Netflix - Data Exploration and Visualisation

[3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[45]:
```
netflix =  pd.read_csv('netflix.csv')
```

Problem Statement

Analyze Netflix's catalog to understand what types of shows and movies are popular globally and in specific countries. Provide recommendations on how Netflix can tailor its content to attract more viewers and grow its business in different regions.

Basic Analysis

[5]:
```
netflix.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
```

[6]:
```
netflix.sample(5)
```

[6]:
```
      show_id     type                         title        director  \
8455   s8456    Movie             The Pirate Fairy    Peggy Holmes
2082   s2083    Movie         Maniyarayile Ashokan    Shamzu Zayba
292     s293    Movie                      Quartet  Dustin Hoffman
7761   s7762  TV Show  Power Rangers Dino Charge             NaN
4728   s4729    Movie                        Joker  Shirish Kunder

                                                   cast         country  \
8455  Mae Whitman, Christina Hendricks, Tom Hiddlest…   United States
2082  Jacob Gregory, S.V. Krishna Shankar, Shine Tom…           India
292   Maggie Smith, Tom Courtenay, Billy Connolly, P…  United Kingdom
7761  Brennan Mejia, Camille Hyde, Yoshi Sudarso, Mi…   United States
4728  Akshay Kumar, Sonakshi Sinha, Shreyas Talpade,…           India

            date_added  release_year rating  duration  \
8455     June 15, 2014          2014      G    78 min
2082   August 31, 2020          2020  TV-14   110 min
292     August 8, 2021          2012  PG-13    98 min
7761  December 2, 2015          2015  TV-Y7  1 Season
4728    August 2, 2018          2012  TV-PG    98 min

                                        listed_in  \
8455                        Children & Family Movies
2082   Comedies, International Movies, Romantic Movies
292              Comedies, Dramas, Independent Movies
7761                                     Kids' TV
4728  Comedies, International Movies, Music & Musicals

                                          description
8455  In this spritely tale, Tinker Bell and her fri…
2082  When his unlucky horoscope doesn't bode well f…
292   To save their posh retirement home, former ope…
7761  In the time of dinosaurs, the ancient and powe…
4728  A remote village situated neither in India or …
```

Shape of the Data Frame

[7]:
```
netflix.shape
```

[7]:
```
(8807, 12)
```

Converting Type, Rating and Country Attributes from object Data Type to category

[8]:
```
netflix['type'] = netflix['type'].astype('category')
netflix['rating'] = netflix['rating'].astype('category')
netflix['country'] = netflix['country'].astype('category')
```

Statistical Summary

[9]:
```
netflix.describe()
```

[9]:
```
       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
```

Finding number of missing values in each column

[10]:
```
netflix.isna().sum()
```

[10]:
```
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
```

Number of movies in each type

[11]:
```
netflix['type'].value_counts()
```

[11]:
```
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

Number of movies directed by each director

[12]:
```
netflix['director'].value_counts()
```

[12]:
```
director
Rajiv Chilaka                     19
Raúl Campos, Jan Suter            18
Marcus Raboy                      16
Suhas Kadav                       16
```

```
Jay Karas                          14
                                   ..
Raymie Muzquiz, Stu Livingston      1
Joe Menendez                        1
Eric Bross                          1
Will Eisenberg                      1
Mozez Singh                         1
Name: count, Length: 4528, dtype: int64
```

Number of movies released in each year

[13]:
```
netflix['release_year'].value_counts()
```

[13]:
```
release_year
2018    1147
2017    1032
2019    1030
2020     953
2016     902
        …
1959       1
1925       1
1961       1
1947       1
1966       1
Name: count, Length: 74, dtype: int64
```

Number of movies released under each rating

[14]:
```
netflix['rating'].value_counts()
```

[14]:
```
rating
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
UR             3
NC-17          3
74 min         1
84 min         1
```

```
66 min         1
Name: count, dtype: int64
```

Move incorrect 'rating' values to the 'duration' column and Replace the incorrect 'rating' values with NaN or a default value

[15]:
```
netflix.loc[netflix['rating'] == '66 min', 'duration'] = '66 min'
netflix.loc[netflix['rating'] == '74 min', 'duration'] = '74 min'
netflix.loc[netflix['rating'] == '84 min', 'duration'] = '84 min'

netflix.loc[netflix['rating'] == '66 min', 'rating'] = None
netflix.loc[netflix['rating'] == '74 min', 'rating'] = None
netflix.loc[netflix['rating'] == '84 min', 'rating'] = None
```

[16]:
```
netflix['rating'].value_counts()
```

[16]:
```
rating
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
UR             3
NC-17          3
74 min         0
84 min         0
66 min         0
Name: count, dtype: int64
```

Extracting the numeric duration (in minutes) from the 'duration' column.

[17]:
```
netflix['duration_numeric'] = np.where(netflix['type'] ==␣
↪'Movie',netflix['duration'].str[:-4],netflix['duration'].str[:-7])
```

Exploding the 'Country', 'Cast', and 'Listed_in' attributes.

[18]:
```
netflix['cast'] = netflix['cast'].str.split(',')
netflix['listed_in'] = netflix['listed_in'].str.split(',')
netflix['country'] = netflix['country'].str.split(',')
```

[19]:
```
netflix = netflix.explode('cast')
netflix = netflix.explode('listed_in')
netflix = netflix.explode('country')
```

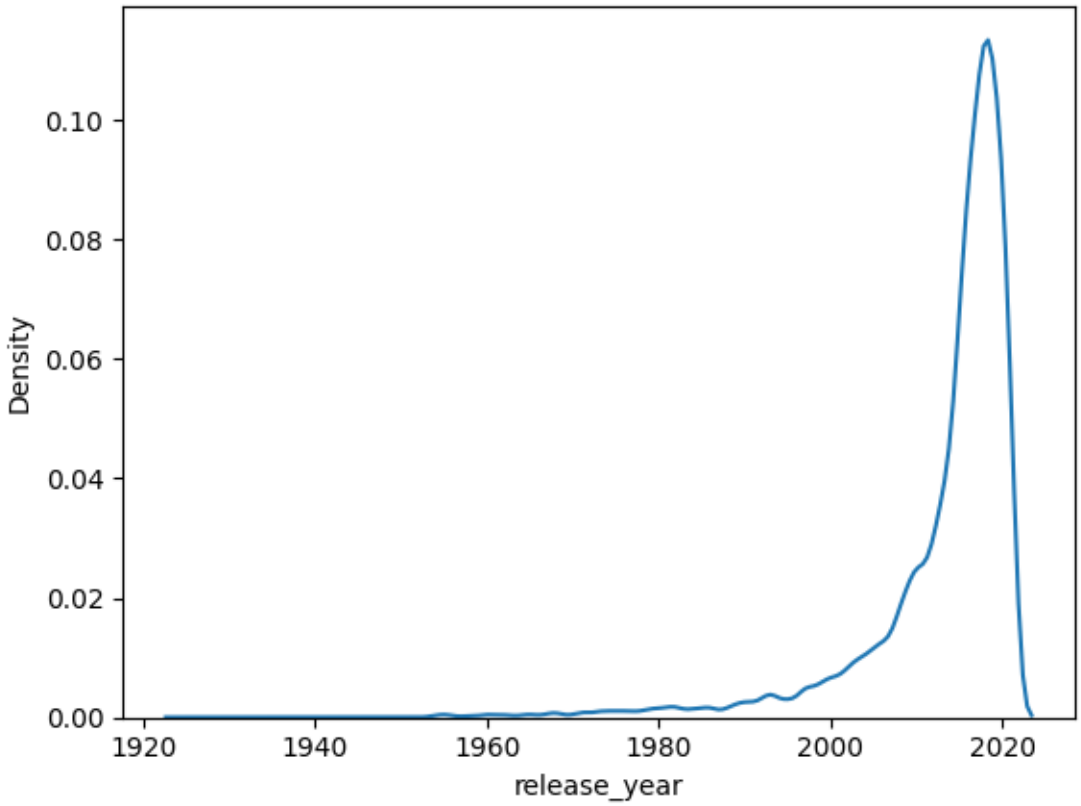
[20]:
```
netflix.shape
```

[20]:
```
(186399, 13)
```

Visual Analysis

[21]:
```
sns.kdeplot(netflix['release_year'])
```

[21]:
```
<Axes: xlabel='release_year', ylabel='Density'>
```



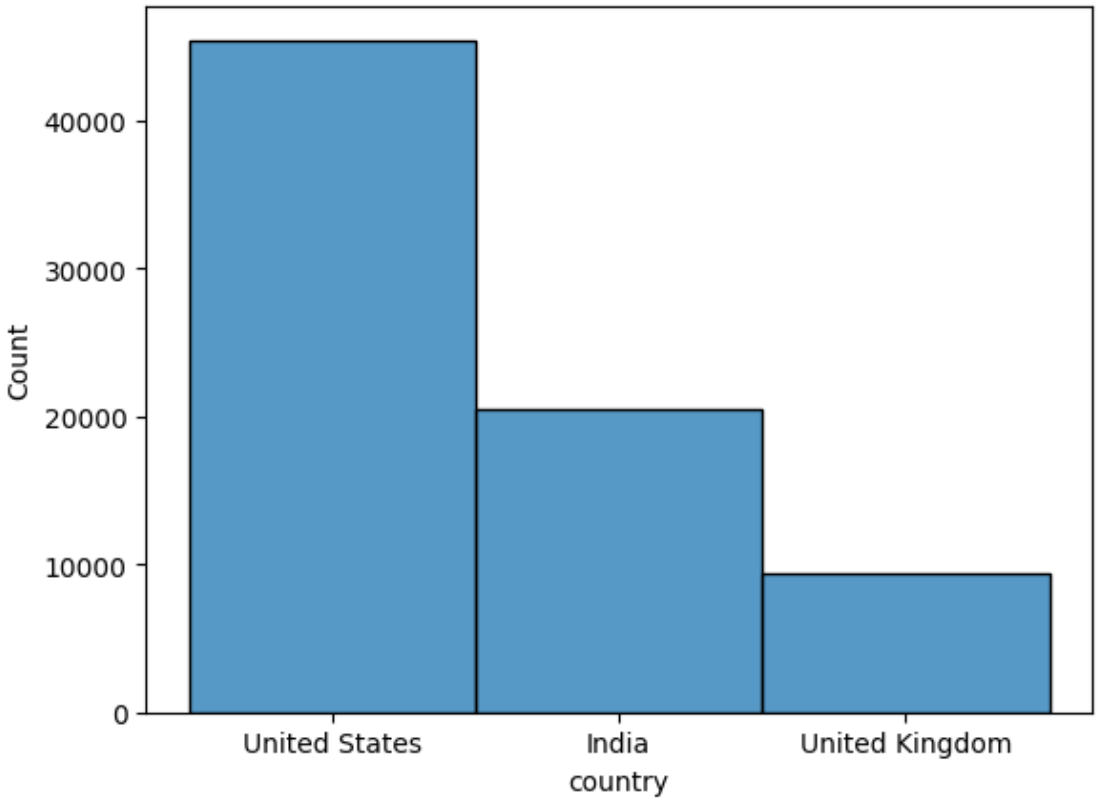
Finding movies released in top 3 countries

[22]:
```
top_3_country = netflix['country'].value_counts().head(3).index
top_3 = netflix.loc[netflix['country'].isin(top_3_country)]
top_3.shape
```

[22]:
```
(75329, 13)
```

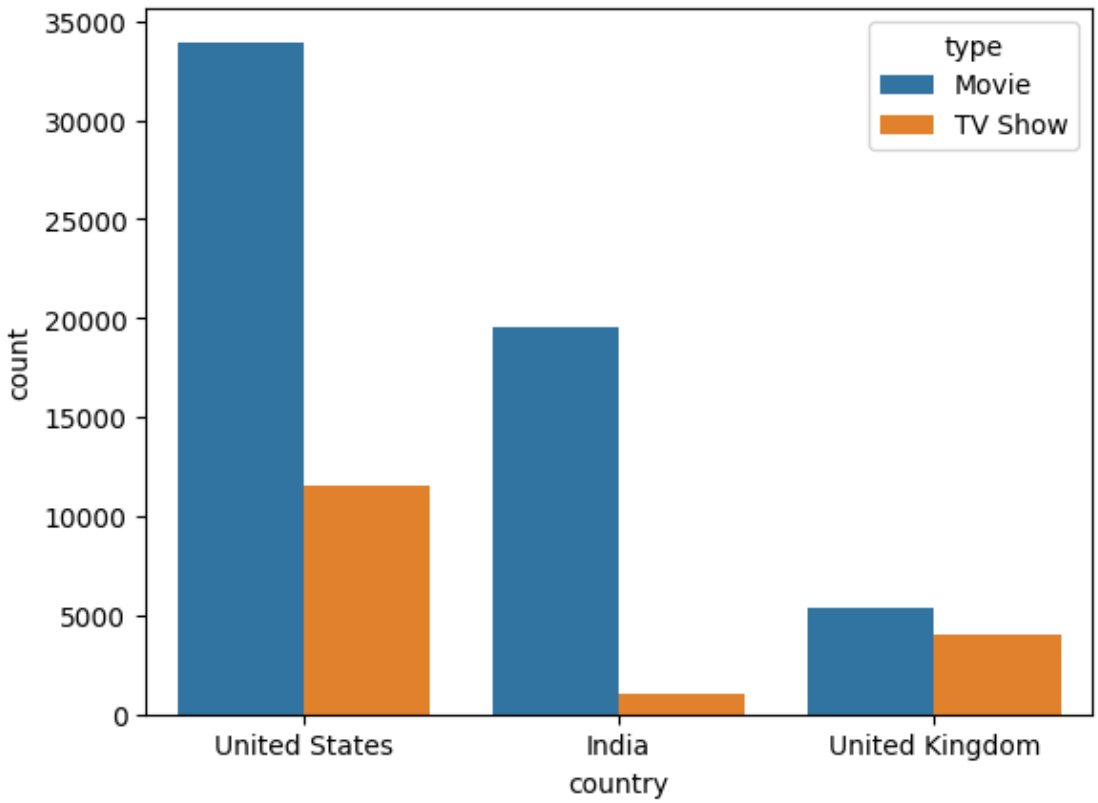
[23]:
```
sns.histplot(top_3['country'])
```

[23]: <Axes: xlabel='country', ylabel='Count'>



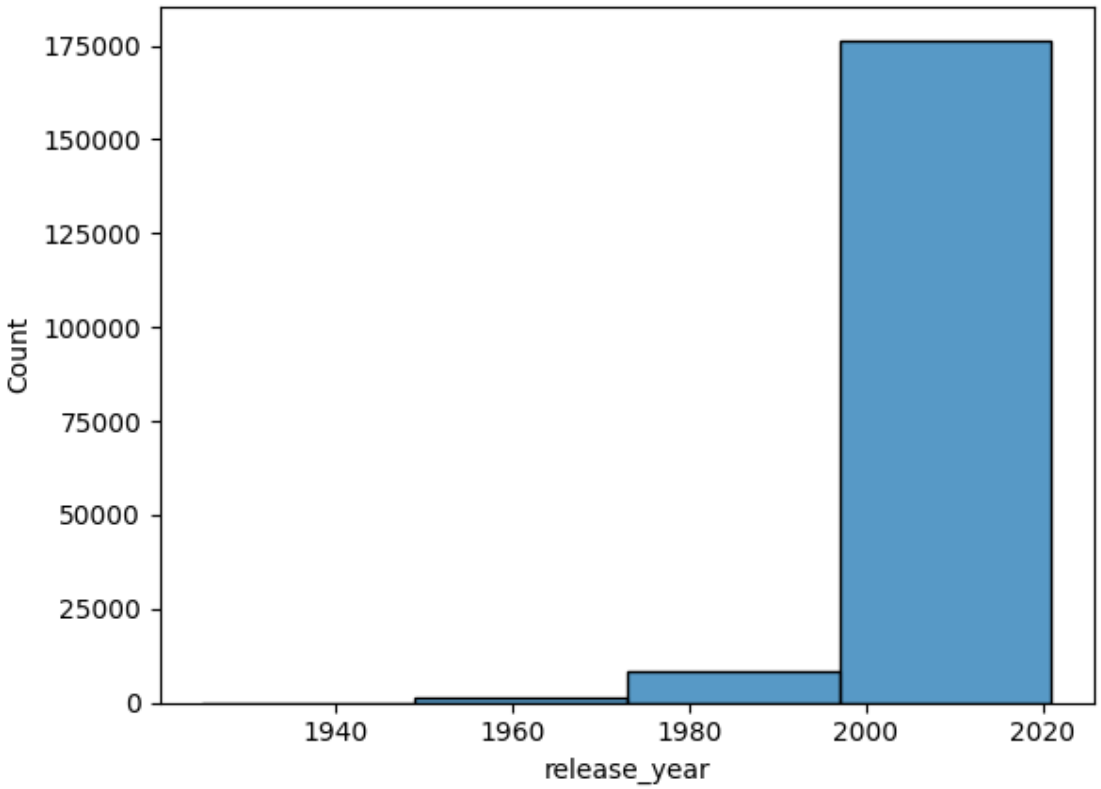
[24]:
```
sns.countplot(x = 'country', hue='type', data=top_3)
```

[24]: <Axes: xlabel='country', ylabel='count'>



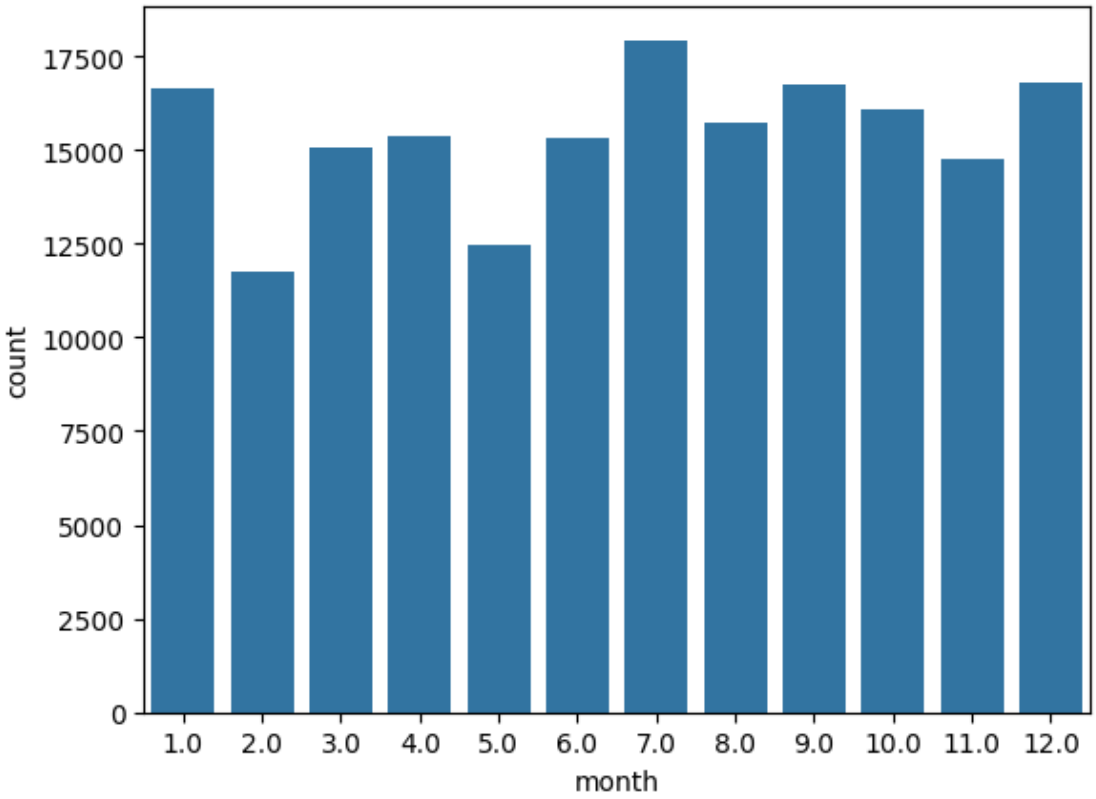
[36]:
```
sns.histplot(netflix['release_year'],bins = 4)
```

[36]: <Axes: xlabel='release_year', ylabel='Count'>



[37]:
```
netflix['month'] = pd.to_datetime(netflix['date_added'], errors='coerce').dt.
  ↪month
sns.barplot(x = netflix['month'].value_counts().index,y = netflix['month'].
  ↪value_counts())
```

[37]: <Axes: xlabel='month', ylabel='count'>



Checking For Outliers

[38]:
```
netflix['duration_numeric'] = netflix['duration_numeric'].astype('int')
q3 = netflix['duration_numeric'].sort_values().quantile(0.75)
q1 = netflix['duration_numeric'].sort_values().quantile(0.25)
iqr = q3 - q1
upper_bound = q3 + 1.5 * iqr
lower_bound = q1 - 1.5 * iqr
upper_bound,lower_bound
```

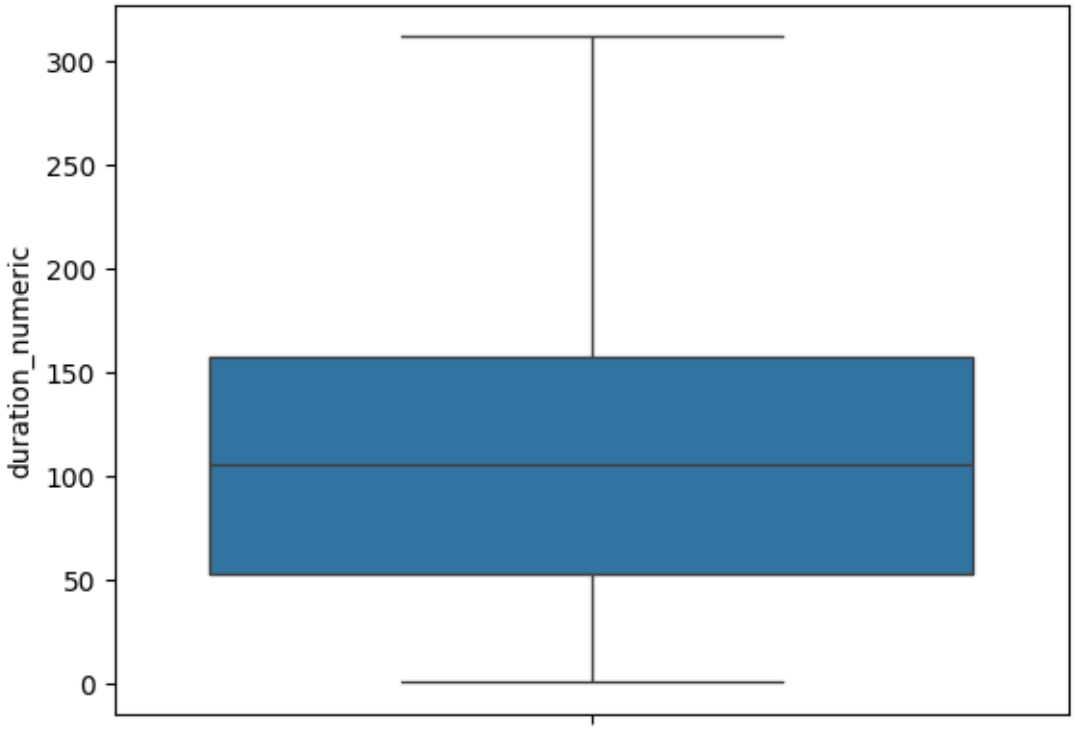
[38]:
```
(275.5, -160.5)
```

[39]:
```
netflix.loc[(netflix['duration_numeric'] > upper_bound) &␣
 ↪(netflix['duration_numeric'] < lower_bound)]
```

[39]:
```
Empty DataFrame
Columns: [show_id, type, title, director, cast, country, date_added,
release_year, rating, duration, listed_in, description, duration_numeric, month]
Index: []
```

[40]:
```
sns.boxplot(netflix['duration_numeric'].value_counts().index)
```

[40]: <Axes: ylabel='duration_numeric'>



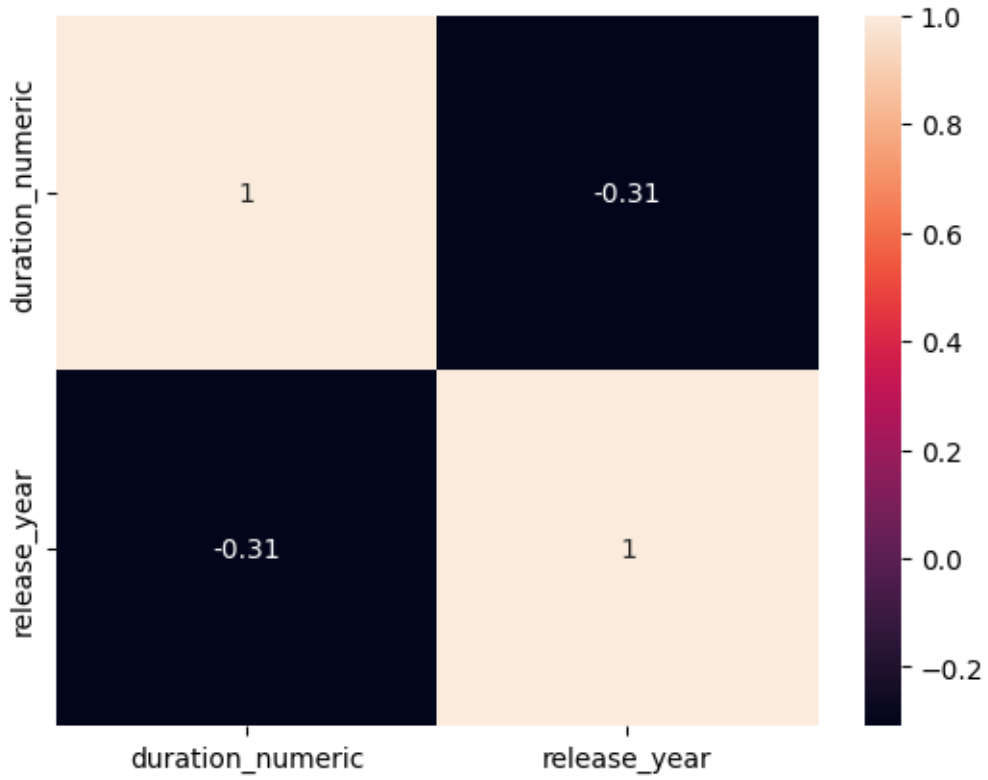
[41]:
```
netflix[['duration_numeric','release_year']].corr()
```

[41]:

	duration_numeric	release_year
duration_numeric	1.00000	-0.30851
release_year	-0.30851	1.00000

[42]:
```
sns.heatmap(netflix[['duration_numeric','release_year']].corr(),annot=True)
```

[42]: <Axes: >



Overview of the range of attributes

[43]:
```
netflix.describe()
```

[43]:

	release_year	duration_numeric	month
count	186399.000000	186399.000000	184662.000000
mean	2013.422792	76.677831	6.649868
std	9.048670	52.294745	3.449333
min	1925.000000	1.000000	1.000000
25%	2012.000000	3.000000	4.000000
50%	2016.000000	95.000000	7.000000
75%	2019.000000	112.000000	10.000000
max	2021.000000	312.000000	12.000000

Movies acted in by each actor in top 3 countries

[44]:
```
top_3['cast'].value_counts()
```

[44]:
```
cast
 Anupam Kher            105
Shah Rukh Khan           73
 Om Puri                 69
```

```
 Boman Irani             68
Akshay Kumar             67
                        …
 Jocelyn Osorio           1
 Eddie J. Fernandez       1
 David Fernandez Jr.      1
 Mauricio Mendoza         1
W. Kamau Bell             1
Name: count, Length: 20363, dtype: int64
```

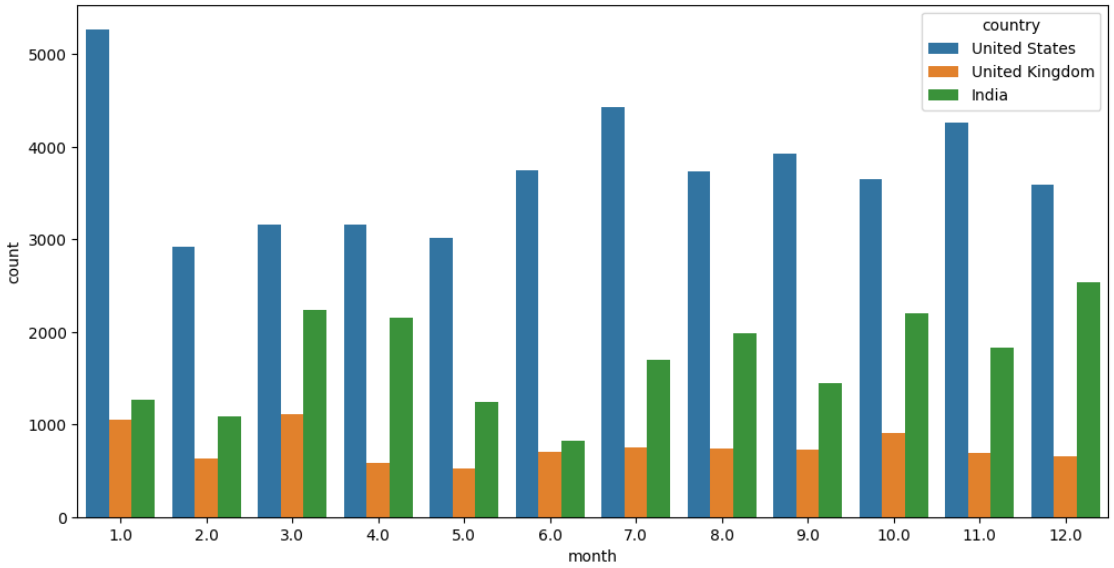
[46]:
```
top_3.head()
```

[46]:
```
  show_id     type                  title         director             cast  \
0      s1    Movie  Dick Johnson Is Dead  Kirsten Johnson              NaN
4      s5  TV Show          Kota Factory              NaN       Mayur More
4      s5  TV Show          Kota Factory              NaN       Mayur More
4      s5  TV Show          Kota Factory              NaN       Mayur More
4      s5  TV Show          Kota Factory              NaN   Jitendra Kumar

         country          date_added  release_year rating   duration  \
0  United States  September 25, 2021          2020  PG-13     90 min
4          India  September 24, 2021          2021  TV-MA  2 Seasons
4          India  September 24, 2021          2021  TV-MA  2 Seasons
4          India  September 24, 2021          2021  TV-MA  2 Seasons
4          India  September 24, 2021          2021  TV-MA  2 Seasons

                listed_in                                        description  \
0           Documentaries  As her father nears the end of his life, filmm…
4  International TV Shows  In a city of coaching centers known to train I…
4       Romantic TV Shows  In a city of coaching centers known to train I…
4             TV Comedies  In a city of coaching centers known to train I…
4  International TV Shows  In a city of coaching centers known to train I…

  duration_numeric
0               90
4                2
4                2
4                2
4                2
```

Movies released per month in the top 3 countries.

[57]:
```
top_3['date_added'] = pd.to_datetime(top_3['date_added'], errors='coerce')
top_3['month'] = top_3['date_added'].dt.month
plt.figure(figsize=(12,6))
sns.countplot(x = 'month',data = top_3,hue='country')
```

[57]: <Axes: xlabel='month', ylabel='count'>



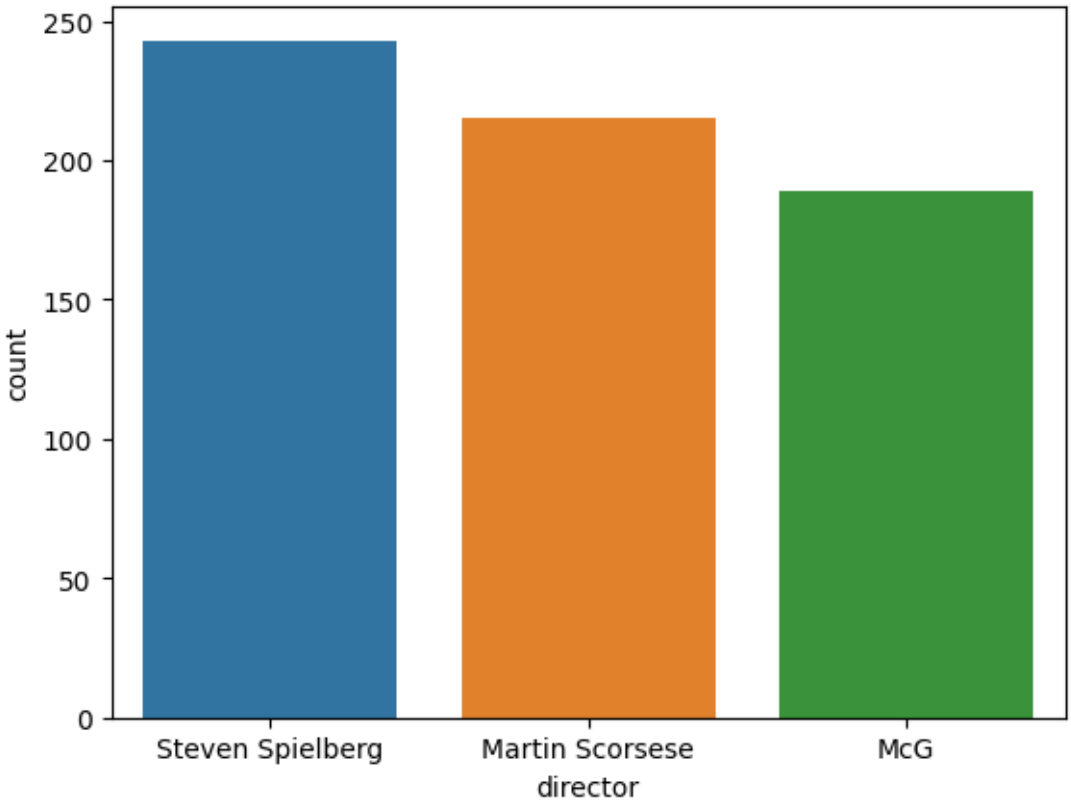
[87]:
```
us = top_3.loc[top_3['country'] == 'United States']['director'].value_counts().
  ↪head(3)
ind = top_3.loc[top_3['country'] == 'India']['director'].value_counts().head(3)
uk = top_3.loc[top_3['country'] == 'United Kingdom']['director'].value_counts().
  ↪head(3)
```

Top 3 directors based on movies from the top 3 countries.

1. United States

[91]:
```
sns.barplot(x=us.index, y=us , hue = us.index)
```

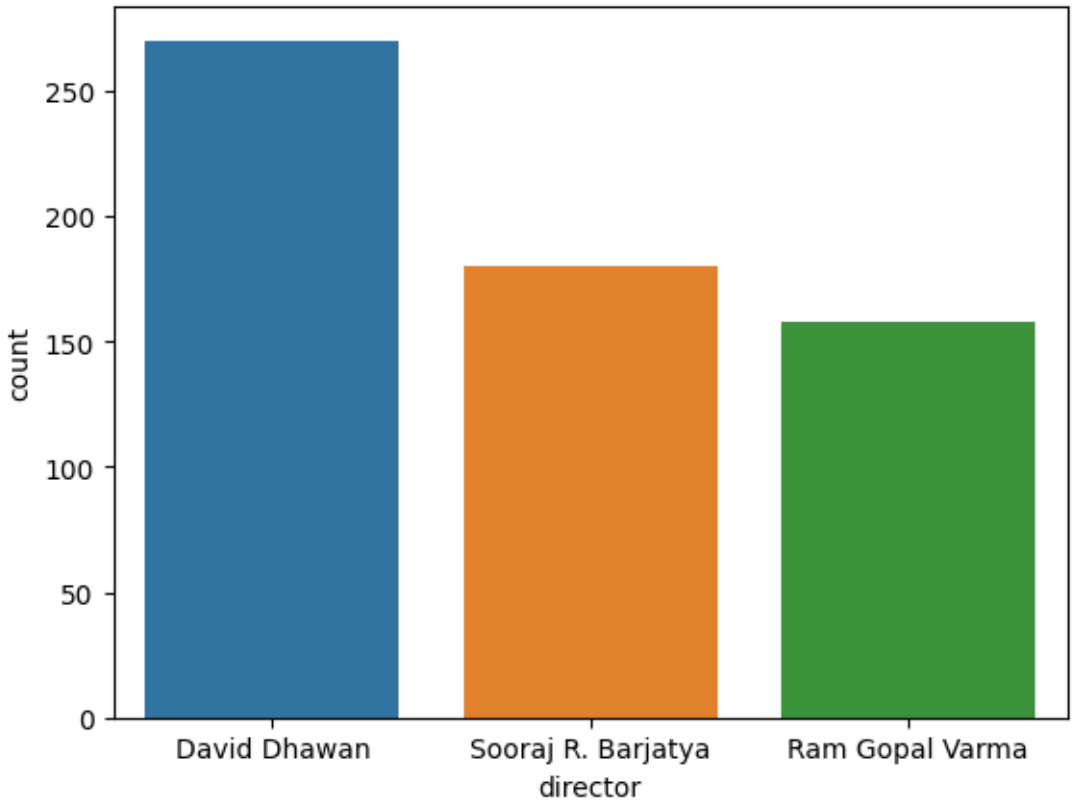
[91]: <Axes: xlabel='director', ylabel='count'>



2. India

[92]:
```
sns.barplot(x=ind.index, y=ind , hue = ind.index)
```

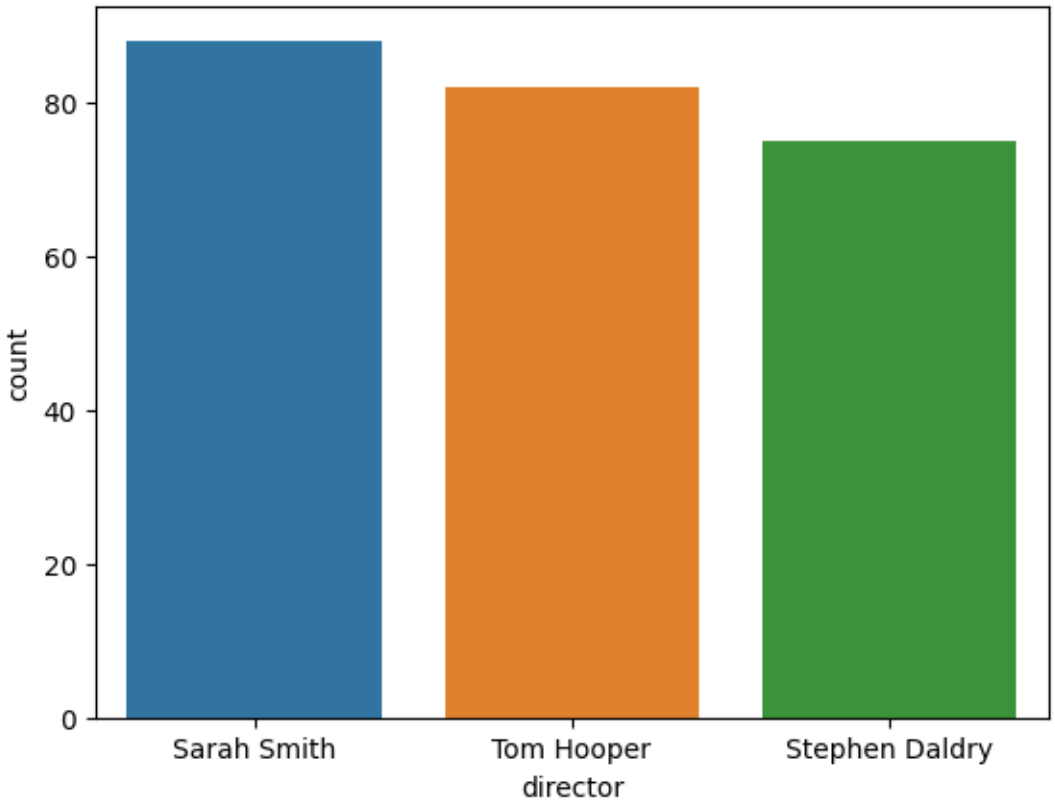
[92]:
```
<Axes: xlabel='director', ylabel='count'>
```



3. United Kindom

[93]:
```
sns.barplot(x=uk.index, y=uk , hue = uk.index)
```

[93]:
```
<Axes: xlabel='director', ylabel='count'>
```

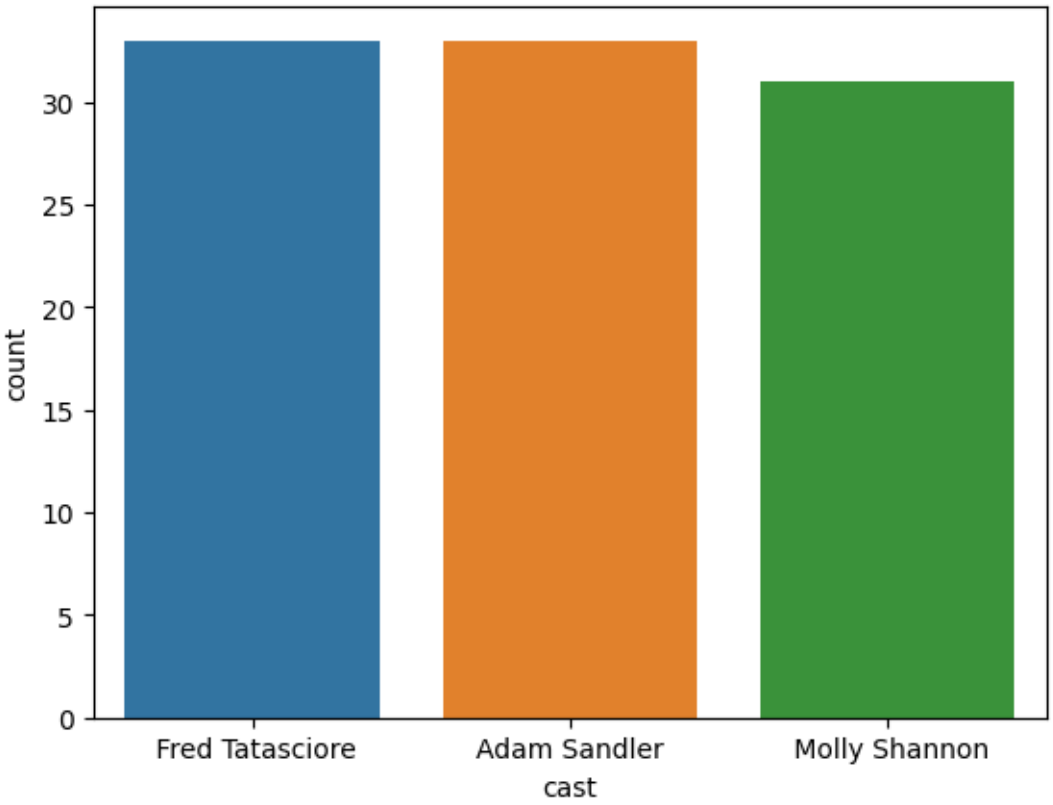


[94]:
```
us_cast = top_3.loc[top_3['country'] == 'United States']['cast'].value_counts().
 ↪head(3)
ind_cast = top_3.loc[top_3['country'] == 'India']['cast'].value_counts().head(3)
uk_cast = top_3.loc[top_3['country'] == 'United Kingdom']['cast'].
 ↪value_counts().head(3)
```

Top 3 actors based on movies from the top 3 countries.

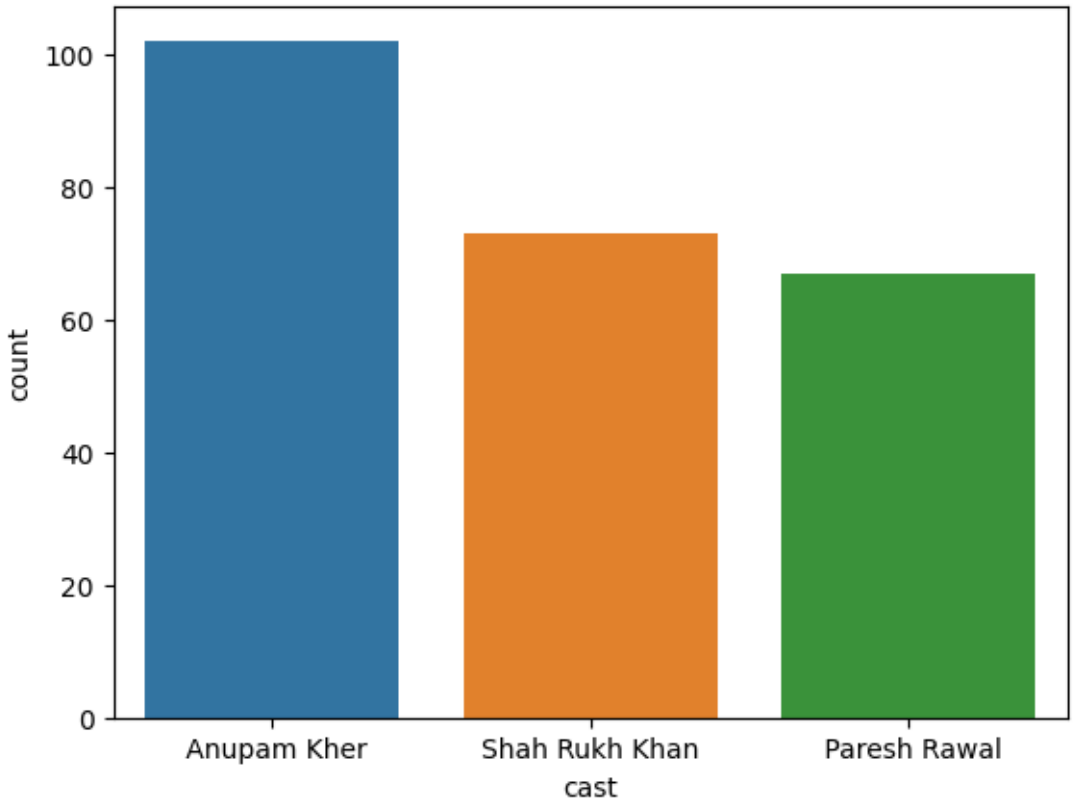
[95]:
```
sns.barplot(x=us_cast.index, y=us_cast , hue = us_cast.index)
```

[95]:
```
<Axes: xlabel='cast', ylabel='count'>
```



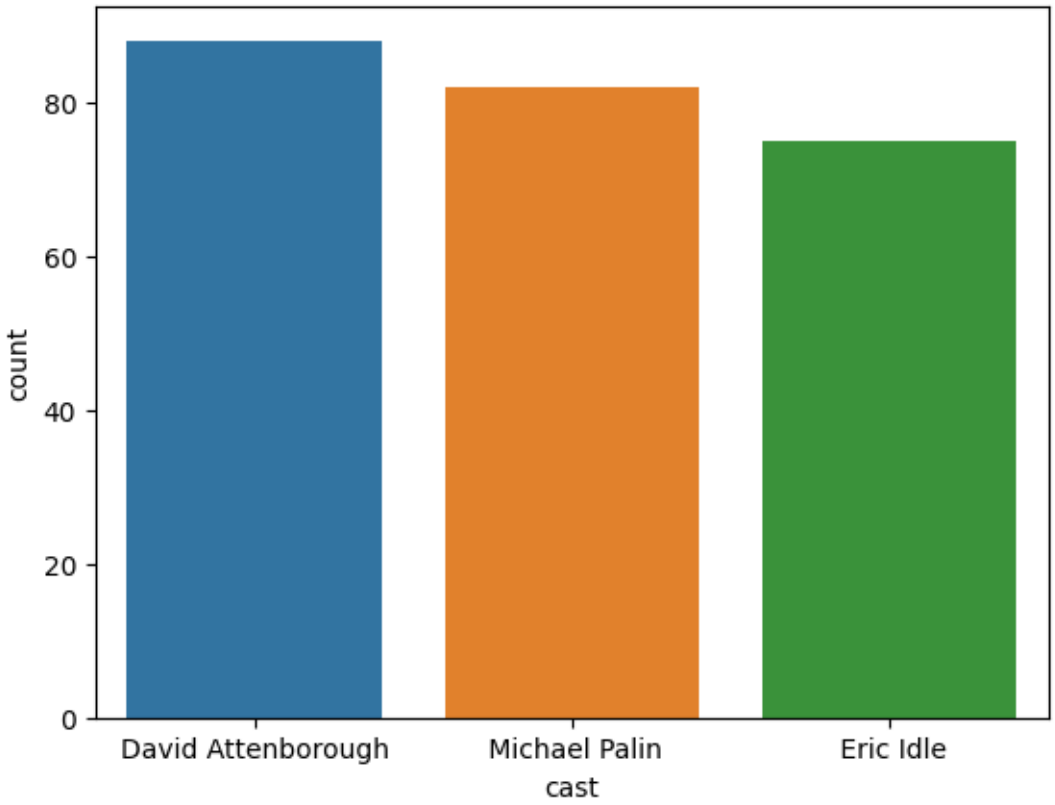
[98]:
```
sns.barplot(x=ind_cast.index, y=ind_cast , hue = ind_cast.index)
```

[98]:
```
<Axes: xlabel='cast', ylabel='count'>
```



[99]:
```
sns.barplot(x=uk_cast.index, y=uk , hue = uk_cast.index)
```

[99]:
```
<Axes: xlabel='cast', ylabel='count'>
```



[109]:
```
us_rating = top_3.loc[top_3['country'] == 'United States']['rating'].
 ↪value_counts().head(3)
ind_rating = top_3.loc[top_3['country'] == 'India']['rating'].value_counts().
 ↪head(3)
uk_rating = top_3.loc[top_3['country'] == 'United Kingdom']['rating'].
 ↪value_counts().head(3)
```

Top 3 rating based on movies from the top 3 countries.

[106]:
```
us_rating
```

[106]:
```
rating
TV-MA    11648
R         9933
PG-13     7250
Name: count, dtype: int64
```

[111]:
```
ind_rating
```

[111]:
```
rating
TV-14    11711
```

```
TV-MA     5209
TV-PG     2991
Name: count, dtype: int64
```

[112]:
```
uk_rating
```

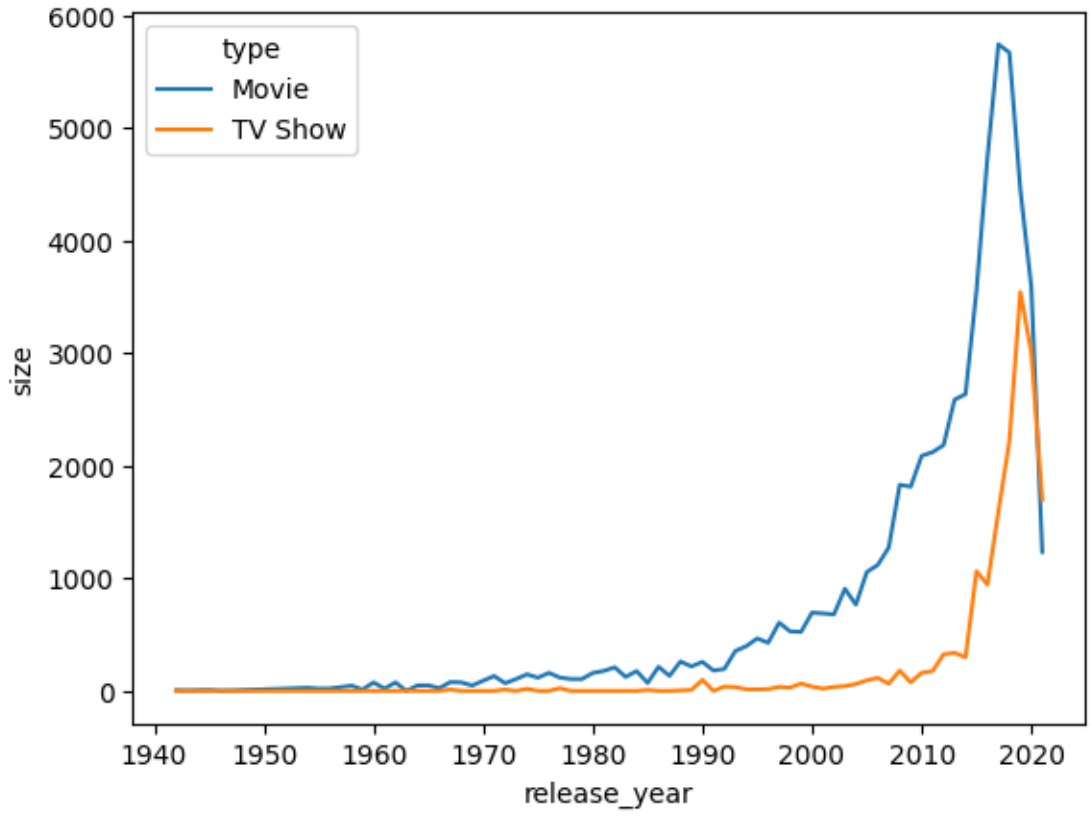
[112]:
```
rating
TV-MA    3727
R        1898
PG-13    1037
Name: count, dtype: int64
```

Trend Analysis

[118]:
```
trend_type = top_3.groupby(['release_year','type'],as_index = False).size()
sns.lineplot(x = 'release_year',y = 'size',hue = 'type',data = trend_type)
```

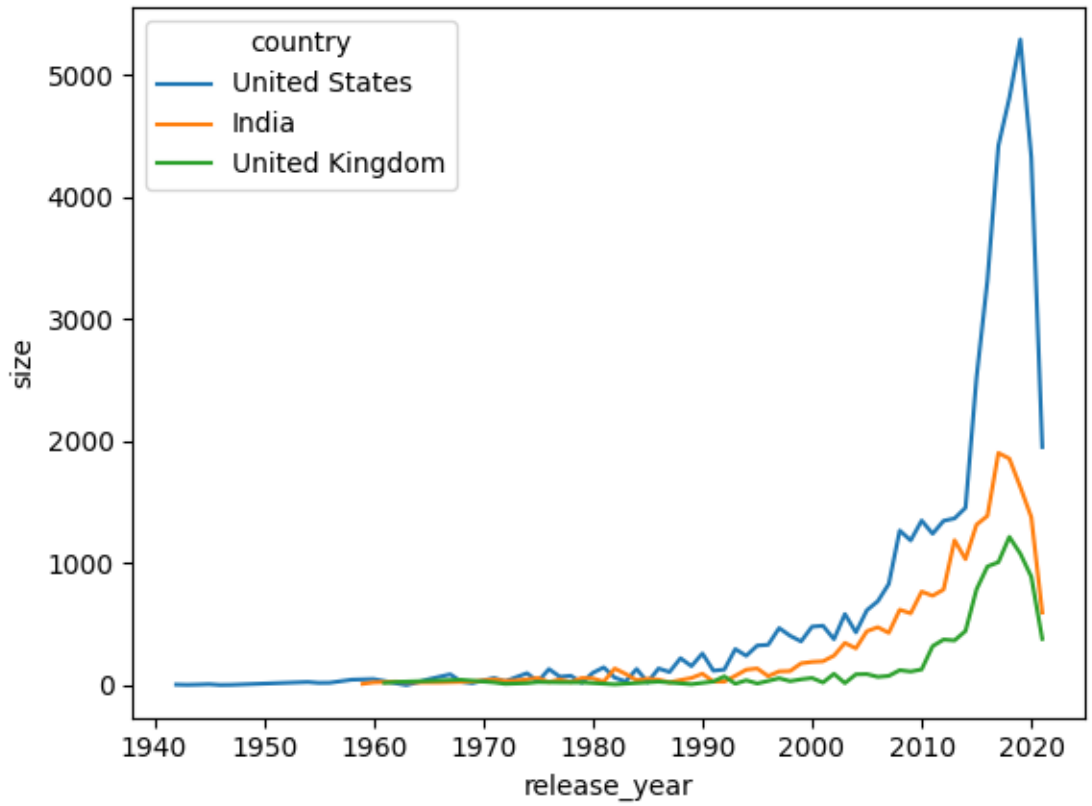
```
<ipython-input-118-1401a4d2a86a>:1: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass
observed=False to retain current behavior or observed=True to adopt the future
default and silence this warning.
  trend_type = top_3.groupby(['release_year','type'],as_index = False).size()
```

[118]:
```
<Axes: xlabel='release_year', ylabel='size'>
```



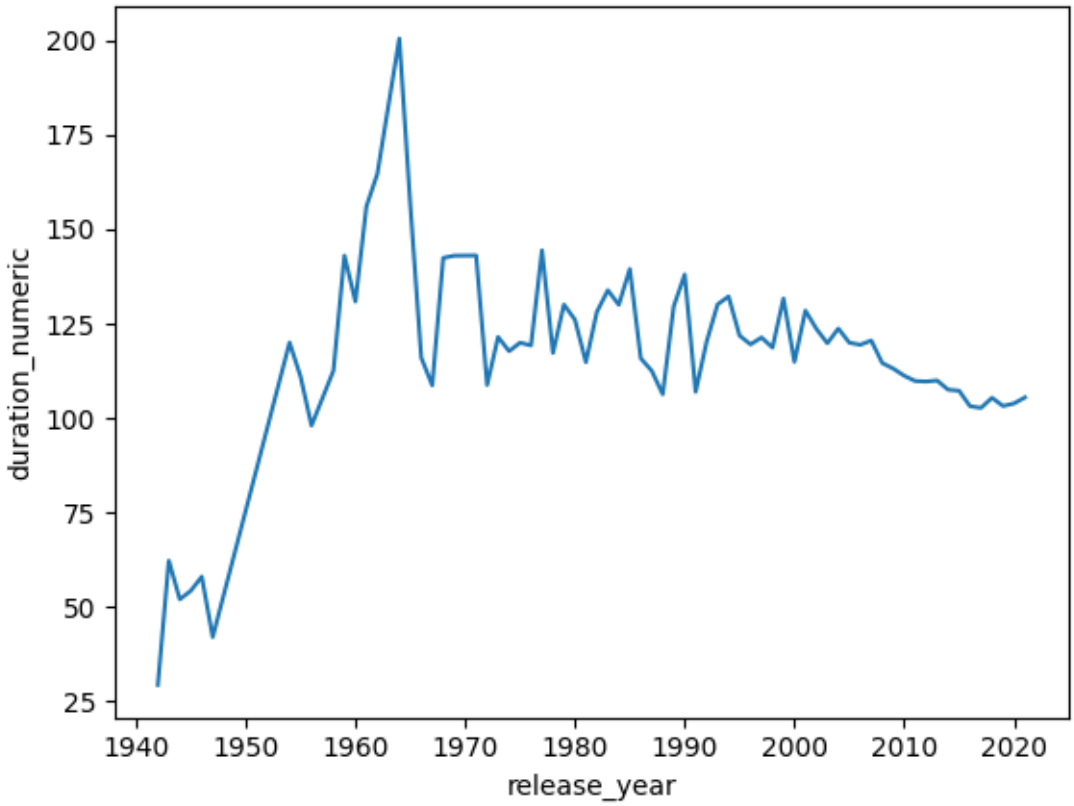
[122]:
```
trend_country = top_3.groupby(['release_year','country'],as_index = False).
 ↪size()
sns.lineplot(x = 'release_year',y = 'size',hue = 'country',data = trend_country)
```

[122]: <Axes: xlabel='release_year', ylabel='size'>



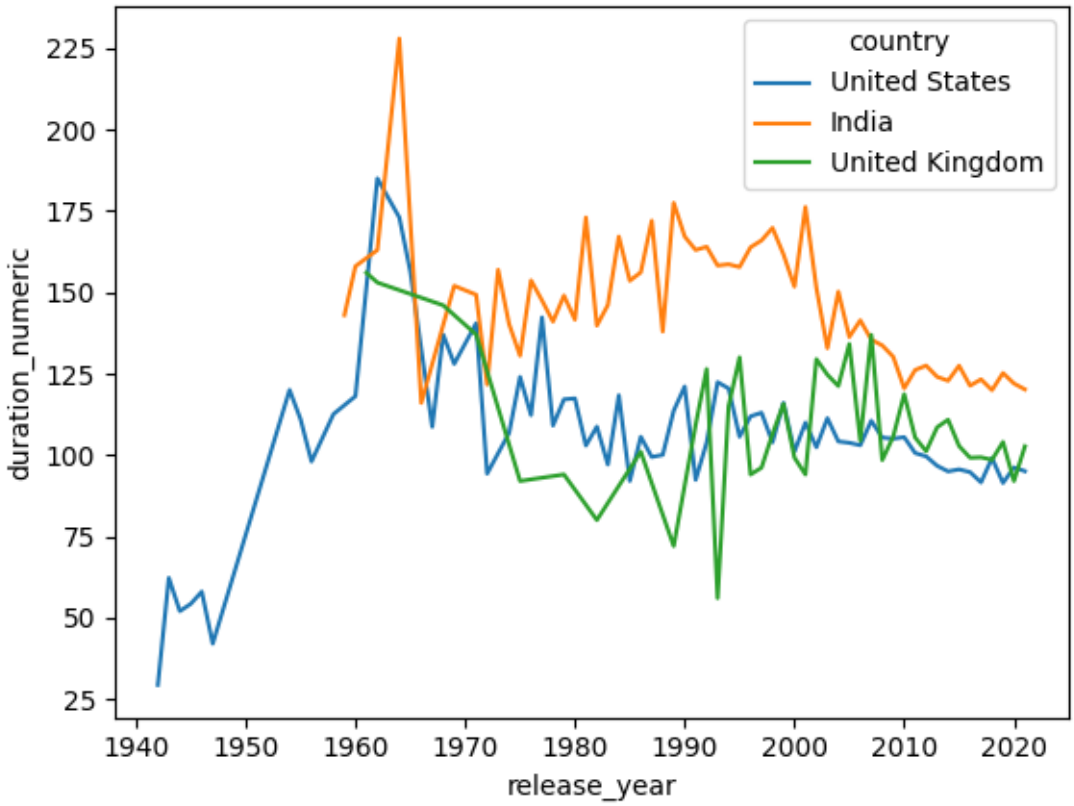
[135]:
```
top_3['duration_numeric']  = top_3['duration_numeric'].astype('int')
duration_trend = top_3[top_3['type'] == 'Movie'].
 ↪groupby('release_year')['duration_numeric'].mean()
sns.lineplot(x = duration_trend.index,y = duration_trend)
```

[135]: <Axes: xlabel='release_year', ylabel='duration_numeric'>



[136]:
```
duration_trend_country_wise = top_3[top_3['type'] == 'Movie'].
 ↪groupby(['release_year','country'],as_index = False)['duration_numeric'].
 ↪mean()
sns.lineplot(x = 'release_year',y = 'duration_numeric', data = ␣
 ↪duration_trend_country_wise, hue = 'country')
```

[136]:
```
<Axes: xlabel='release_year', ylabel='duration_numeric'>
```



Recommendations Based on Analysis

1. **Top 3 Countries by Movie Releases:**
   The top 3 countries where most movies are released are the United States, India, and the United Kingdom. We can leverage the popularity of shows from these regions to broadcast on Netflix, as they are more likely to be watched by audiences from these countries.

2. **Movies vs. TV Shows:**
   In all three countries, movies are released more frequently than TV shows, so we should focus on broadcasting movies rather than TV shows.

3. **Release Timing:**

   - In the **United States**, most movies are added to Netflix in the 1st month, so we can consider releasing more shows at this time.
   - In **India**, the 12th month sees the most movie releases.
   - In the **United Kingdom**, the 3rd month is the peak period for movie releases.

4. **Directors with Popular Movies:**

   - In the **United States**, movies directed by Steven Spielberg, Martin Scorsese, and McG are released more frequently.
   - In **India**, movies directed by David Dhawan, Sooraj R. Barjatya, and Ram Gopal Varma are released more often.

- In the **United Kingdom**, films directed by Sharon Smith, Tom Hooper, and Stephen Daldry have higher release rates.

Therefore, we should prioritize broadcasting movies directed by these filmmakers.

5. **Actors in Popular Movies:**

   - In the **United States**, movies featuring Fred Tatasciore, Adam Sandler, and Molly Shannon are more frequent.
   - In **India**, films featuring Anupam Kher, Shah Rukh Khan, and Paresh Rawal are released more often.
   - In the **United Kingdom**, movies featuring David Attenborough, Michael Palin, and Eric Idle have higher release frequencies.

   We should prioritize movies featuring these actors.

6. **Movie Ratings:**

   - In the **United States**, the most common movie ratings are **TV-MA**, **R**, and **PG-13**.
   - In **India**, the most popular ratings are **TV-14**, **TV-MA**, and **TV-PG**.
   - In the **United Kingdom**, the common ratings are **TV-MA**, **R**, and **PG-13**.

   We should focus on releasing movies with these ratings.

7. **Duration of Movies and Shows:**
   In recent years, the duration of movies and shows in all three countries has been decreasing due to the reduced attention span of audiences. To cater to this shift, we should focus on broadcasting shows with shorter durations.
