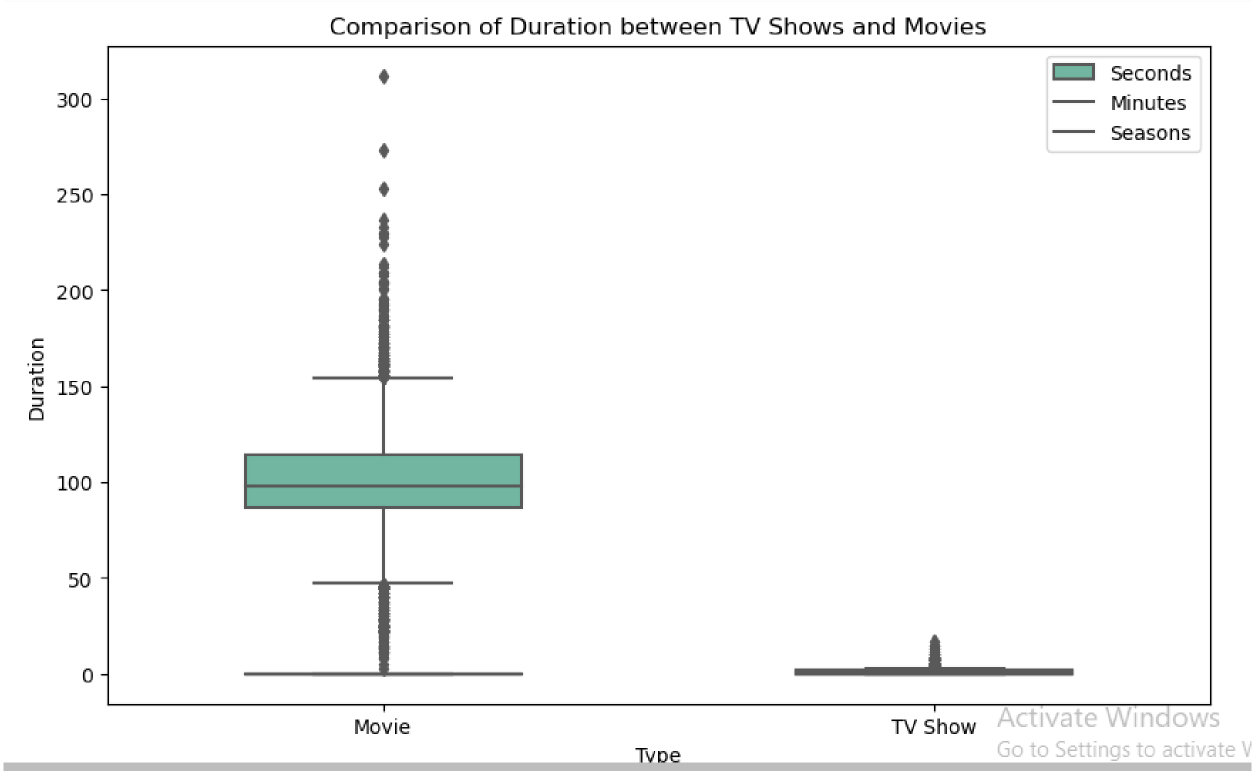
```
#Q4.a: Investigate the relationship between content duration and viewership, considering its impact on user ratings.
# Convert 'duration' to numerical values (seconds, minutes, seasons) and create new columns
data['duration_sec'] = data['duration'].apply(lambda x: int(x.split(' ')[0]) if 'sec' in x else 0)
data['duration_min'] = data['duration'].apply(lambda x: int(x.split(' ')[0]) if 'min' in x else 0)
data['duration_season'] = data['duration'].apply(lambda x: int(x.split(' ')[0]) if 'Season' in x else 0)

# Plotting the relationship between duration and viewer ratings
plt.figure(figsize=(12, 6))
sns.scatterplot(x='duration_sec', y='rating', data=data, label='Seconds', alpha=0.5)
sns.scatterplot(x='duration_min', y='rating', data=data, label='Minutes', alpha=0.5)
sns.scatterplot(x='duration_season', y='rating', data=data, label='Seasons', alpha=0.5)
plt.title('Relationship between Content Duration and Viewer Ratings')
plt.xlabel('Duration')
plt.ylabel('Viewer Rating')
plt.legend()
plt.show()
```



Insights: The duration of movies/tv shows were given in three forms: seconds, minutes and seasons. So we have converted the duration column into numerical and in three categories. Most data lies in approx. 70-160 minutes. So, general duration must be kept in this range.

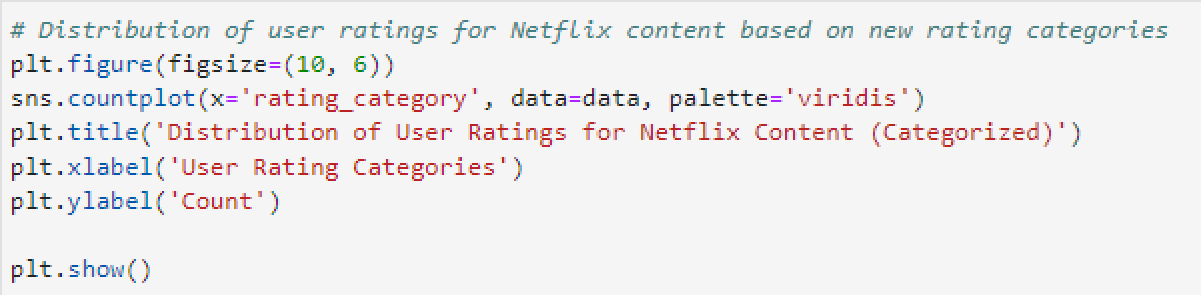


```
#Q4. b: Comparing the duration of TV shows to that of movies
plt.figure(figsize=(10, 6))
sns.boxplot(x='type', y='duration_sec', data=data, palette='Set2', width=0.5)
sns.boxplot(x='type', y='duration_min', data=data, palette='Set2', width=0.5)
sns.boxplot(x='type', y='duration_season', data=data, palette='Set2', width=0.5)

# Set legend
plt.legend(['Seconds', 'Minutes', 'Seasons'])

plt.title('Comparison of Duration between TV Shows and Movies')
plt.xlabel('Type')
plt.ylabel('Duration')
plt.show()
```

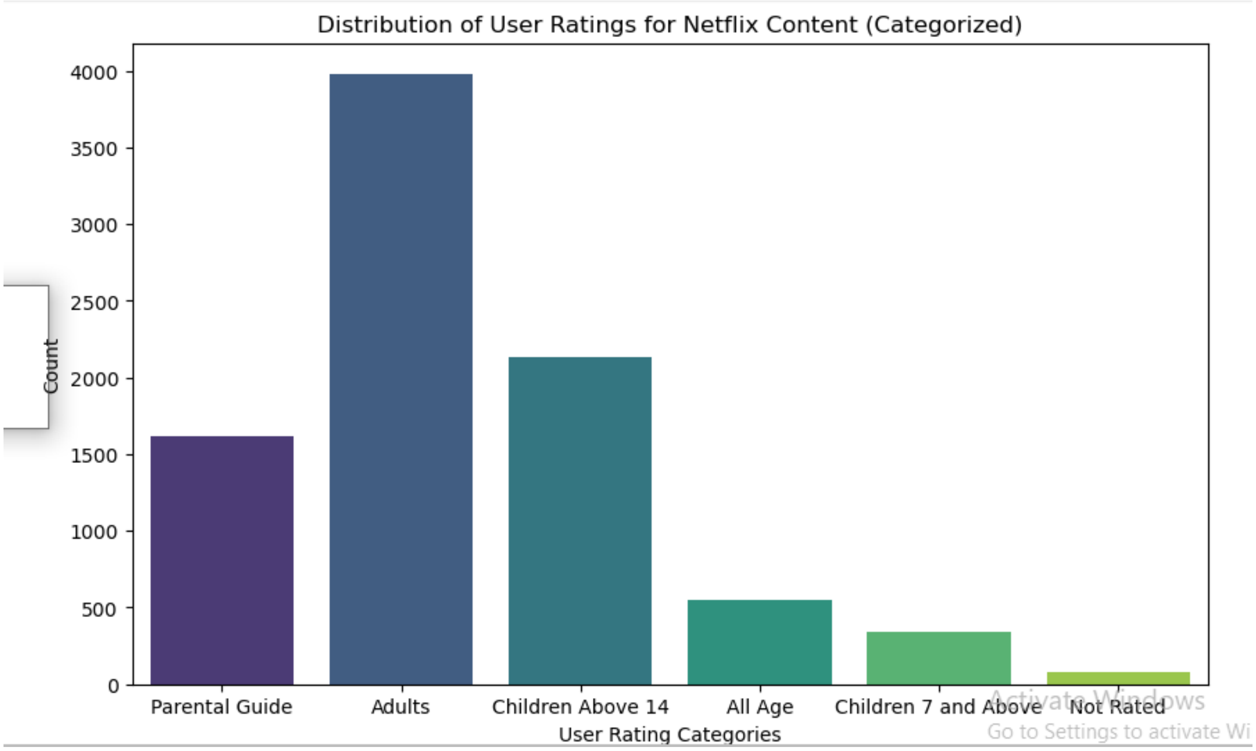


Insights: The duration of movies should be kept generally for the duration of around 160 minutes and that of tv shows around 20-30 minutes.

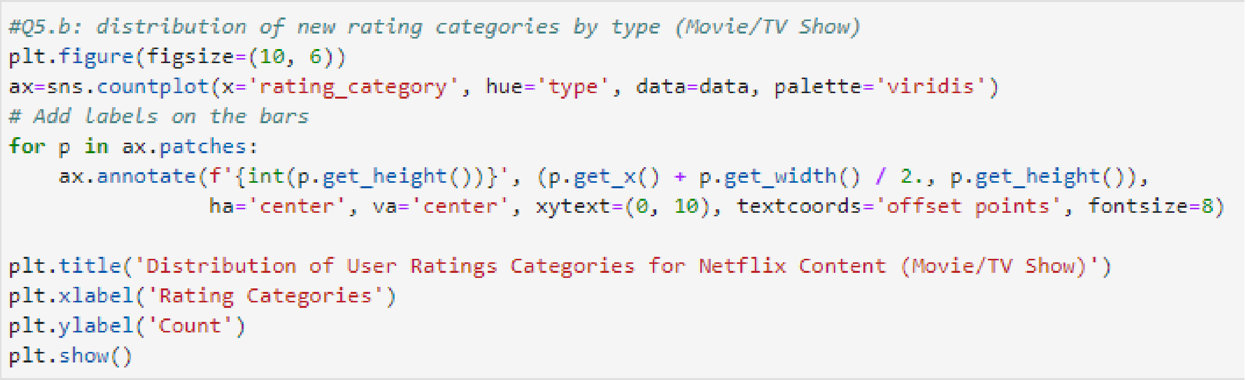
v. **Ratings and it's impact:** Analyze the ratings of movies and tv shows to provide data-driven insights on the types of shows or movies to launch. Questions: What is the distribution of ratings for Netflix content? What is the trend of various ratings over the last 40 years?



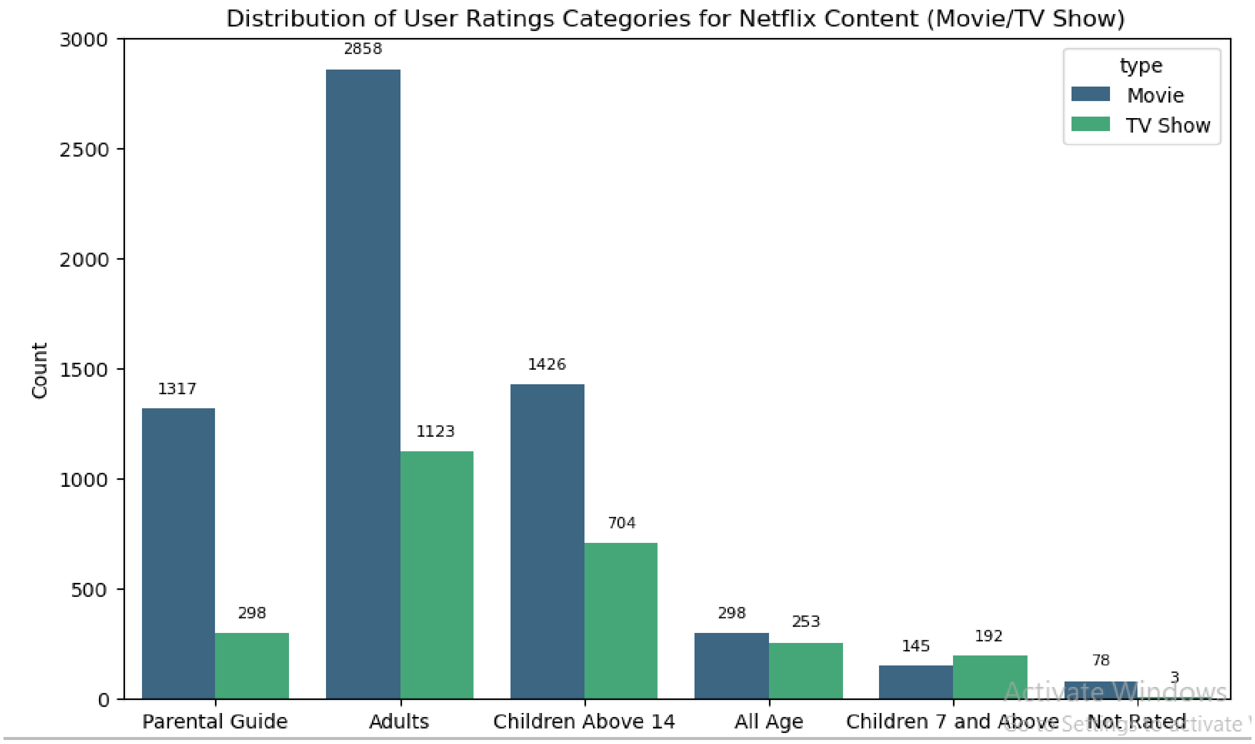
```
# Distribution of user ratings for Netflix content based on new rating categories
plt.figure(figsize=(10, 6))
sns.countplot(x='rating_category', data=data, palette='viridis')
plt.title('Distribution of User Ratings for Netflix Content (Categorized)')
plt.xlabel('User Rating Categories')
plt.ylabel('Count')

plt.show()
```



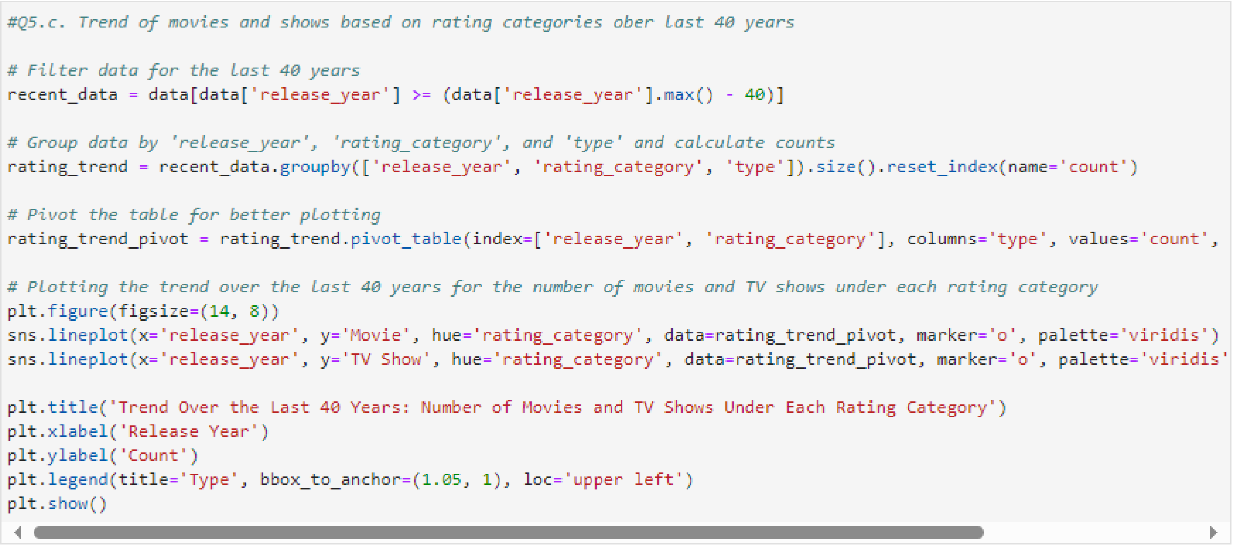
Insights: Maximum movies/tv shows are made for adults. Next mostly launched movies/tv shows are for children of age 14 or above and the next most launched are for children but with parental guide. So we can generally the content is from 14 years and above and for children very less content is launched. May be next they can try launching more of kids content too as these days kids are more involved in watching tv and mobiles.



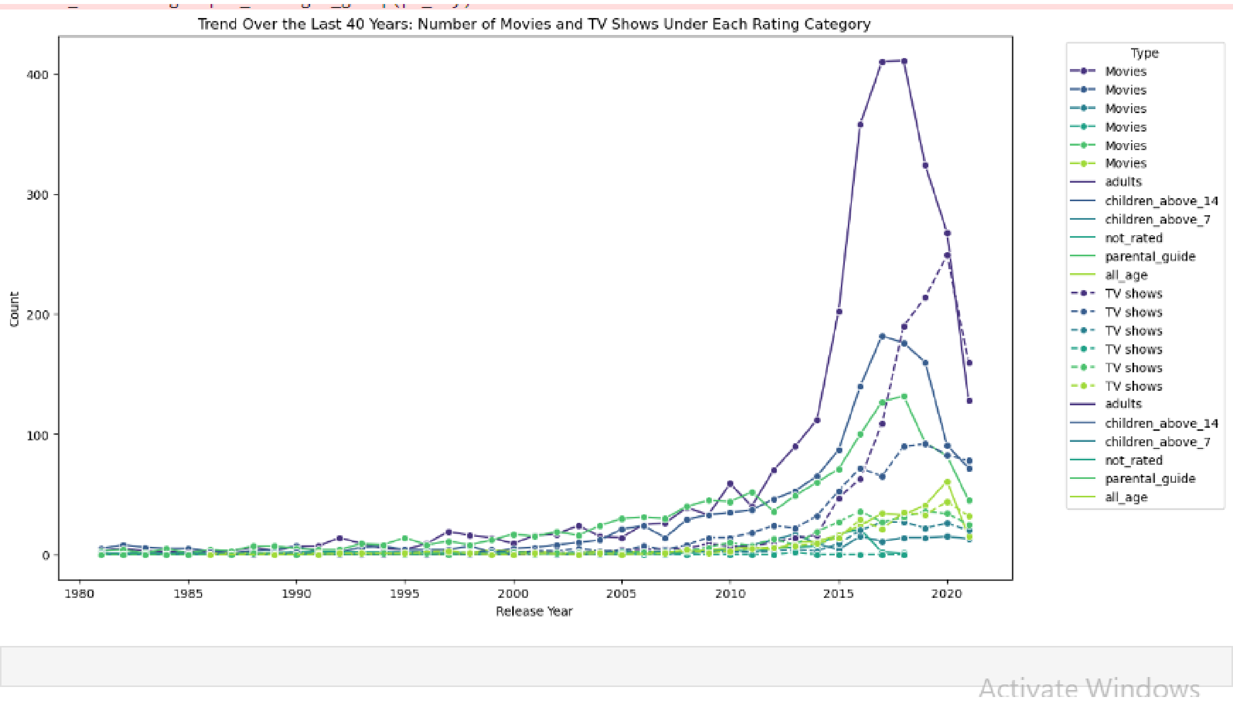
```
#Q5.b: distribution of new rating categories by type (Movie/TV Show)
plt.figure(figsize=(10, 6))
ax=sns.countplot(x='rating_category', hue='type', data=data, palette='viridis')
# Add labels on the bars
for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset points', fontsize=8)

plt.title('Distribution of User Ratings Categories for Netflix Content (Movie/TV Show)')
plt.xlabel('Rating Categories')
plt.ylabel('Count')
plt.show()
```



Insights: We can see for most of the rating categories movies are launched more. For all age movies as well tv shows count are almost similar and for children of 7 and above tv shows are more. That means in kids sections tv shows are liked. Netflix can try launching tv shows for children of and above 14 so that can attract more kids and they will keep on watching Netflix as tv shows are in some series.

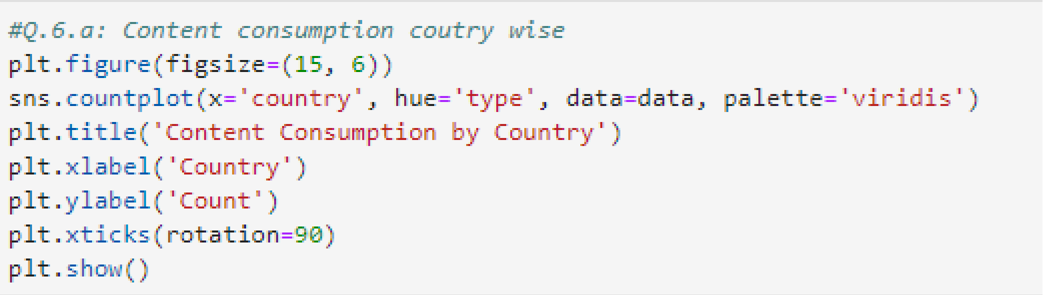


```
#Q5.c. Trend of movies and shows based on rating categories ober last 40 years

# Filter data for the last 40 years
recent_data = data[data['release_year'] >= (data['release_year'].max() - 40)]

# Group data by 'release_year', 'rating_category', and 'type' and calculate counts
rating_trend = recent_data.groupby(['release_year', 'rating_category', 'type']).size().reset_index(name='count')

# Pivot the table for better plotting
rating_trend_pivot = rating_trend.pivot_table(index=['release_year', 'rating_category'], columns='type', values='count',

# Plotting the trend over the last 40 years for the number of movies and TV shows under each rating category
plt.figure(figsize=(14, 8))
sns.lineplot(x='release_year', y='Movie', hue='rating_category', data=rating_trend_pivot, marker='o', palette='viridis')
sns.lineplot(x='release_year', y='TV Show', hue='rating_category', data=rating_trend_pivot, marker='o', palette='viridis'

plt.title('Trend Over the Last 40 Years: Number of Movies and TV Shows Under Each Rating Category')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.legend(title='Type', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

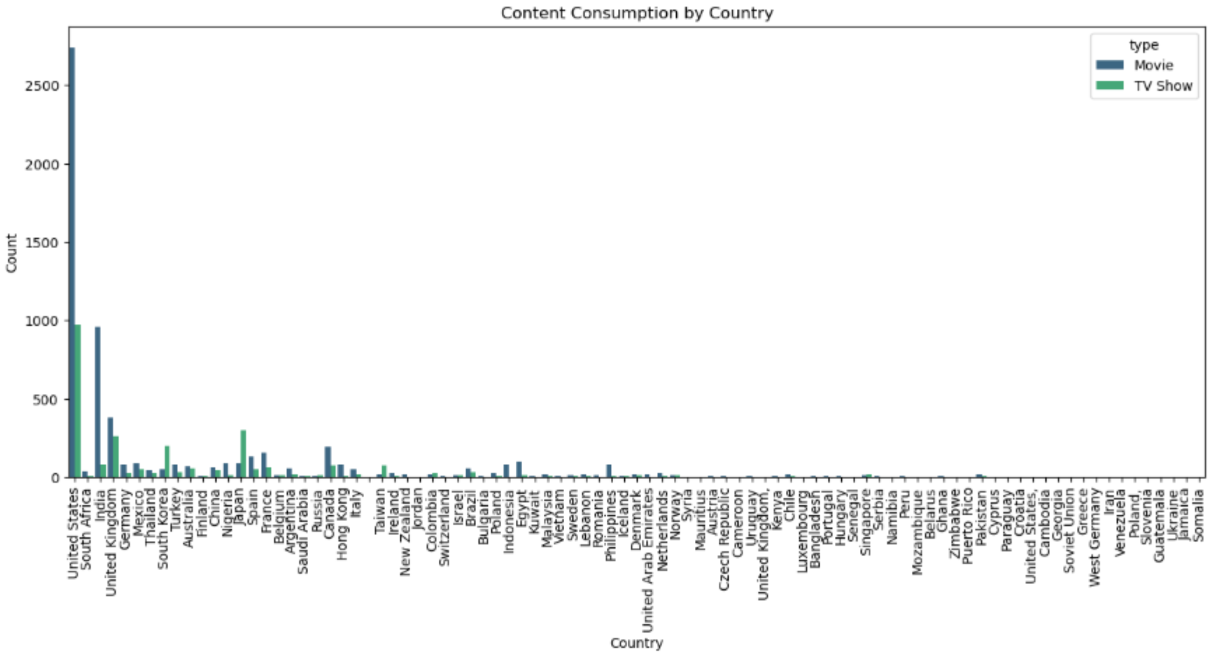


Insights: From the year 1981-2010 there was similar growth in all types of rating movies and shows. But after 2010 we can visualize a huge hike in adult movies which started falling down after around 2016. There was a good hike in adult tv shows too but this also came down after around 2020. We need to find the reason why this huge hike obtained by adult movies/shows was not maintained and work upon it to gain more business.

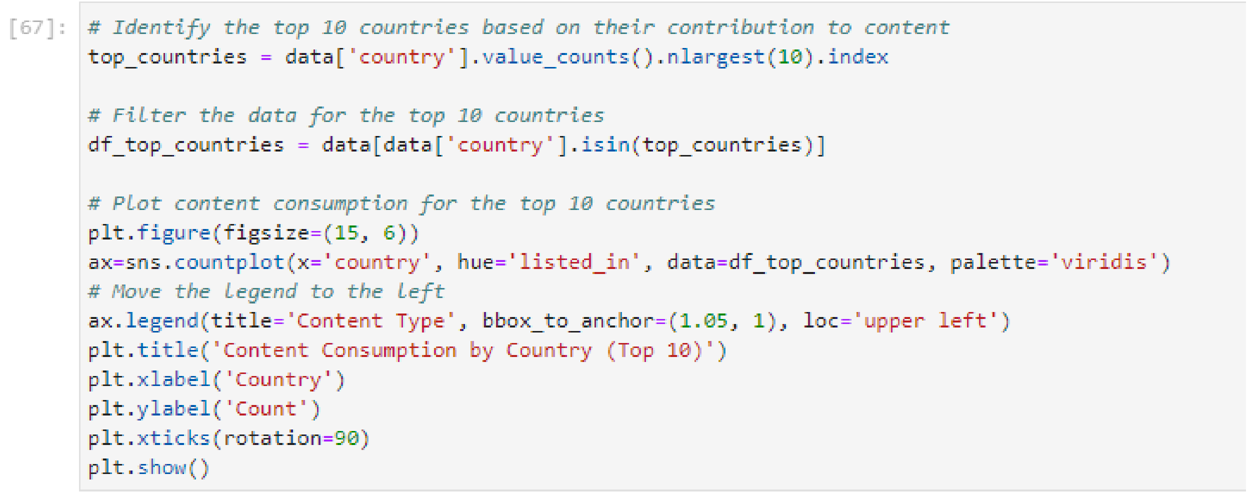
vi. **Demographic Analysis for Targeted Content:** Understand how user demographics influence content preferences for targeted and personalized production. Questions: Are there demographic trends that correlate with specific content preferences? How does location impact content consumption patterns?



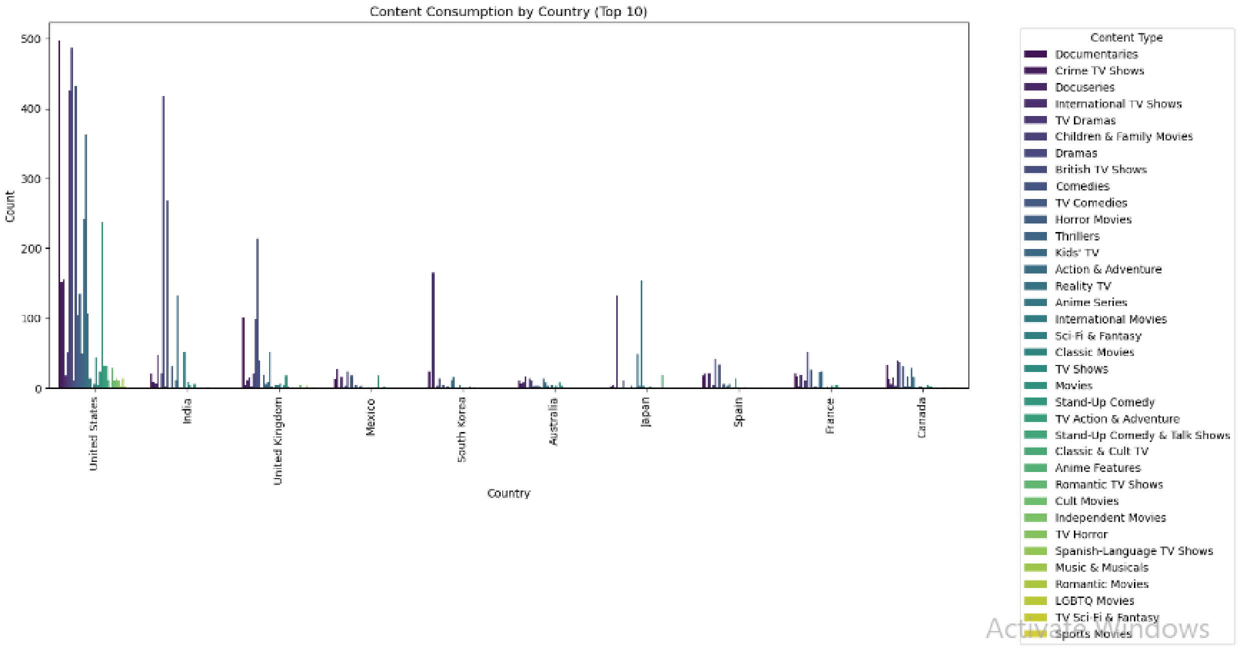
```
#Q.6.a: Content consumption coutry wise
plt.figure(figsize=(15, 6))
sns.countplot(x='country', hue='type', data=data, palette='viridis')
plt.title('Content Consumption by Country')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



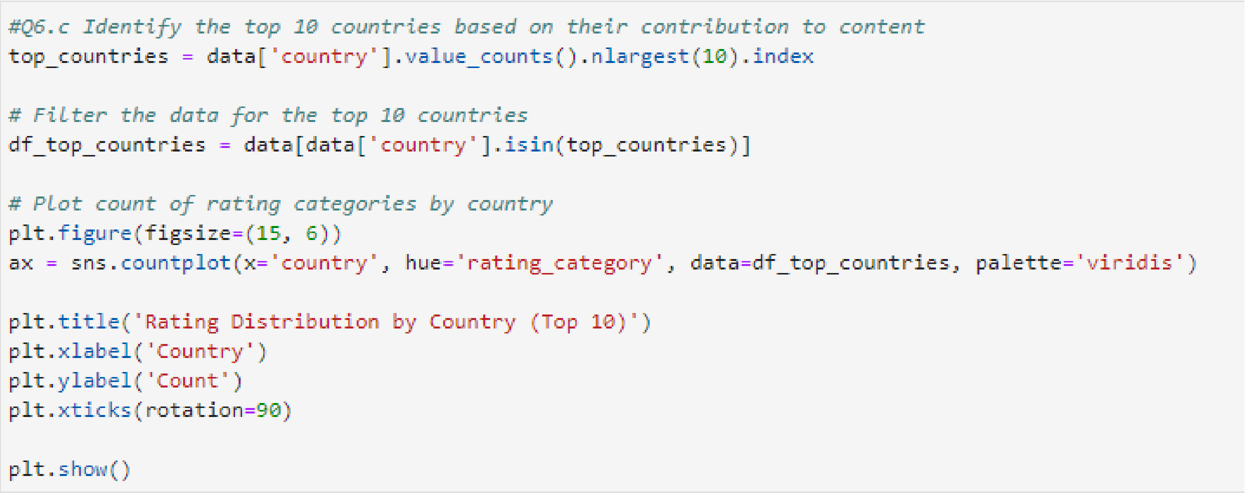
Insights: We can see that the USA is the most valuable country for Netflix as it has the maximum movies and tv shows launched, followed by India and UK. Japan and South Korea have a good number of tv shows popular among them, so Netflix can work more upon launch tv shows in these areas and movies in the USA, India and Uk as comparative to others.



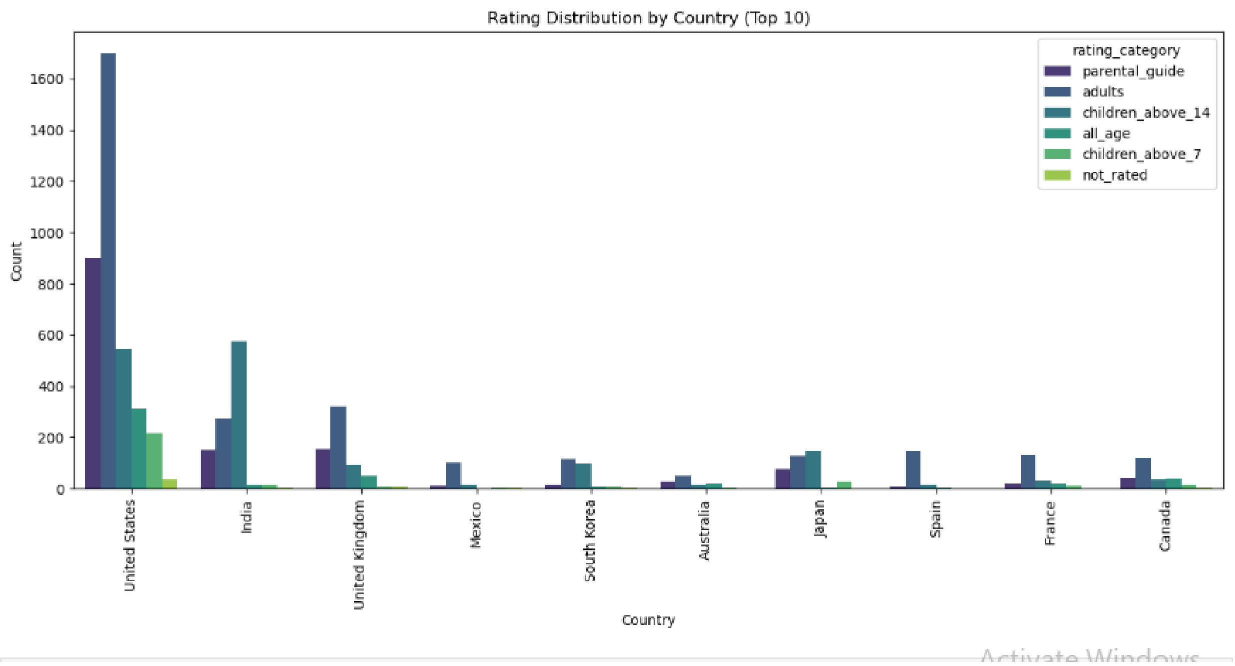
```python
# Identify the top 10 countries based on their contribution to content
top_countries = data['country'].value_counts().nlargest(10).index

# Filter the data for the top 10 countries
df_top_countries = data[data['country'].isin(top_countries)]

# Plot content consumption for the top 10 countries
plt.figure(figsize=(15, 6))
ax=sns.countplot(x='country', hue='listed_in', data=df_top_countries, palette='viridis')
# Move the legend to the left
ax.legend(title='Content Type', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.title('Content Consumption by Country (Top 10)')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



Insights: We have the list of top ten countries. Based on the content consumption of each country Netflix must launched their content in those countries so that everyone get what they are looking for and thus increase the number of users and watch time.

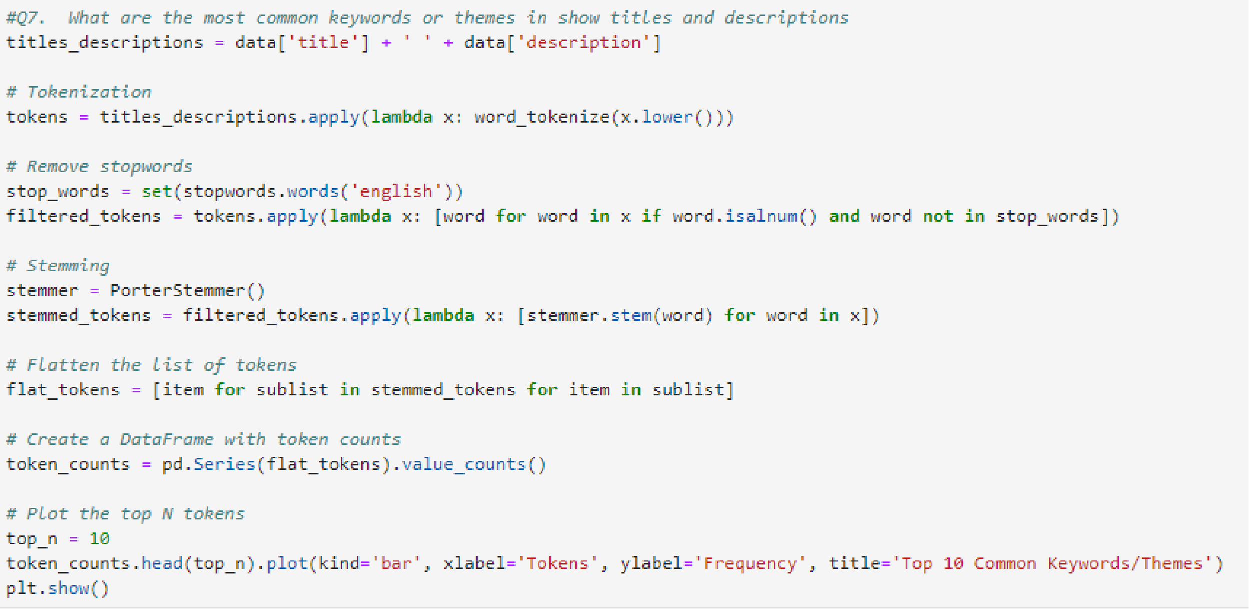


```
#Q6.c Identify the top 10 countries based on their contribution to content
top_countries = data['country'].value_counts().nlargest(10).index

# Filter the data for the top 10 countries
df_top_countries = data[data['country'].isin(top_countries)]

# Plot count of rating categories by country
plt.figure(figsize=(15, 6))
ax = sns.countplot(x='country', hue='rating_category', data=df_top_countries, palette='viridis')

plt.title('Rating Distribution by Country (Top 10)')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=90)

plt.show()
```

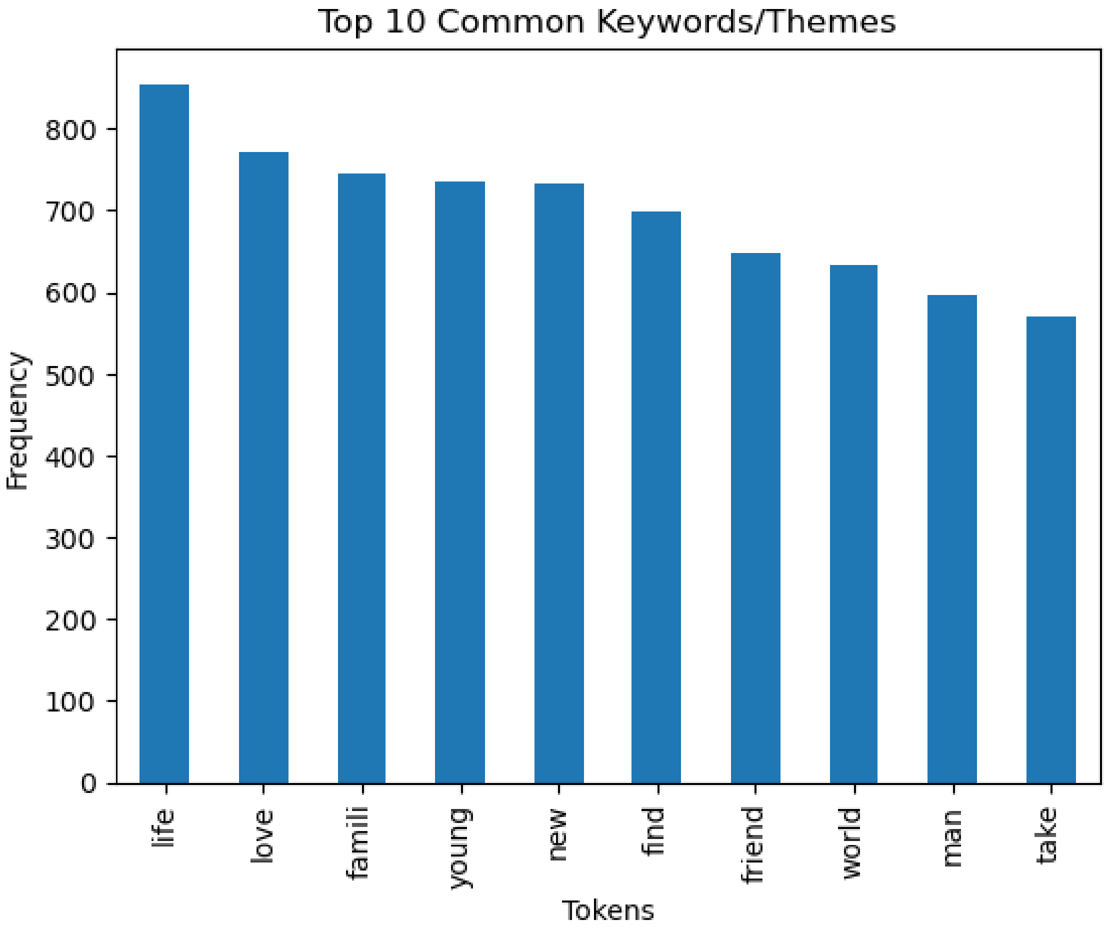


Insights: It's reflecting top ten countries and showing the distribution of ratings in all those countries. As we can see in the USA the most popular type is adults where as in India the most popular are those for the children of and above 14 years and in UK its adults. So based on countries preferences Netflix must launch those types of movies/tv shows.

vii. **Keyword Analysis:** Identify common keywords or themes in show titles and descriptions to inform content themes and trends. Questions: What are the most common keywords or themes in show titles and descriptions?

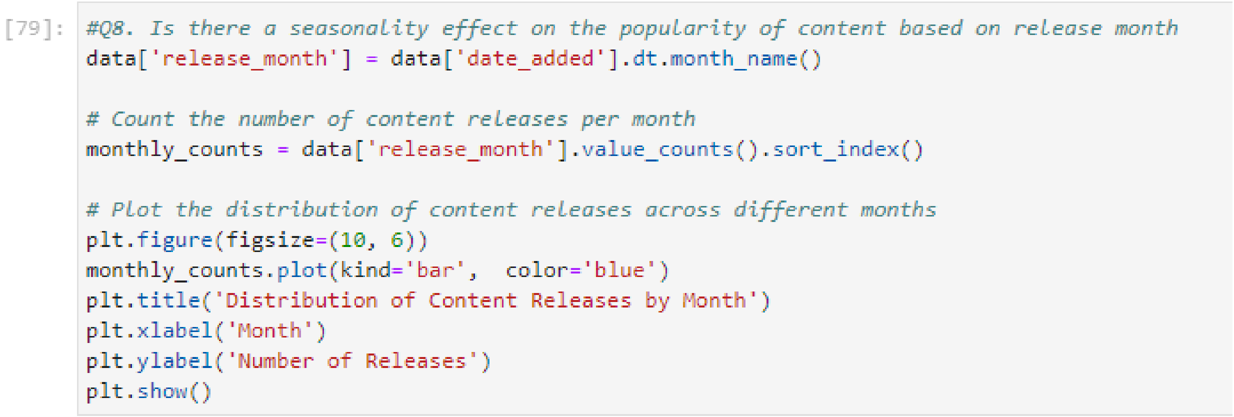


```
#Q7.  What are the most common keywords or themes in show titles and descriptions
titles_descriptions = data['title'] + ' ' + data['description']

# Tokenization
tokens = titles_descriptions.apply(lambda x: word_tokenize(x.lower()))

# Remove stopwords
stop_words = set(stopwords.words('english'))
filtered_tokens = tokens.apply(lambda x: [word for word in x if word.isalnum() and word not in stop_words])

# Stemming
stemmer = PorterStemmer()
stemmed_tokens = filtered_tokens.apply(lambda x: [stemmer.stem(word) for word in x])

# Flatten the list of tokens
flat_tokens = [item for sublist in stemmed_tokens for item in sublist]

# Create a DataFrame with token counts
token_counts = pd.Series(flat_tokens).value_counts()

# Plot the top N tokens
top_n = 10
token_counts.head(top_n).plot(kind='bar', xlabel='Tokens', ylabel='Frequency', title='Top 10 Common Keywords/Themes')
plt.show()
```

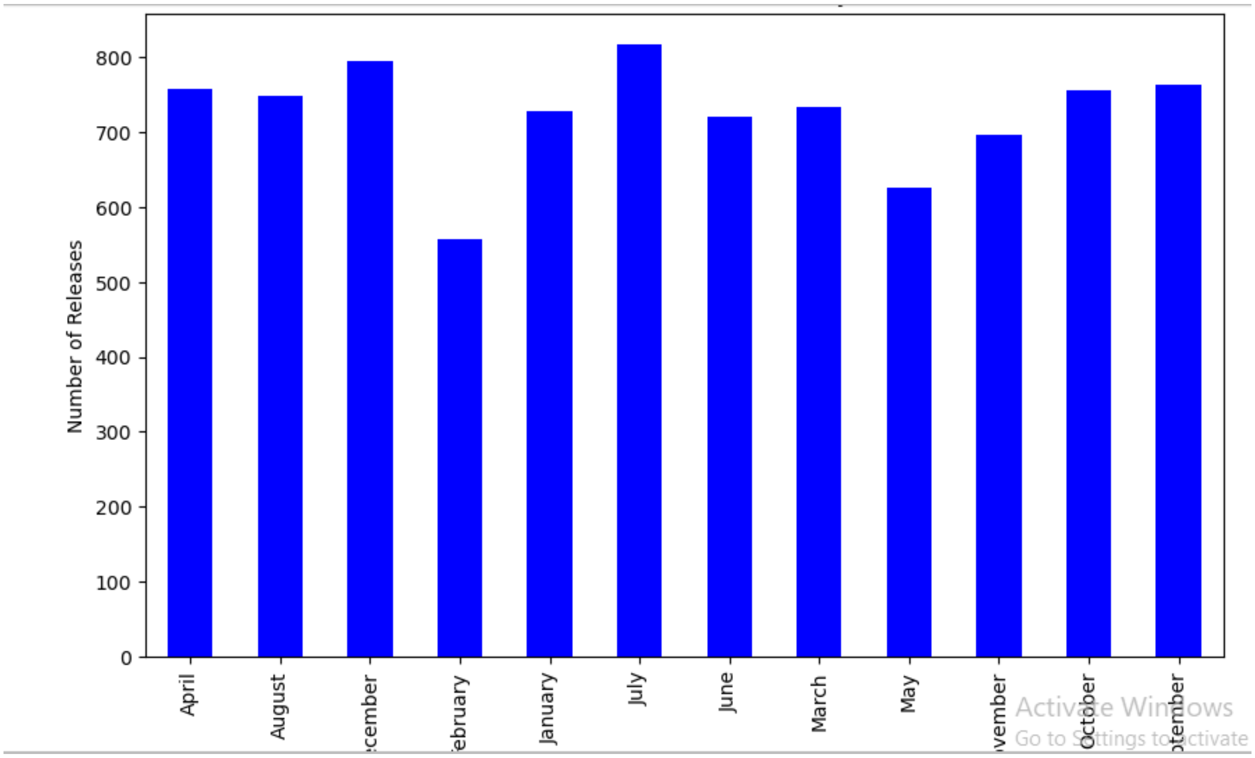


Insights: We have got top ten keywords which include life , love, family at the top three positions. Try to keep these keywords in title or description so that there are more chances for the user to click on the movies/shows.

viii. **Release Month Influence:** Investigate if there is a relationship between the release month and show popularity, guiding strategic release timing. Questions: Is there a seasonality effect on the popularity of content based on release month? Do certain months see higher viewer engagement?

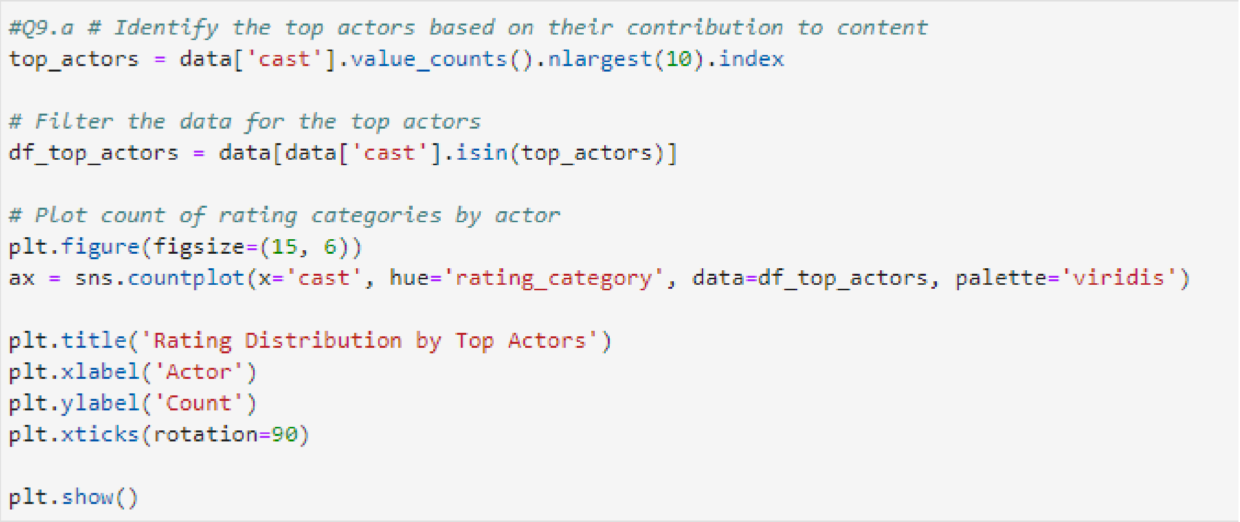


```
[79]: #Q8. Is there a seasonality effect on the popularity of content based on release month
      data['release_month'] = data['date_added'].dt.month_name()

      # Count the number of content releases per month
      monthly_counts = data['release_month'].value_counts().sort_index()

      # Plot the distribution of content releases across different months
      plt.figure(figsize=(10, 6))
      monthly_counts.plot(kind='bar',  color='blue')
      plt.title('Distribution of Content Releases by Month')
      plt.xlabel('Month')
      plt.ylabel('Number of Releases')
      plt.show()
```

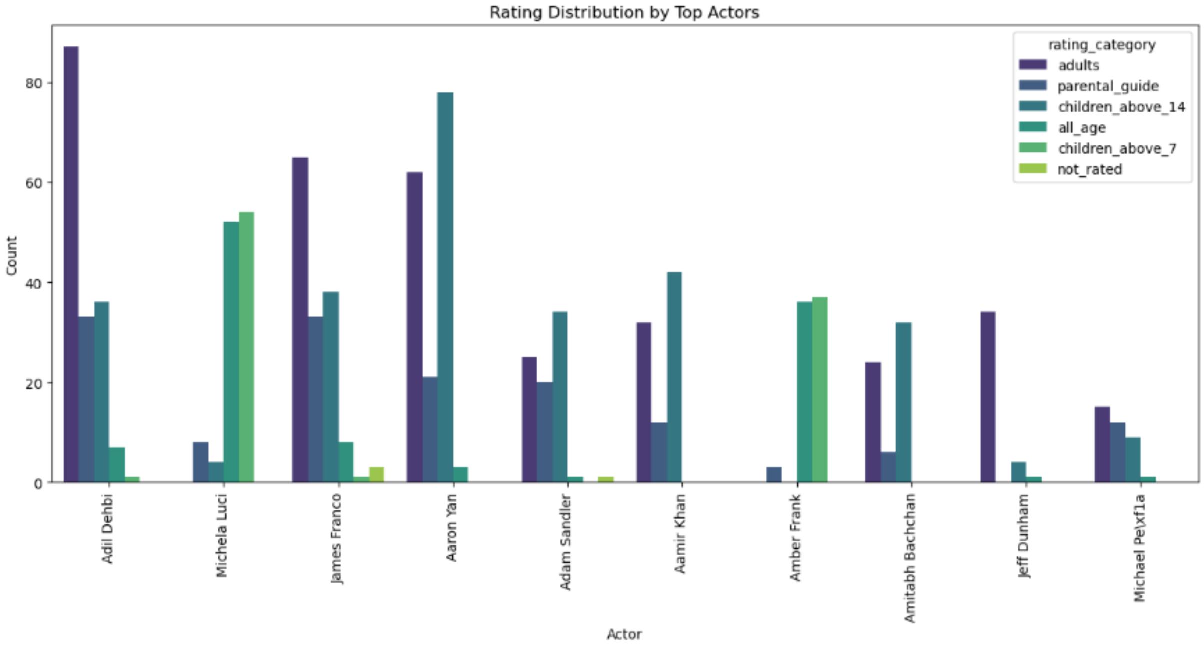


Insights: This is to show month wise trend of releasing a movie/show. Although there is not that much variance in the data so there is not much seasonality seen. But still we can say July is the best month to launch. We should avoid launching in February and March as per the data. The reason can be the exams of students. So we should keep the launch during these months to the lowest and try launching on other months.

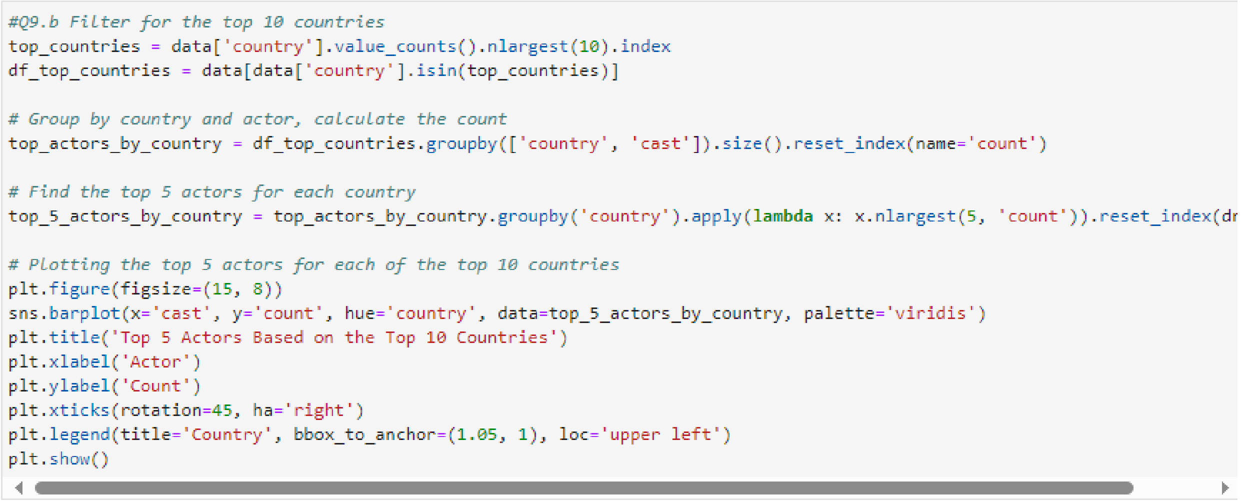
ix. **Actor/Director Association with Highly Rated Shows:** Explore which actors or directors are consistently associated with highly-rated content. Questions: Who are the actors or directors frequently associated with high ratings? Is there a correlation between specific talent and content success?



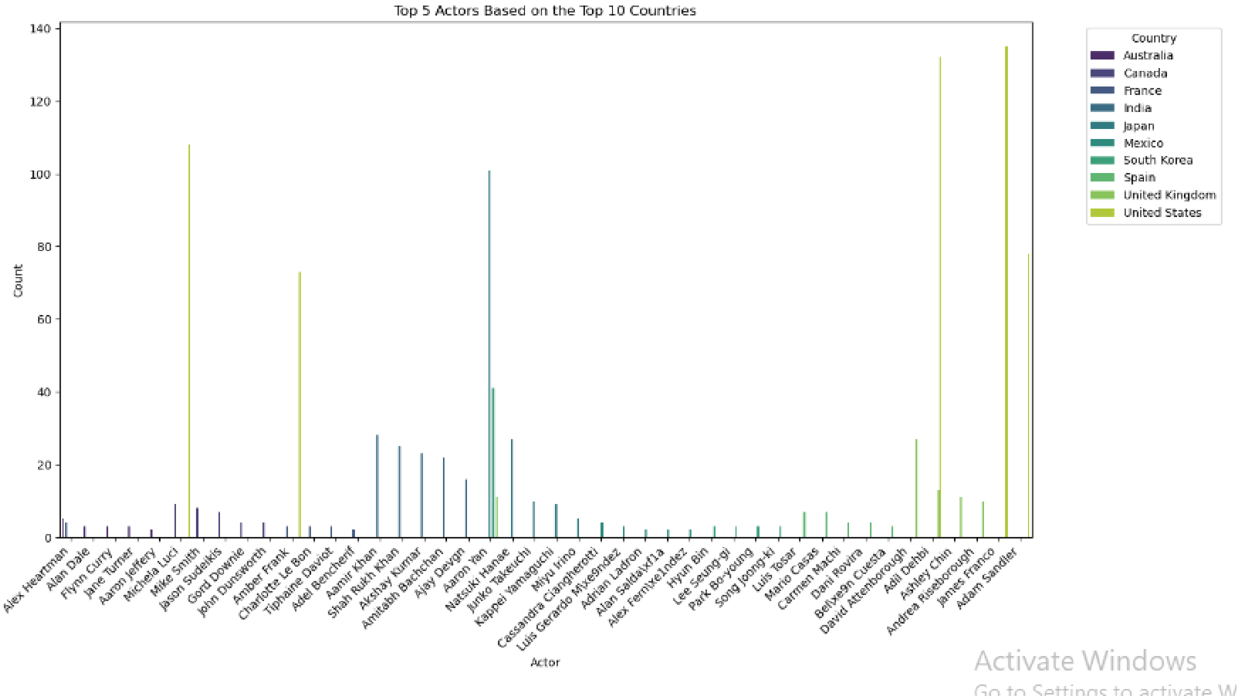
```
#Q9.a # Identify the top actors based on their contribution to content
top_actors = data['cast'].value_counts().nlargest(10).index

# Filter the data for the top actors
df_top_actors = data[data['cast'].isin(top_actors)]

# Plot count of rating categories by actor
plt.figure(figsize=(15, 6))
ax = sns.countplot(x='cast', hue='rating_category', data=df_top_actors, palette='viridis')

plt.title('Rating Distribution by Top Actors')
plt.xlabel('Actor')
plt.ylabel('Count')
plt.xticks(rotation=90)

plt.show()
```



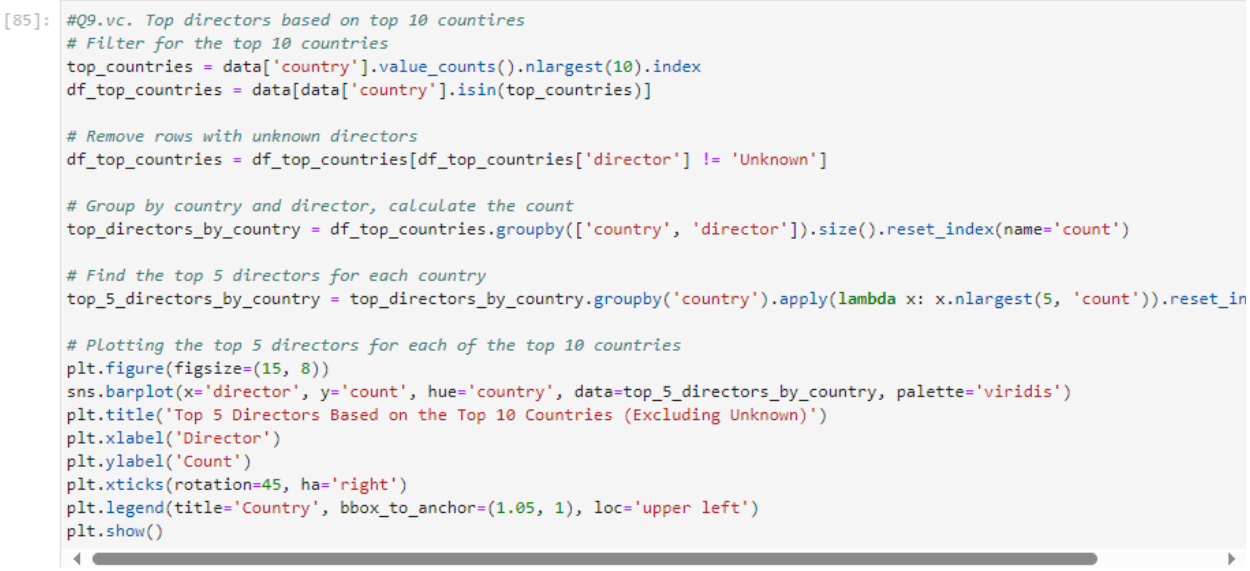
Insights: These are the top 10 actors and the distribution of the movies they have done along with the numbers. So these actors are more likely preferred by the users. When launching new content, it is advisable to consider the preferences of viewers. For example- for adults movies Adil Dehbi is preferred, for parental guide Aaron Yan is preferred.



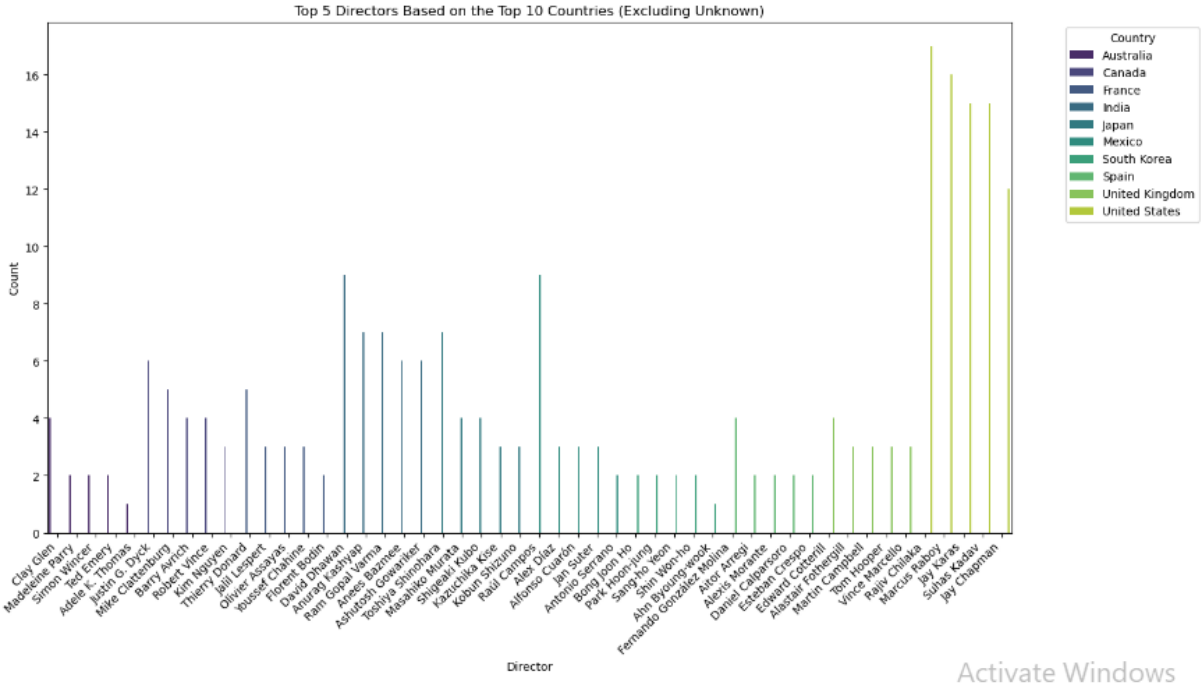
```
#Q9.b Filter for the top 10 countries
top_countries = data['country'].value_counts().nlargest(10).index
df_top_countries = data[data['country'].isin(top_countries)]

# Group by country and actor, calculate the count
top_actors_by_country = df_top_countries.groupby(['country', 'cast']).size().reset_index(name='count')

# Find the top 5 actors for each country
top_5_actors_by_country = top_actors_by_country.groupby('country').apply(lambda x: x.nlargest(5, 'count')).reset_index(dr

# Plotting the top 5 actors for each of the top 10 countries
plt.figure(figsize=(15, 8))
sns.barplot(x='cast', y='count', hue='country', data=top_5_actors_by_country, palette='viridis')
plt.title('Top 5 Actors Based on the Top 10 Countries')
plt.xlabel('Actor')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



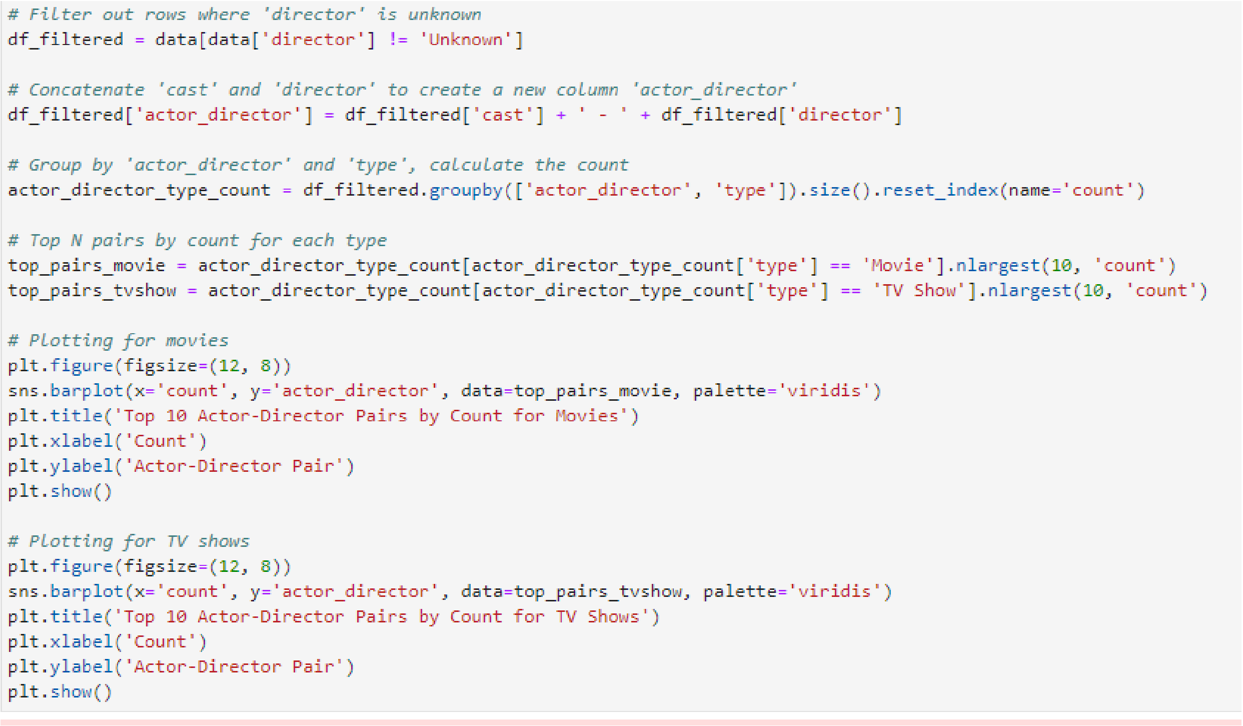
Insights: Now I am taken into account country wise popular actors so that this may help Netflix to launch movies in various countries based on the users choice.



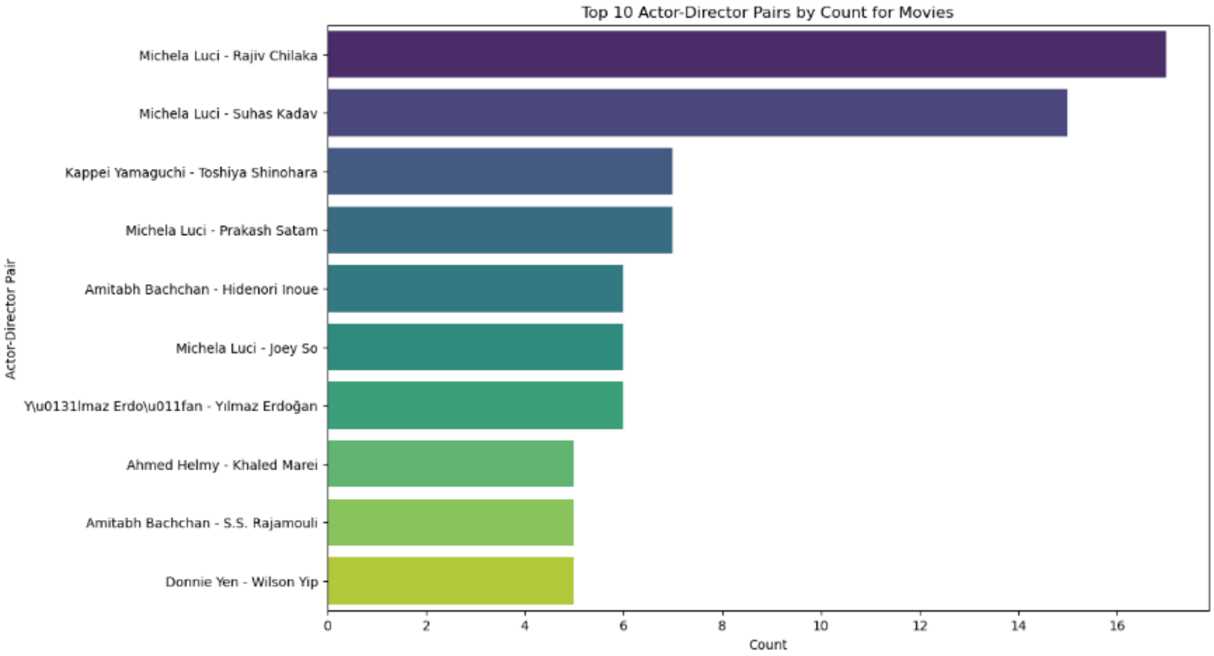
```
[85]: #Q9.vc. Top directors based on top 10 countries
      # Filter for the top 10 countries
      top_countries = data['country'].value_counts().nlargest(10).index
      df_top_countries = data[data['country'].isin(top_countries)]

      # Remove rows with unknown directors
      df_top_countries = df_top_countries[df_top_countries['director'] != 'Unknown']

      # Group by country and director, calculate the count
      top_directors_by_country = df_top_countries.groupby(['country', 'director']).size().reset_index(name='count')

      # Find the top 5 directors for each country
      top_5_directors_by_country = top_directors_by_country.groupby('country').apply(lambda x: x.nlargest(5, 'count')).reset_in

      # Plotting the top 5 directors for each of the top 10 countries
      plt.figure(figsize=(15, 8))
      sns.barplot(x='director', y='count', hue='country', data=top_5_directors_by_country, palette='viridis')
      plt.title('Top 5 Directors Based on the Top 10 Countries (Excluding Unknown)')
      plt.xlabel('Director')
      plt.ylabel('Count')
      plt.xticks(rotation=45, ha='right')
      plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
      plt.show()
```

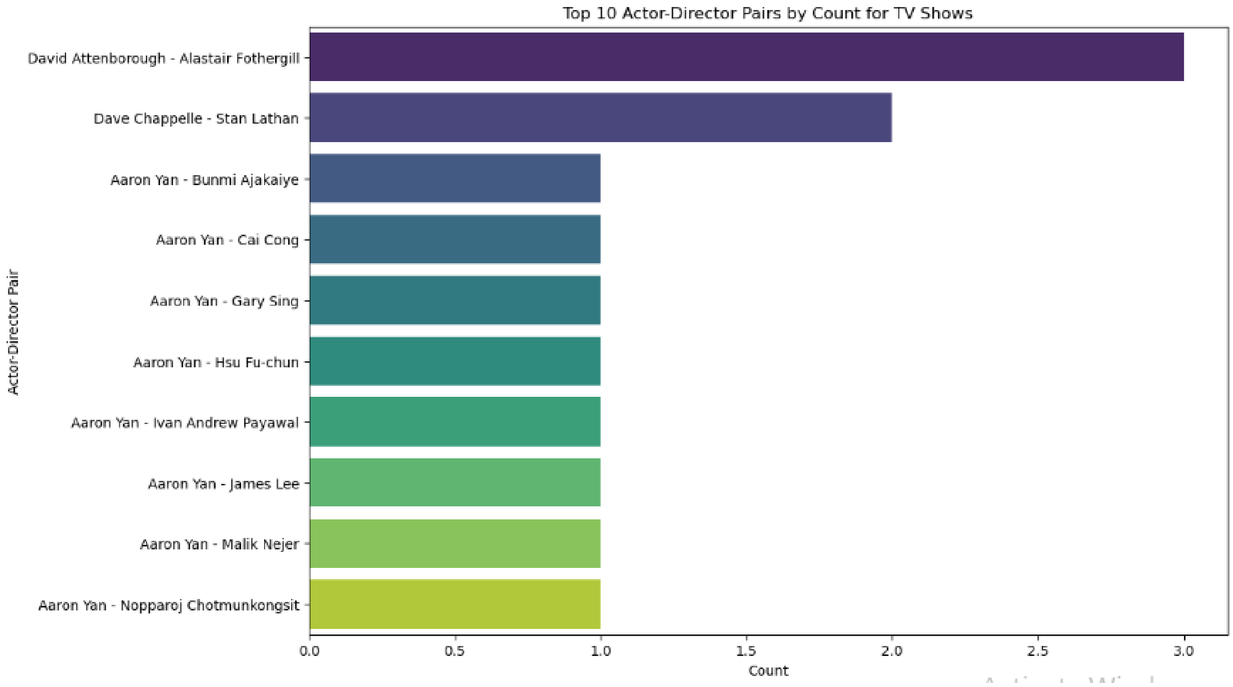


Insights: Now I have clubbed directors and countries so as to get the top directors country wise. We can try launching more movies directed by these preferred directors.



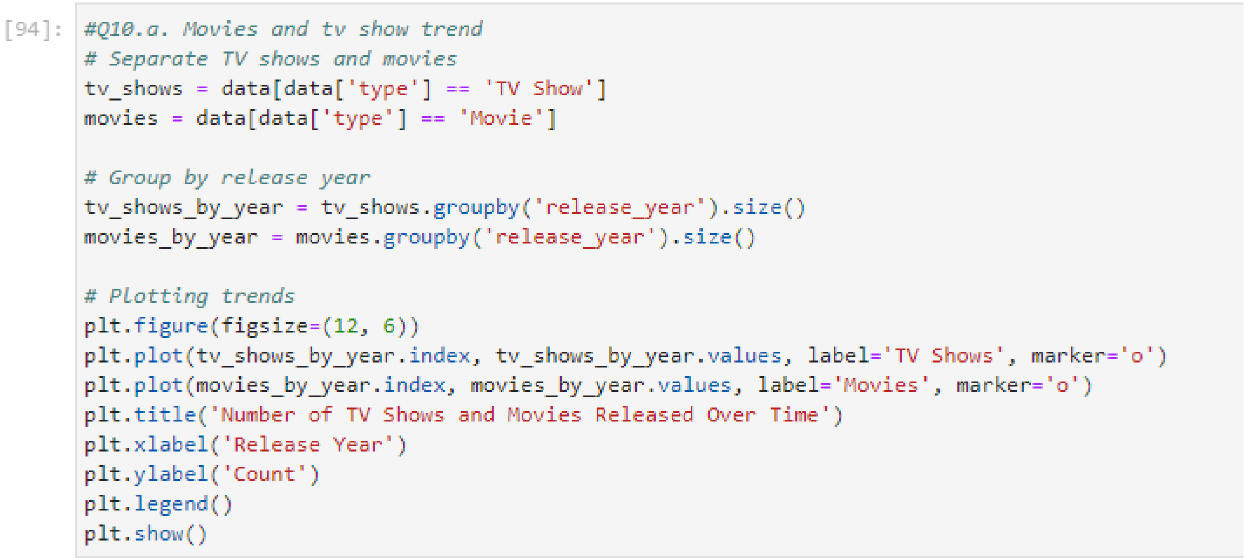
```
# Filter out rows where 'director' is unknown
df_filtered = data[data['director'] != 'Unknown']

# Concatenate 'cast' and 'director' to create a new column 'actor_director'
df_filtered['actor_director'] = df_filtered['cast'] + ' - ' + df_filtered['director']

# Group by 'actor_director' and 'type', calculate the count
actor_director_type_count = df_filtered.groupby(['actor_director', 'type']).size().reset_index(name='count')

# Top N pairs by count for each type
top_pairs_movie = actor_director_type_count[actor_director_type_count['type'] == 'Movie'].nlargest(10, 'count')
top_pairs_tvshow = actor_director_type_count[actor_director_type_count['type'] == 'TV Show'].nlargest(10, 'count')

# Plotting for movies
plt.figure(figsize=(12, 8))
sns.barplot(x='count', y='actor_director', data=top_pairs_movie, palette='viridis')
plt.title('Top 10 Actor-Director Pairs by Count for Movies')
plt.xlabel('Count')
plt.ylabel('Actor-Director Pair')
plt.show()

# Plotting for TV shows
plt.figure(figsize=(12, 8))
sns.barplot(x='count', y='actor_director', data=top_pairs_tvshow, palette='viridis')
plt.title('Top 10 Actor-Director Pairs by Count for TV Shows')
plt.xlabel('Count')
plt.ylabel('Actor-Director Pair')
plt.show()
```





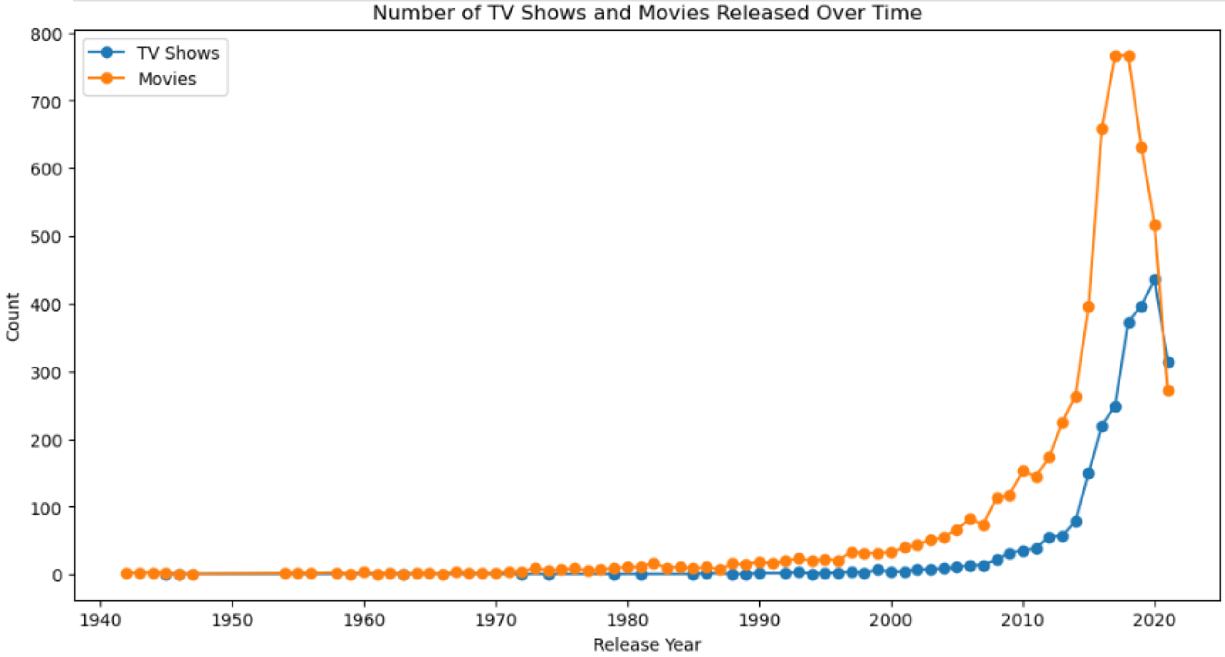
Insights: In this analysis, we delve into the top actor-director pairs, distinguishing between movies and TV shows. This insight is crucial for understanding user preferences, allowing content producers to make informed decisions about launching new movies and shows that align with the most favored actor-director pairs, ensuring content resonates well with the audience.

x. **Uncovering Growth Opportunities:** Assess growth opportunities by comparing TV shows vs. movies to determine focus areas for expansion. Questions: How do the popularity and growth trends differ between TV shows and movies? Are there specific content categories where Netflix could explore growth opportunities?



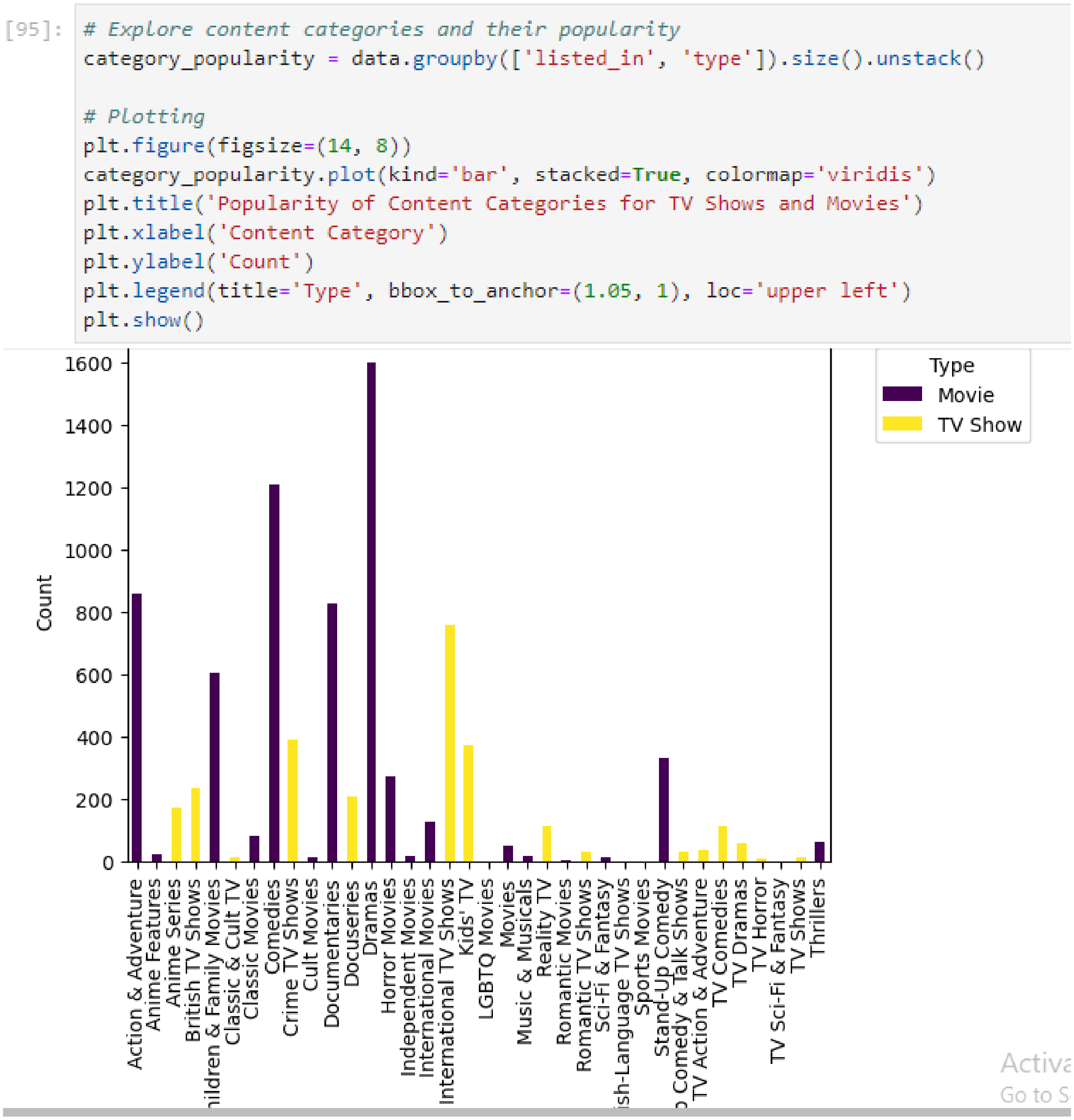
[94]:

```
#Q10.a. Movies and tv show trend
# Separate TV shows and movies
tv_shows = data[data['type'] == 'TV Show']
movies = data[data['type'] == 'Movie']

# Group by release year
tv_shows_by_year = tv_shows.groupby('release_year').size()
movies_by_year = movies.groupby('release_year').size()

# Plotting trends
plt.figure(figsize=(12, 6))
plt.plot(tv_shows_by_year.index, tv_shows_by_year.values, label='TV Shows', marker='o')
plt.plot(movies_by_year.index, movies_by_year.values, label='Movies', marker='o')
plt.title('Number of TV Shows and Movies Released Over Time')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.legend()
plt.show()
```



Insights: From 1942 to 1971, Netflix primarily introduced movies. However, starting in 1971, the inclusion of TV shows marked a significant shift. Between 1971 and 2000, both movies and TV shows were introduced at comparable rates. Subsequently, from 2000 onwards, movies experienced a substantial surge, peaking around 2016. In contrast, the growth in TV shows was more gradual, reaching a plateau by 2020. Post-2020, there was a decline in the introduction of both movies and TV shows.

This historical trend suggests that while movies witnessed a notable increase in production, TV shows gained popularity steadily over time. As TV shows tend to keep users engaged with Netflix for longer durations, a balanced approach, incorporating both TV shows and movies, seems beneficial for sustaining user interest and content diversity.



```
[95]: # Explore content categories and their popularity
      category_popularity = data.groupby(['listed_in', 'type']).size().unstack()

      # Plotting
      plt.figure(figsize=(14, 8))
      category_popularity.plot(kind='bar', stacked=True, colormap='viridis')
      plt.title('Popularity of Content Categories for TV Shows and Movies')
      plt.xlabel('Content Category')
      plt.ylabel('Count')
      plt.legend(title='Type', bbox_to_anchor=(1.05, 1), loc='upper left')
      plt.show()
```

Insights: The most popular genre for movies is Dramas whereas for tv shows is International Tv shows. Based on these different genre's popularity for movies and tv shows Netflix must launch more.

6. **Business Insights:**

Based on the comprehensive analysis of Netflix content, it is evident that the platform initially focused on introducing movies, later diversifying to include TV shows. Surprisingly, TV shows gained popularity swiftly, even though they were introduced after movies. However, a declining trend was observed for both TV shows after 2020 and movies after approximately 2016.

Examining genres, it was identified that 'Dramas' are the most prevalent genre for movies, while 'International TV shows' dominate the TV show category. Ratings analysis revealed a preference for adult-rated content, with a scarcity of kids-oriented movies and shows.

Key insights from keyword analysis highlighted that the top three commonly used keywords in titles and descriptions are 'life,' 'love,' and 'family.' Furthermore, strategic considerations for content release indicated that July is an optimal month, while February and March are less favorable.

In summary, the data-driven analysis provides valuable insights for Netflix content strategy, emphasizing the popularity of dramas, international TV shows, and adult-rated content. Additionally, it guides decisions on content release timing, emphasizing the significance of strategic planning for continued success.

7. **Recommendations:** Based on the gathered insights, it is advisable for Netflix to maintain a balanced approach by continuing to release both movies and TV shows. The inclusion of TV shows, especially those catered towards children, can significantly enhance user engagement over an extended period. It is crucial to consider user preferences for actors, directors, genres, and specific countries when planning new releases.

   While observing a decline in the trend for both movies and TV shows after 2020, it is imperative to investigate and address the underlying reasons. Identifying and rectifying these issues could potentially reinvigorate the popularity of Netflix's content.

   For optimal results, Netflix should strategically launch movies based on the preferences of actors and directors in specific countries. Additionally, considering the seasonal preferences of viewers, with a cautious approach to releasing fewer movies in February and March, and maximizing releases in July, could contribute to increased viewership.

   To enhance discoverability, incorporating top keywords into titles and descriptions is essential. Prioritizing countries with significant user bases, such as the USA, India, UK, Canada, and South Korea, can be a key focus for expanding and solidifying Netflix's presence in these regions.