netflix-bussiness-case

May 27, 2024

[88]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[89]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[90]:
```
df=pd.read_csv('/content/netflix-case study.csv')
```

[91]:
```
df.head(5)
```

[91]:
```
  show_id     type                   title         director  \
0      s1    Movie    Dick Johnson Is Dead  Kirsten Johnson
1      s2  TV Show           Blood & Water              NaN
2      s3  TV Show               Ganglands  Julien Leclercq
3      s4  TV Show   Jailbirds New Orleans              NaN
4      s5  TV Show            Kota Factory              NaN

                                                cast        country  \
0                                                NaN  United States
1  Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…   South Africa
2  Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…            NaN
3                                                NaN            NaN
4  Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…          India

          date_added  release_year rating   duration  \
0  September 25, 2021          2020  PG-13     90 min
1  September 24, 2021          2021  TV-MA  2 Seasons
2  September 24, 2021          2021  TV-MA   1 Season
3  September 24, 2021          2021  TV-MA   1 Season
4  September 24, 2021          2021  TV-MA  2 Seasons

                                           listed_in  \
0                                      Documentaries
```

```
1     International TV Shows, TV Dramas, TV Mysteries
2   Crime TV Shows, International TV Shows, TV Act…
3                              Docuseries, Reality TV
4   International TV Shows, Romantic TV Shows, TV …

                                         description
0   As her father nears the end of his life, filmm…
1   After crossing paths at a party, a Cape Town t…
2   To protect his family from a powerful drug lor…
3   Feuds, flirtations and toilet talk go down amo…
4   In a city of coaching centers known to train I…
```

[92]:
```
df.shape
```

[92]:
```
(8807, 12)
```

[93]:
```
df.isnull().sum()
```

[93]:
```
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
```

Splitting the directors by title

[94]:
```
df_director_r=pd.DataFrame(df['director'].apply(lambda x: str(x).split(',')).
 ↪tolist(),index=df['title'])
```

[95]:
```
df_director=df_director_r.stack().reset_index()
df_director.drop('level_1',axis=1,inplace=True)
df_director.rename(columns = {0:'director'},inplace= True)
df_director.head(5)
```

[95]:
```
                   title          director
0   Dick Johnson Is Dead  Kirsten Johnson
1          Blood & Water              nan
2              Ganglands  Julien Leclercq
3  Jailbirds New Orleans              nan
```

```
4           Kota Factory                  nan
```

Splitting cast by title

[96]:
```
df_cast_r=pd.DataFrame(df['cast'].apply(lambda x: str(x).split(',')).
 ↪tolist(),index=df['title'])
```

[97]:
```
df_cast=df_cast_r.stack().reset_index()
df_cast.drop('level_1',axis=1,inplace=True)
df_cast.rename(columns = {0:'cast'},inplace= True)
df_cast.head(5)
```

[97]:
```
                  title            cast
0  Dick Johnson Is Dead             nan
1         Blood & Water      Ama Qamata
2         Blood & Water     Khosi Ngema
3         Blood & Water   Gail Mabalane
4         Blood & Water  Thabang Molaba
```

Splitting country by tittle

[98]:
```
df_country_r=pd.DataFrame(df['country'].apply(lambda x: str(x).split(',')).
 ↪tolist(),index=df['title'])
df_country=df_country_r.stack().reset_index()
df_country.drop('level_1',axis=1,inplace= True)
df_country.rename(columns={0:'country'},inplace=True)
df_country.head(5)
```

[98]:
```
                   title        country
0   Dick Johnson Is Dead  United States
1          Blood & Water   South Africa
2              Ganglands            nan
3  Jailbirds New Orleans            nan
4           Kota Factory          India
```

Splitting listed_in by title

[99]:
```
df_listed_in_r=pd.DataFrame(df['listed_in'].apply(lambda x: str(x).split(',')).
 ↪tolist(),index=df['title'])
df_listed_in=df_listed_in_r.stack().reset_index()
df_listed_in.drop('level_1',axis=1,inplace=True)
df_listed_in.rename(columns={0:'listed_in'},inplace=True)
df_listed_in.head(5)
```

[99]:
```
                  title               listed_in
0  Dick Johnson Is Dead           Documentaries
1         Blood & Water  International TV Shows
2         Blood & Water               TV Dramas
```

```
3          Blood & Water            TV Mysteries
4              Ganglands          Crime TV Shows
```

```
[100]: df.columns
```

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

Merging the columns

```
[101]: df_new1=df_director.merge(df_cast,how='inner',on = 'title')
       df_new1
       df_new2=df_listed_in.merge(df_country,how='inner',on='title')
       df_new2
       df_new=df_new1.merge(df_new2,how= 'inner',on='title')
       df_new.head(5)
```

```
[101]:                    title          director          cast  \
       0  Dick Johnson Is Dead  Kirsten Johnson           nan
       1         Blood & Water               nan    Ama Qamata
       2         Blood & Water               nan    Ama Qamata
       3         Blood & Water               nan    Ama Qamata
       4         Blood & Water               nan  Khosi Ngema

                      listed_in        country
       0          Documentaries  United States
       1  International TV Shows   South Africa
       2              TV Dramas   South Africa
       3           TV Mysteries   South Africa
       4  International TV Shows   South Africa
```

```
[102]: df.columns
```

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

Creating the df_final DataFrame

```
[103]: df_final=df_new.merge(df[['show_id', 'type', 'title',  'date_added',
              'release_year', 'rating', 'duration',␣
        ↪'description']],how='inner',on='title')
       df_final.sample(5)
```

```
[103]:                          title          director             cast  \
       133435                  Results   Andrew Bujalski  Brooklyn Decker
       74241                      MFKZ   Shojiro Nishimi     Kenn Michael
```

```
88840   Neon Genesis Evangelion                 nan   Takehito Koyasu
41563            Alone/Together  Antoinette Jadaone     Adrian Alandy
77979               Shot Caller     Ric Roman Waugh    Omari Hardwick

                      listed_in        country show_id     type  \
133435       Independent Movies  United States   s5890    Movie
74241                  Comedies          Japan   s3102    Movie
88840    International TV Shows          Japan   s3725  TV Show
41563      International Movies    Philippines   s1733    Movie
77979                    Dramas  United States   s3236    Movie

               date_added  release_year rating  duration  \
133435   October 22, 2015          2015      R   105 min
74241   December 25, 2019          2017      R    95 min
88840       June 21, 2019          1995  TV-MA  1 Season
41563    November 5, 2020          2019  TV-14   103 min
77979   November 24, 2019          2017      R   121 min

                                              description
133435  After an ugly divorce, a wealthy but unhappy m…
74241   Mysterious goons chase teenage slacker Angelin…
88840   Fifteen years after the Second Impact, Shinji …
41563   Eight years after their breakup, college sweet…
77979   Trying to go straight, a once-successful busin…
```

Replacing ‘Nan’ values with Unknown

[104]:

```
df_final['director'].replace(['nan'],['Unknown Director'],inplace = True)
df_final['cast'].replace(['nan'],['Unknown Cast'],inplace = True)
df_final['date_added'].replace(['nan'],['Unknown Date'],inplace = True)
df_final['country'].replace(['nan'],['Unknown Country'],inplace=True)

df_final.head(5)
```

[104]:

```
                  title          director          cast  \
0  Dick Johnson Is Dead   Kirsten Johnson  Unknown Cast
1         Blood & Water  Unknown Director    Ama Qamata
2         Blood & Water  Unknown Director    Ama Qamata
3         Blood & Water  Unknown Director    Ama Qamata
4         Blood & Water  Unknown Director   Khosi Ngema

                listed_in        country show_id     type          date_added  \
0           Documentaries  United States      s1    Movie  September 25, 2021
1  International TV Shows   South Africa      s2  TV Show  September 24, 2021
2               TV Dramas   South Africa      s2  TV Show  September 24, 2021
3            TV Mysteries   South Africa      s2  TV Show  September 24, 2021
4  International TV Shows   South Africa      s2  TV Show  September 24, 2021
```

```
   release_year rating   duration  \
0          2020  PG-13     90 min   
1          2021  TV-MA  2 Seasons   
2          2021  TV-MA  2 Seasons   
3          2021  TV-MA  2 Seasons   
4          2021  TV-MA  2 Seasons   

                                         description  
0  As her father nears the end of his life, filmm…  
1  After crossing paths at a party, a Cape Town t…  
2  After crossing paths at a party, a Cape Town t…  
3  After crossing paths at a party, a Cape Town t…  
4  After crossing paths at a party, a Cape Town t…  
```

Checking for null values

[105]:
```
df_final[df_final['duration'].isnull()]
```

[105]:
```
                                       title    director        cast  \
126582                       Louis C.K. 2017  Louis C.K.  Louis C.K.   
131648                 Louis C.K.: Hilarious  Louis C.K.  Louis C.K.   
131782  Louis C.K.: Live at the Comedy Store  Louis C.K.  Louis C.K.   

       listed_in        country show_id   type          date_added  \
126582    Movies  United States   s5542  Movie       April 4, 2017   
131648    Movies  United States   s5795  Movie  September 16, 2016   
131782    Movies  United States   s5814  Movie     August 15, 2016   

        release_year  rating duration  \
126582          2017  74 min      NaN   
131648          2010  84 min      NaN   
131782          2015  66 min      NaN   

                                              description  
126582  Louis C.K. muses on religion, eternal love, gi…  
131648  Emmy-winning comedy writer Louis C.K. brings h…  
131782  The comic puts his trademark hilarious/thought…  
```

[106]:
```
df_final[df_final['duration'].isnull()]
```

[106]:
```
                                       title    director        cast  \
126582                       Louis C.K. 2017  Louis C.K.  Louis C.K.   
131648                 Louis C.K.: Hilarious  Louis C.K.  Louis C.K.   
131782  Louis C.K.: Live at the Comedy Store  Louis C.K.  Louis C.K.   

       listed_in        country show_id   type          date_added  \
126582    Movies  United States   s5542  Movie       April 4, 2017
131648    Movies  United States   s5795  Movie  September 16, 2016
131782    Movies  United States   s5814  Movie     August 15, 2016

        release_year  rating duration  \
126582          2017  74 min      NaN
131648          2010  84 min      NaN
131782          2015  66 min      NaN

                                              description
126582  Louis C.K. muses on religion, eternal love, gi…
131648  Emmy-winning comedy writer Louis C.K. brings h…
131782  The comic puts his trademark hilarious/thought…
```

[107]:

```
import pandas as pd

# Assuming df_final is already defined and loaded with the relevant data

# Fill missing 'duration' values with 'rating' values
df_final['duration'] = df_final['duration'].fillna(df_final['rating'])

# Update specific rows in 'rating' column
df_final.loc[126582, 'rating'] = "unknown rating"
df_final.loc[131648, 'rating'] = "unknown rating"
df_final.loc[131782, 'rating'] = "unknown rating"
```

[108]:

```
for ele in df_final[df_final['rating'].isnull()]:
    df_final['rating'] = df_final['rating'].fillna('unknown rating')
```

[109]:

```
for ele in df_final[df_final['date_added'].isnull()]:
    df_final['date_added'] = df_final['date_added'].fillna('unknown date')
```

Confirming no further nulls in the df_final

[110]:

```
df_final.isnull().sum()
```

[110]:

```
title           0
director        0
cast            0
listed_in       0
country         0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
```

```
description     0
dtype: int64
```

Final data frame after cleaning as ‘df_final’

[111]:
```
df_final.head()
```

[111]:
```
                  title          director          cast  \
0  Dick Johnson Is Dead   Kirsten Johnson  Unknown Cast
1         Blood & Water  Unknown Director    Ama Qamata
2         Blood & Water  Unknown Director    Ama Qamata
3         Blood & Water  Unknown Director    Ama Qamata
4         Blood & Water  Unknown Director   Khosi Ngema

                listed_in        country show_id     type          date_added  \
0           Documentaries  United States      s1    Movie  September 25, 2021
1  International TV Shows   South Africa      s2  TV Show  September 24, 2021
2               TV Dramas   South Africa      s2  TV Show  September 24, 2021
3            TV Mysteries   South Africa      s2  TV Show  September 24, 2021
4  International TV Shows   South Africa      s2  TV Show  September 24, 2021

   release_year rating   duration  \
0          2020  PG-13     90 min
1          2021  TV-MA  2 Seasons
2          2021  TV-MA  2 Seasons
3          2021  TV-MA  2 Seasons
4          2021  TV-MA  2 Seasons

                                         description
0  As her father nears the end of his life, filmm…
1  After crossing paths at a party, a Cape Town t…
2  After crossing paths at a party, a Cape Town t…
3  After crossing paths at a party, a Cape Town t…
4  After crossing paths at a party, a Cape Town t…
```

1. What type of content is available in different countries?

[112]:
```
#Non-graphical analysis
type_uni=df_final['type'].nunique()
country_uni=df_final['country'].nunique()
print(f'The no.of unique values of type of content is :{type_uni} ')
print(f'The no.of unique country content :{country_uni} ')
```

```
The no.of unique values of type of content is :2
The no.of unique country content :198
```

[113]:
```
country_wise=df_final.groupby(['country'])['type'].value_counts()
country_wise_df=country_wise.reset_index(name='counts')
```

```python
country_wise_df

# Remove the 'counts' column
country_wise_df.drop(columns=['counts'], inplace=True,errors='ignore')

# Merge rows for countries with both "Movie" and "TV Show"
country_wise_df = country_wise_df.groupby('country')['type'].apply(lambda x: '␣
  ↪& '.join(map(str, x))).reset_index()

# Function to map types to numeric categories
def map_category(types):
    if 'Movie' in types and 'TV Show' in types:
        return 2
    elif 'Movie' in types:
        return 0
    elif 'TV Show' in types:
        return 1
    else:
        return None

# Apply the mapping function to create the 'category' column
country_wise_df['category'] = country_wise_df['type'].apply(map_category)

country_wise_df.drop(columns=['type'], inplace=True)
country_wise_df.rename(columns={'category': 'type'},inplace=True)
country_wise_df
```

[113]:

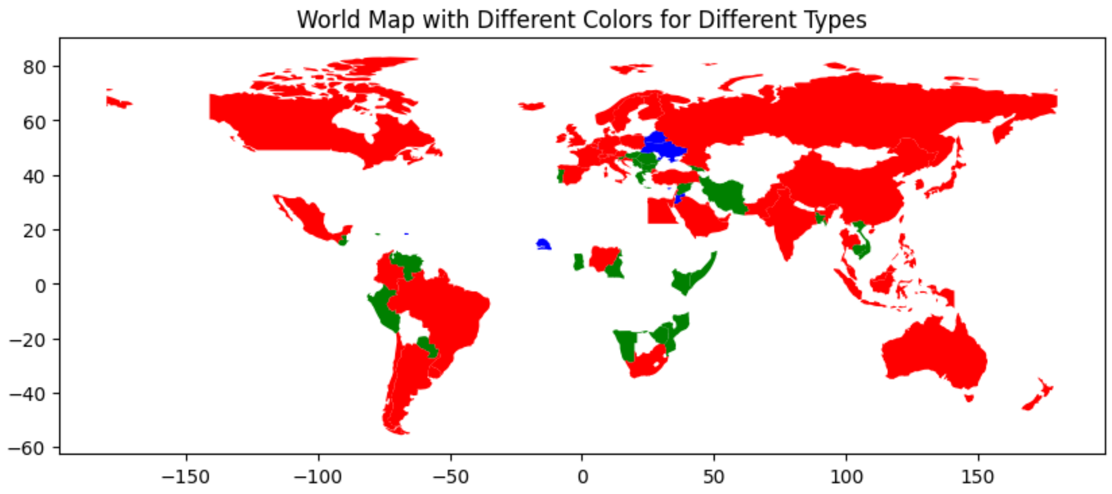
```
          country  type
0                     2
1     Afghanistan     0
2         Albania     0
3         Algeria     0
4          Angola     0
..            ...   ...
193       Uruguay     2
194     Venezuela     0
195       Vietnam     0
196  West Germany     0
197      Zimbabwe     0

[198 rows x 2 columns]
```

[114]:

```python
import geopandas as gpd


world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

```

```
# Merge world map with the data
world = world.merge(country_wise_df, how='left', left_on='name',␣
 ↪right_on='country')

# Plotting
fig, ax = plt.subplots(1, 1, figsize=(10, 6))

# Plot countries with type 0 in one color
world[world['type'] == 0].plot(ax=ax, color='green', legend=True)

# Plot countries with type 1 in another color
world[world['type'] == 1].plot(ax=ax, color='blue', legend=True)

# Plot countries with type 2 in another color
world[world['type'] == 2].plot(ax=ax, color='red', legend=True)

ax.set_title('World Map with Different Colors for Different Types')
plt.show()
```

```
<ipython-input-114-a2bd7936e57b>:4: FutureWarning: The geopandas.dataset module
is deprecated and will be removed in GeoPandas 1.0. You can get the original
'naturalearth_lowres' data from
https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.
  world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
```



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

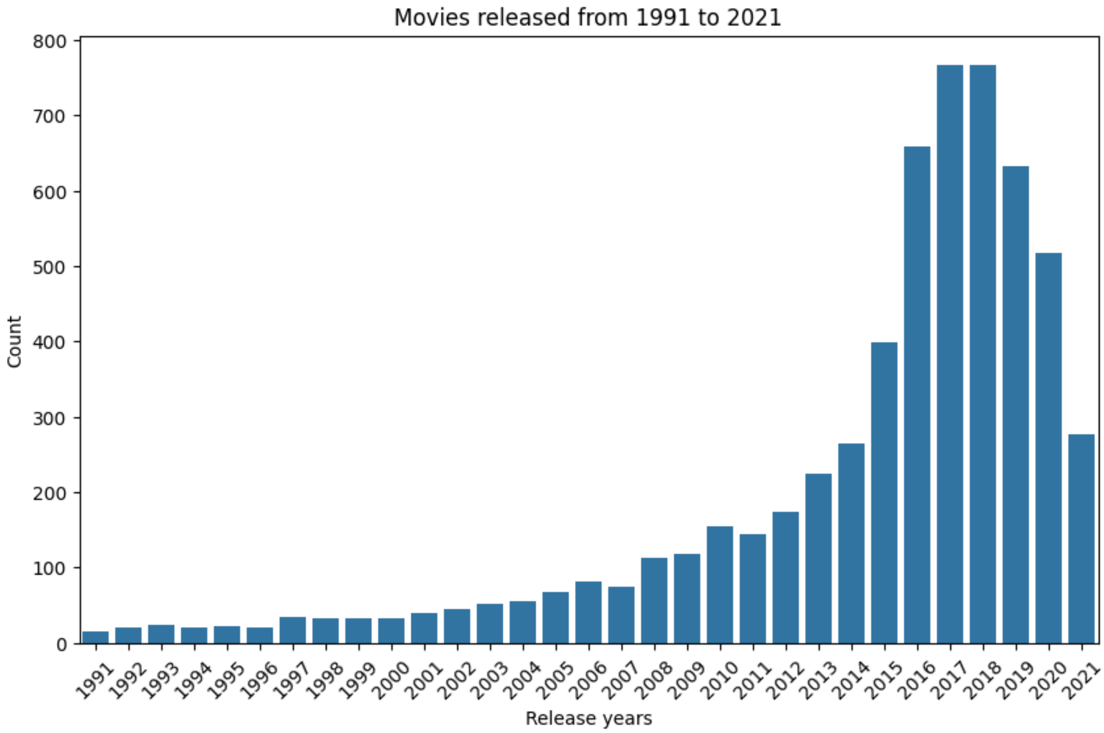
```
[115]: # Extract relevant columns and remove duplicates
movies_yr = df_final[['title', 'type', 'release_year']].copy()
movies_yr.drop_duplicates(subset=['title'], inplace=True)
movies_yr.reset_index(drop=True, inplace=True)

# Group by type and release year, then count the occurrences
movies_yr = movies_yr.groupby(['type', 'release_year']).size().
 ↪reset_index(name='count')

# Filter the DataFrame to include only movies
movies_yr_df = movies_yr[movies_yr['type'] == 'Movie'].copy()

# Sort the DataFrame by release year
movies_yr_df.sort_values(by='release_year', inplace=True)

# Define the time period for the last 30 years
end_year = 2021
start_year = end_year - 30

# Filter the DataFrame for the specified time period
movies_yr_df = movies_yr_df[
    (movies_yr_df['release_year'] >= start_year) &
    (movies_yr_df['release_year'] <= end_year)
]

# Display the first few rows of the filtered DataFrame
movies_yr_df.head()
```

[115]:

```
     type  release_year  count
42  Movie          1991     16
43  Movie          1992     20
44  Movie          1993     24
45  Movie          1994     20
46  Movie          1995     23
```

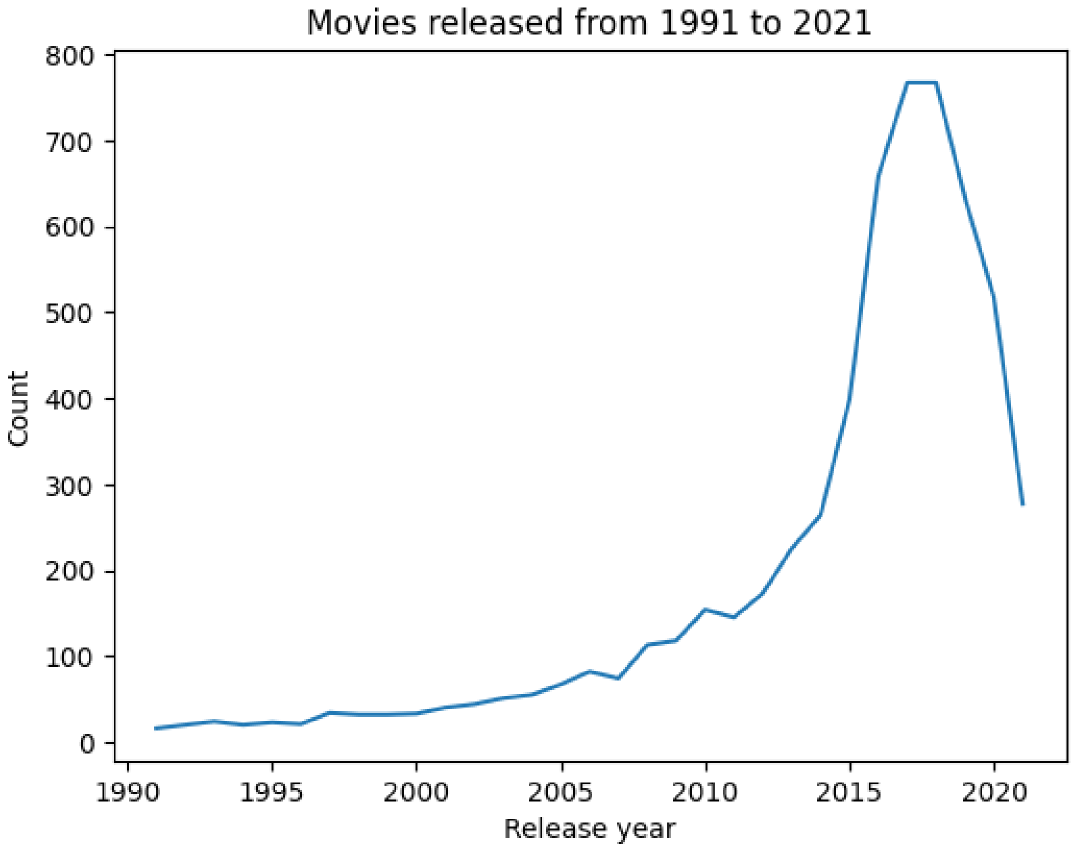
[116]:

```
plt.figure(figsize=(10, 6))
sns.barplot(x='release_year',y='count',data=movies_yr_df)
plt.xlabel('Release years')
plt.ylabel('Count')
plt.title('Movies released from 1991 to 2021')
plt.xticks(rotation=45)
plt.show()
```



[117]:

```
sns.lineplot(x='release_year',y='count',data=movies_yr_df)
plt.title('Movies released from 1991 to 2021')
plt.xlabel('Release year')
plt.ylabel('Count')
plt.show()
```



3.Comparison of tv shows vs. movies.

[118]:

```
# Extracting only title, type and release_year columns from cleaned data
movie_shows = df_final[['title', 'type', 'release_year']].copy()

# Deleting duplicate rows
movie_shows.drop_duplicates(subset=['title'], inplace=True)

# Resetting the index after deleting duplicate rows
movie_shows = movie_shows.reset_index(drop=True)

# Grouping by type (Movies and TV shows) and value counting the number of␣
  ↪movies and TV shows per year
movie_shows = movie_shows.groupby('type')['release_year'].value_counts()

# Resetting index again
movie_shows = movie_shows.reset_index(name='count')

# Splitting the dataframe by movie and TV show
movie_shows_df = movie_shows[movie_shows['type'] == 'Movie'].copy()
```

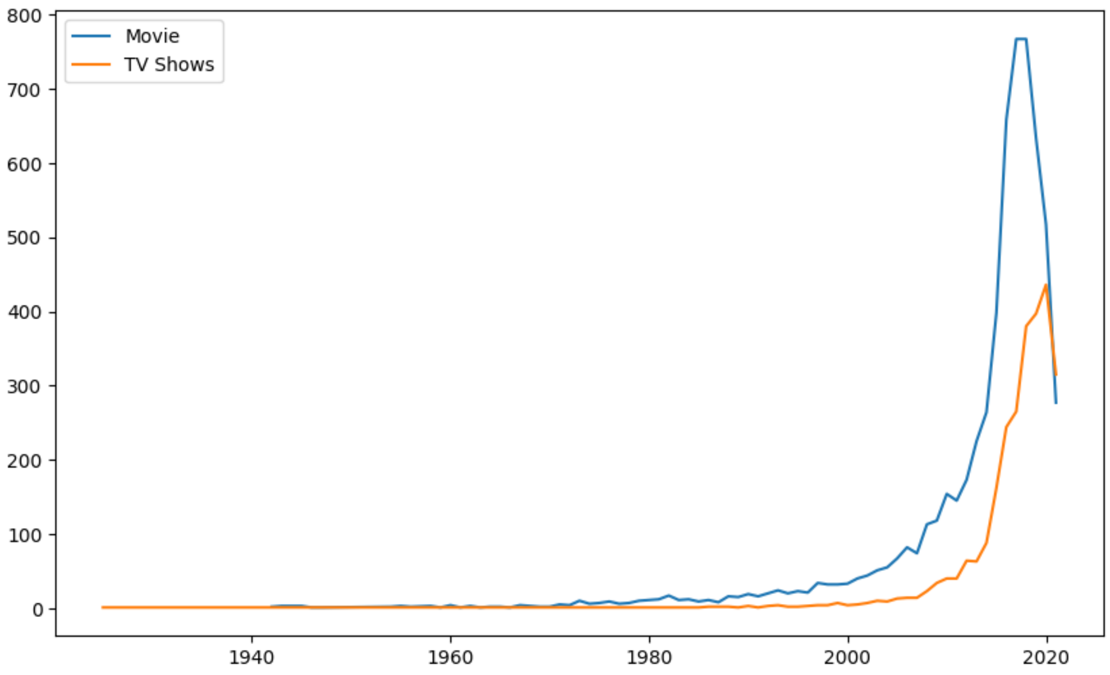
```
tv_shows_df = movie_shows[movie_shows['type'] == 'TV Show'].copy()

# Sorting the dataframes by release year
movie_shows_df.sort_values(by='release_year', inplace=True)
tv_shows_df.sort_values(by='release_year', inplace=True)

movie_shows_df.head(), tv_shows_df.head()
```

[118]:

```
(     type  release_year  count
 62  Movie          1942      2
 59  Movie          1943      3
 58  Movie          1944      3
 56  Movie          1945      3
 68  Movie          1946      1,
         type  release_year  count
 107  TV Show          1925      1
 106  TV Show          1945      1
 108  TV Show          1946      1
 110  TV Show          1963      1
 111  TV Show          1967      1)
```

[119]:

```
plt.figure(figsize=(10, 6))
# Plot the dataframes
plt.plot(movie_shows_df['release_year'], movie_shows_df['count'], label='Movie')
plt.plot(tv_shows_df['release_year'], tv_shows_df['count'], label='TV Shows')

# Add a legend
plt.legend()

# Show the plot
plt.show()
```



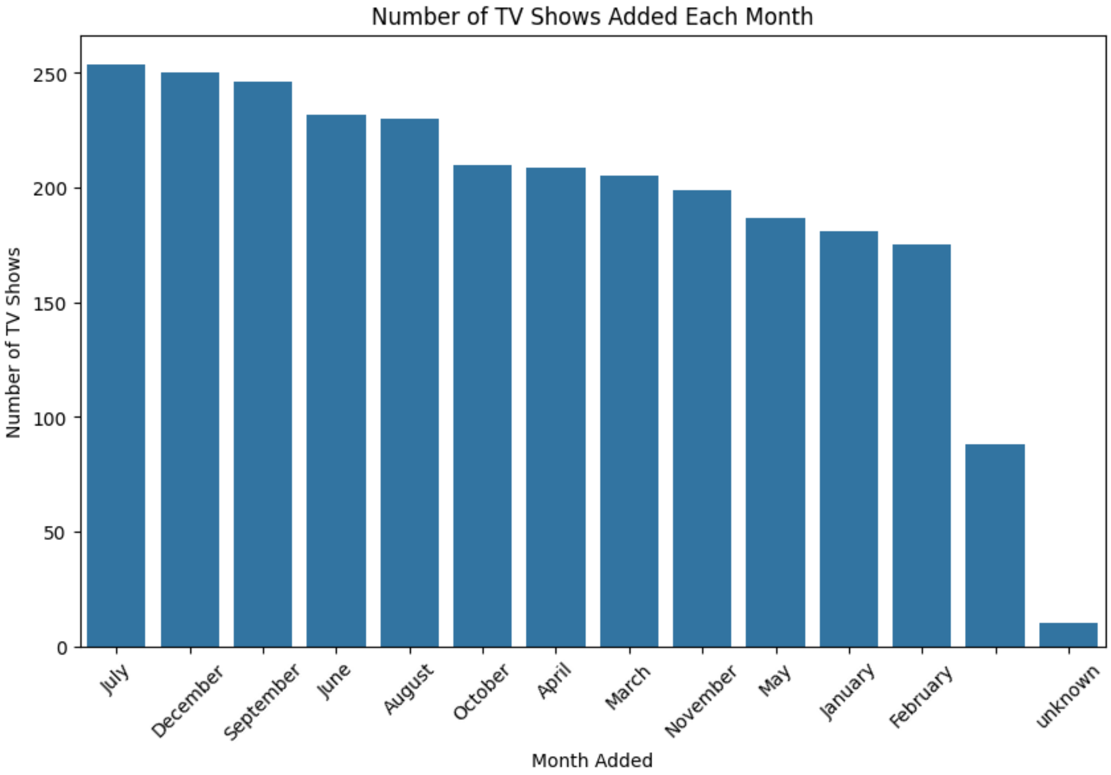
4. What is the best time to launch a TV show?

[120]:

```
# Extract relevant columns and remove duplicates
best_time = df_final[['title', 'date_added', 'type']].copy()
best_time.drop_duplicates(subset=['title'], inplace=True)
best_time.reset_index(drop=True, inplace=True)

# Remove rows where type is 'Movie'
best_time = best_time[best_time['type'] != 'Movie'].copy()

# Extract the month from the 'date_added' column
best_time['month_added'] = best_time['date_added'].apply(lambda x: x.split('␣
 ↪')[0])

# Count the number of TV shows added each month
best = best_time['month_added'].value_counts().reset_index()
best.columns = ['month_added', 'count']

# Extract the top 5 months
top_months = best.nlargest(5, 'count')['month_added'].tolist()
print(f"The best time to launch a TV show is: \n{top_months}")
```

```
The best time to launch a TV show is:
['July', 'December', 'September', 'June', 'August']
```

```
[121]: plt.figure(figsize=(10, 6))
       sns.barplot(x='month_added', y='count', data=best, order=best['month_added'])
       plt.xlabel('Month Added')
       plt.ylabel('Number of TV Shows')
       plt.title('Number of TV Shows Added Each Month')
       plt.xticks(rotation=45)
       plt.show()
```

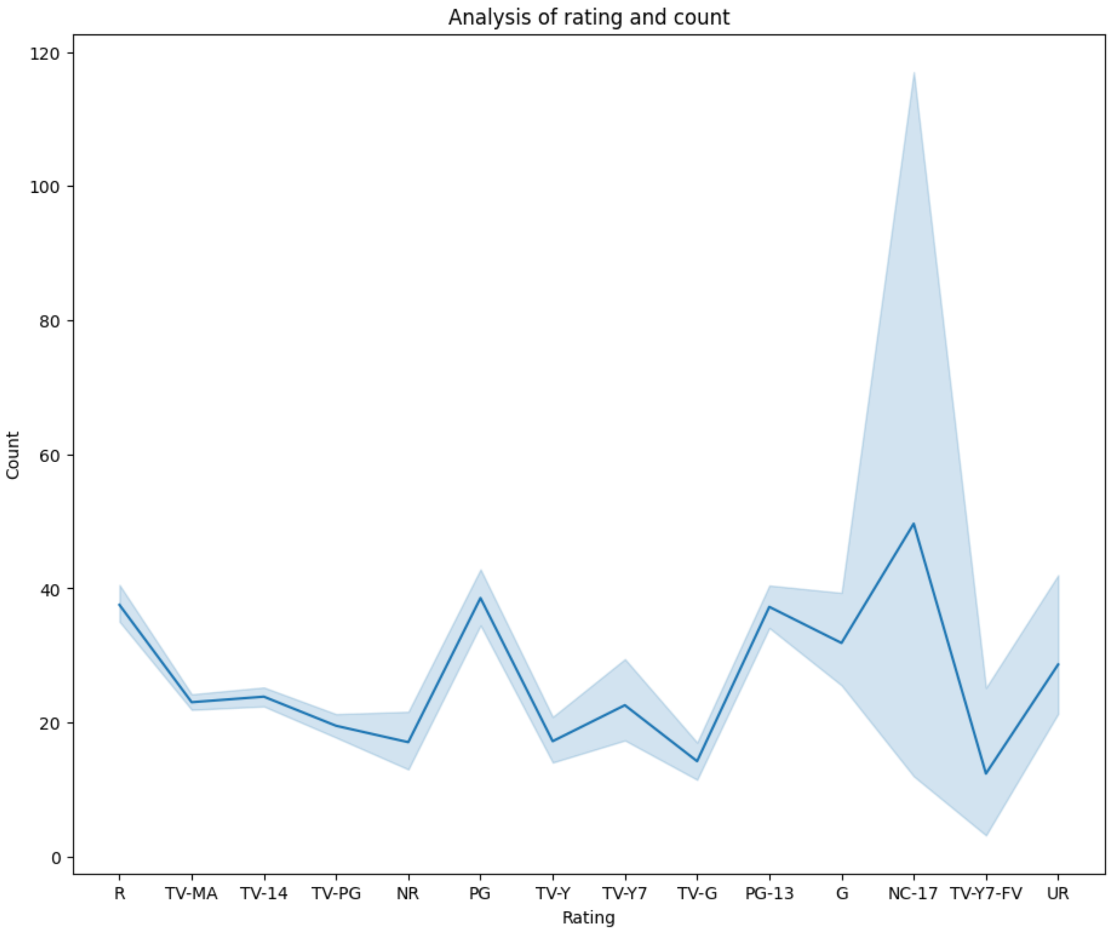


5. Analysis of actors/directors of different types of shows/movies.

```
[122]: analysis_dir = df_final[['director','rating','type','title']].copy()
       analysis_dir.drop_duplicates(subset=['title'], inplace=True)
       analysis_dir.reset_index(drop=True, inplace=True)

       analysis_dir = df_final.groupby(['director','rating','type']).size().
        ↪reset_index(name='count').sort_values(by=['director', 'rating','type'])
       movie_shows_dir_df = analysis_dir[analysis_dir['type'] == 'Movie'].copy()
       movie_shows_dir_df.drop(movie_shows_dir_df[movie_shows_dir_df['rating'] ==␣
        ↪'unknown rating'].index,inplace=True)

       movie_shows_dir_df.head(5)
```

```
[122]:                    director rating   type  count
       0             Aaron Moorhead      R  Movie      6
       1             Aaron Moorhead  TV-MA  Movie     18
       2                Aaron Woolf  TV-14  Movie      2
       3   Abbas Alibhai Burmawalla  TV-14  Movie     18
       4           Abdullah Al Noor  TV-MA  Movie     30
```

```
[123]: plt.figure(figsize=(11, 9))
       sns.lineplot(x='rating',y='count',data=movie_shows_dir_df)
       plt.title('Analysis of rating and count')
       plt.xlabel('Rating')
       plt.ylabel('Count')
       plt.show()
```



6. Does Netflix has more focus on TV Shows than movies in recent years

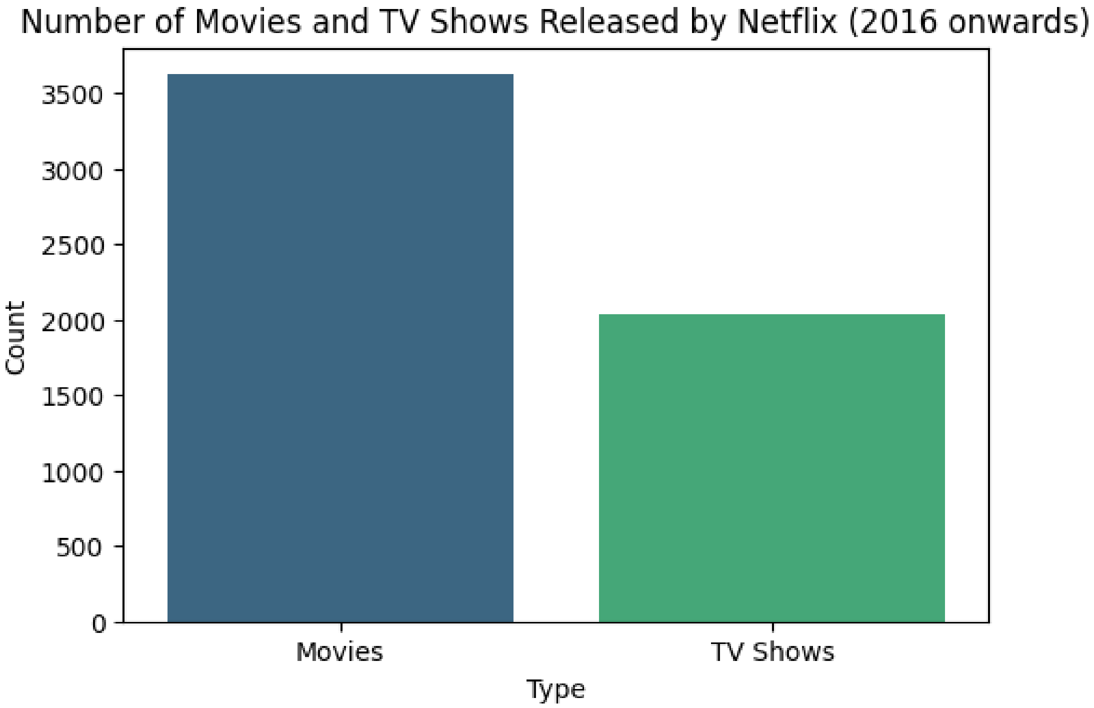
```
[124]: recent_years = df_final[df_final['release_year'] >= 2016]
       recent_years = recent_years.drop_duplicates(subset=["title"])
```

```python
type_counts = recent_years['type'].value_counts()
movie = type_counts.get('Movie', 0)
tv_show = type_counts.get('TV Show', 0)
print(f"Number of movies in recent years: {movie}")
print(f"Number of TV shows in recent years: {tv_show}")

if movie > tv_show:
    print("Netflix has more focus on movies in recent years.")
elif movie < tv_show:
    print("Netflix has more focus on TV shows in recent years.")
```

```
Number of movies in recent years: 3619
Number of TV shows in recent years: 2037
Netflix has more focus on movies in recent years.
```

[125]:

```python
plt.figure(figsize=(6, 4))
sns.barplot(x=['Movies', 'TV Shows'], y=[movie, tv_show], hue=['Movies', 'TV Shows'], palette='viridis', legend=False)
plt.xlabel('Type')
plt.ylabel('Count')
plt.title('Number of Movies and TV Shows Released by Netflix (2016 onwards)')
plt.show()
```



7 .Understanding what content is available in different countries

[126]:

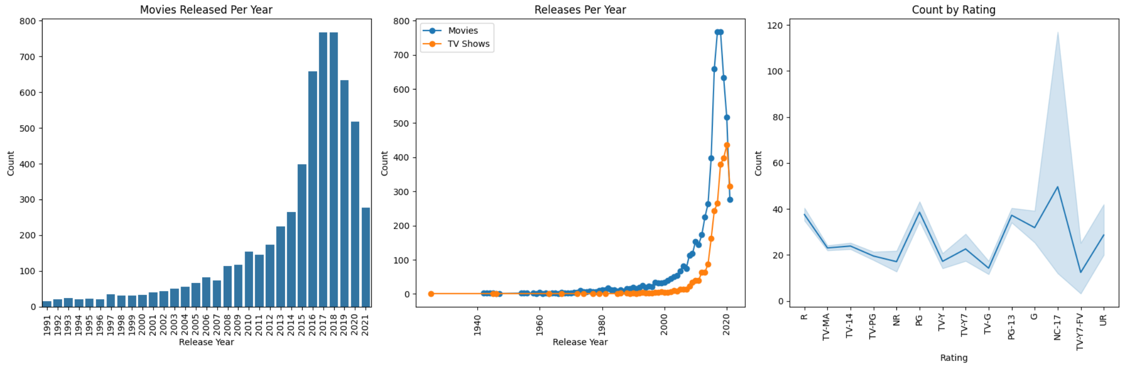
```
df_understanding=df_final.groupby(['country'])['listed_in'].value_counts()
df_understanding_1=df_understanding.reset_index(name='counts')
df_understanding_1.sample(5)
```

[126]:

```
                   country                listed_in  counts
936   United Arab Emirates                Thrillers       8
2180           Switzerland     Faith & Spirituality       1
141                 Canada                  Kids' TV      58
2007                Russia   TV Action & Adventure       8
1773              Malaysia              TV Comedies       7
```

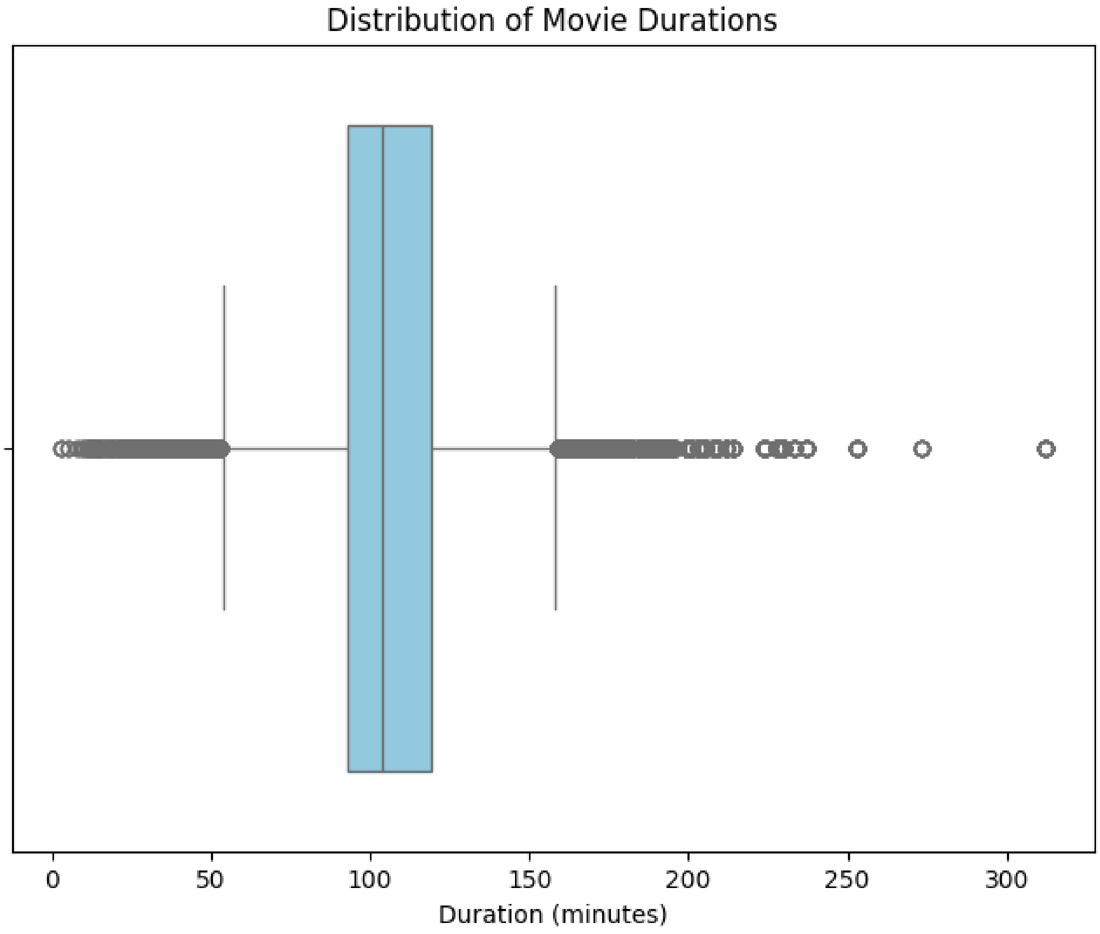
[128]:

```
# Create a figure and subplots
plt.figure(figsize=(18, 6))

# Plot first subplot
plt.subplot(1, 3, 1)
sns.barplot(x='release_year', y='count', data=movies_yr_df)
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.title('Movies Released Per Year')
plt.xticks(rotation=90)

# Plot second subplot
plt.subplot(1, 3, 2)
plt.plot(movie_shows_df['release_year'], movie_shows_df['count'],␣
 ↪label='Movies', marker='o')
plt.plot(tv_shows_df['release_year'], tv_shows_df['count'], label='TV Shows',␣
 ↪marker='o')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.title('Releases Per Year')
plt.legend()
plt.xticks(rotation=90)

# Plot third subplot
plt.subplot(1, 3, 3)
sns.lineplot(x='rating', y='count', data=movie_shows_dir_df)
plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Count by Rating')
plt.xticks(rotation=90)

# Adjust layout to prevent overlap
plt.tight_layout()

# Show the plot
plt.show()
```



Box plot for the distribution of movie duration.

```
[129]: # Filter out TV shows
movies_data = df_final[df_final['type'] == 'Movie'].copy()

# Convert duration to numeric (remove 'min' and convert to integer)
movies_data['duration'] = movies_data['duration'].str.replace(' min', '').
  ↪astype(int)

# Create a boxplot
plt.figure(figsize=(8, 6))
sns.boxplot(x=movies_data['duration'], color='skyblue')
plt.title('Distribution of Movie Durations')
plt.xlabel('Duration (minutes)')
plt.show()
```



0.0.1 1. ***Content Availability in Different Countries***

Insight: - Different countries have access to different types of content, including movies, TV shows, or both.

Recommendation: - *Localization Strategy:* Tailor content offerings to specific countries based on local preferences. For instance, if a country shows a higher preference for TV shows, focus on acquiring or producing more TV shows for that market. - *Content Diversity:* Ensure a mix of both movies and TV shows in each market to cater to a broader audience and avoid saturating the market with one type of content.

0.0.2 2. ***Trends in Movie Releases Over the Last 20-30 Years***

Insight: - There is a noticeable trend in the number of movies released over the years, with fluctuations possibly influenced by industry changes and global events.

Recommendation: - *Historical Analysis:* Use historical data to predict future trends in movie releases and plan content acquisition accordingly. - *Seasonal Releases:* Identify peak times for movie releases and schedule major releases during these periods to maximize viewership.

0.0.3 3. ***Comparison of TV Shows vs. Movies***

Insight: - There has been a notable difference in the release patterns of movies and TV shows over the years.

Recommendation: - *Content Balance:* Maintain a balance between movies and TV shows to cater to different audience preferences. - *Genre Analysis:* Analyze popular genres within both categories to optimize content acquisition and production.

0.0.4 4. ***Best Time to Launch a TV Show***

Insight: - Certain months have higher counts of TV show releases, indicating possibly favorable times for launching new shows.

Recommendation: - *Optimal Release Timing:* Plan the launch of new TV shows during months with historically high viewership to capitalize on audience engagement. - *Marketing Campaigns:* Align marketing efforts with these optimal times to maximize the impact of new releases.

0.0.5 5. ***Analysis of Actors/Directors in Different Types of Shows/Movies***

Insight: - Certain directors and actors are associated with specific types of content and ratings.

Recommendation: - *Talent Acquisition:* Focus on collaborating with directors and actors who have a proven track record of success in the type of content Netflix wants to promote. - *Star Power:* Use well-known actors and directors to attract viewership, especially in new markets or for new content types.

0.0.6 6. ***Focus on TV Shows vs. Movies in Recent Years***

Insight: - There has been a shift in focus towards TV shows in recent years.

Recommendation: - *Strategic Focus:* If the trend shows a growing interest in TV shows, continue to invest in high-quality TV show production and acquisition. - *Content Expansion:* Consider expanding the TV show library, including different genres and international TV shows, to cater to diverse tastes.

0.0.7 7. ***Understanding Content Available in Different Countries***

Insight: - The content available varies significantly between countries, reflecting local tastes and preferences.

Recommendation: - *Localized Content:* Increase investment in local content production and acquisitions to cater to regional preferences. - *Global Distribution:* Ensure popular content is available globally, with localized subtitles and dubbing to enhance accessibility and viewership.

0.0.8 General Recommendations:

- *Data-Driven Decisions:* Continuously use data analytics to monitor viewer preferences and trends, enabling more informed decisions about content acquisition and production.
- *User Feedback:* Regularly collect and analyze user feedback to understand their preferences and improve the content library accordingly.

- *Marketing Strategies:* Tailor marketing strategies based on regional insights and viewing habits to maximize engagement and subscription rates.
- *Content Innovation:* Explore new and innovative content formats, such as interactive shows or short-form content, to stay ahead in the competitive streaming market.

By leveraging these insights and recommendations, Netflix can optimize its content strategy to enhance user engagement, satisfaction, and overall market competitiveness.