Netflix: which type of shows to produce and how to grow the business

In [1]:

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

In [2]:

```
netflix_data = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/
```

Understanding the Data

Checking for Nulls, Duplicates

Data Type and Non - Null Counts

Non-Null Counts, Data Type

- There are a lot of nulls in Director, cast and country columns but only few in date_added, rating and duration columns
- The data type of date_added is not datetime

In [3]:

```
1
2 netflix_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
```

Null value counts

In [4]:

```
1
2 netflix_data.isna().sum()
```

Out[4]:

```
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
```

Null value percentage per column

In [5]:

```
netflix_data.isna().apply(lambda x: round((x.sum()/x.size)*100, 2))
```

Out[5]:

```
show_id          0.00
type             0.00
title            0.00
director        29.91
cast             9.37
country          9.44
date_added       0.11
release_year     0.00
rating           0.05
duration         0.03
listed_in        0.00
description      0.00
dtype: float64
```

Duplicates

- There are no duplicated rows
- There are no duplicates in the Title of the show

In [6]:

```
netflix_data.loc[netflix_data.duplicated()]
```

Out[6]:

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	liste

In [7]:

```
netflix_data.loc[netflix_data[["title"]].duplicated(), "title"]
```

Out[7]:

```
Series([], Name: title, dtype: object)
```

Null Value Check

In [8]:

```
1
2 isna_full = netflix_data.isna()
3 isna_rows = isna_full.any(axis= 1)
4 null_rows = isna_rows[isna_rows > 0].index
5 isna_cols = isna_full.any(axis= 0)
6 null_cols = isna_cols[isna_cols > 0].index
7
8 null_rows, null_cols
```

Out[8]:

```
(Int64Index([   0,    1,    2,    3,    4,    5,    6,   10,   11,   13,
             ...
             8775, 8780, 8783, 8784, 8785, 8795, 8796, 8797, 8800, 8803],
            dtype='int64', length=3475),
 Index(['director', 'cast', 'country', 'date_added', 'rating', 'duration'],
dtype='object'))
```

The nuils in rating, duration

- The Nulls in duration are due to the values being recorded in the rating column

In [9]:

```
1
2 netflix_data.loc[isna_full[["rating", "duration"]].any(axis= 1), :]
```

Out[9]:

	show_id	type	title	director	cast	country	date_added	release_year
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015
5989	s5990	Movie	13TH: A Conversation with Oprah Winfrey & Ava ...	NaN	Oprah Winfrey, Ava DuVernay	NaN	January 26, 2017	2017
6827	s6828	TV Show	Gargantia on the Verdurous Planet	NaN	Kaito Ishikawa, Hisako Kanemoto, Ai Kayano, Ka...	Japan	December 1, 2016	2013
7312	s7313	TV Show	Little Lunch	NaN	Flynn Curry, Olivia Deeble, Madison Lu, Oisín ...	Australia	February 1, 2018	2015
7537	s7538	Movie	My Honor Was Loyalty	Alessandro Pepe	Leone Frisa, Paolo Vaccarino, Francesco Miglio...	Italy	March 1, 2017	2015

In [10]:

```
1
2 duration_nulls = netflix_data.loc[isna_full[["duration"]].any(axis= 1), :]
3
4 duration_nulls
```

Out[10]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	74 min	
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	84 min	
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	66 min	

The nulls in date_added

- There seems to be no appearent reason behing the Nulls in date_added

In [11]:

```
1
2 netflix_data.loc[isna_full[["date_added"]].any(axis= 1), :].head()
```

Out[11]:

	show_id	type	title	director	cast	country	date_added	release_year	ratir
6066	s6067	TV Show	A Young Doctor's Notebook and Other Stories	NaN	Daniel Radcliffe, Jon Hamm, Adam Godley, Chris...	United Kingdom	NaN	2013	T M
6174	s6175	TV Show	Anthony Bourdain: Parts Unknown	NaN	Anthony Bourdain	United States	NaN	2018	T P
6795	s6796	TV Show	Frasier	NaN	Kelsey Grammer, Jane Leeves, David Hyde Pierce...	United States	NaN	2003	T P
6806	s6807	TV Show	Friends	NaN	Jennifer Aniston, Courteney Cox, Lisa Kudrow, ...	United States	NaN	2003	TV-1
6901	s6902	TV Show	Gunslinger Girl	NaN	Yuuka Nanri, Kanako Mitsuhashi, Eri Sendai, Am...	Japan	NaN	2008	TV-1

The nulls in director, cast and country

- There seems to be no immediately apparent reason behind the Nulls in director, cast, country

In [12]:

```
1
2 netflix_data.loc[isna_full[["director", "cast", "country"]].any(axis= 1), :].head()
```

Out[12]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA

In [13]:

```
1
2 netflix_data.loc[isna_full[["director", "cast", "country"]].all(axis= 1), :].head()
```

Out[13]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	Sea
10	s11	TV Show	Vendetta: Truth, Lies and The Mafia	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	Sea
14	s15	TV Show	Crime Stories: India Detectives	NaN	NaN	NaN	September 22, 2021	2021	TV-MA	Sea
74	s75	TV Show	The World's Most Amazing Vacation Rentals	NaN	NaN	NaN	September 14, 2021	2021	TV-PG	Seas
123	s124	TV Show	Luv Kushh	NaN	NaN	NaN	September 2, 2021	2012	TV-Y7	Sea

Null Value and Datatype correction

Null value correction in duration

- The null values in duration are replaced with their corresponding values from rating
- Now there are more nulls in rating

In [14]:

```
netflix_data.loc[isna_full[["duration"]].any(axis= 1), "duration"] = duration_nulls["ra

netflix_data.loc[isna_full[["duration"]].any(axis= 1), "rating"] = duration_nulls["dura

netflix_data.loc[isna_full[["duration"]].any(axis= 1), :]
```

Out[14]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	NaN	74
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	NaN	84
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	NaN	66

Recheck of Null value counts

In [15]:

```
netflix_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        8800 non-null   object
 9   duration      8807 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

In [16]:

```
1
2 netflix_data.isna().sum()
```

Out[16]:

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             7
duration           0
listed_in          0
description        0
dtype: int64
```

In [17]:

```
1
2 netflix_data.isna().apply(lambda x: round((x.sum()/x.size)*100, 2))
```

Out[17]:

```
show_id          0.00
type             0.00
title            0.00
director        29.91
cast             9.37
country          9.44
date_added       0.11
release_year     0.00
rating           0.08
duration         0.00
listed_in        0.00
description      0.00
dtype: float64
```

Datatype correction in date_added

- The values in date_added are changed to datetime datatype

In [18]:

```
1
2 netflix_data["date_added"] = pd.to_datetime(netflix_data["date_added"], infer_datetime_
3
4 netflix_data.head()
```

Out[18]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA

In [19]:

```
netflix_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   datetime64[ns]
 7   release_year  8807 non-null   int64         
 8   rating        8800 non-null   object        
 9   duration      8807 non-null   object        
 10  listed_in     8807 non-null   object        
 11  description   8807 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(10)
memory usage: 825.8+ KB
```

Null value correction in date_added

In [20]:

```
isna_full = netflix_data.isna()
isna_rows = isna_full.any(axis= 1)
null_rows = isna_rows[isna_rows > 0].index
isna_cols = isna_full.any(axis= 0)
null_cols = isna_cols[isna_cols > 0].index

null_rows, null_cols
```

Out[20]:

```
(Int64Index([   0,    1,    2,    3,    4,    5,    6,   10,   11,   13,
             ...
             8775, 8780, 8783, 8784, 8785, 8795, 8796, 8797, 8800, 8803],
            dtype='int64', length=3475),
 Index(['director', 'cast', 'country', 'date_added', 'rating'], dtype='objec
t'))
```

In [21]:

```
1
2 netflix_data.loc[isna_full["date_added"], :].head()
```

Out[21]:

	show_id	type	title	director	cast	country	date_added	release_year	ratir
6066	s6067	TV Show	A Young Doctor's Notebook and Other Stories	NaN	Daniel Radcliffe, Jon Hamm, Adam Godley, Chris...	United Kingdom	NaT	2013	T M
6174	s6175	TV Show	Anthony Bourdain: Parts Unknown	NaN	Anthony Bourdain	United States	NaT	2018	T P
6795	s6796	TV Show	Frasier	NaN	Kelsey Grammer, Jane Leeves, David Hyde Pierce...	United States	NaT	2003	T P
6806	s6807	TV Show	Friends	NaN	Jennifer Aniston, Courteney Cox, Lisa Kudrow, ...	United States	NaT	2003	TV-1
6901	s6902	TV Show	Gunslinger Girl	NaN	Yuuka Nanri, Kanako Mitsuhashi, Eri Sendai, Am...	Japan	NaT	2008	TV-1

distribution of day, month added

- Most of the movies and TV Shows are released on start of the month
- Mosat of the movies or shows are released on the start or end of the year

In [22]:

```
netflix_data.dropna().date_added.dt.day.value_counts().head(5)
```

Out[22]:

```
1     1519
15     408
2      208
16     188
31     165
Name: date_added, dtype: int64
```

In [23]:

```
netflix_data.dropna().date_added.dt.month.value_counts().head(5)
```

Out[23]:

```
10    491
12    490
1     489
4     471
3     469
Name: date_added, dtype: int64
```

Strategy for imputation

- The nulls in date added are imputed to be on the strart date of the release_year

Note: As we move forward we will find that most movies are added tonetflix on the year they're released

In [24]:

```
import datetime as dt
```

In [25]:

```
pd.to_datetime([dt.date(2022, 1, 1)], infer_datetime_format=True)
```

Out[25]:

```
DatetimeIndex(['2022-01-01'], dtype='datetime64[ns]', freq=None)
```

In [26]:

```
1
2 temp = netflix_data.loc[isna_full["date_added"], "release_year"].apply(lambda x: dt.dat
3
4 temp = pd.to_datetime(temp, infer_datetime_format= True)
5
6
7 netflix_data.loc[isna_full["date_added"], "date_added"] = temp
8
9 netflix_data.loc[isna_full["date_added"], "date_added"]
```

Out[26]:

```
6066   2013-01-01
6174   2018-01-01
6795   2003-01-01
6806   2003-01-01
6901   2008-01-01
7196   2010-01-01
7254   2012-01-01
7406   2016-01-01
7847   2015-01-01
8182   2015-01-01
Name: date_added, dtype: datetime64[ns]
```

In [27]:

```
1
2 netflix_data["date_added"] = pd.to_datetime(netflix_data["date_added"], infer_datetime_
3
4 netflix_data.head()
```

Out[27]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	

Dropping description column

- Since the analysis of show description is not currently interesting, it has been dropped

In [28]:

```
1
2 netflix_data.drop(["description"], axis= 1, inplace= True)
```

In [29]:

```
1
2 netflix_data.columns
```

Out[29]:

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

null values correction in rating

- based on the below image, the rating for the null values can be put as
  - NR for movies
  - TV-MA for TV Shows

Local Rating Values	Kids *(All)*	Older Kids *(7+)*	Teens *(13+)*	Young Adults *(16+)*	Adults *(18+)*
MPAA (Movies)	G	PG	PG-13		NC-17
					NR
					Unrated
					R
TVPG (TV)	TV-G	TV-Y7		TV-14	TV-MA
		TV-Y7-FV			
	TV-Y	TV-PG			

In [30]:

```
1
2 netflix_data.loc[(isna_full.rating) & (netflix_data.type == "Movie"), "rating"] = "NR"
3 netflix_data.loc[(isna_full.rating) & (netflix_data.type == "TV Show"), "rating"] = "TV
```

In [31]:

```
1
2 netflix_data.loc[isna_full.rating]
```

Out[31]:

	show_id	type	title	director	cast	country	date_added	release_year
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04	2017
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	2016-09-16	2010
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	2016-08-15	2015
5989	s5990	Movie	13TH: A Conversation with Oprah Winfrey & Ava ...	NaN	Oprah Winfrey, Ava DuVernay	NaN	2017-01-26	2017
6827	s6828	TV Show	Gargantia on the Verdurous Planet	NaN	Kaito Ishikawa, Hisako Kanemoto, Ai Kayano, Ka...	Japan	2016-12-01	2013
7312	s7313	TV Show	Little Lunch	NaN	Flynn Curry, Olivia Deeble, Madison Lu, Oisín ...	Australia	2018-02-01	2015
7537	s7538	Movie	My Honor Was Loyalty	Alessandro Pepe	Leone Frisa, Paolo Vaccarino, Francesco Miglio...	Italy	2017-03-01	2015

Null Value Re-check

In [32]:

```
1
2 netflix_data.head()
```

Out[32]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA

In [33]:

```
1
2 netflix_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 11 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    8807 non-null   datetime64[ns]
 7   release_year  8807 non-null   int64
 8   rating        8807 non-null   object
 9   duration      8807 non-null   object
 10  listed_in     8807 non-null   object
dtypes: datetime64[ns](1), int64(1), object(9)
memory usage: 757.0+ KB
```

In [34]:

```
1
2 netflix_data.isna().sum()
```

Out[34]:

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added         0
release_year       0
rating             0
duration           0
listed_in          0
dtype: int64
```

In [35]:

```
1
2 netflix_data.isna().apply(lambda x: round((x.sum()/x.size)*100, 2))
```

Out[35]:

```
show_id          0.00
type             0.00
title            0.00
director        29.91
cast             9.37
country          9.44
date_added       0.00
release_year     0.00
rating           0.00
duration         0.00
listed_in        0.00
dtype: float64
```

In [36]:

```
1
2 isna_full = netflix_data.isna()
3 isna_rows = isna_full.any(axis= 1)
4 null_rows = isna_rows[isna_rows > 0].index
5 isna_cols = isna_full.any(axis= 0)
6 null_cols = isna_cols[isna_cols > 0].index
7
8 null_rows, null_cols
```

Out[36]:

```
(Int64Index([   0,    1,    2,    3,    4,    5,    6,   10,   11,   13,
             ...
             8775, 8780, 8783, 8784, 8785, 8795, 8796, 8797, 8800, 8803],
            dtype='int64', length=3471),
 Index(['director', 'cast', 'country'], dtype='object'))
```

In [37]:

```
1
2 netflix_data.loc[isna_rows, isna_cols]
```

Out[37]:

	director	cast	country
0	Kirsten Johnson	NaN	United States
1	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa
2	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN
3	NaN	NaN	NaN
4	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India
...	...	...	...
8795	NaN	Mike Liscio, Emily Bauer, Billy Bob Thompson, ...	Japan, Canada
8796	NaN	Gökhan Atalay, Payidar Tüfekçioglu, Baran Akbu...	Turkey
8797	NaN	Michael Johnston, Jessica Gee-George, Christin...	United States, France, South Korea, Indonesia
8800	NaN	Sanam Saeed, Fawad Khan, Ayesha Omer, Mehreen ...	Pakistan
8803	NaN	NaN	NaN

3471 rows × 3 columns

Distribution of null vs non-null columns

- ignoring the null values in rating

In [38]:

```
1
2 netflix_data.loc[isna_rows].show_id.nunique()
```

Out[38]:

```
3471
```

nulls vs show type

- Most of the nulls are for TV Shows

In [39]:

```
1
2 netflix_data.loc[isna_rows].type.value_counts()
```

Out[39]:

```
TV Show    2529
Movie       942
Name: type, dtype: int64
```

In [40]:

```
1
2 netflix_data.loc[isna_rows].type.value_counts() / 3475 * 100
```

Out[40]:

```
TV Show    72.776978
Movie      27.107914
Name: type, dtype: float64
```

nulls vs listed genre

- Most of the nulls are for International TV Shows, Documentaries, Kid's TV

In [41]:

```
1
2 netflix_data.loc[isna_rows].listed_in.value_counts().head(10)
```

Out[41]:

```
Kids' TV                                                 216
Documentaries                                            203
Documentaries, International Movies                      122
International TV Shows, TV Dramas                        115
Reality TV                                                94
Kids' TV, TV Comedies                                     94
Crime TV Shows, International TV Shows, TV Dramas         93
International TV Shows, Romantic TV Shows, TV Dramas      89
International TV Shows, Romantic TV Shows, TV Comedies    88
Anime Series, International TV Shows                      83
Name: listed_in, dtype: int64
```

In [42]:

```
1
2 netflix_data.loc[isna_rows].listed_in.str.split(", ").explode().value_counts().head(10)
```

Out[42]:

```
International TV Shows    1264
TV Dramas                  711
TV Comedies                551
Documentaries              478
Kids' TV                   438
Crime TV Shows             432
International Movies       383
Docuseries                 381
Romantic TV Shows          349
Reality TV                 252
Name: listed_in, dtype: int64
```

nulls vs rating

- Most of the nulls are for TV-MA (50 %)

In [43]:

```
1
2 netflix_data.loc[isna_rows].rating.value_counts().head(10)
```

Out[43]:

```
TV-MA    1387
TV-14     946
TV-PG     432
TV-Y7     258
TV-Y      231
TV-G      136
NR         23
R          21
PG-13      20
PG         12
Name: rating, dtype: int64
```

In [44]:

```
netflix_data.loc[isna_rows].rating.value_counts() / 3475 * 100
```

Out[44]:

```
TV-MA       39.913669
TV-14       27.223022
TV-PG       12.431655
TV-Y7        7.424460
TV-Y         6.647482
TV-G         3.913669
NR           0.661871
R            0.604317
PG-13        0.575540
PG           0.345324
TV-Y7-FV     0.086331
G            0.028777
NC-17        0.028777
Name: rating, dtype: float64
```

nulls vs release_year

- Most of the nulls are for the recent years (2018- 2021)

In [45]:

```
netflix_data.loc[isna_rows].release_year.value_counts().head(10)
```

Out[45]:

```
2019    511
2020    511
2018    499
2021    431
2017    374
2016    325
2015    209
2014    110
2013     91
2012     74
Name: release_year, dtype: int64
```

director nulls vs genre

- Most of the nulls are for International TV, Dramas, Comedies, Kid's TV

In [46]:

```
isna_full.director.sum()
```

Out[46]:

```
2634
```

In [47]:

```
netflix_data.loc[isna_full.director].listed_in.str.split(", ").explode().value_counts()
```

Out[47]:

```
International TV Shows    1223
TV Dramas                  702
TV Comedies                539
Kids' TV                   433
Crime TV Shows             401
Romantic TV Shows          341
Docuseries                 335
Reality TV                 249
British TV Shows           228
Anime Series               165
Name: listed_in, dtype: int64
```

director nulls vs rating

- Most of the nulls are for TV-MA

In [48]:

```
netflix_data.loc[isna_full.director].rating.value_counts().head(10)
```

Out[48]:

```
TV-MA       1092
TV-14        703
TV-PG        325
TV-Y7        202
TV-Y         195
TV-G         102
NR             6
R              4
TV-Y7-FV       2
PG-13          1
Name: rating, dtype: int64
```

director nulls vs release year

- Most of the nulls are for recent years (2017 - 2021)

In [49]:

```
netflix_data.loc[isna_full.director].release_year.value_counts().head(10)
```

Out[49]:

```
2020    405
2019    401
2018    387
2021    295
2017    259
2016    249
2015    160
2014     85
2013     63
2012     62
Name: release_year, dtype: int64
```

cast nulls vs genre

- Most of the nulls are for Documentaries, International, Reality, Crime, Kid's TV, Science & nature

In [50]:

```
isna_full.cast.sum()
```

Out[50]:

```
825
```

In [51]:

```
netflix_data.loc[isna_full.cast].listed_in.value_counts().head(10)
```

Out[51]:

```
Documentaries                                    183
Documentaries, International Movies              117
Docuseries                                        47
Crime TV Shows, Docuseries                        36
Reality TV                                        31
Documentaries, Sports Movies                      30
Crime TV Shows, Docuseries, International TV Shows    24
Kids' TV                                          23
Documentaries, International Movies, Sports Movies    21
Documentaries, Music & Musicals                   20
Name: listed_in, dtype: int64
```

In [52]:

```
netflix_data.loc[isna_full.cast].listed_in.str.split(", ").explode().value_counts().hea
```

Out[52]:

```
Documentaries             424
Docuseries                207
International Movies      178
International TV Shows    109
Reality TV                 92
Crime TV Shows             75
Sports Movies              54
British TV Shows           45
Kids' TV                   42
Science & Nature TV        35
Name: listed_in, dtype: int64
```

cast nulls vs rating

- Most of the nulls are for TV-MA, TV-14, TV-PG

In [53]:

```
netflix_data.loc[isna_full.cast].rating.value_counts().head(10)
```

Out[53]:

```
TV-MA    326
TV-14    205
TV-PG    144
TV-Y      39
TV-G      37
TV-Y7     24
NR        17
PG-13     13
R          9
PG         8
Name: rating, dtype: int64
```

cast nulls vs release year

- Most of the nulls are for recently released movies (2017 - 2021)

In [54]:

```
netflix_data.loc[isna_full.cast].release_year.value_counts().head(10)
```

Out[54]:

```
2020    126
2018    121
2017    120
2019    113
2021     98
2016     98
2015     46
2014     24
2013     20
2012     12
Name: release_year, dtype: int64
```

country nulls vs genre

- Most of the nulls are for International, Dramas, Children & Families, Kid's TV

In [55]:

```
isna_full.country.sum()
```

Out[55]:

```
831
```

In [56]:

```
netflix_data.loc[isna_full.country].listed_in.value_counts().head(10)
```

Out[56]:

```
Children & Family Movies                                    70
Kids' TV                                                    44
International TV Shows, TV Dramas                           36
Stand-Up Comedy                                             31
International TV Shows, Romantic TV Shows, TV Comedies      28
International TV Shows, Romantic TV Shows, TV Dramas        27
Dramas, International Movies                                25
Movies                                                      23
Documentaries, International Movies                         21
Comedies, International Movies                              21
Name: listed_in, dtype: int64
```

In [57]:

```
1
2 netflix_data.loc[isna_full.country].listed_in.str.split(", ").explode().value_counts().
```

Out[57]:

```
International TV Shows      223
International Movies        209
Dramas                      110
Children & Family Movies    106
TV Dramas                   100
Comedies                     94
Kids' TV                     81
TV Comedies                  80
Documentaries                75
Romantic TV Shows            71
Name: listed_in, dtype: int64
```

country nulls vs rating

- Most of the nulls are for TV-MA, TV-14

In [58]:

```
1
2 netflix_data.loc[isna_full.country].rating.value_counts().head(10)
```

Out[58]:

```
TV-MA       276
TV-14       230
TV-Y7        98
TV-PG        90
TV-Y         80
TV-G         30
R            11
PG-13         8
PG            6
TV-Y7-FV      1
Name: rating, dtype: int64
```

country nulls vs release year

- Most of the nulls are for recently released Shows

In [59]:

```
netflix_data.loc[isna_full.country].release_year.value_counts().head(10)
```

Out[59]:

```
2021    209
2019    117
2018    109
2020    101
2017     66
2016     64
2015     44
2014     19
2013     18
2010     16
Name: release_year, dtype: int64
```

nulls vs genres

- Nearly half the null values are from the genres that are generic and may involve, anonymous people

In [60]:

```
netflix_data.loc[isna_full[["director", "cast", "country"]].any(axis= 1), :].show_id.nu
```

Out[60]:

```
3471
```

In [61]:

```
temp = netflix_data.loc[isna_full[["director", "cast", "country"]].any(axis= 1), :]["li


temp_nulls = netflix_data.loc[isna_full[["director", "cast", "country"]].any(axis= 1),

temp_nulls.show_id.nunique()
```

Out[61]:

```
1560
```

In [62]:

```
1
2 temp_nulls.listed_in.str.split(", ").explode().value_counts().head(10)
```

Out[62]:

```
Documentaries            478
Kids' TV                 438
Docuseries               381
Reality TV               252
International TV Shows   220
International Movies     181
TV Comedies              161
British TV Shows         151
Crime TV Shows           107
Science & Nature TV       91
Name: listed_in, dtype: int64
```

Null value correction in cast, director

- Based on the null value analysis above we may conclude that the null values in cast, director are so because they people involved were mostly general public/animals/animation or anonymous
- hence, we may replace all the null values as "anonymous" in cast and director

In [63]:

```
1
2 netflix_data.loc[isna_full.director, "director"] = "Anonymous"
3 netflix_data.loc[isna_full.cast, "cast"] = "Anonymous"
4
5 netflix_data.loc[isna_full[["director", "cast"]].any(axis= 1)].head()
```

Out[63]:

	show_id	type	title	director	cast	country	date_added	release_year	ratir
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anonymous	United States	2021-09-25	2020	P(
1	s2	TV Show	Blood & Water	Anonymous	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	T' M
3	s4	TV Show	Jailbirds New Orleans	Anonymous	Anonymous	NaN	2021-09-24	2021	T' M
4	s5	TV Show	Kota Factory	Anonymous	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	T' M
10	s11	TV Show	Vendetta: Truth, Lies and The Mafia	Anonymous	Anonymous	NaN	2021-09-24	2021	T' M

Null value correction in country

- The nulls in the country may be replaced with the most popular country

In [64]:

```
netflix_data.loc[~isna_full.country].country.value_counts().head(5)
```

Out[64]:

```
United States     2818
India              972
United Kingdom     419
Japan              245
South Korea        199
Name: country, dtype: int64
```

In [65]:

```
netflix_data.loc[isna_full.country, "country"] = "United States"

netflix_data.loc[isna_full[["director", "cast", "country"]].any(axis= 1)].head()
```

Out[65]:

	show_id	type	title	director	cast	country	date_added	release_year	ratir
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anonymous	United States	2021-09-25	2020	P(
1	s2	TV Show	Blood & Water	Anonymous	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	T M
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	2021	T M
3	s4	TV Show	Jailbirds New Orleans	Anonymous	Anonymous	United States	2021-09-24	2021	T M
4	s5	TV Show	Kota Factory	Anonymous	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	T M

Column transformation for Analysis

Transforming duration to numeric

- For movies: we can get rid of mins
- For TV Shows: we can get rid of Season/s

In [66]:

```
1
2 netflix_data.head()
```

Out[66]:

	show_id	type	title	director	cast	country	date_added	release_year	ratir
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anonymous	United States	2021-09-25	2020	P(
1	s2	TV Show	Blood & Water	Anonymous	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	T M
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	2021	T M
3	s4	TV Show	Jailbirds New Orleans	Anonymous	Anonymous	United States	2021-09-24	2021	T M
4	s5	TV Show	Kota Factory	Anonymous	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	T M

In [67]:

```
1
2 netflix_data["duration"].apply(lambda x: x.split(" ")[0]).value_counts(dropna= False).i
3
4
```

Out[67]:

```
0
```

In [68]:

```
1
2 netflix_data["duration"] = pd.to_numeric(netflix_data["duration"].apply(lambda x: x.spl
```

In [69]:

```
1
2 netflix_data
```

Out[69]:

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anonymous	United States	2021-09-25	2020
1	s2	TV Show	Blood & Water	Anonymous	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	2021
3	s4	TV Show	Jailbirds New Orleans	Anonymous	Anonymous	United States	2021-09-24	2021
4	s5	TV Show	Kota Factory	Anonymous	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021
...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	2019-11-20	2007
8803	s8804	TV Show	Zombie Dumb	Anonymous	Anonymous	United States	2019-07-01	2018
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	2019-11-01	2009
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	2020-01-11	2006
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	2019-03-02	2015

8807 rows × 11 columns

Adding min_age column

- For a more insightfull analysis, the min_age for a rating can be derived based on th ebelow image

Local Rating Values	Kids *(All)*	Older Kids *(7+)*	Teens *(13+)*	Young Adults *(16+)*	Adults *(18+)*
MPAA (Movies)	G	PG	PG-13		NC-17
					NR
					Unrated
					R
TVPG (TV)	TV-G	TV-Y7		TV-14	TV-MA
		TV-Y7-FV			
	TV-Y	TV-PG			

In [70]:

```
def min_age_rating(rating):
    if (rating == "G") | (rating == "TV-G") | (rating == "TV-Y"):
        return 0
    if (rating == "PG") | (rating == "TV-Y7") | (rating == "TV-Y7-FV") | (rating == "TV
        return 7
    if (rating == "PG-13"):
        return 13
    if (rating == "TV-14"):
        return 16
    if (rating == "NC-17") | (rating == "NR") | (rating == "UR") | (rating == "R") | (r
        return 18

netflix_data["min_age"] = netflix_data["rating"].apply(min_age_rating)

netflix_data.head()
```

Out[70]:

	show_id	type	title	director	cast	country	date_added	release_year	ratir
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anonymous	United States	2021-09-25	2020	P(
1	s2	TV Show	Blood & Water	Anonymous	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	T M
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	2021	T M
3	s4	TV Show	Jailbirds New Orleans	Anonymous	Anonymous	United States	2021-09-24	2021	T M
4	s5	TV Show	Kota Factory	Anonymous	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	T M

Recheck of Null value counts

- All nulls have been imputed

In [71]:

```
1
2 netflix_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      8807 non-null   object        
 4   cast          8807 non-null   object        
 5   country       8807 non-null   object        
 6   date_added    8807 non-null   datetime64[ns]
 7   release_year  8807 non-null   int64         
 8   rating        8807 non-null   object        
 9   duration      8807 non-null   int64         
 10  listed_in     8807 non-null   object        
 11  min_age       8807 non-null   int64         
dtypes: datetime64[ns](1), int64(3), object(8)
memory usage: 825.8+ KB
```

In [72]:

```
1
2 netflix_data.isna().sum()
```

Out[72]:

```
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
min_age         0
dtype: int64
```

In [73]:

```
1
2 netflix_data.isna().apply(lambda x: round((x.sum()/x.size)*100, 2))
```

Out[73]:

```
show_id         0.0
type            0.0
title           0.0
director        0.0
cast            0.0
country         0.0
date_added      0.0
release_year    0.0
rating          0.0
duration        0.0
listed_in       0.0
min_age         0.0
dtype: float64
```

Unnesting of nested columns

- We are going to unnest the columns: cast, director, country, listed_in

In [74]:

```
1
2  netflix_data_listed = netflix_data.copy()
3
4  netflix_data_listed["director"] = netflix_data_listed["director"].str.split(", ")
5
6  netflix_data_listed["cast"] = netflix_data_listed["cast"].str.split(", ")
7
8  netflix_data_listed["country"] = netflix_data_listed["country"].str.split(", ")
9
10 netflix_data_listed["listed_in"] = netflix_data_listed["listed_in"].str.split(", ")
11
12 netflix_data_listed.head()
```

Out[74]:

	show_id	type	title	director	cast	country	date_added	release_year	ra
0	s1	Movie	Dick Johnson Is Dead	[Kirsten Johnson]	[Anonymous]	[United States]	2021-09-25	2020	
1	s2	TV Show	Blood & Water	[Anonymous]	[Ama Qamata, Khosi Ngema, Gail Mabalane, Thaba...	[South Africa]	2021-09-24	2021	
2	s3	TV Show	Ganglands	[Julien Leclercq]	[Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nab...	[United States]	2021-09-24	2021	
3	s4	TV Show	Jailbirds New Orleans	[Anonymous]	[Anonymous]	[United States]	2021-09-24	2021	
4	s5	TV Show	Kota Factory	[Anonymous]	[Mayur More, Jitendra Kumar, Ranjan Raj, Alam ...	[India]	2021-09-24	2021	

In [75]:

```
netflix_data_full = netflix_data_listed.explode(
                                                "director",
                                                ignore_index= True
                                                ).explode(
                                                        "cast",
                                                        ignore_index= True
                                                        ).explode(
                                                                "country",
                                                                ignore_index= True
                                                                ).explode(
                                                                        "listed_in",
                                                                        ignore_index= T
```

In [76]:

```
netflix_data_full.head()
```

Out[76]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anonymous	United States	2021-09-25	2020	PG-13
1	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA
2	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA
3	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA
4	s2	TV Show	Blood & Water	Anonymous	Khosi Ngema	South Africa	2021-09-24	2021	TV-MA

Recheck of null value counts

In [77]:

```
1
2 netflix_data_full.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201991 entries, 0 to 201990
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   show_id       201991 non-null  object
 1   type          201991 non-null  object
 2   title         201991 non-null  object
 3   director      201991 non-null  object
 4   cast          201991 non-null  object
 5   country       201991 non-null  object
 6   date_added    201991 non-null  datetime64[ns]
 7   release_year  201991 non-null  int64
 8   rating        201991 non-null  object
 9   duration      201991 non-null  int64
 10  listed_in     201991 non-null  object
 11  min_age       201991 non-null  int64
dtypes: datetime64[ns](1), int64(3), object(8)
memory usage: 18.5+ MB
```

In [78]:

```
1
2 netflix_data_full.isna().sum()
```

Out[78]:

```
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
min_age         0
dtype: int64
```

Analysing the Data

Full data value count by type

In [79]:

```
df = netflix_data_full

df.head()
```

Out[79]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anonymous	United States	2021-09-25	2020	PG-13
1	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA
2	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA
3	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA
4	s2	TV Show	Blood & Water	Anonymous	Khosi Ngema	South Africa	2021-09-24	2021	TV-MA

In [80]:

```
df.show_id.nunique()
```

Out[80]:

```
8807
```

In [81]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201991 entries, 0 to 201990
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype         
---  ------        --------------   -----         
 0   show_id       201991 non-null  object        
 1   type          201991 non-null  object        
 2   title         201991 non-null  object        
 3   director      201991 non-null  object        
 4   cast          201991 non-null  object        
 5   country       201991 non-null  object        
 6   date_added    201991 non-null  datetime64[ns]
 7   release_year  201991 non-null  int64         
 8   rating        201991 non-null  object        
 9   duration      201991 non-null  int64         
 10  listed_in     201991 non-null  object        
 11  min_age       201991 non-null  int64         
dtypes: datetime64[ns](1), int64(3), object(8)
memory usage: 18.5+ MB
```

In [82]:

```
df.describe()
```

Out[82]:

	release_year	duration	min_age
count	201991.000000	201991.000000	201991.000000
mean	2013.452891	77.688749	14.696838
std	9.003933	51.488067	4.906162
min	1925.000000	1.000000	0.000000
25%	2012.000000	4.000000	13.000000
50%	2016.000000	95.000000	18.000000
75%	2019.000000	112.000000	18.000000
max	2021.000000	312.000000	18.000000

In [83]:

```
df.describe(include= "object", exclude= "int")
```

Out[83]:

	show_id	type	title	director	cast	country	rating	listed_in
count	201991	201991	201991	201991	201991	201991	201991	201991
unique	8807	2	8807	4994	36440	127	14	42
top	s7165	Movie	Kahlil Gibran's The Prophet	Anonymous	Anonymous	United States	TV-MA	Dramas
freq	700	145843	700	50643	2146	71246	73925	29775

In [84]:

```
df.type.unique()
```

Out[84]:

```
array(['Movie', 'TV Show'], dtype=object)
```

In [85]:

```
netflix_data.type.value_counts()
```

Out[85]:

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

In [86]:

```
df.groupby(["type"]).nunique()
```

Out[86]:

type	show_id	title	director	cast	country	date_added	release_year	rating	duration	lis
Movie	6131	6131	4778	25952	122	1533	73	14	205	
TV Show	2676	2676	300	14864	66	1018	46	9	15	

In [87]:

```
(6131/8807) * 100
```

Out[87]:

```
69.61507891449983
```

In [88]:

```
df[df.type == "TV Show"].show_id.nunique()
```

Out[88]:

```
2676
```

Movie Data Analysis - Unnested

The following is the analysis of movies data only for unnested columns

- "show_id", "type", "title", "date_added", "release_year", "rating", "duration", "min_age"

In [89]:

```
netflix_data_listed_movies = netflix_data_listed[netflix_data_listed.type == "Movie"].c

netflix_data_listed_movies_nonest = netflix_data_listed_movies[
                                    ["show_id", "type", "title", "date_added", "release
                                    ].copy()

netflix_data_listed_movies_nonest.head()
```

Out[89]:

	show_id	type	title	date_added	release_year	rating	duration	min_age
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90	13
6	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	7
7	s8	Movie	Sankofa	2021-09-24	1993	TV-MA	125	18
9	s10	Movie	The Starling	2021-09-24	2021	PG-13	104	13
12	s13	Movie	Je Suis Karl	2021-09-23	2021	TV-MA	127	18

In [90]:

```
df = netflix_data_listed_movies_nonest.copy()
```

Univariate Analysis:

What is the distribution of:

- Ratings
- duration
- release year
- added month, year, day of month, day of year, day of week
- difference between released and added year

In [91]:

```
df["rating"].unique()
```

Out[91]:

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

No of movies by rating

Observations:

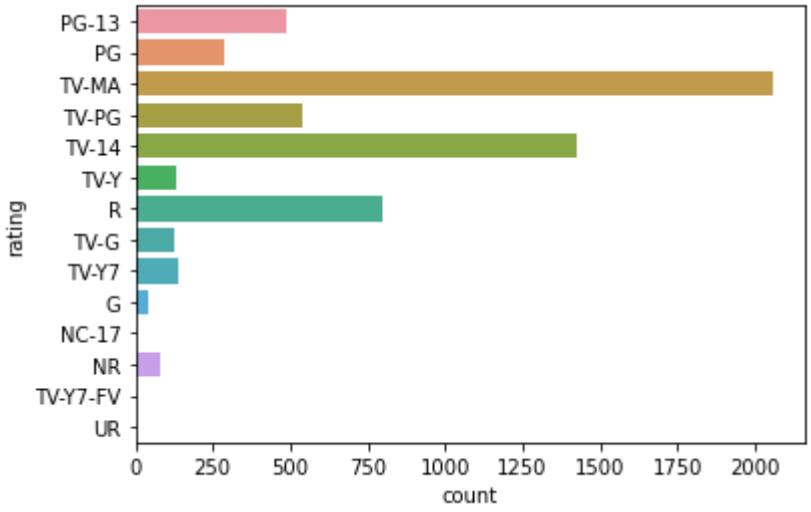
- The graph below shows that most no of movies belong to 1. TV-MA, 2. TV-14, 3. R ratings

In [92]:

```
1
2 sns.countplot(data= df, y= "rating")
```

Out[92]:

<AxesSubplot:xlabel='count', ylabel='rating'>



No of movies by min_age

Observations:

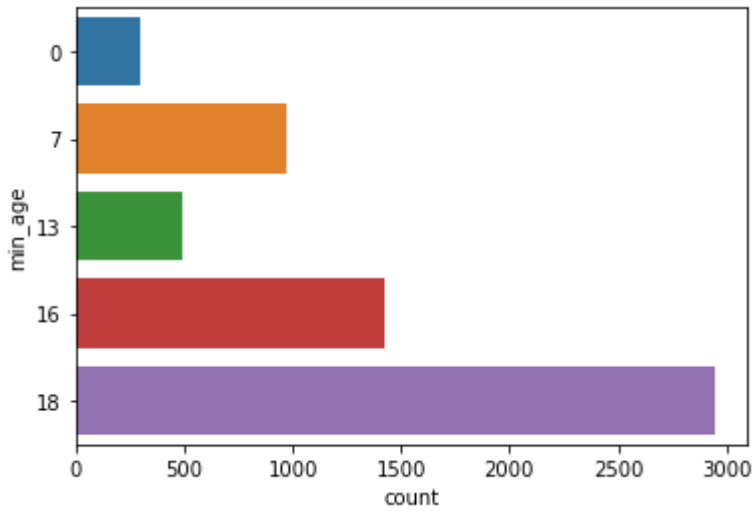
- Most of the movies are for people of age 1. 18+, 2. 16+, 3. 7+

In [93]:

```
1
2 sns.countplot(data= df, y= "min_age")
```

Out[93]:

<AxesSubplot:xlabel='count', ylabel='min_age'>



No of movies by duration

Observation:

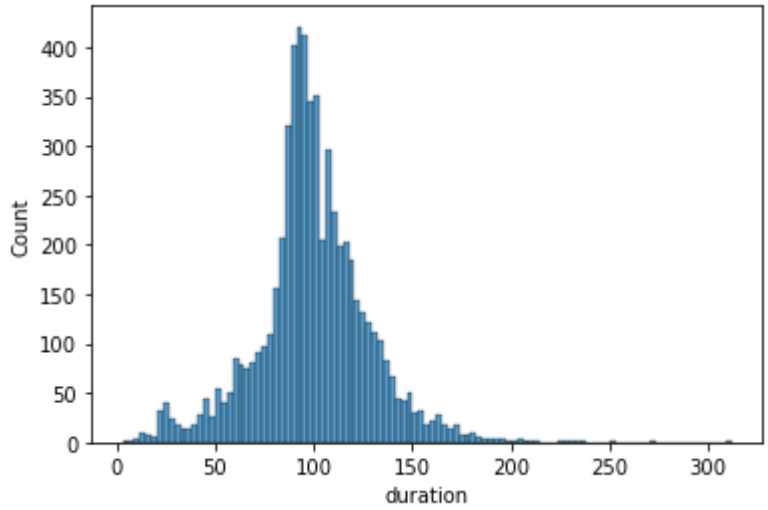
- most of the movies are of 90 - 120 mins duration i.e. 1 1/2 to 2 hr duration

In [94]:

```
1
2 sns.histplot(data= df, x= "duration")
```

Out[94]:

```
<AxesSubplot:xlabel='duration', ylabel='Count'>
```



No of movies by duration

Observation:

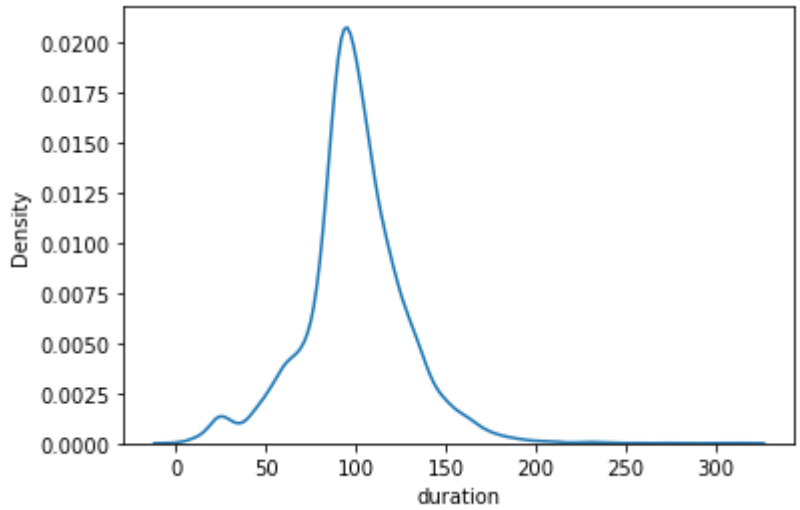
- most of the movies are of 90 - 120 mins duration i.e. 1 1/2 to 2 hr duration

In [95]:

```
1
2 sns.kdeplot(data= df, x= "duration")
```

Out[95]:

```
<AxesSubplot:xlabel='duration', ylabel='Density'>
```



No of movies by release year

Observation:

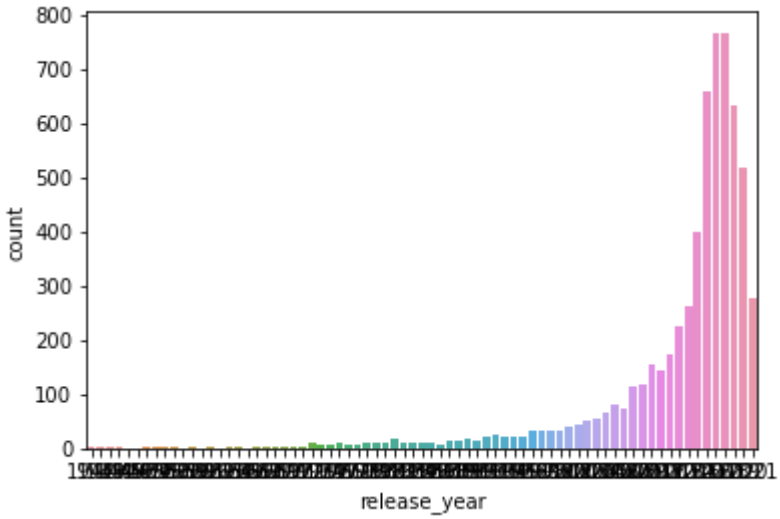
- most of the movies on netflix are released in recent years i.e. very few old movies

In [96]:

```
1
2 sns.countplot(data= df, x= "release_year")
```

Out[96]:

```
<AxesSubplot:xlabel='release_year', ylabel='count'>
```



No of movies by release year

Observation:

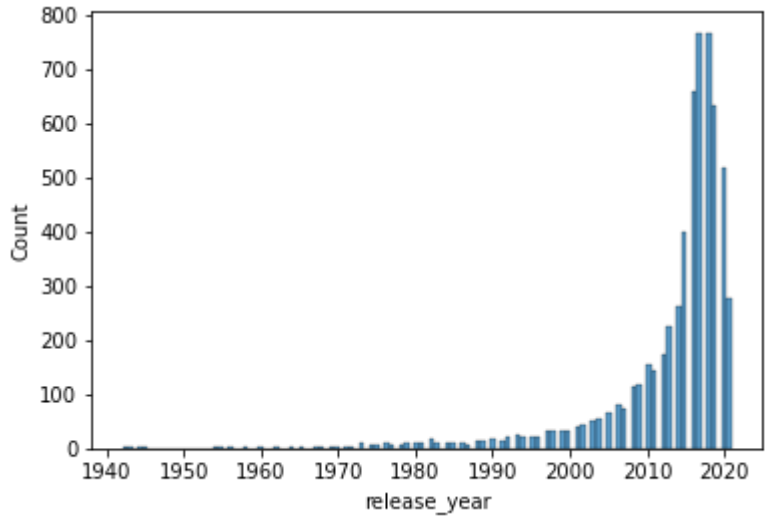
- most of the movies on netflix are released in recent years i.e. there are very few old movies

In [97]:

```
1
2 sns.histplot(data= df, x= "release_year")
```

Out[97]:

```
<AxesSubplot:xlabel='release_year', ylabel='Count'>
```



No of movies by release year

Observation:

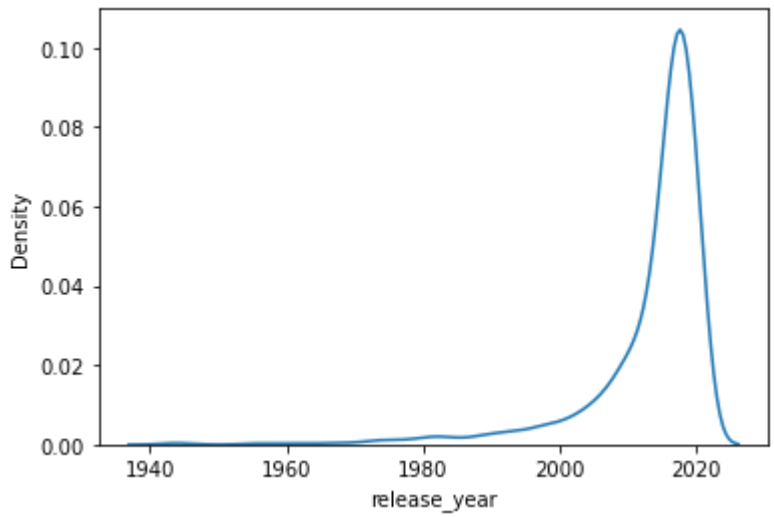
- most of the movies on netflix are released in recent years i.e. there are very few old movies

In [98]:

```
1
2 sns.kdeplot(data= df, x= "release_year")
```

Out[98]:

```
<AxesSubplot:xlabel='release_year', ylabel='Density'>
```



No of movies by release year

Observation:

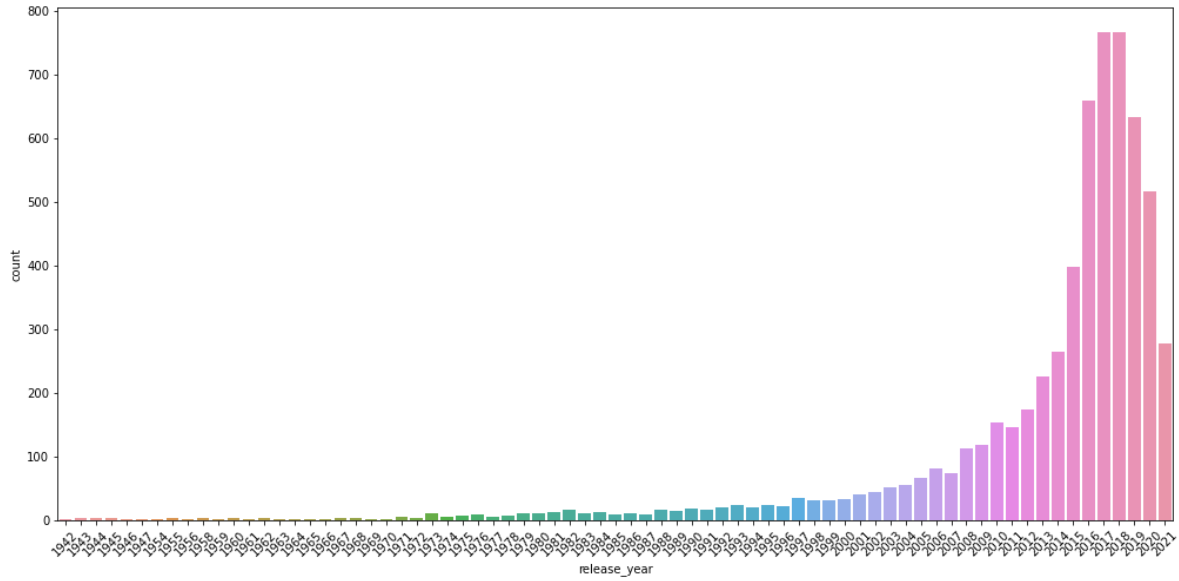
- The no of recently released movies are lesser than the the movies released 2 to 3 years ago on Netflix

In [99]:

```
1
2 plt.figure(figsize=(17,8))
3 plt.xticks(rotation=45)
4 sns.countplot(data= df, x= "release_year")
```

Out[99]:

```
<AxesSubplot:xlabel='release_year', ylabel='count'>
```



No of movies by added date

Observation:

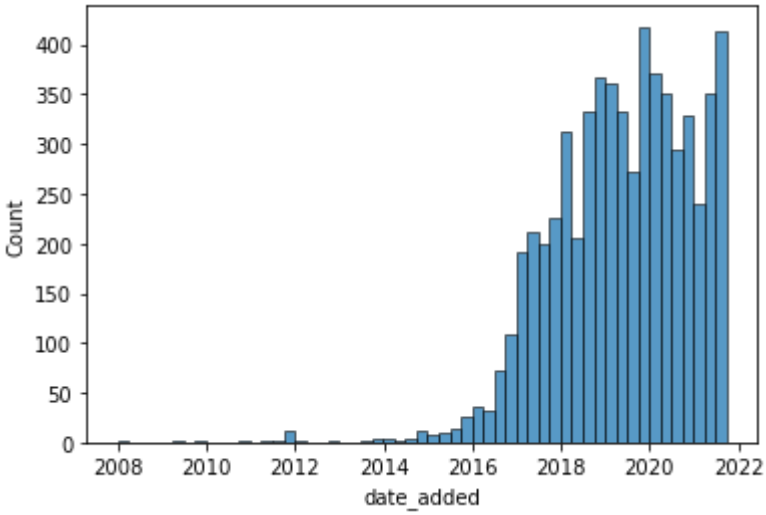
- The no. of movies added increaed rapidly from 2014 to 2016 and then recently stabilized

In [100]:

```
1
2 sns.histplot(data= df, x= "date_added")
```

Out[100]:

```
<AxesSubplot:xlabel='date_added', ylabel='Count'>
```



No of movies by added date

Observation:

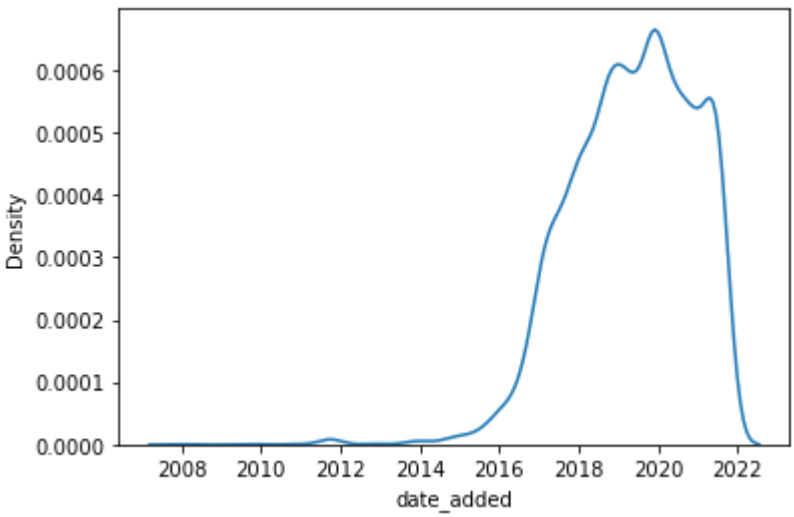
- The no. of movies added increaed rapidly from 2014 to 2016 and then recently stabilized

In [101]:

```
1
2 sns.kdeplot(data= df, x= "date_added")
```

Out[101]:

```
<AxesSubplot:xlabel='date_added', ylabel='Density'>
```



In [102]:

```
df["day_of_year_added"] = df["date_added"].dt.dayofyear
df["day_of_month_added"] = df["date_added"].dt.day
df["day_of_week_added"] = df["date_added"].dt.dayofweek
df["month_added"] = df["date_added"].dt.month
df["quarter_added"] = df["date_added"].dt.quarter
df["year_added"] = df["date_added"].dt.year
df["released_decade"] = df["release_year"].apply(lambda x: x - (x%10))

df.head()
```

Out[102]:

	show_id	type	title	date_added	release_year	rating	duration	min_age	day_of_ye
0	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90	13	
6	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91	7	
7	s8	Movie	Sankofa	2021-09-24	1993	TV-MA	125	18	
9	s10	Movie	The Starling	2021-09-24	2021	PG-13	104	13	
12	s13	Movie	Je Suis Karl	2021-09-23	2021	TV-MA	127	18	

No of movies by added date

Observation:

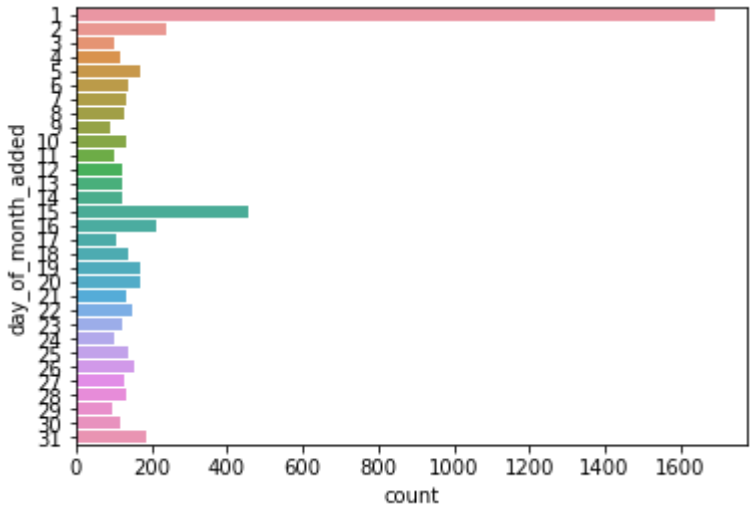
- Most movies are added on the start of the month or the middle of the month
- no. of movis added pn the rest of the days of the month is similar

In [103]:

```
1
2 sns.countplot(data= df, y= "day_of_month_added")
```

Out[103]:

```
<AxesSubplot:xlabel='count', ylabel='day_of_month_added'>
```



No of movies by added day of week

Observation:

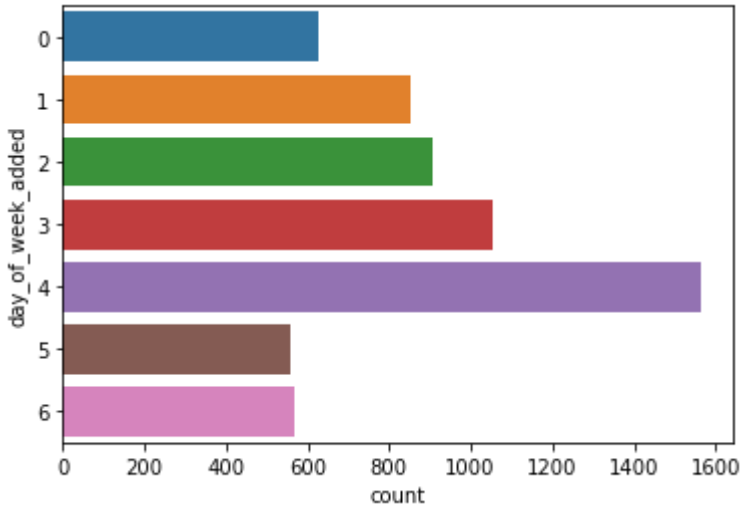
- Most movies are added on the 5th day of the week (Friday) or the 4th day of the week (Thurday) i.e. just before weekend

In [104]:

```
1
2 sns.countplot(data= df, y= "day_of_week_added")
```

Out[104]:

```
<AxesSubplot:xlabel='count', ylabel='day_of_week_added'>
```



No of movies by added month of the year

Observation:

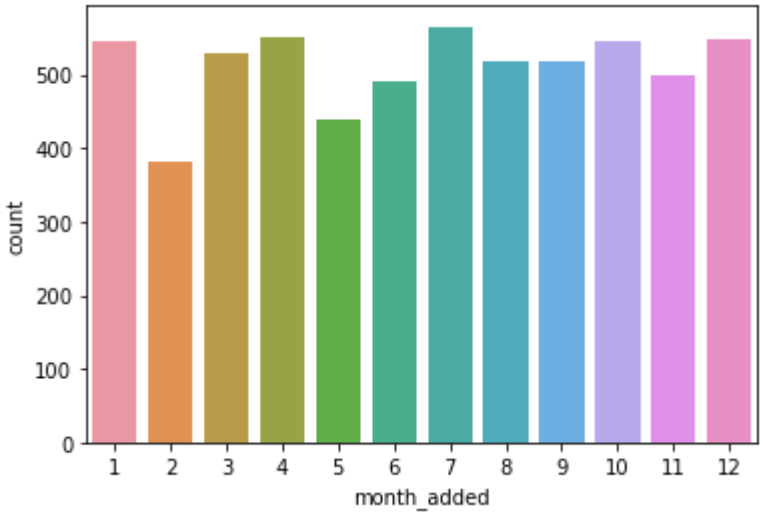
- The no. of movies added is very low in february/ may
- the no. of movies added on other months are similar

In [105]:

```
1
2 sns.countplot(data= df, x= "month_added")
```

Out[105]:

```
<AxesSubplot:xlabel='month_added', ylabel='count'>
```

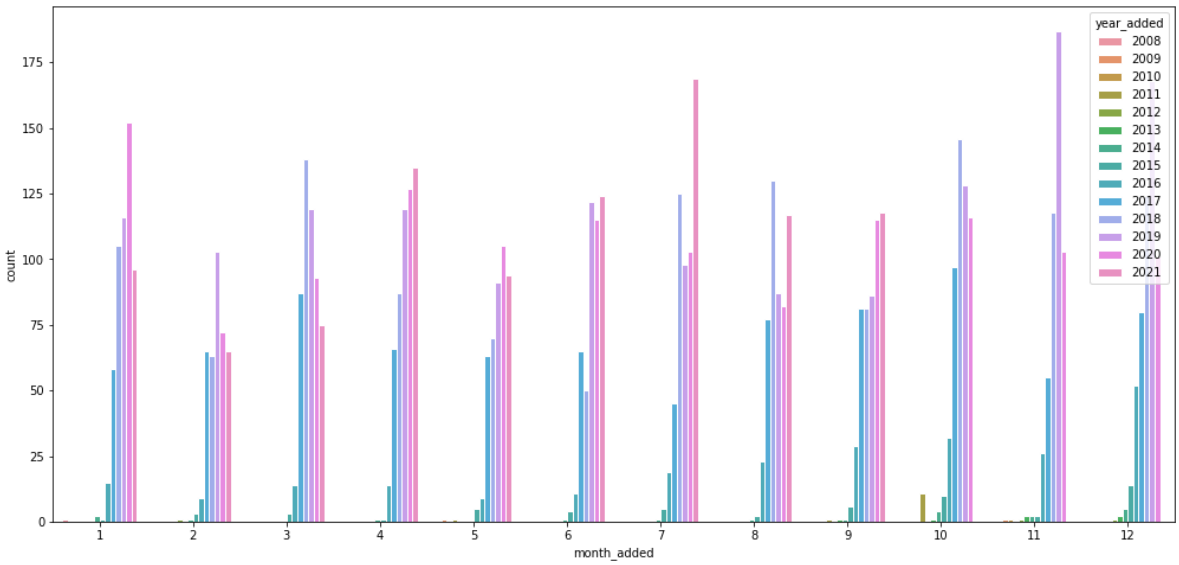


In [106]:

```
1
2 plt.figure(figsize= (17, 8))
3
4 sns.countplot(data= df, x= "month_added", hue= "year_added", edgecolor= "white")
```

Out[106]:

```
<AxesSubplot:xlabel='month_added', ylabel='count'>
```



No of movies by added quarter of the year

Observation:

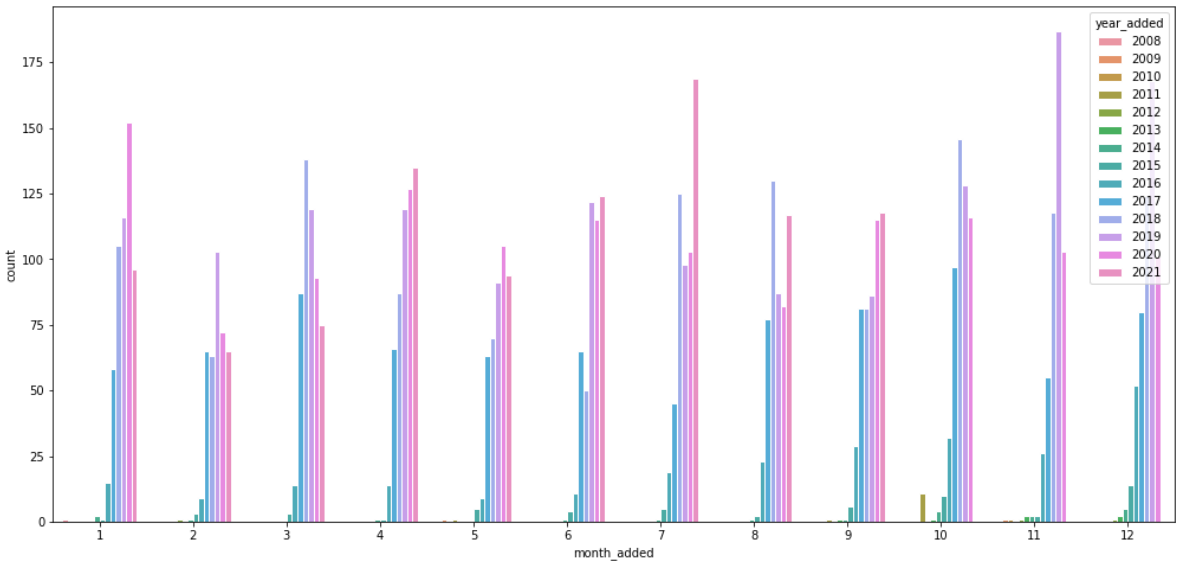
- The no. of movies added is moderately high in the 4th quarter
- the no. of movies added on other quarters are similar

In [107]:

```
1
2 plt.figure(figsize= (17, 8))
3
4 sns.countplot(data= df, x= "month_added", hue= "year_added", edgecolor= "white")
```

Out[107]:

<AxesSubplot:xlabel='month_added', ylabel='count'>

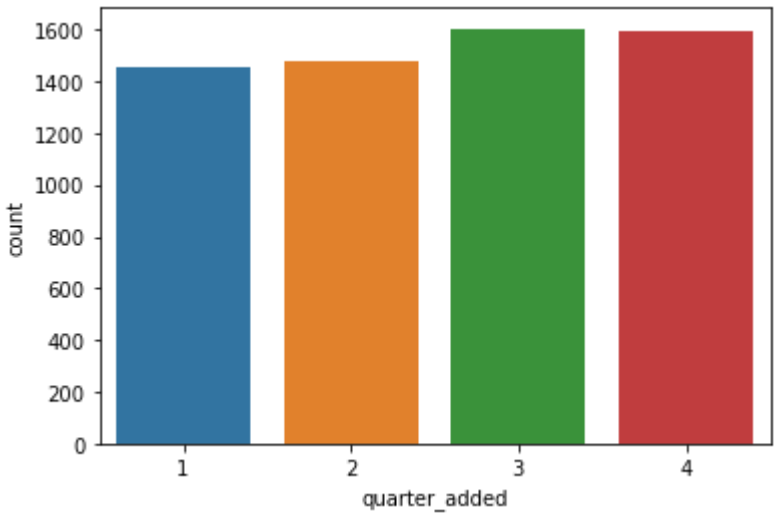


In [108]:

```
1
2 sns.countplot(data= df, x= "quarter_added")
```

Out[108]:

<AxesSubplot:xlabel='quarter_added', ylabel='count'>

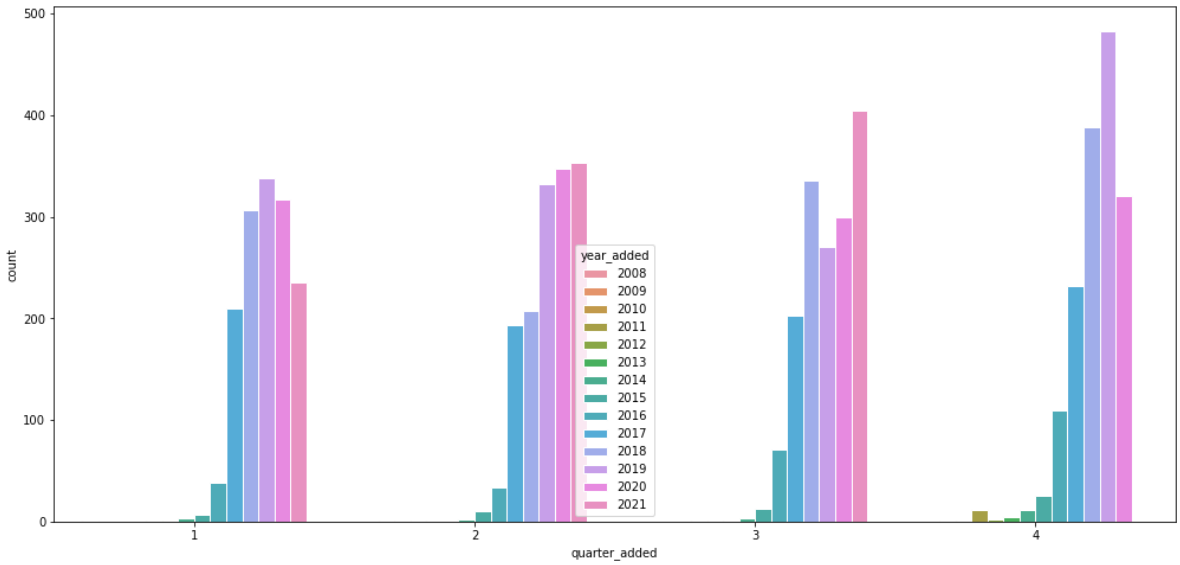


In [109]:

```
1
2 plt.figure(figsize= (17, 8))
3
4 sns.countplot(data= df, x= "quarter_added", hue= "year_added", edgecolor= "white")
```

Out[109]:

```
<AxesSubplot:xlabel='quarter_added', ylabel='count'>
```



distribution of duration of movies

Observation:

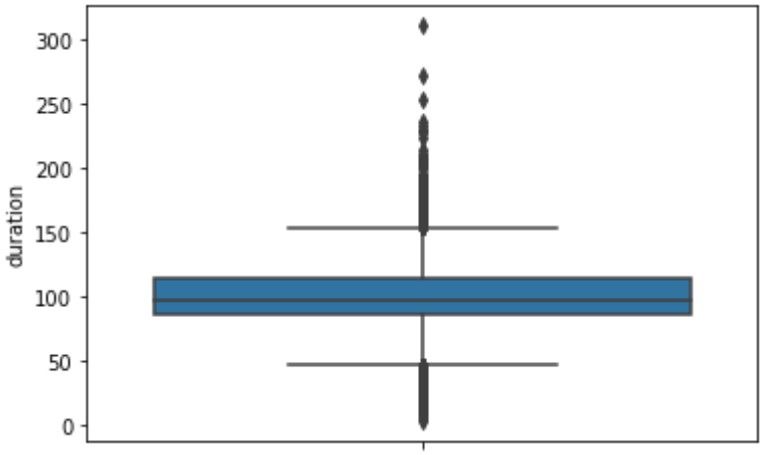
- most movies are around 100 min long
- There are only a few very short and very long movies

In [110]:

```
1
2 sns.boxplot(data= df, y= "duration")
```

Out[110]:

<AxesSubplot:ylabel='duration'>

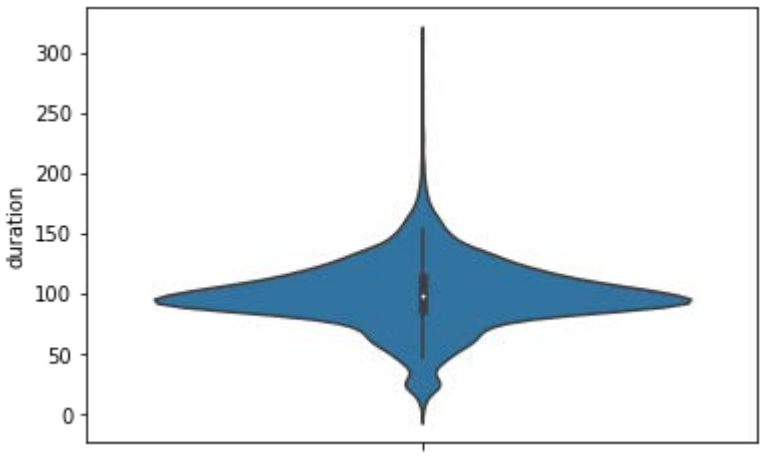


In [111]:

```
1
2 sns.violinplot(data= df, y= "duration")
```

Out[111]:

<AxesSubplot:ylabel='duration'>



distribution of release year of movies

Observation:

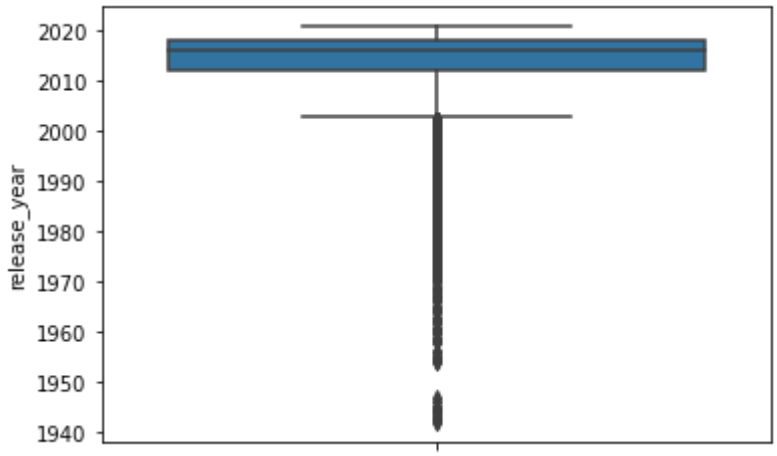
- most movies are released during 2010 - 2020
- There are only a few old movies

In [112]:

```
1
2 sns.boxplot(data= df, y= "release_year")
```

Out[112]:

<AxesSubplot:ylabel='release_year'>

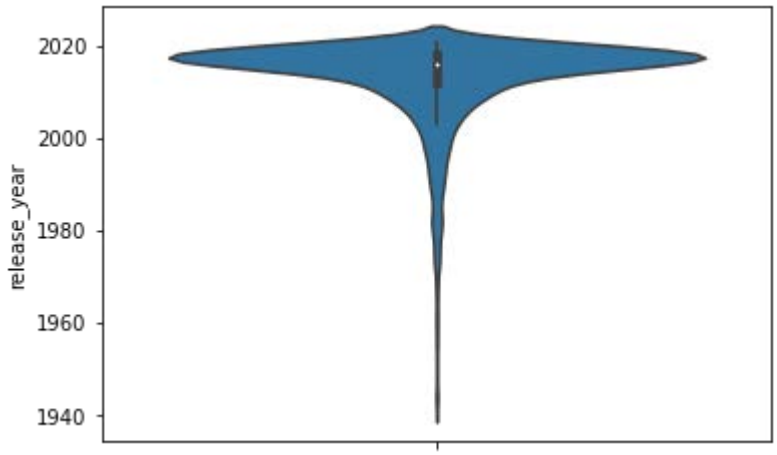


In [113]:

```
1
2 sns.violinplot(data= df, y= "release_year")
```

Out[113]:

<AxesSubplot:ylabel='release_year'>



distribution of day of year added of movies

Observation:

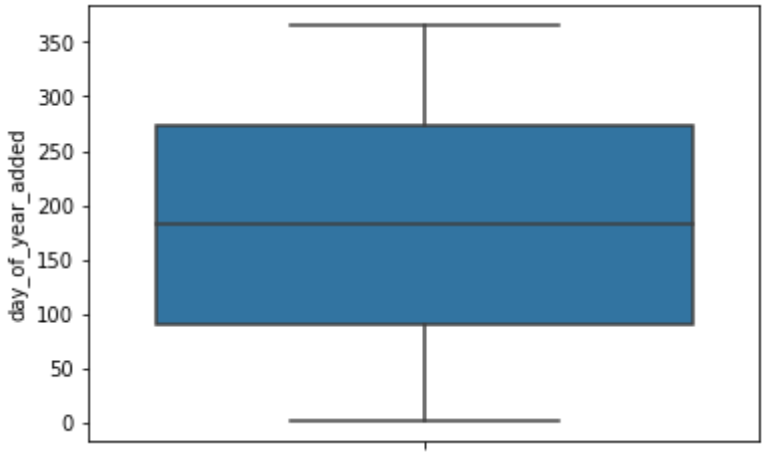
- nothing

In [114]:

```
1
2 sns.boxplot(data= df, y= "day_of_year_added")
```

Out[114]:

```
<AxesSubplot:ylabel='day_of_year_added'>
```



distribution of day of month added of movies

Observation:

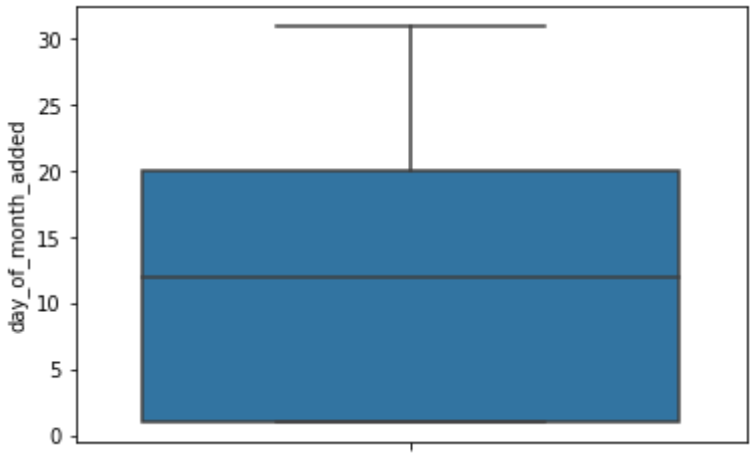
- most movies are added during the first half of th month

In [115]:

```
1
2 sns.boxplot(data= df, y= "day_of_month_added")
```

Out[115]:

```
<AxesSubplot:ylabel='day_of_month_added'>
```

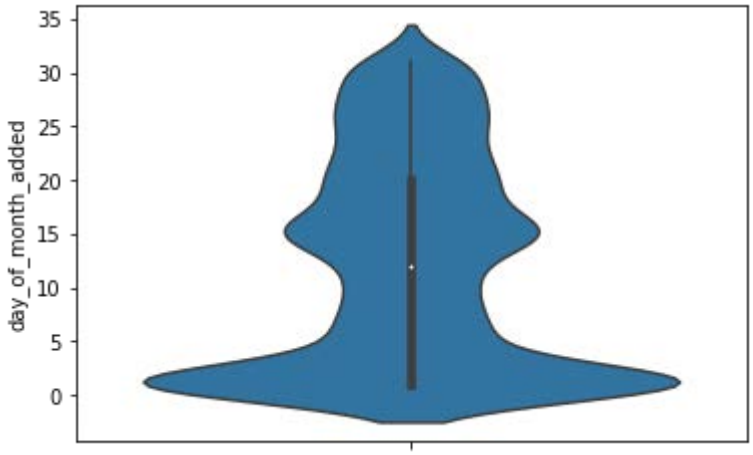


In [116]:

```
1
2 sns.violinplot(data= df, y= "day_of_month_added")
```

Out[116]:

```
<AxesSubplot:ylabel='day_of_month_added'>
```



distribution of added year of movies

Observation:

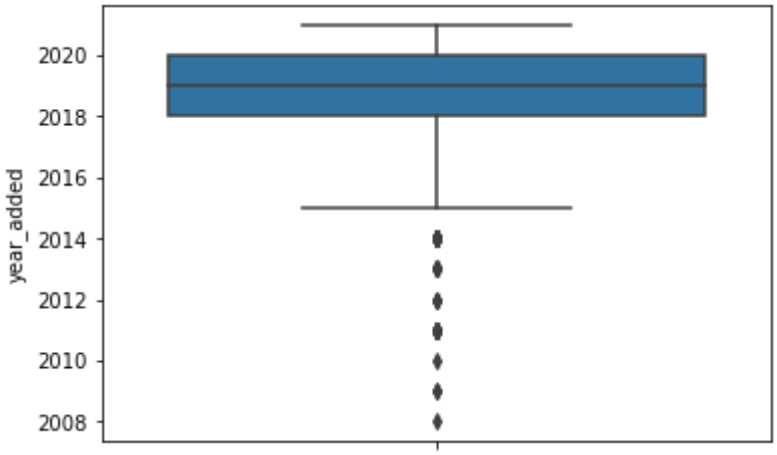
- most movies are added during the last couple years (after 2018)
- the no. of movies on netflix before 2015 were very low

In [117]:

```
1
2 sns.boxplot(data= df, y= "year_added")
```

Out[117]:

```
<AxesSubplot:ylabel='year_added'>
```

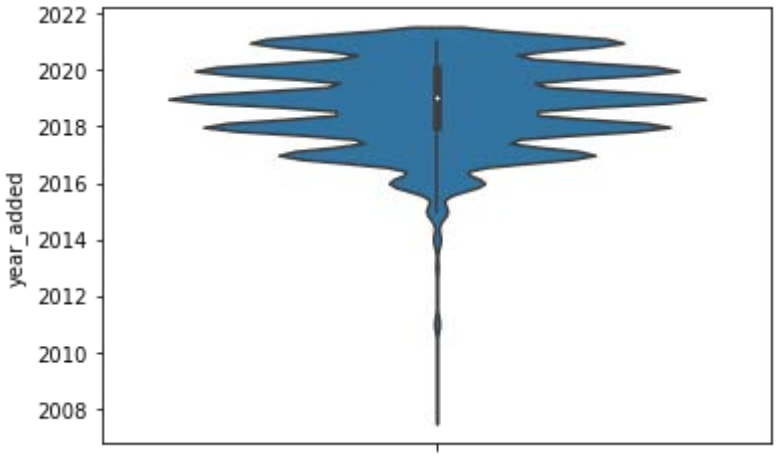


In [118]:

```
1
2 sns.violinplot(data= df, y= "year_added")
```

Out[118]:

<AxesSubplot:ylabel='year_added'>



distribution of added day of week for movies

Observation:

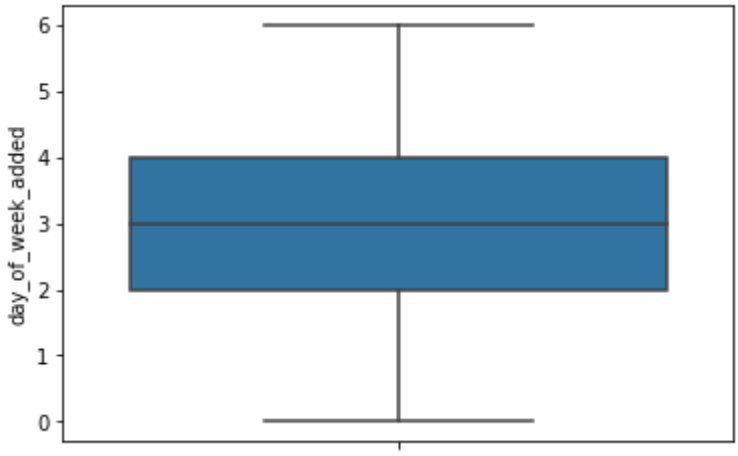
- most movies are added in the middle of the week

In [119]:

```
1
2 sns.boxplot(data= df, y= "day_of_week_added")
```

Out[119]:

<AxesSubplot:ylabel='day_of_week_added'>



distribution of added quarter of month for movies

Observation:

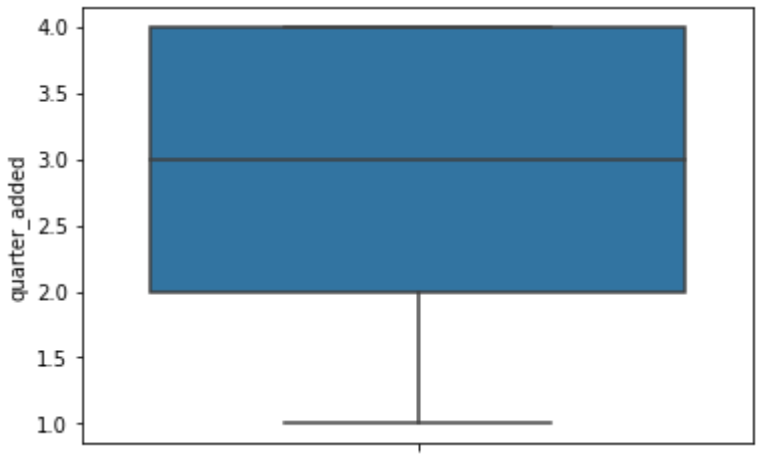
- most movies are added during the last 2 quarters (maybe)

In [120]:

```
1
2 sns.boxplot(data= df, y= "quarter_added")
```

Out[120]:

```
<AxesSubplot:ylabel='quarter_added'>
```



no. of movies by year added

Observation:

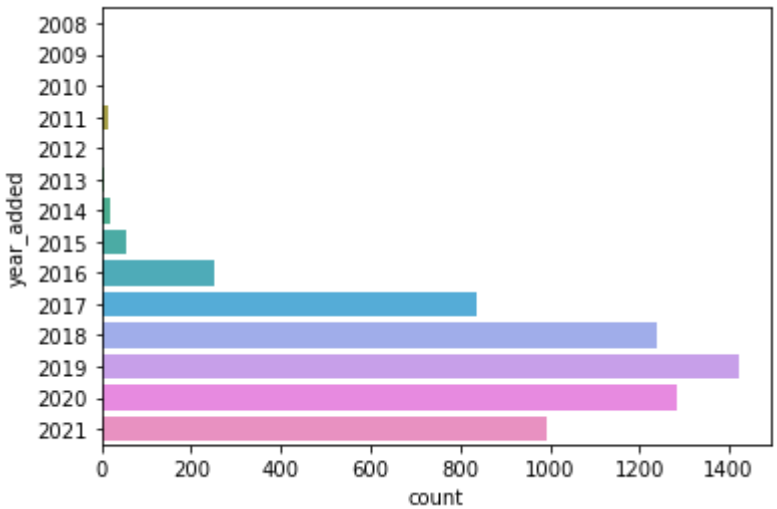
- most movies are added in the last couple years (after 2018)
- the no of movies aded per yaer is decreasing over the last years
- the no of movies added was increasing from 2014 - 2019

In [121]:

```
1
2 sns.countplot(data= df, y= "year_added")
```

Out[121]:

```
<AxesSubplot:xlabel='count', ylabel='year_added'>
```

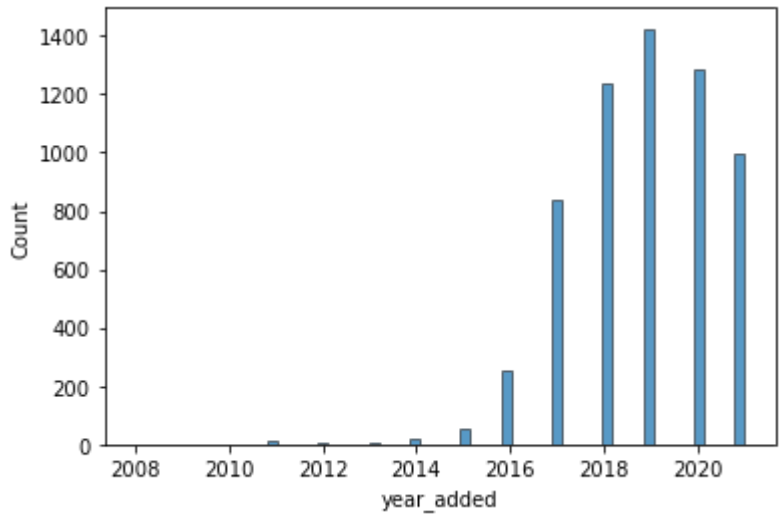


In [122]:

```
sns.histplot(data= df, x= "year_added")
```

Out[122]:

<AxesSubplot:xlabel='year_added', ylabel='Count'>

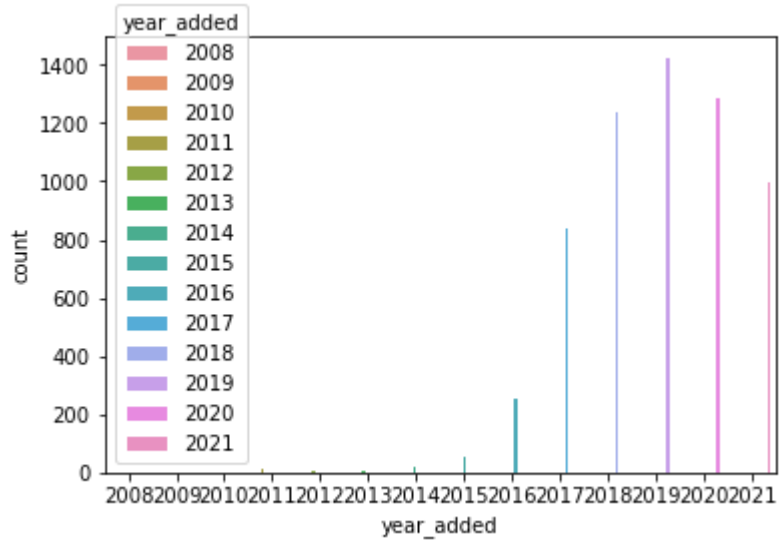


In [123]:

```
sns.countplot(data= df, x= "year_added", hue= "year_added")
```

Out[123]:

<AxesSubplot:xlabel='year_added', ylabel='count'>



no. of movies by year added and month of year

Observation:

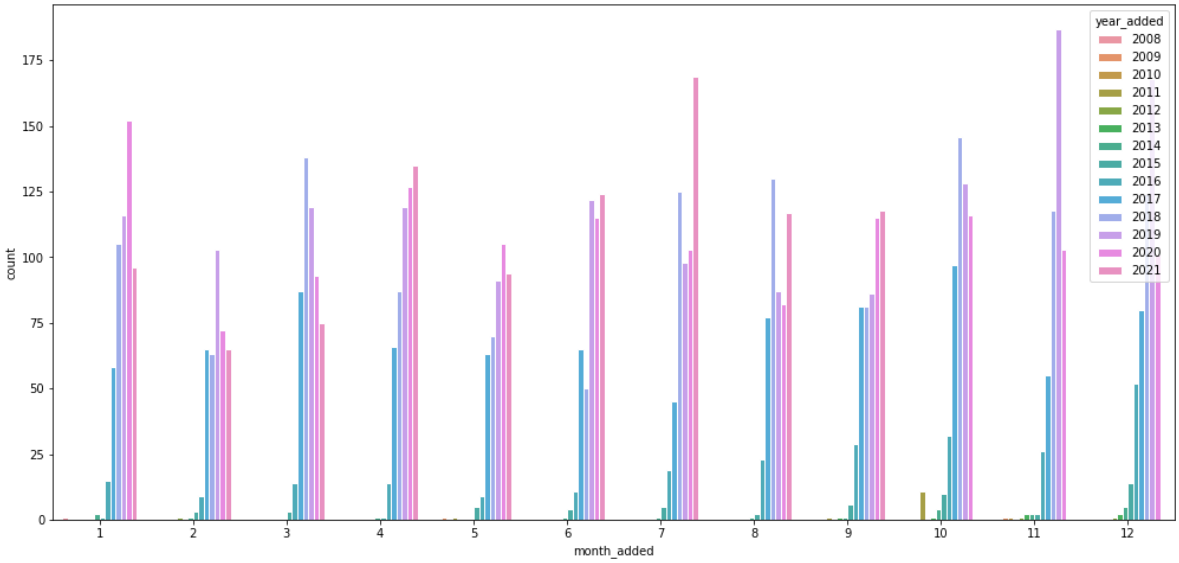
- the no of movies added each month is decreasing for the months in 1st half of the yaer, each year
- increasing for the month in 2nd half of the year each year in recent years

In [124]:

```
plt.figure(figsize= (17, 8))

sns.countplot(data= df, x= "month_added", hue= "year_added", edgecolor= "white")
```

Out[124]:

```
<AxesSubplot:xlabel='month_added', ylabel='count'>
```



no. of movies by release decade

Observation:

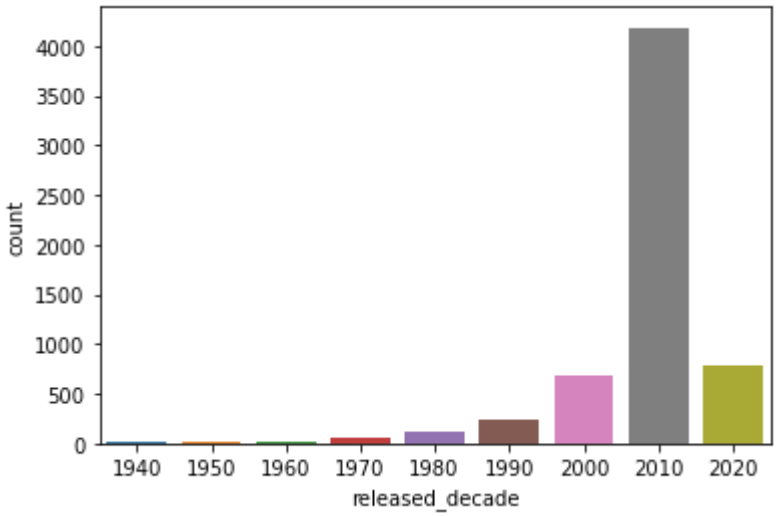
- Most movies on netflix are from the 2010's

In [125]:

```
sns.countplot(data= df, x= "released_decade")
```

Out[125]:

```
<AxesSubplot:xlabel='released_decade', ylabel='count'>
```

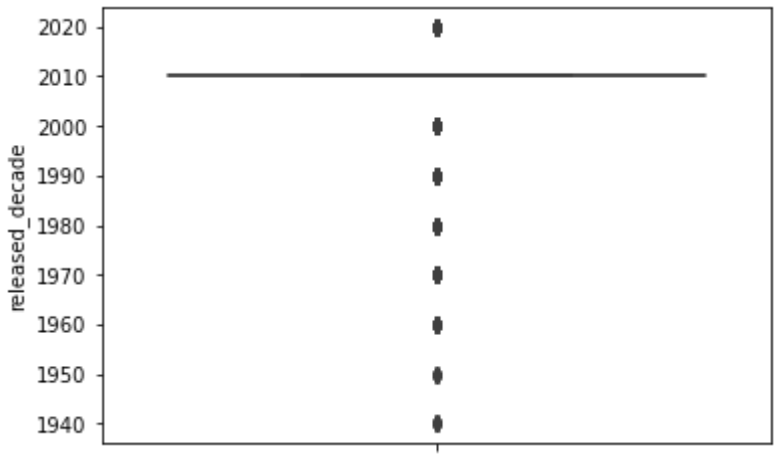


In [126]:

```
1
2 sns.boxplot(data= df, y= "released_decade")
```

Out[126]:

```
<AxesSubplot:ylabel='released_decade'>
```



Bivariate Analysis

duration of movies by release_year

Observation:

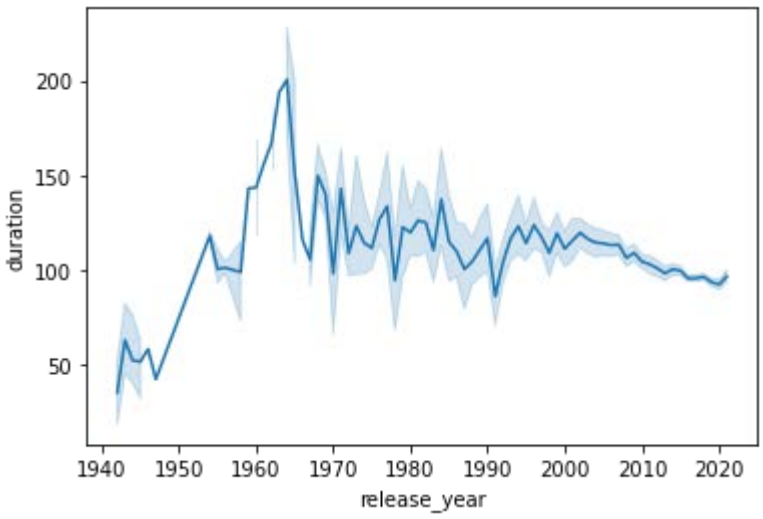
- The few movies (on Netflix) that released during the 60's were longer
- Rest of the movies were around 100 min long irrespective their release yaer

In [127]:

```
1
2 sns.lineplot(data= df, x= "release_year", y= "duration")
```

Out[127]:

```
<AxesSubplot:xlabel='release_year', ylabel='duration'>
```

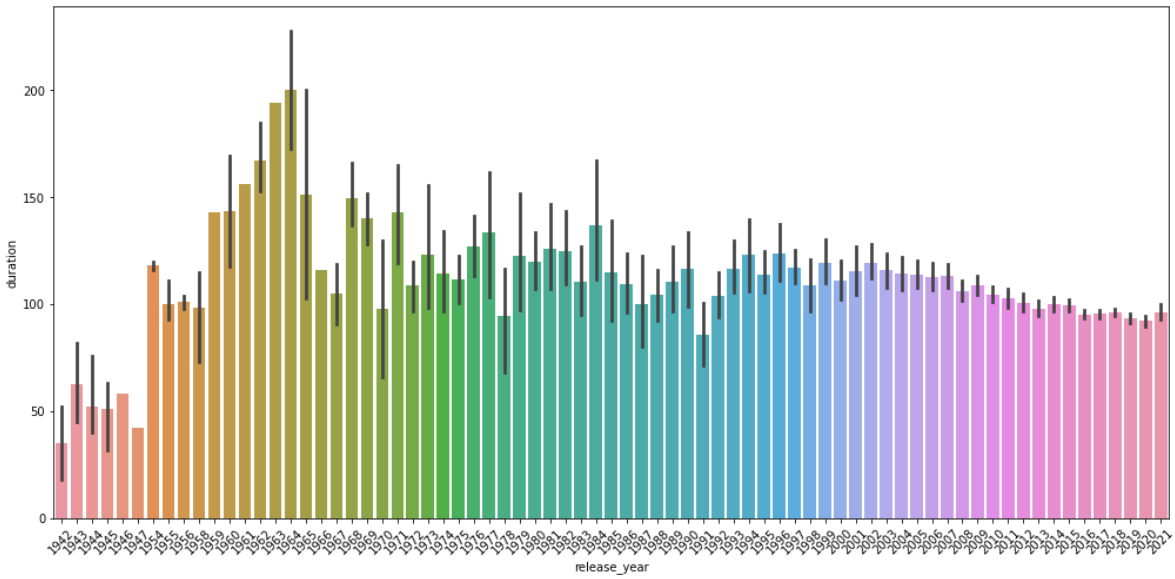


In [128]:

```
1
2 plt.figure(figsize=(17,8))
3 plt.xticks(rotation=45)
4 sns.barplot(data= df, x= "release_year", y= "duration", estimator= np.mean)
```

Out[128]:

<AxesSubplot:xlabel='release_year', ylabel='duration'>

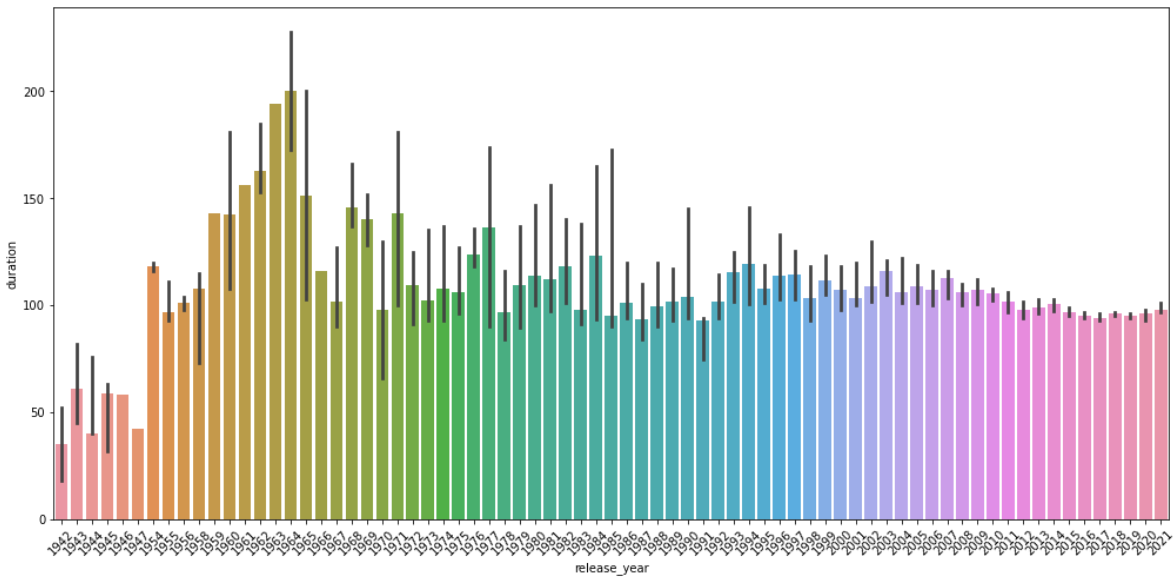


In [129]:

```
1
2 plt.figure(figsize=(17,8))
3 plt.xticks(rotation=45)
4 sns.barplot(data= df, x= "release_year", y= "duration", estimator= np.median)
```

Out[129]:

<AxesSubplot:xlabel='release_year', ylabel='duration'>



duration of movies by release_decade

Observation:

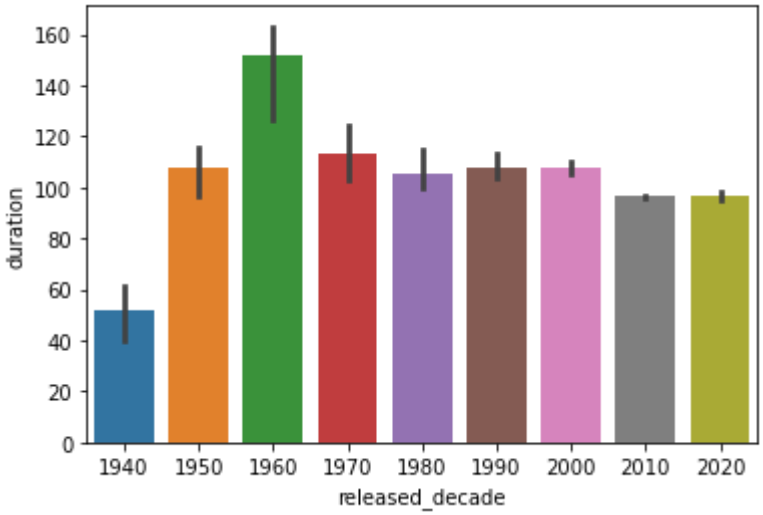
- the few movies (on Netflix) that released during the 40's were much shorter in duration

In [130]:

```
1
2 sns.barplot(data= df, x= "released_decade", y= "duration", estimator= np.median)
```

Out[130]:

```
<AxesSubplot:xlabel='released_decade', ylabel='duration'>
```

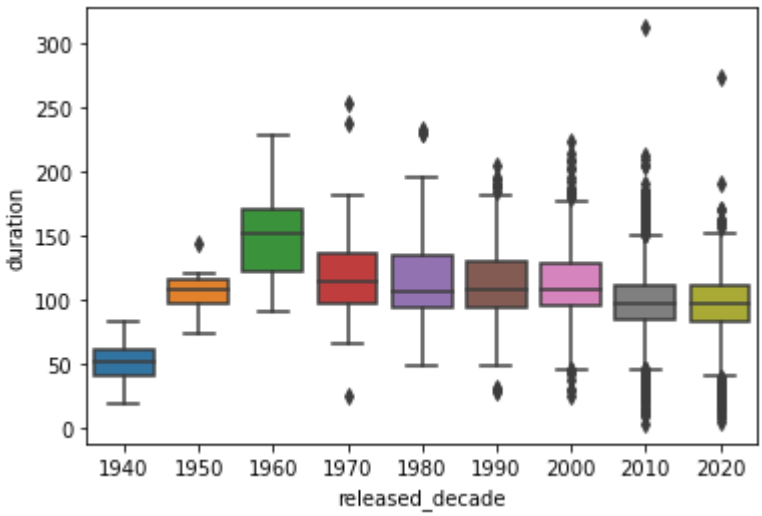


In [131]:

```
1
2 sns.boxplot(data= df, x= "released_decade", y= "duration")
```

Out[131]:

```
<AxesSubplot:xlabel='released_decade', ylabel='duration'>
```

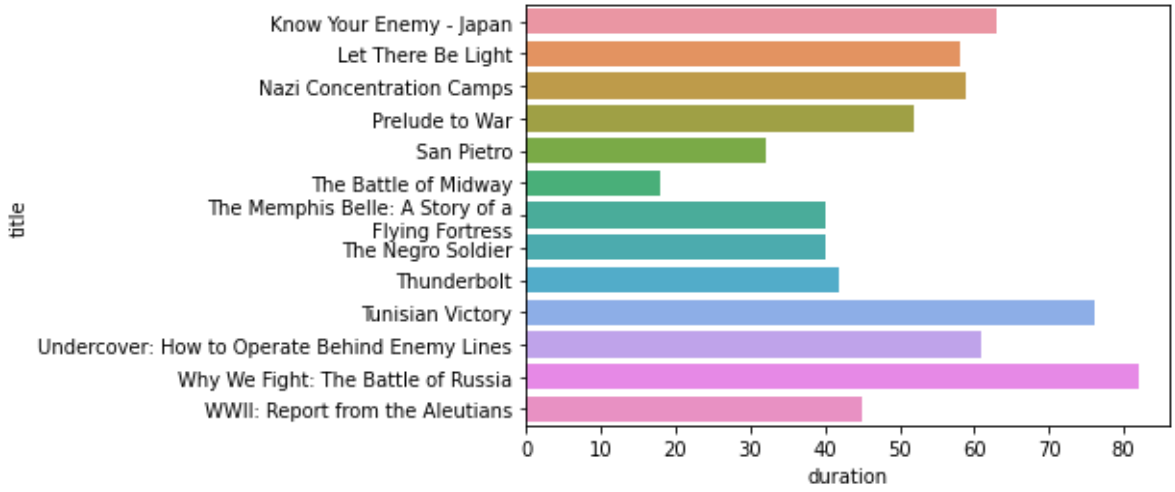


In [132]:

```
1
2 sns.barplot(data= df[df.released_decade == 1940], y= "title", x= "duration")
```

Out[132]:

```
<AxesSubplot:xlabel='duration', ylabel='title'>
```



duration of movies by rating

Observation:

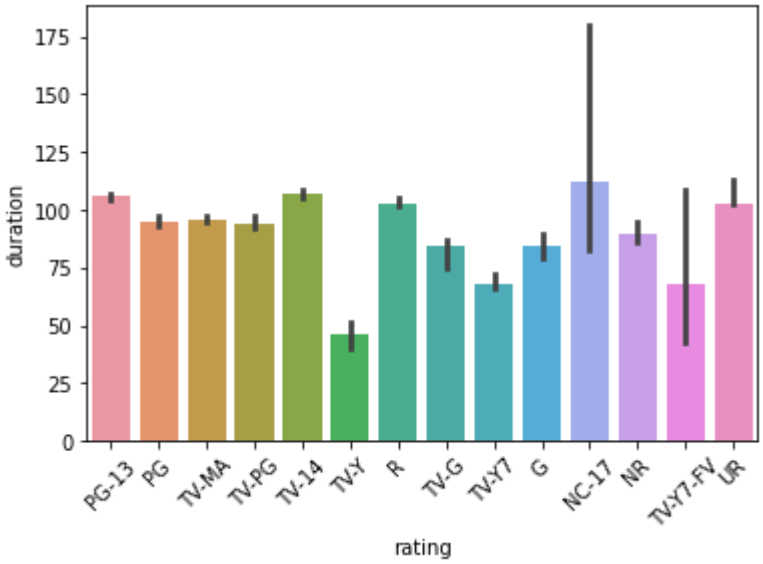
- the movies of rating TV-Y are of shorter duration (for kids maybe)

In [133]:

```
1
2 plt.xticks(rotation = 45)
3 sns.barplot(data= df, x= "rating", y= "duration", estimator= np.median)
```

Out[133]:

<AxesSubplot:xlabel='rating', ylabel='duration'>

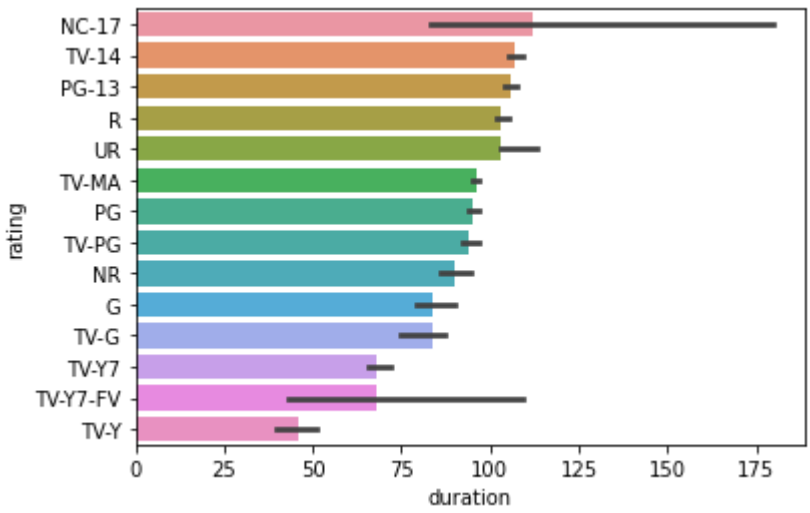


In [134]:

```
1
2 sns.barplot(data= df, y= "rating", x= "duration", estimator= np.median,
3             order = df.groupby(["rating"]).median().sort_values(["duration"], ascending
```

Out[134]:

<AxesSubplot:xlabel='duration', ylabel='rating'>



year added by release yaer

Observation:

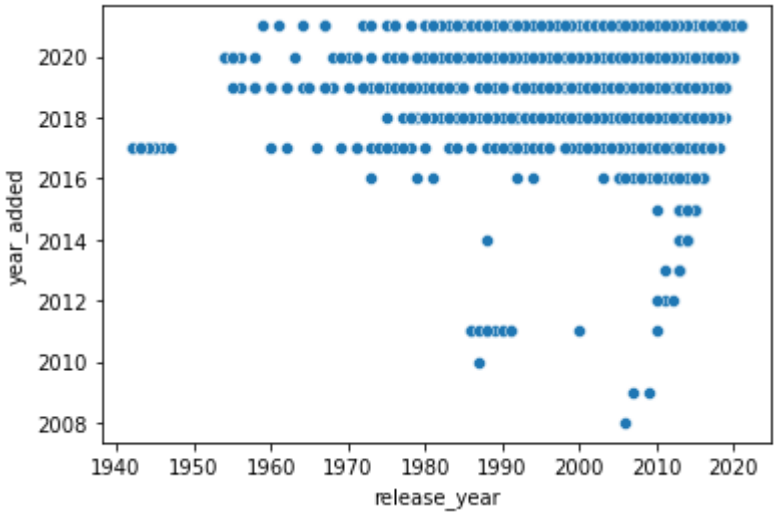
- most movies are added in last few years (after 2018)
- most of the movies that are added released in 2000's and 2010's

In [135]:

```
1
2 sns.scatterplot(data= df, x= "release_year", y= "year_added")
```

Out[135]:

```
<AxesSubplot:xlabel='release_year', ylabel='year_added'>
```



total duration of content added by added day of week

Observation:

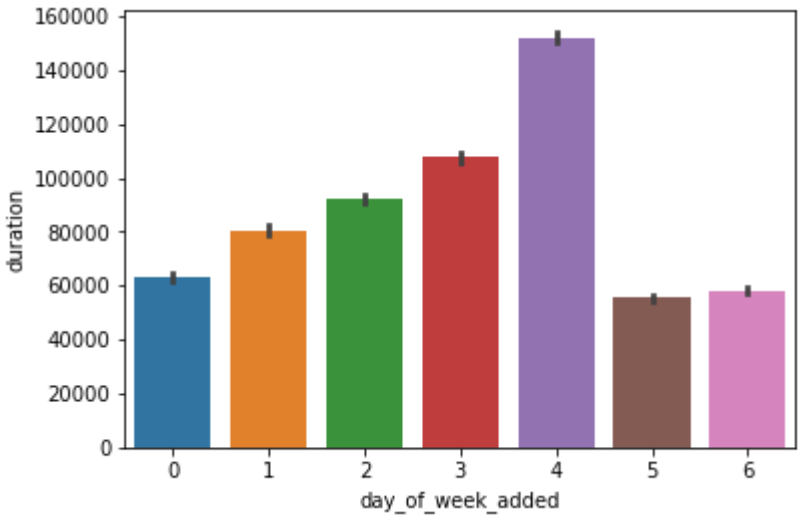
- the total amount of duration of movies added is more on wek days than weekends

In [136]:

```
1
2 sns.barplot(data= df, x= "day_of_week_added", y= "duration", estimator= np.sum)
```

Out[136]:

```
<AxesSubplot:xlabel='day_of_week_added', ylabel='duration'>
```



total duration of content added by added day of week and year

Observation:

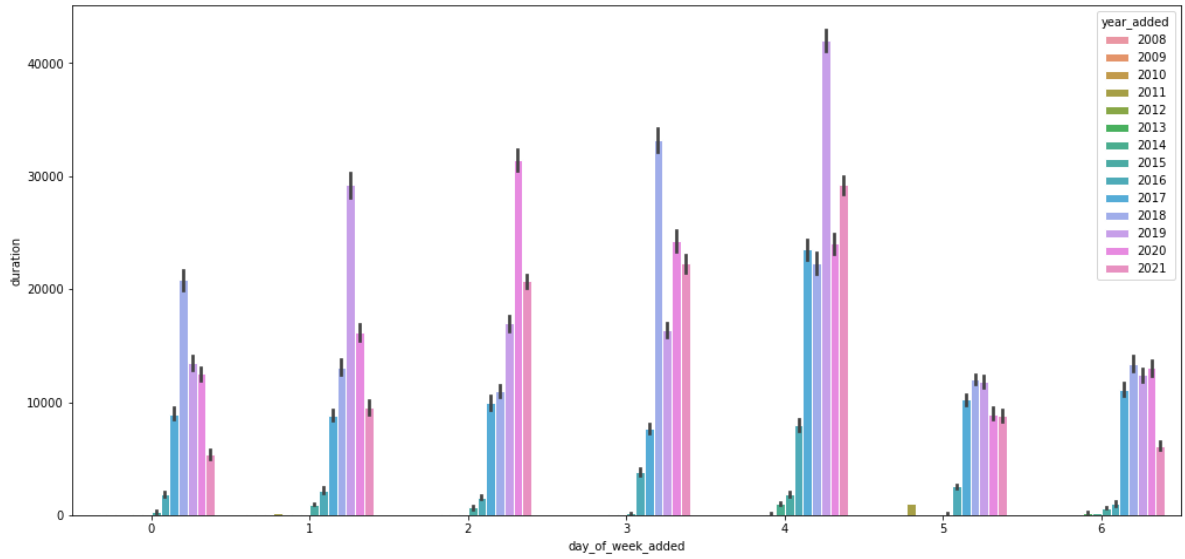
- the total amount of duration of movies added on weekdays had seen a peak in 2018 and then a sudden dop followed by increase

In [137]:

```
1
2 plt.figure(figsize= (17, 8))
3
4 sns.barplot(data= df, x= "day_of_week_added", y= "duration", hue= "year_added", estimat
```

Out[137]:

```
<AxesSubplot:xlabel='day_of_week_added', ylabel='duration'>
```



no. of movies less than 50 min long vs release_decade

Observation:

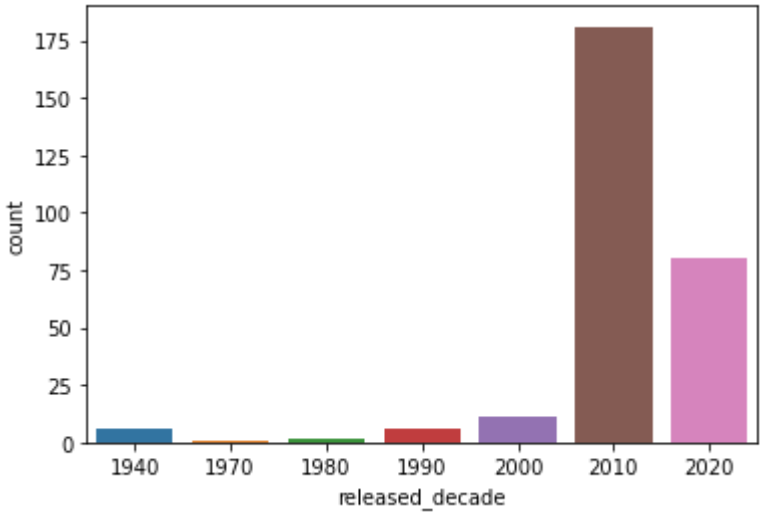
- The no. of moview that are less than 50 min are also belong to the 2010's

In [138]:

```
sns.countplot(data= df[df.duration <= 50], x= "released_decade")
```

Out[138]:

```
<AxesSubplot:xlabel='released_decade', ylabel='count'>
```



total duration vs added day of month

Observation:

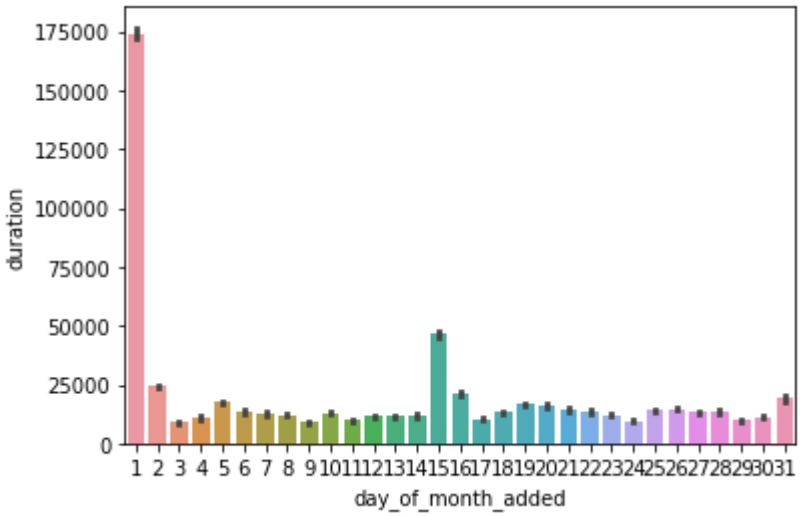
- The total duration of content that was added on start of a given month is significantly higher than that ofany day of month

In [139]:

```
sns.barplot(data= df, x= "day_of_month_added", y= "duration", estimator= np.sum)
```

Out[139]:

```
<AxesSubplot:xlabel='day_of_month_added', ylabel='duration'>
```



release_year vs rating

Observation:

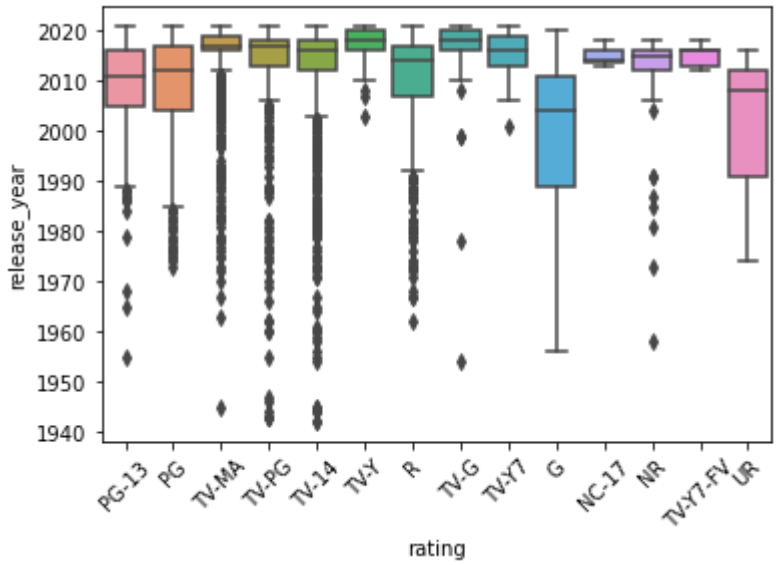
- The relese_year is independent of the rationg of the movie

In [140]:

```
1
2 plt.xticks(rotation= 45)
3 sns.boxplot(data= df, x= "rating", y= "release_year")
4
```

Out[140]:

<AxesSubplot:xlabel='rating', ylabel='release_year'>

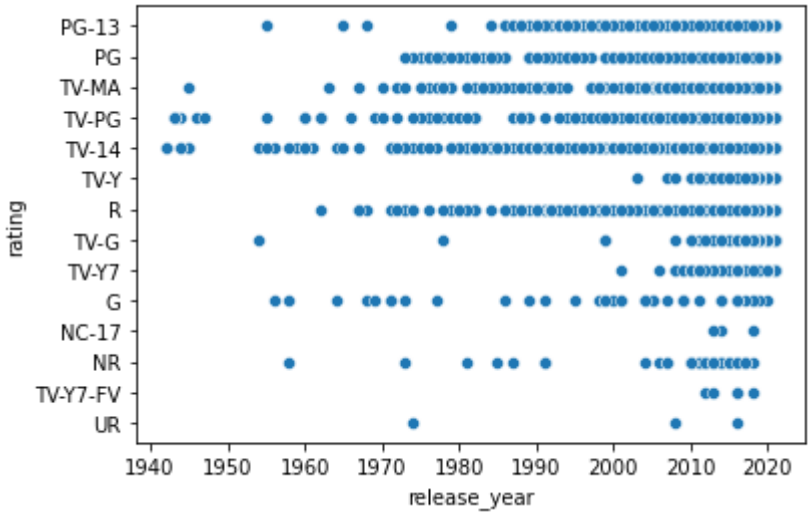


In [141]:

```
1
2 sns.scatterplot(data= df, y= "rating", x= "release_year")
```

Out[141]:

<AxesSubplot:xlabel='release_year', ylabel='rating'>



added day of month/ week vs rating

Observation:

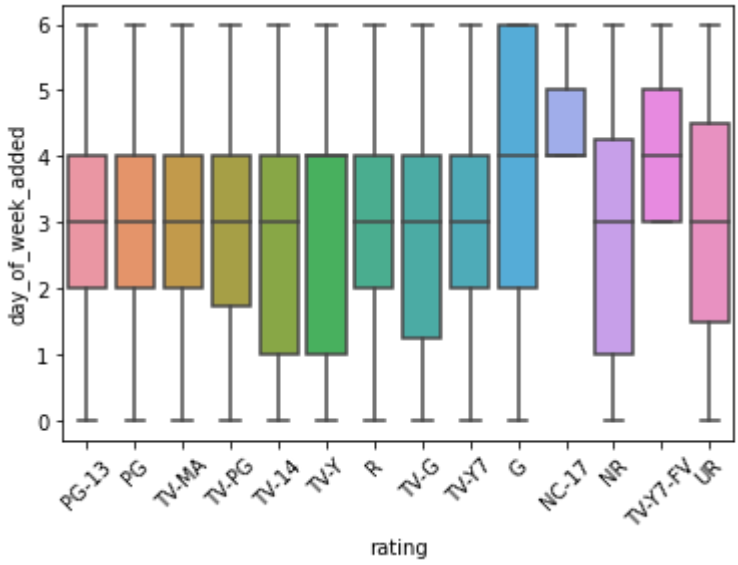
- the added day of week or month is independent of the rating of the movie

In [142]:

```
1
2 plt.xticks(rotation= 45)
3 sns.boxplot(data= df, x= "rating", y= "day_of_week_added")
```

Out[142]:

```
<AxesSubplot:xlabel='rating', ylabel='day_of_week_added'>
```

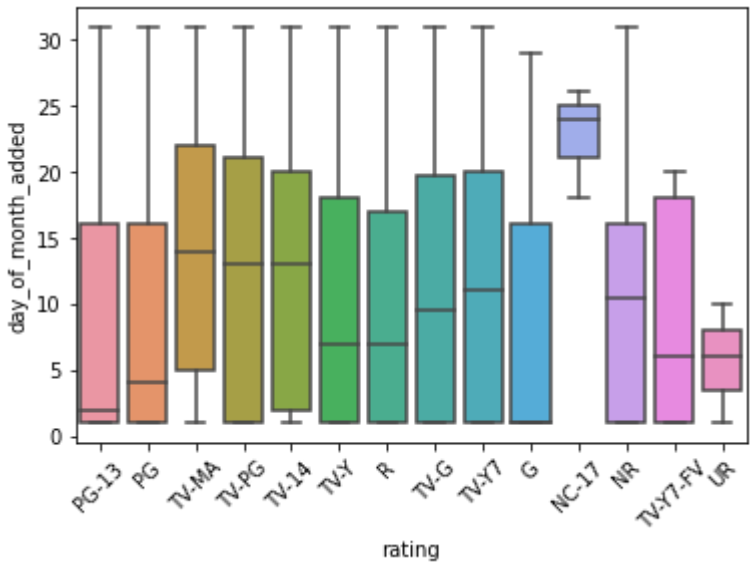


In [143]:

```
1
2 plt.xticks(rotation= 45)
3 sns.boxplot(data= df, x= "rating", y= "day_of_month_added")
```

Out[143]:

```
<AxesSubplot:xlabel='rating', ylabel='day_of_month_added'>
```



distribution of duration vs min_age

Observation:

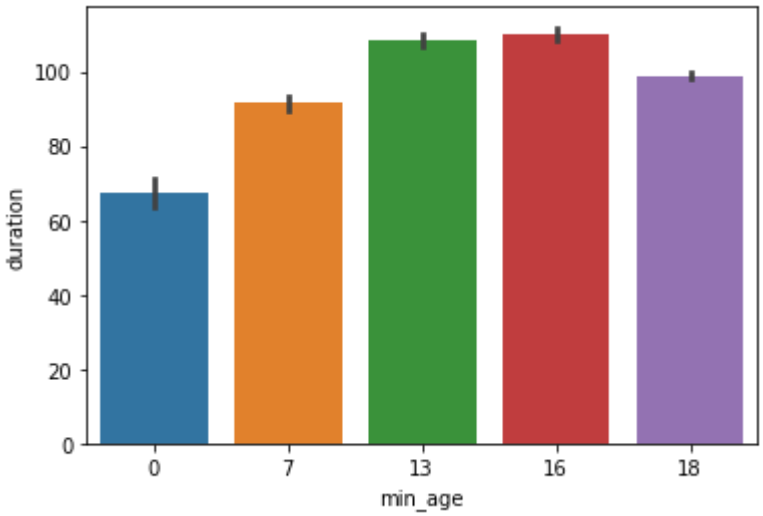
- the mean duration of the movies for 16+, 13+, 18+ is longer than for other age groups
- the duration for the genreic age group (0+) is lower than for any other age group
- the total duration of content is the highest for 18+, 16+ and 7+ age groups

In [144]:

```
1
2 sns.barplot(data= df, x= "min_age", y= "duration", estimator= np.mean)
```

Out[144]:

<AxesSubplot:xlabel='min_age', ylabel='duration'>

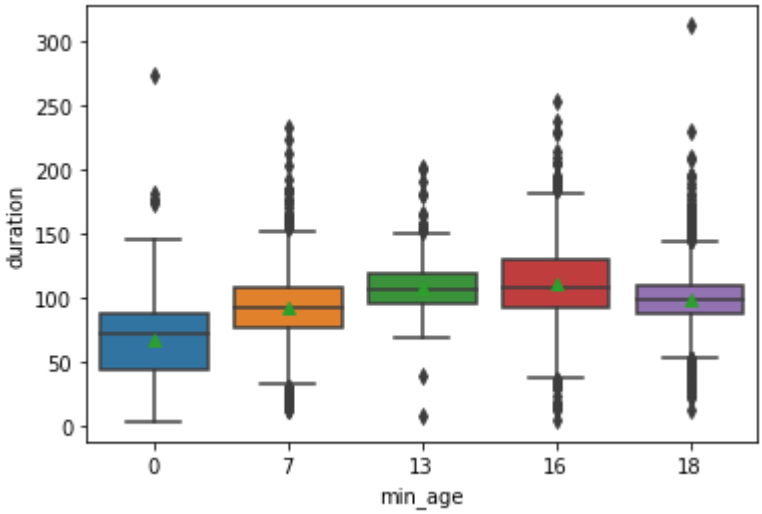


In [145]:

```
1
2 sns.boxplot(data= df, x= "min_age", y= "duration", showmeans= True)
```

Out[145]:

<AxesSubplot:xlabel='min_age', ylabel='duration'>

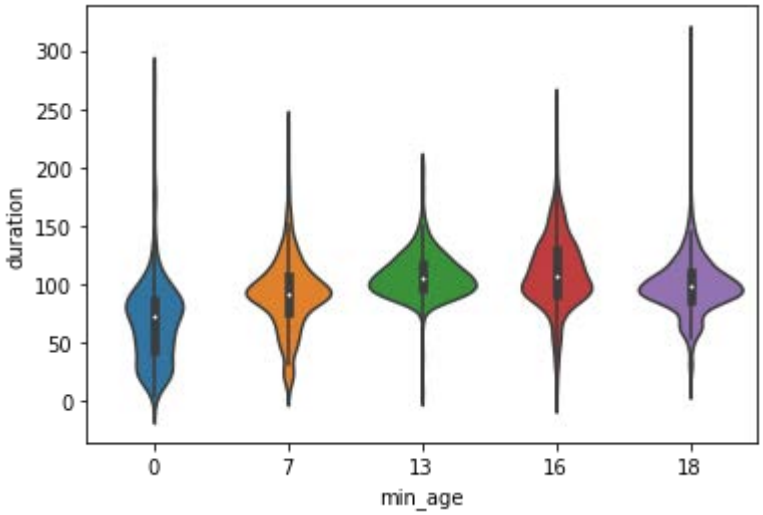


In [146]:

```
1
2 sns.violinplot(data= df, x= "min_age", y= "duration", showmeans= True)
```

Out[146]:

<AxesSubplot:xlabel='min_age', ylabel='duration'>

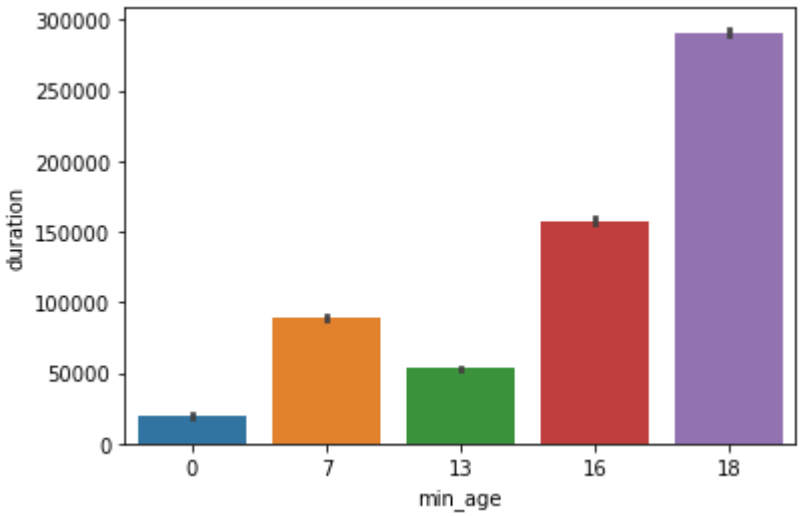


In [147]:

```
1
2 sns.barplot(data= df, x= "min_age", y= "duration", estimator= np.sum)
```

Out[147]:

<AxesSubplot:xlabel='min_age', ylabel='duration'>



distribution of release_year/ added year/ day of week/ month vs min_age

Observation:

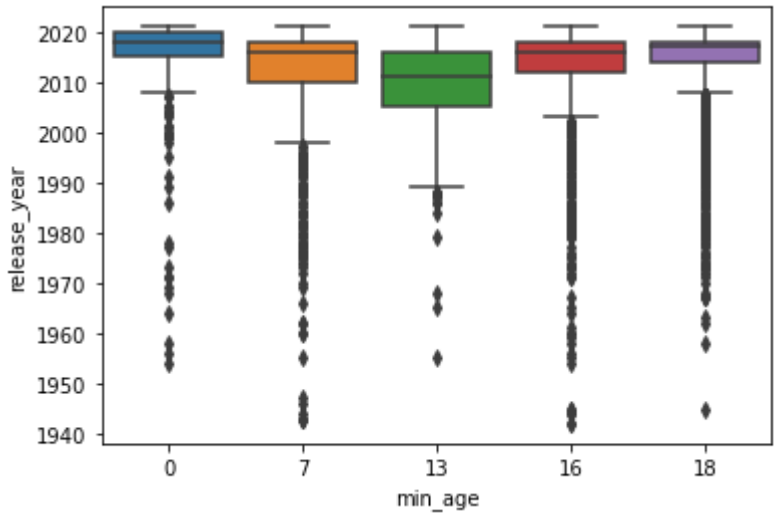
- The release_year/ added year are independent of min_age
- The day of week added id independent for age groups 0+, 7+, 13+ but tends to be on Thur, Friday for 16+ and 18+ age groups

In [148]:

```
1
2 sns.boxplot(data= df, x= "min_age", y= "release_year")
```

Out[148]:

```
<AxesSubplot:xlabel='min_age', ylabel='release_year'>
```

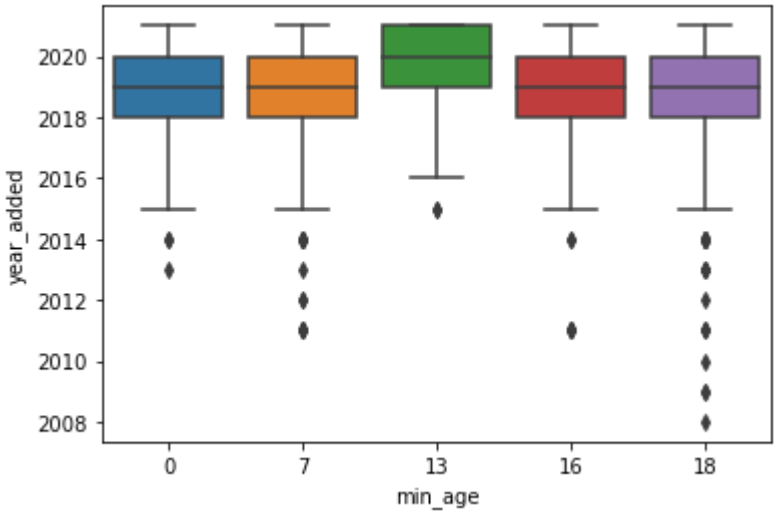


In [149]:

```
1
2 sns.boxplot(data= df, x= "min_age", y= "year_added")
```

Out[149]:

```
<AxesSubplot:xlabel='min_age', ylabel='year_added'>
```

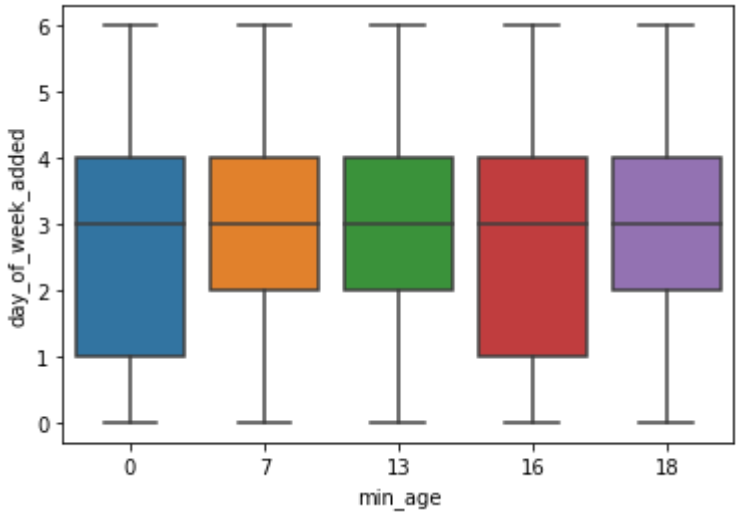


In [150]:

```
1
2 sns.boxplot(data= df, x= "min_age", y= "day_of_week_added")
```

Out[150]:

<AxesSubplot:xlabel='min_age', ylabel='day_of_week_added'>

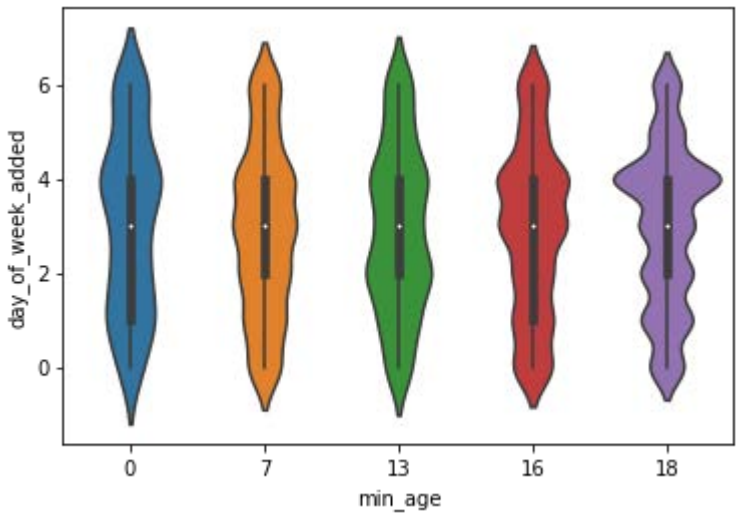


In [151]:

```
1
2 sns.violinplot(data= df, x= "min_age", y= "day_of_week_added")
```

Out[151]:

<AxesSubplot:xlabel='min_age', ylabel='day_of_week_added'>

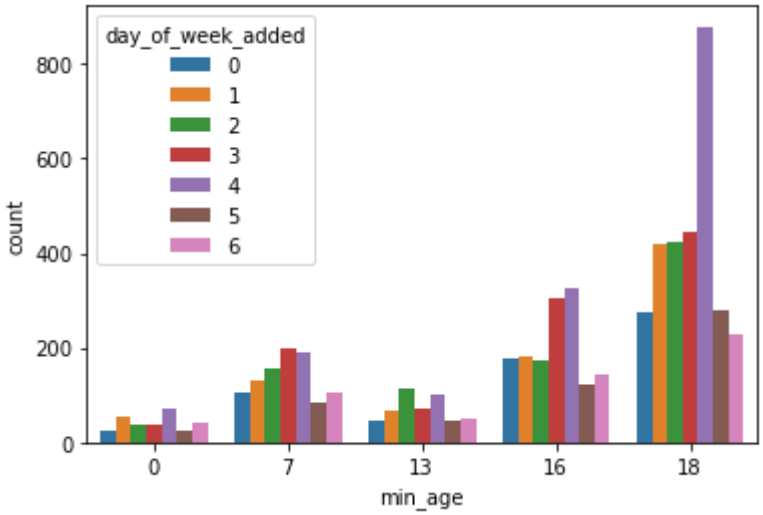


In [152]:

```
1
2 sns.countplot(data= df, x= "min_age", hue= "day_of_week_added")
```

Out[152]:

```
<AxesSubplot:xlabel='min_age', ylabel='count'>
```

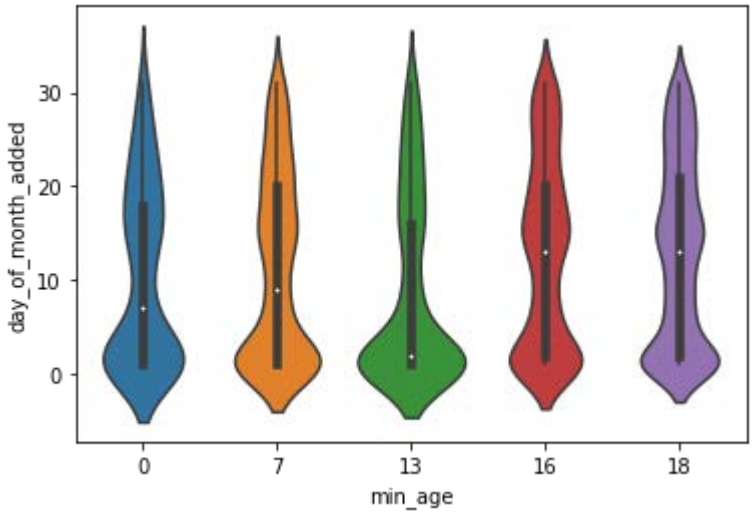


In [153]:

```
1
2 sns.violinplot(data= df, x= "min_age", y= "day_of_month_added")
```

Out[153]:

```
<AxesSubplot:xlabel='min_age', ylabel='day_of_month_added'>
```



Movie Data Analysis: nested included

- The following analysis includes the columns "cast", "director", "country", "listed_in"

In [154]:

```
1
2 netflix_data_full_movies = netflix_data_full[netflix_data_full.type == "Movie"].copy()
3
4 netflix_data_full_movies.head()
```

Out[154]:

	show_id	type	title	director	cast	country	date_added	release_year	ratin
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anonymous	United States	2021-09-25	2020	PG 1
159	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Vanessa Hudgens	United States	2021-09-24	2021	P(
160	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Kimiko Glenn	United States	2021-09-24	2021	P(
161	s7	Movie	My Little Pony: A New Generation	Robert Cullen	James Marsden	United States	2021-09-24	2021	P(
162	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Sofia Carson	United States	2021-09-24	2021	P(

In [155]:

```
1
2 df = netflix_data_full_movies.copy()
```

In [156]:

```
df["release_decade"] = df["release_year"].apply(lambda x: x - (x%10))

df.head()
```

Out[156]:

	show_id	type	title	director	cast	country	date_added	release_year	ratin
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anonymous	United States	2021-09-25	2020	PG 1
159	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Vanessa Hudgens	United States	2021-09-24	2021	P
160	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Kimiko Glenn	United States	2021-09-24	2021	P
161	s7	Movie	My Little Pony: A New Generation	Robert Cullen	James Marsden	United States	2021-09-24	2021	P
162	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Sofia Carson	United States	2021-09-24	2021	P

In [157]:

```
df.columns = ['show_id', 'type', 'title', 'directors', 'actors', 'country', 'date_added
              'release_year', 'rating', 'duration', 'genres', 'min_age', 'release_decad

df.columns
```

Out[157]:

```
Index(['show_id', 'type', 'title', 'directors', 'actors', 'country',
       'date_added', 'release_year', 'rating', 'duration', 'genres', 'min_ag
e',
       'release_decade'],
      dtype='object')
```

Univariate analysis

In [158]:

```
test_df = df.copy(); test_df["count"] = 1

test_df.groupby("count").nunique()[["show_id", "title", "directors", "actors", "country
                                    "release_year", "genres", "release_decade"]]
```

Out[158]:

count	show_id	title	directors	actors	country	release_year	genres	release_decade
1	6131	6131	4778	25952	122	73	20	9

Bivariate Analysis

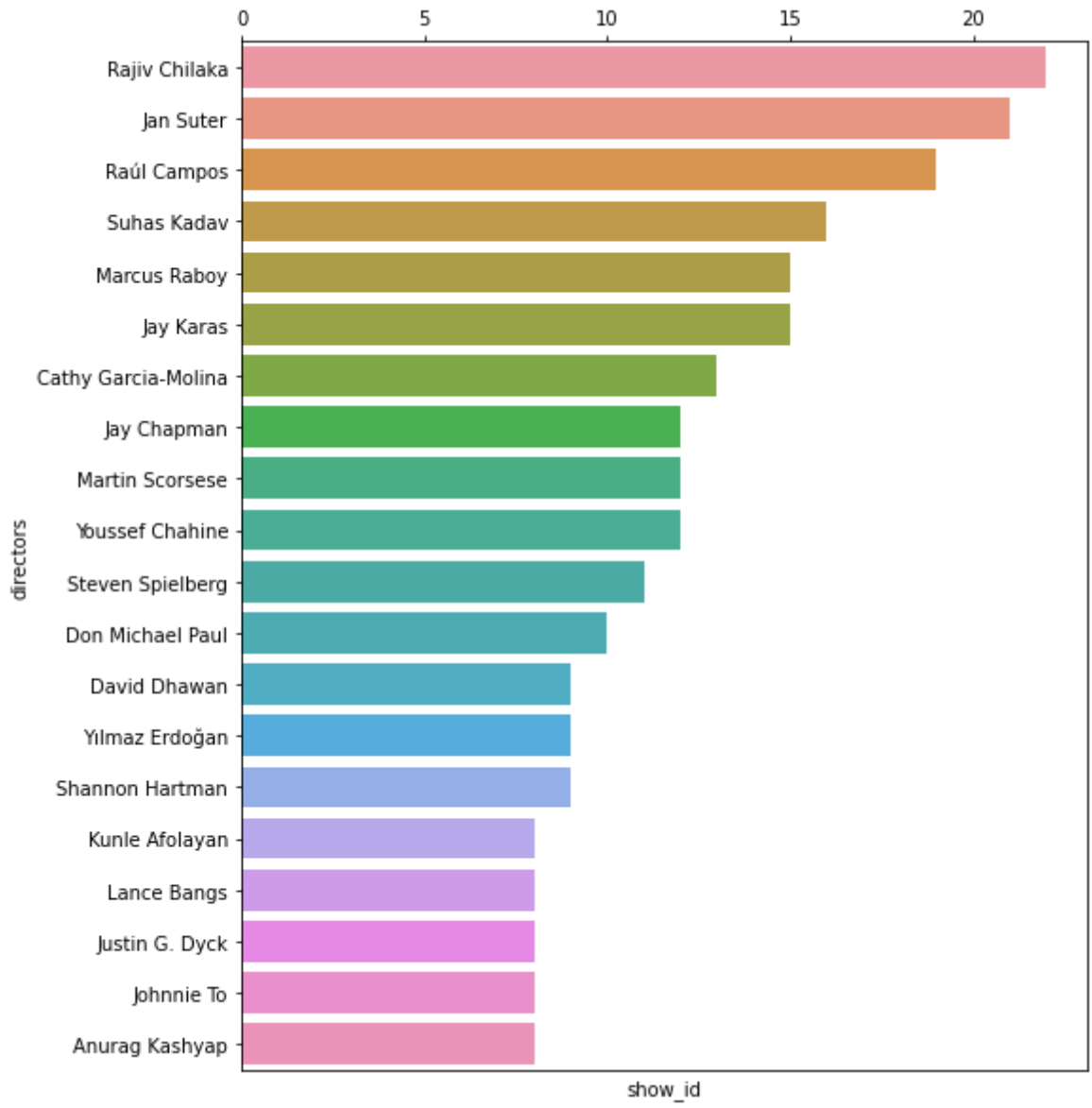
no. of movies vs directors

Observation:

- There are only a handful of directors (among 4167) who have done more than 10 movies

In [159]:

```
1
2 plt_df = df.loc[df.directors != "Anonymous"].groupby(["directors"]).nunique().reset_inc
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "directors", x= "show_id")
7
8 ax.xaxis.tick_top()
```



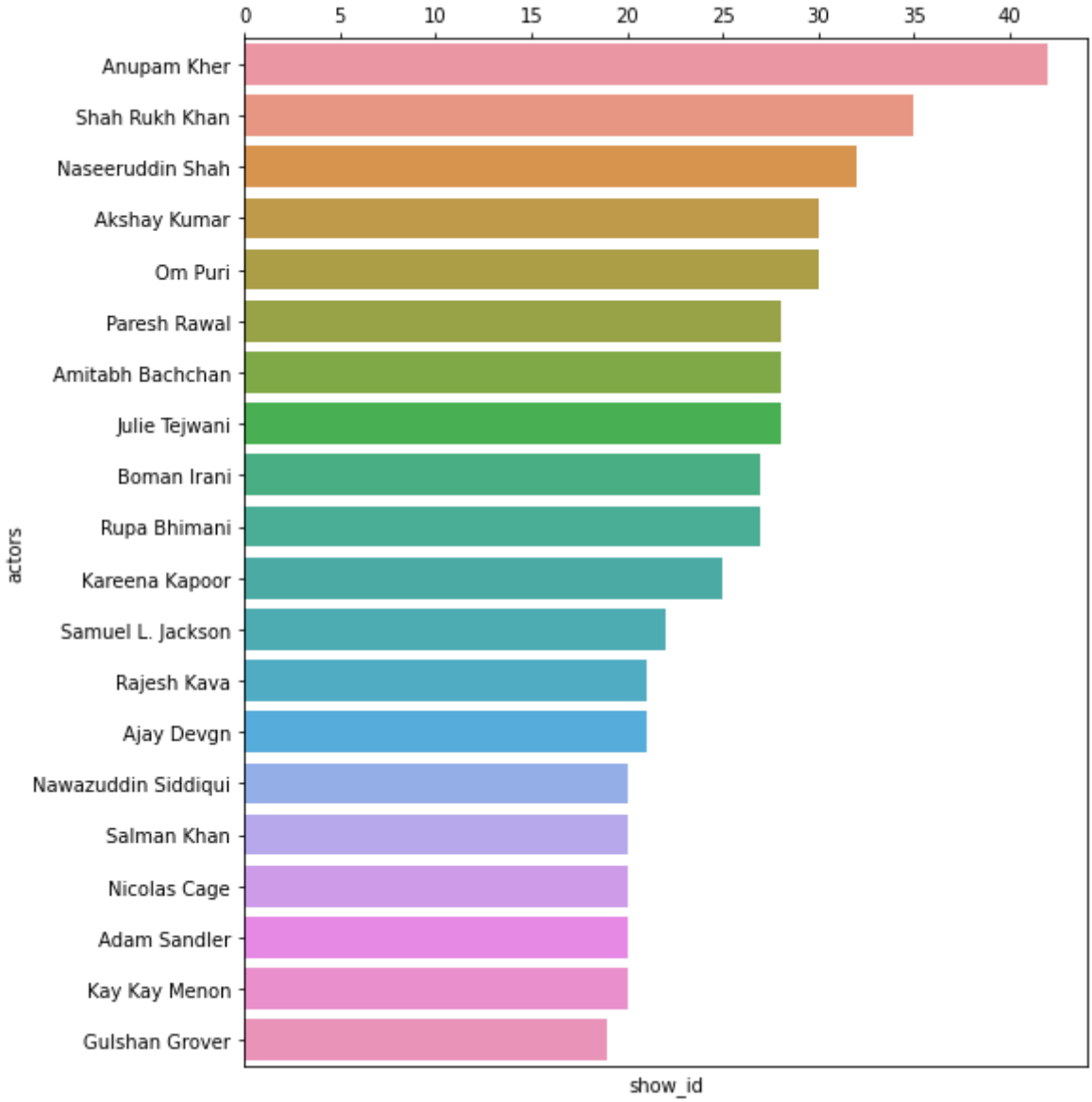
no. of movies per actor

Observation:

- There are only a handful of actors (among 24000) who have done more than 20 movies

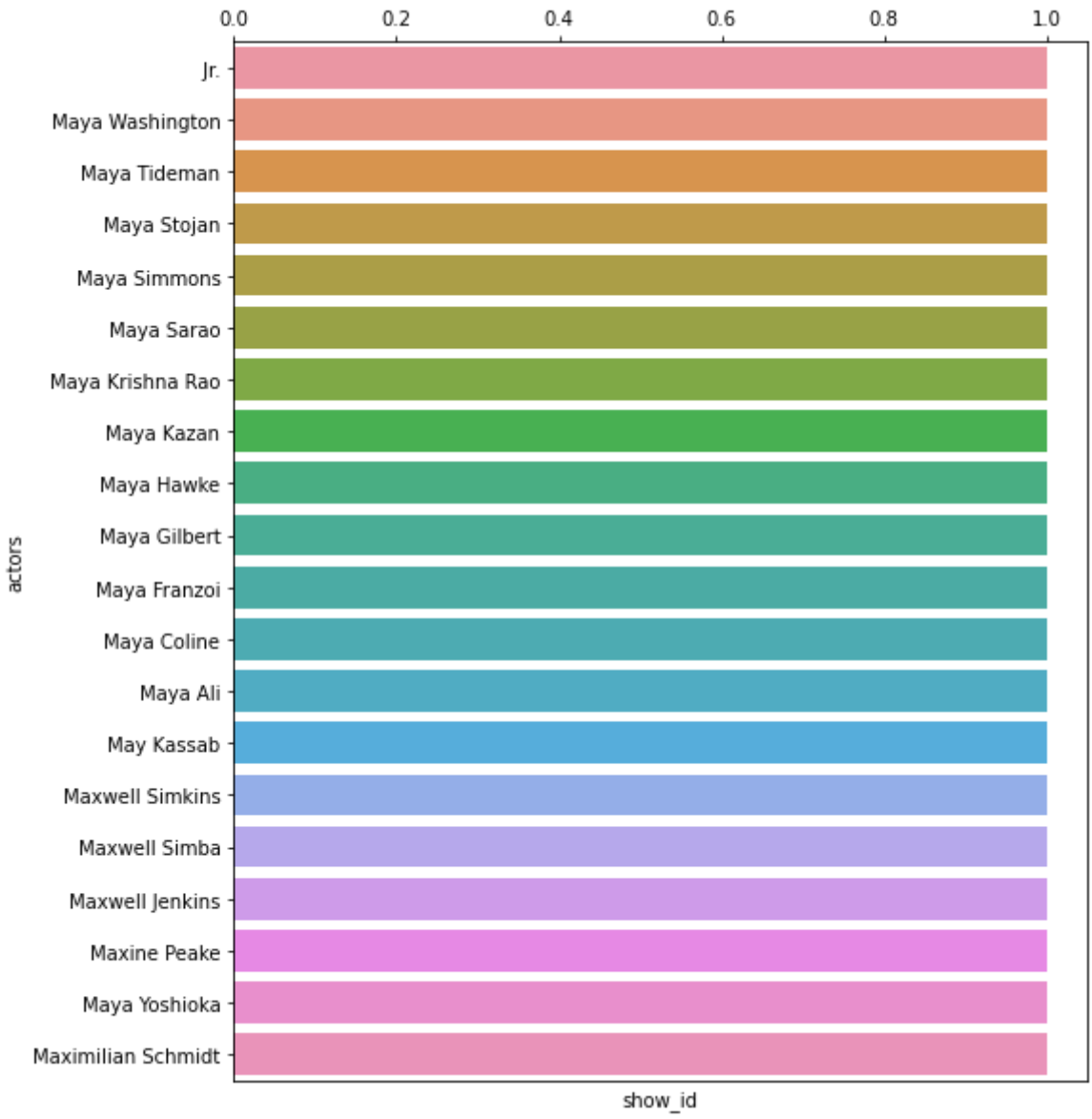
In [160]:

```
1
2 plt_df = df.loc[df.actors != "Anonymous"].groupby(["actors"]).nunique().reset_index().s
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "actors", x= "show_id")
7
8 ax.xaxis.tick_top()
```



In [161]:

```
plt_df = df.loc[df.actors != "Anonymous"].groupby(["actors"]).nunique().reset_index().s

plt.figure(figsize= (8, 10))

ax = sns.barplot(data= plt_df, y= "actors", x= "show_id")

ax.xaxis.tick_top()
```



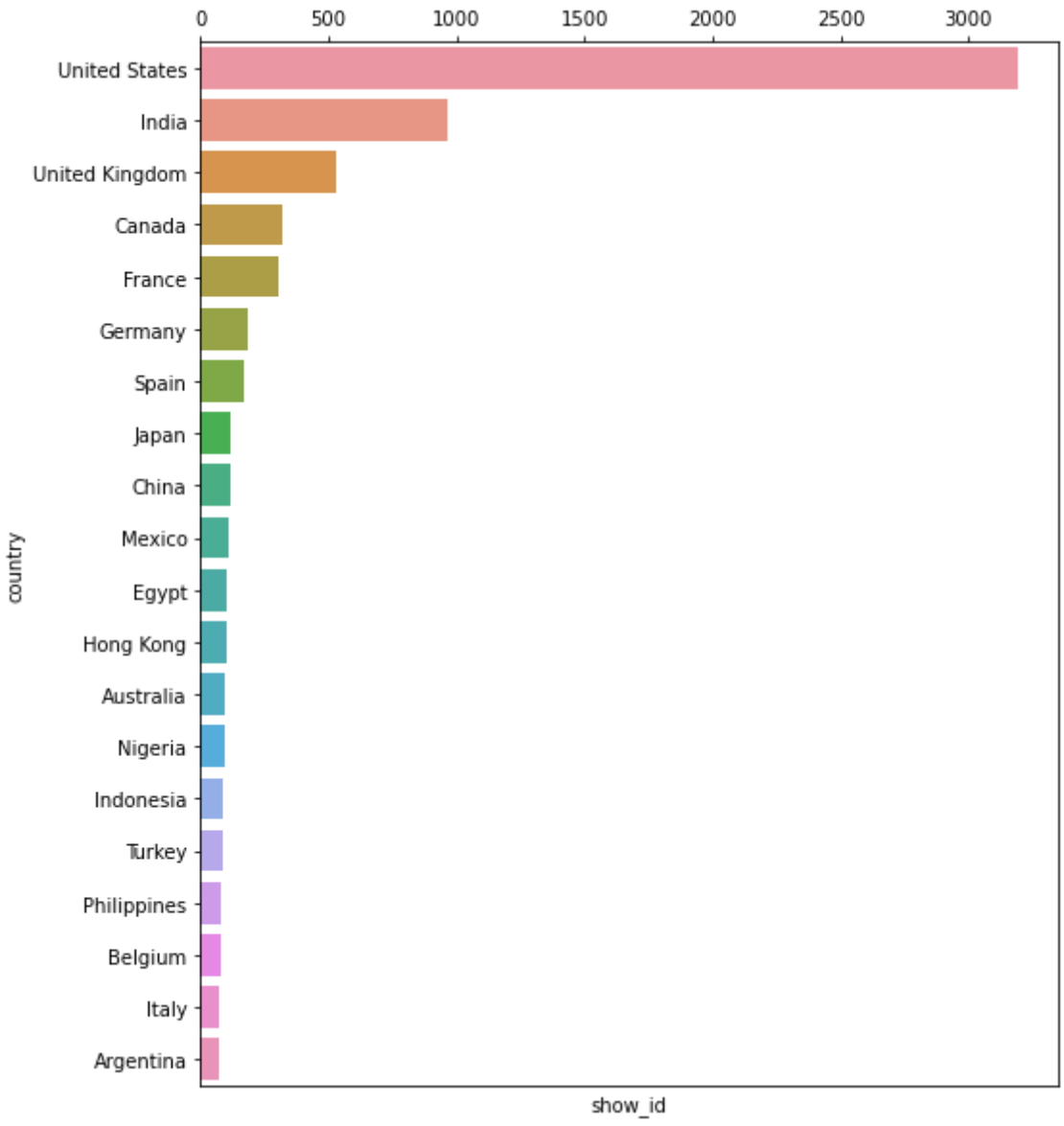
no. of movies per country

Observation:

- Most movies are availabe only for US, India, UK
- Ethiopia, Latvia, Jamaica and few other countries have only one movie streaming

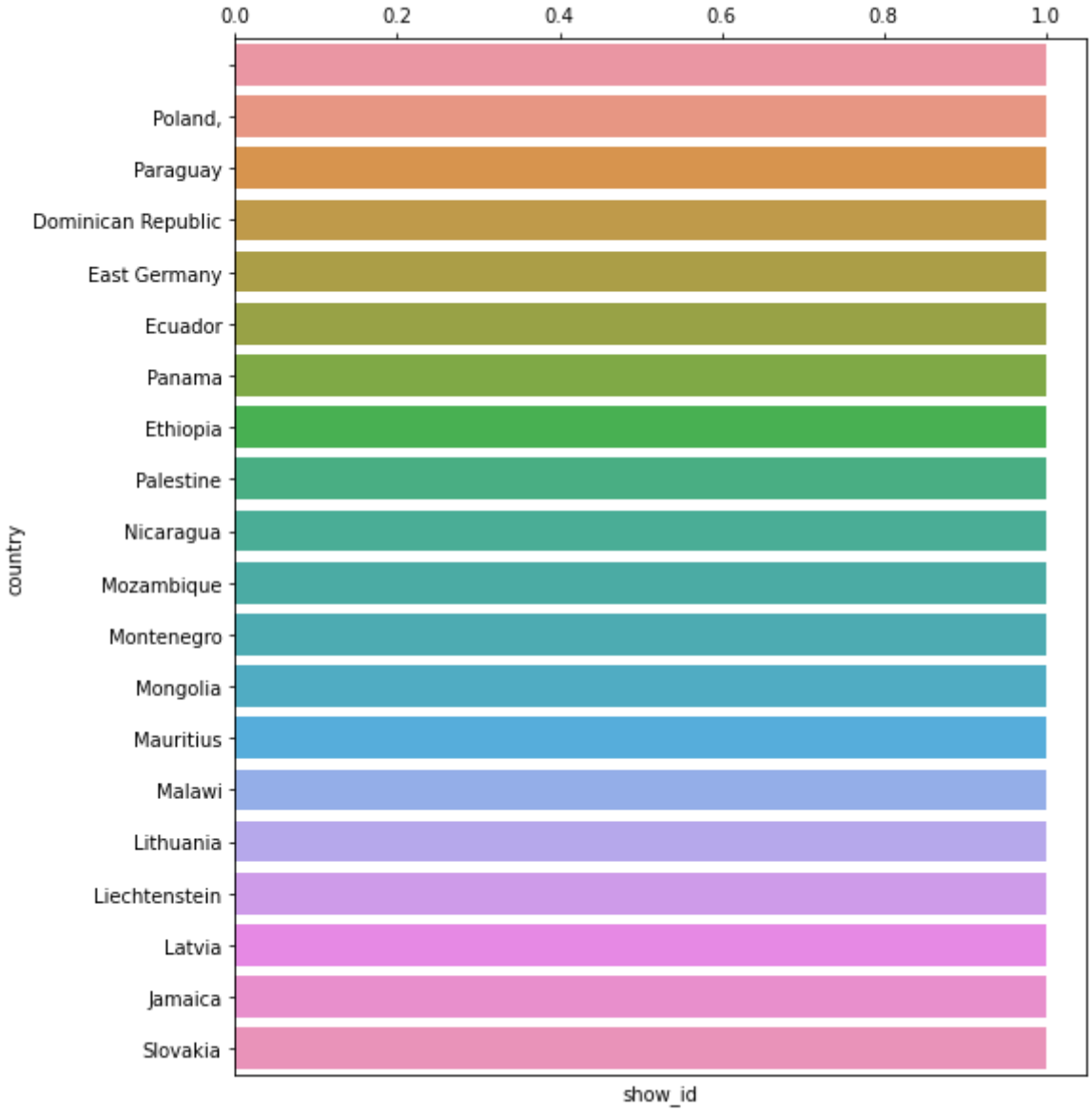
In [162]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["show_id"], ascer
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "show_id")
7
8 ax.xaxis.tick_top()
```



In [163]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["show_id"], ascer
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "show_id")
7
8 ax.xaxis.tick_top()
```



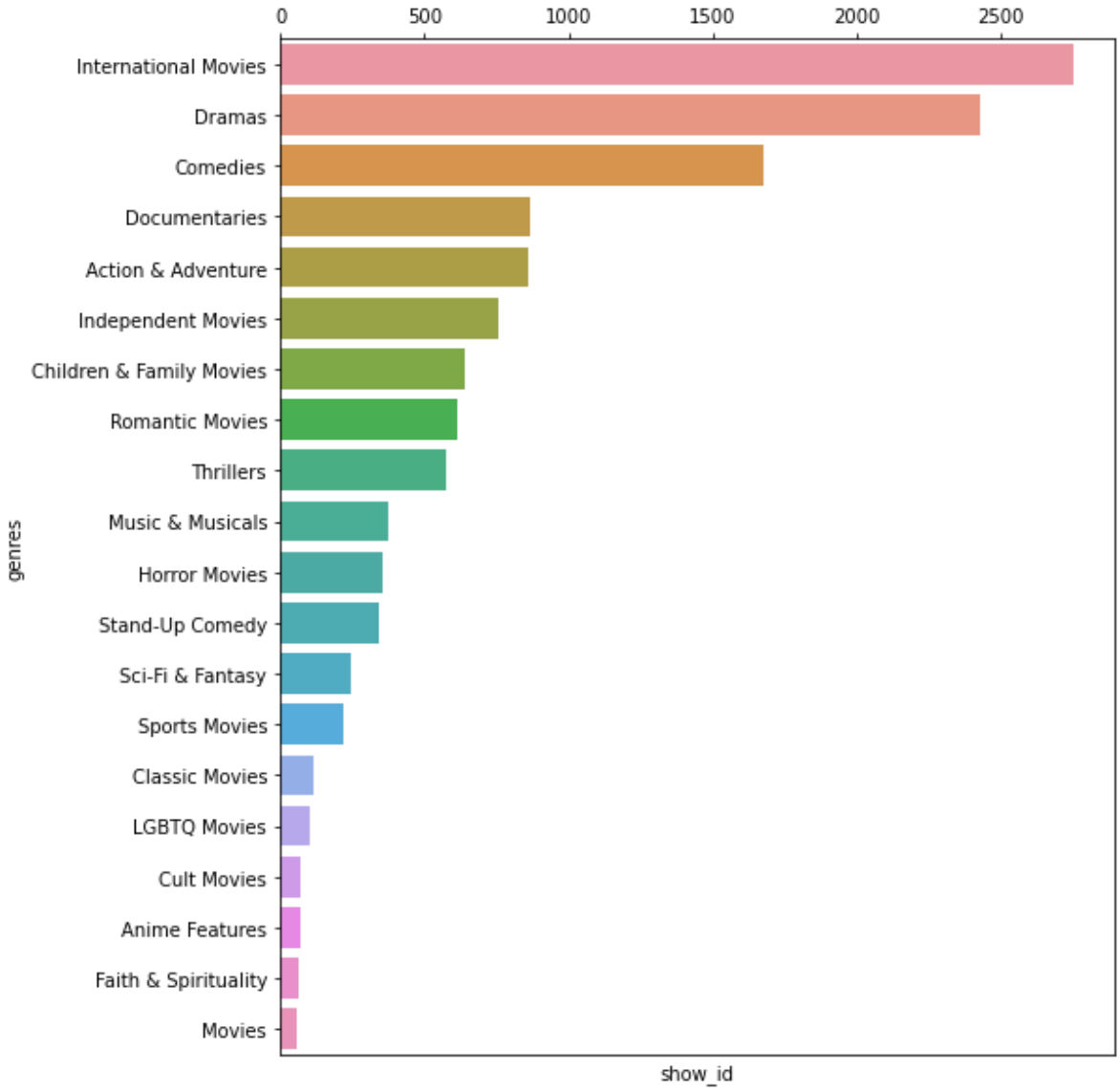
no. of movies per genre

Observation:

- Most movies belong to the genre of International, Dramas, Comedies
- Very few movies belong to the genre of Classic, Cult and Anime

In [164]:

```
1
2 plt_df = df.groupby(["genres"]).nunique().reset_index().sort_values(["show_id"], ascend
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "genres", x= "show_id")
7
8 ax.xaxis.tick_top()
```



In [165]:

```
1
2 df.genres.unique()
```

Out[165]:

```
array(['Documentaries', 'Children & Family Movies', 'Dramas',
       'Independent Movies', 'International Movies', 'Comedies',
       'Thrillers', 'Romantic Movies', 'Music & Musicals',
       'Horror Movies', 'Sci-Fi & Fantasy', 'Action & Adventure',
       'Classic Movies', 'Anime Features', 'Sports Movies', 'Cult Movies',
       'Faith & Spirituality', 'LGBTQ Movies', 'Stand-Up Comedy',
       'Movies'], dtype=object)
```

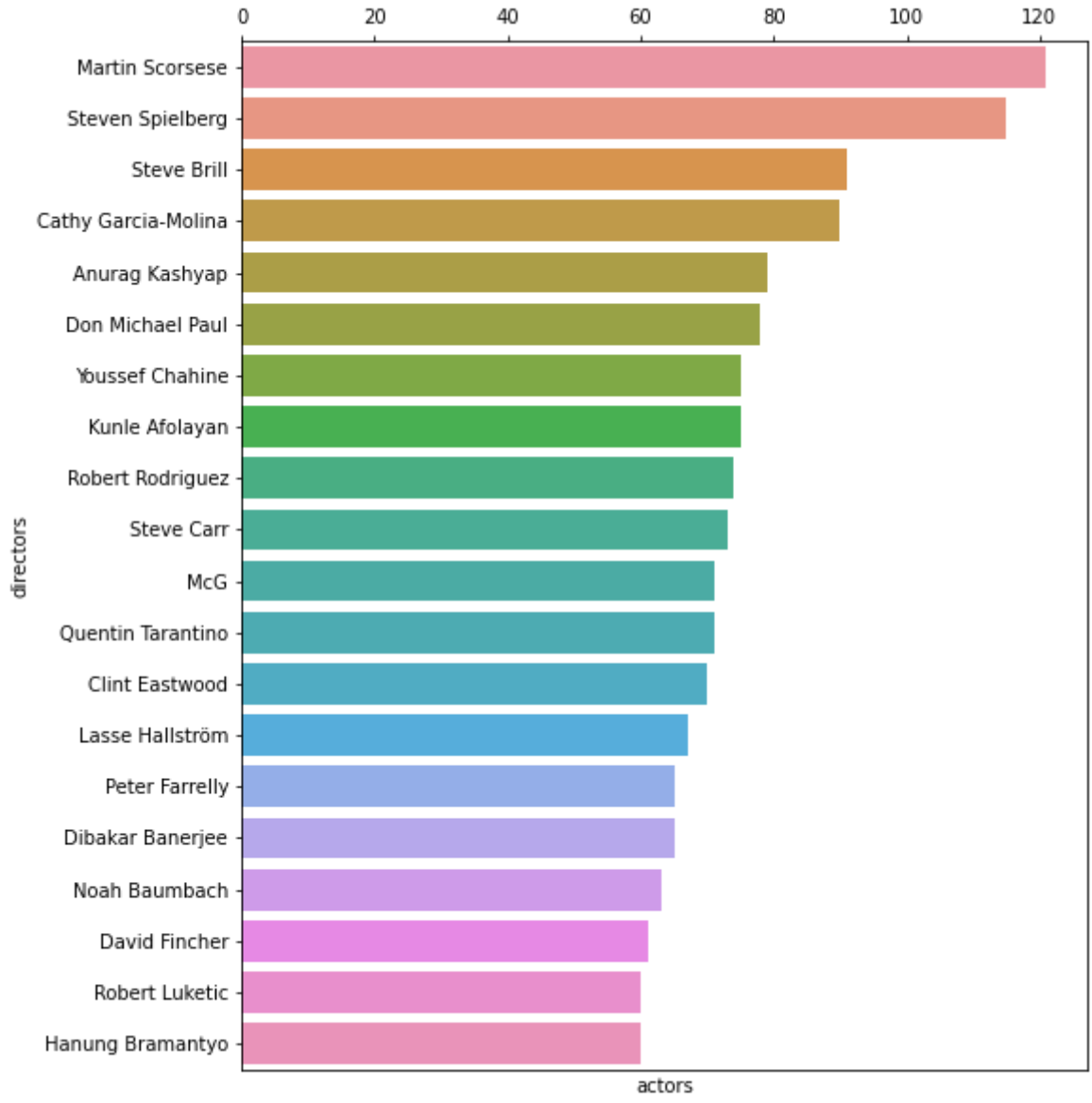
no. of actors worked with per director

Observation:

- Around 20-30 directors (among 4000) worked with more than 60 actors (among 24000)

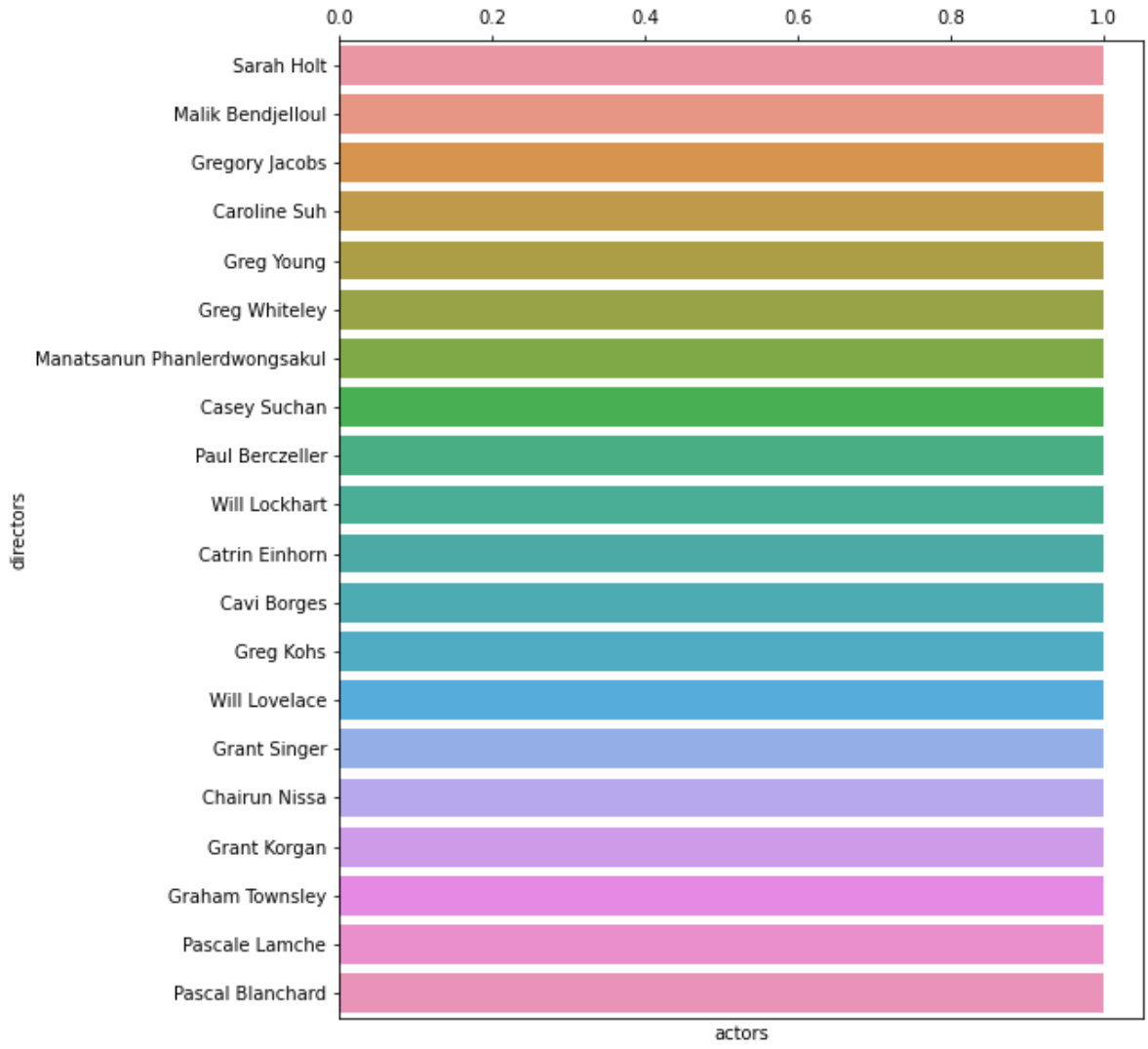
In [166]:

```
1
2 plt_df = df.loc[df.directors != "Anonymous"].groupby(["directors"]).nunique().reset_inc
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "directors", x= "actors")
7
8 ax.xaxis.tick_top()
```



In [167]:

```
1
2 plt_df = df.loc[df.directors != "Anonymous"].groupby(["directors"]).nunique().reset_inc
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "directors", x= "actors")
7
8 ax.xaxis.tick_top()
```



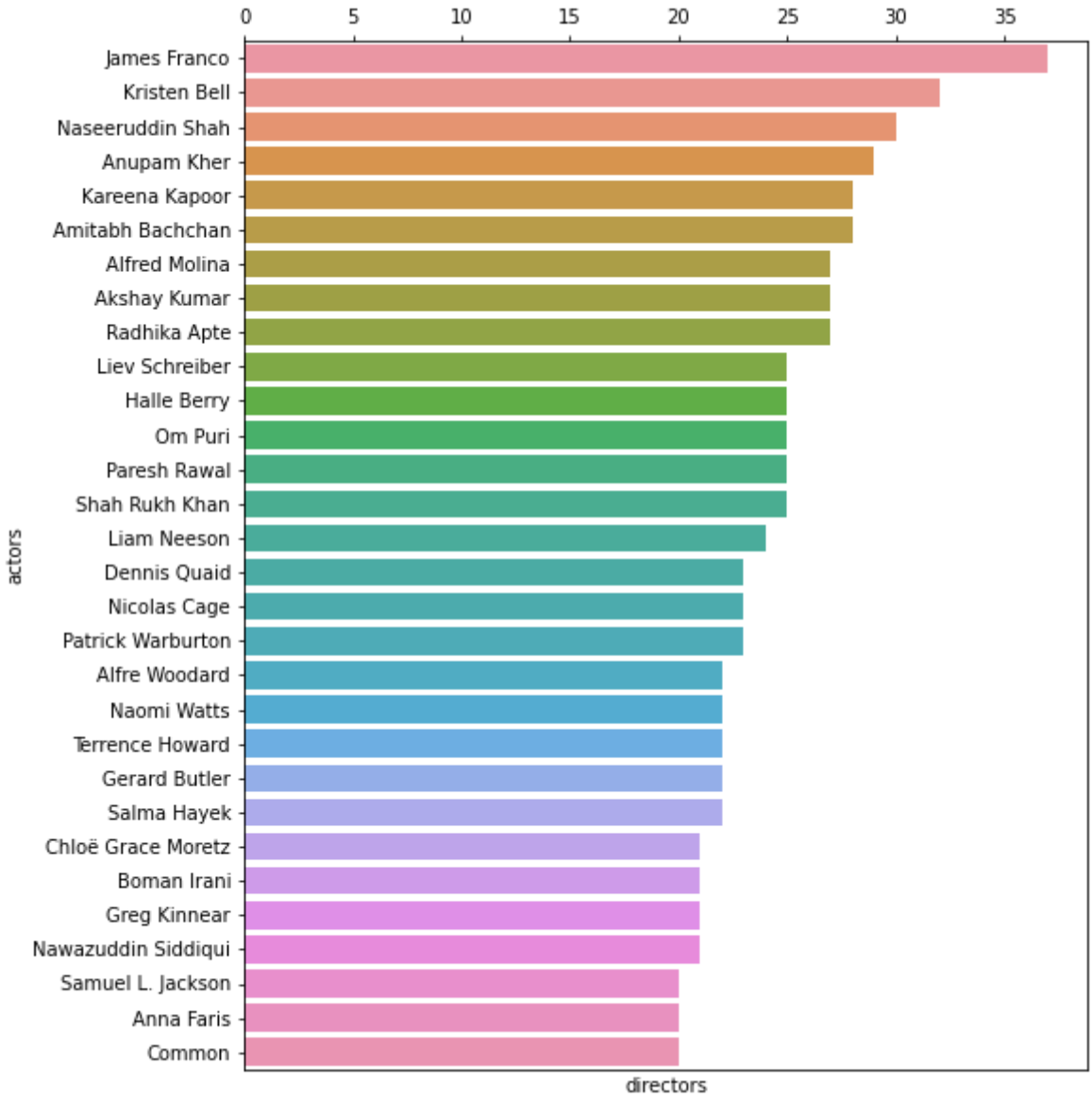
no. of directors worked with per actor

Observation:

- Around 20-30 actors (among 24000) worked with more than 20 directors (among 4000)

In [168]:

```
1
2 plt_df = df.loc[df.actors != "Anonymous"].groupby(["actors"]).nunique().reset_index().s
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "actors", x= "directors")
7
8 ax.xaxis.tick_top()
```



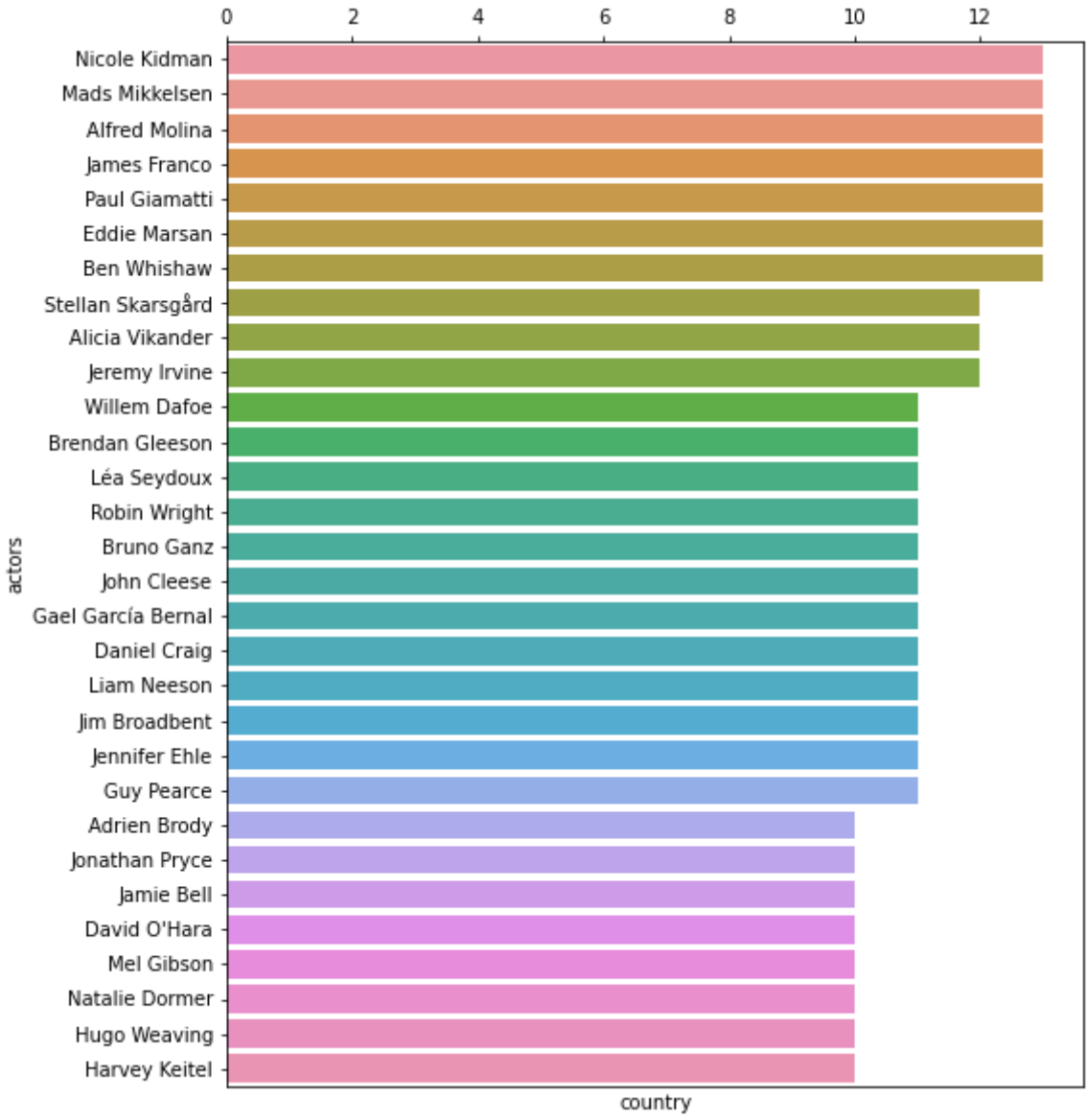
no. of countries streaming per actor

Observation:

- Around 20-30 actors (among 24000) are streaming (popular) in pore than 10 countries

In [169]:

```
1
2 plt_df = df.loc[df.actors != "Anonymous"].groupby(["actors"]).nunique().reset_index().s
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "actors", x= "country")
7
8 ax.xaxis.tick_top()
```



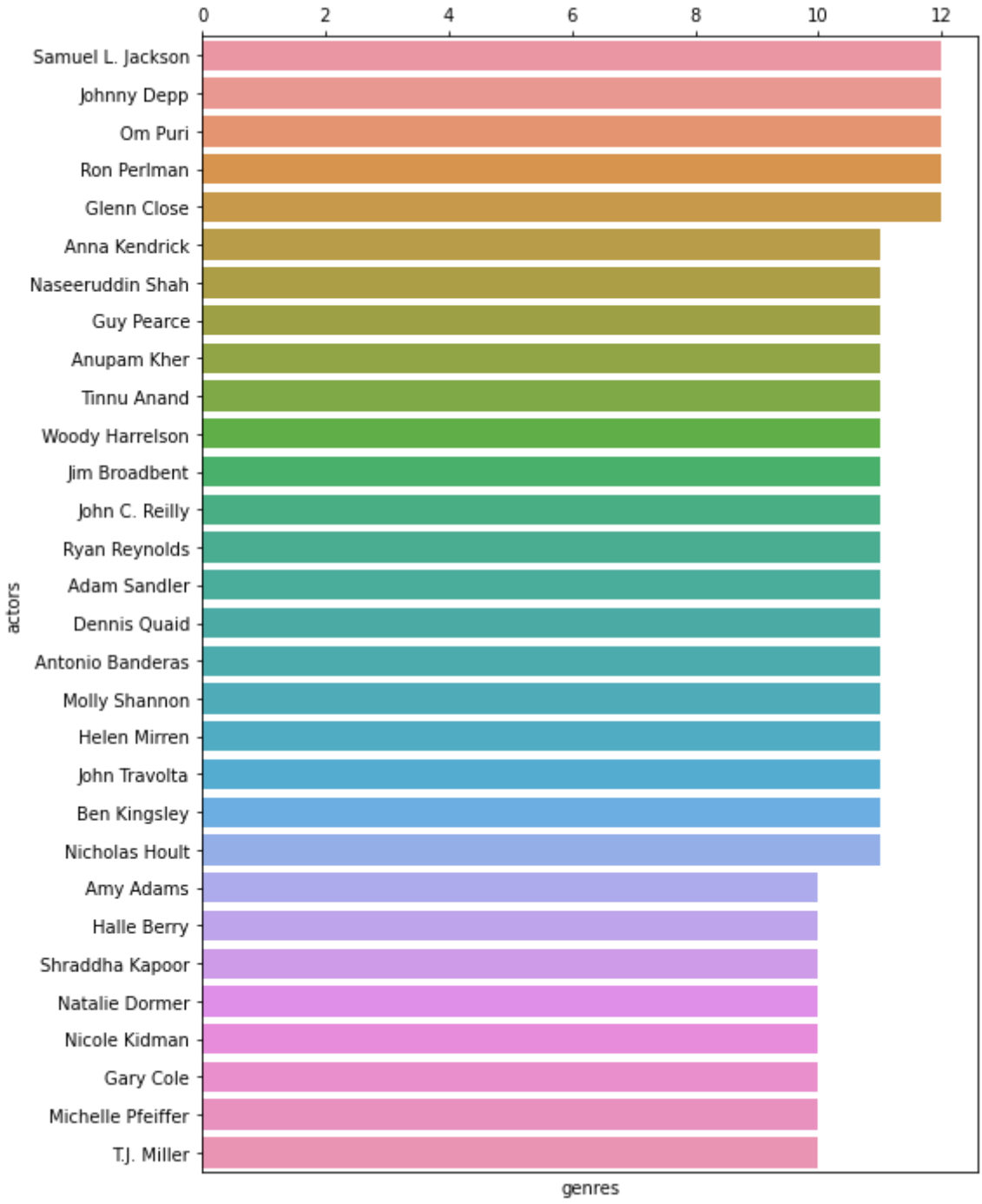
no. of genres per actor

Observation:

- Around 20-30 actors (among 24000) are genres (versatile) in more than 10 genres

In [170]:

```
1
2 plt_df = df.loc[df.actors != "Anonymous"].groupby(["actors"]).nunique().reset_index().s
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "actors", x= "genres")
7
8 ax.xaxis.tick_top()
```



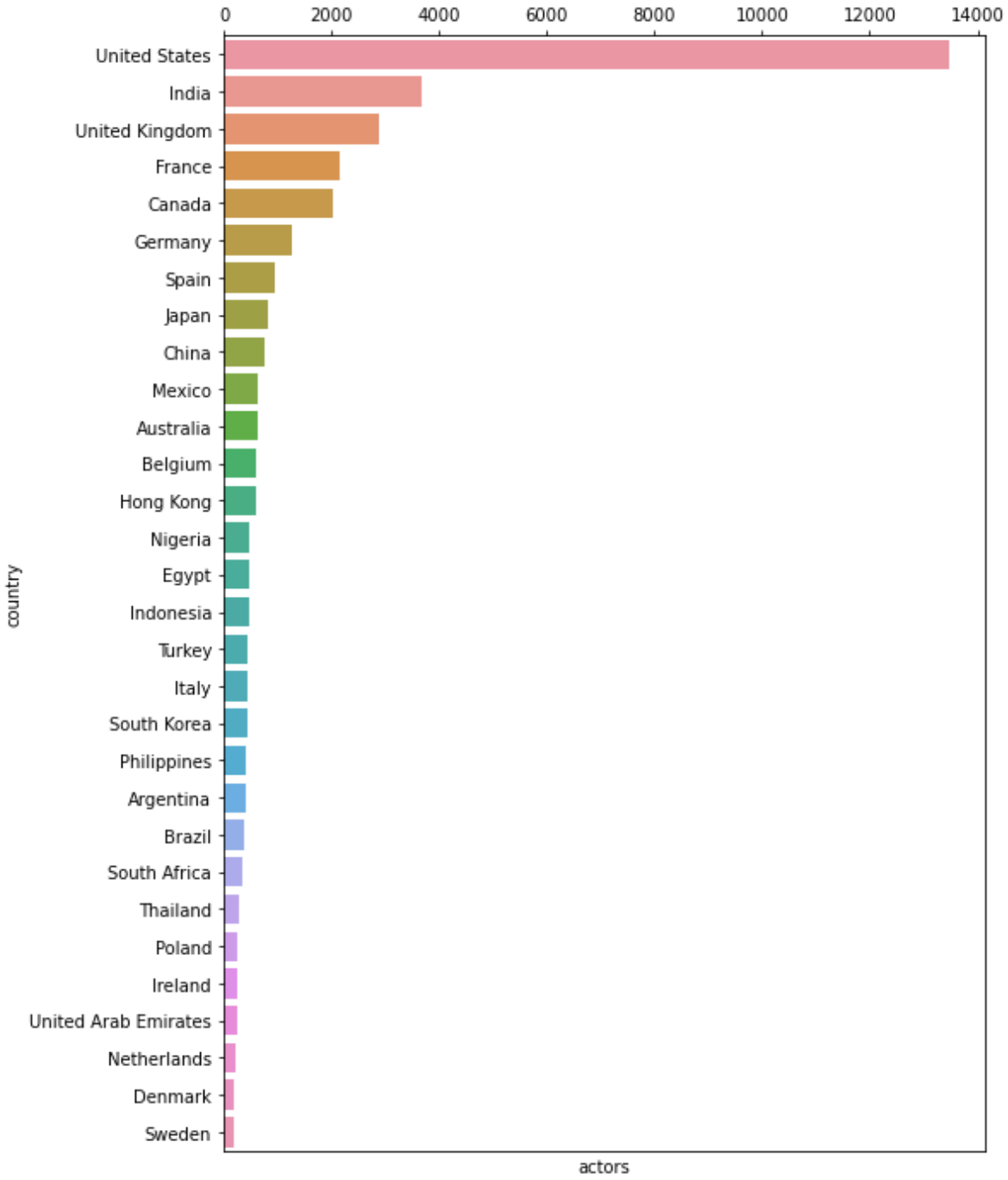
no. of actors/directors per country

Observation:

- Most actors/directors are streaming on US, India, UK, France, Canada, Germany
- The countries such as Panama, Iraq, Afghanistan, Vatican, Sri Lanka have only 1 actor/director streaming

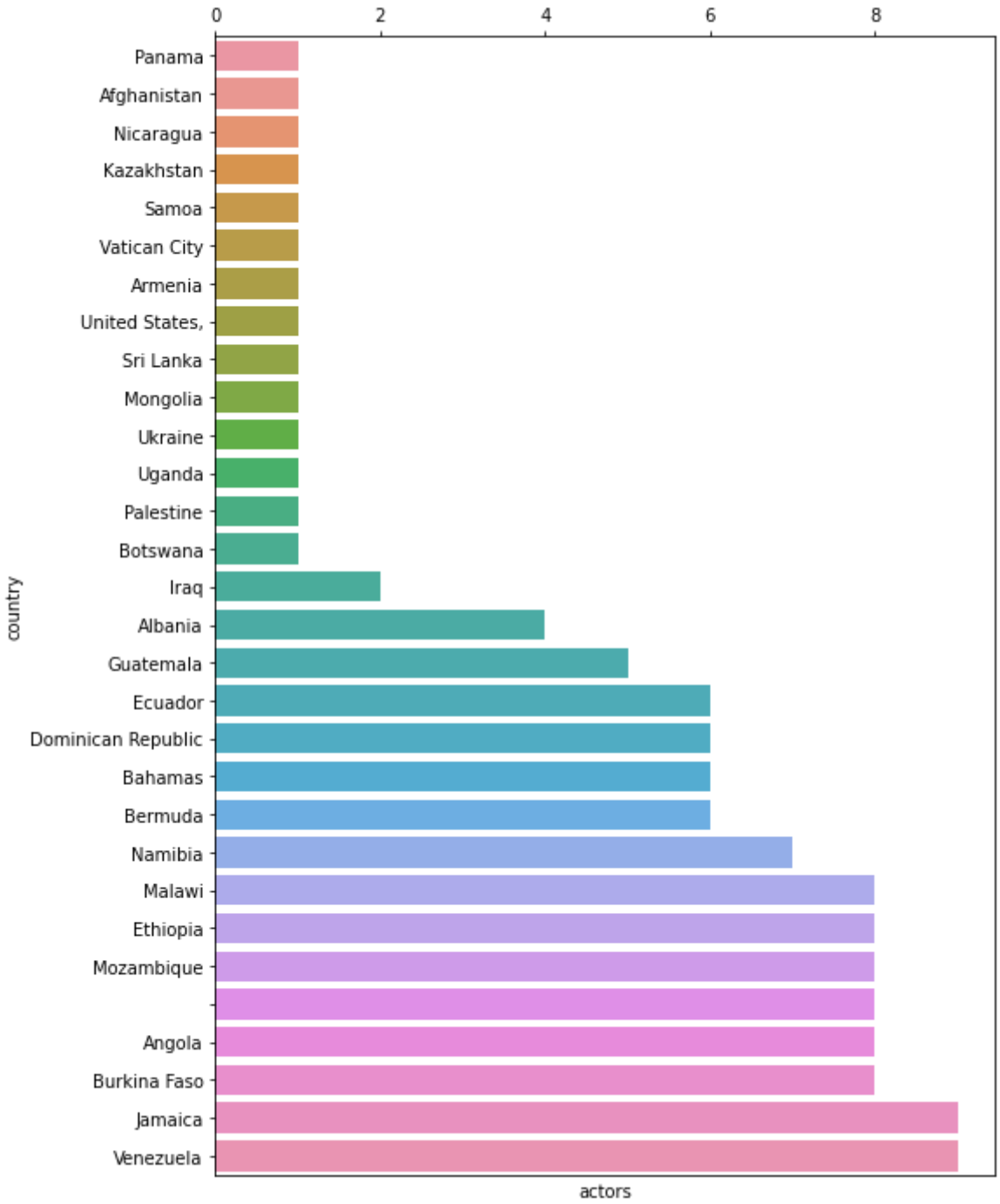
In [171]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["actors"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "actors")
7
8 ax.xaxis.tick_top()
```



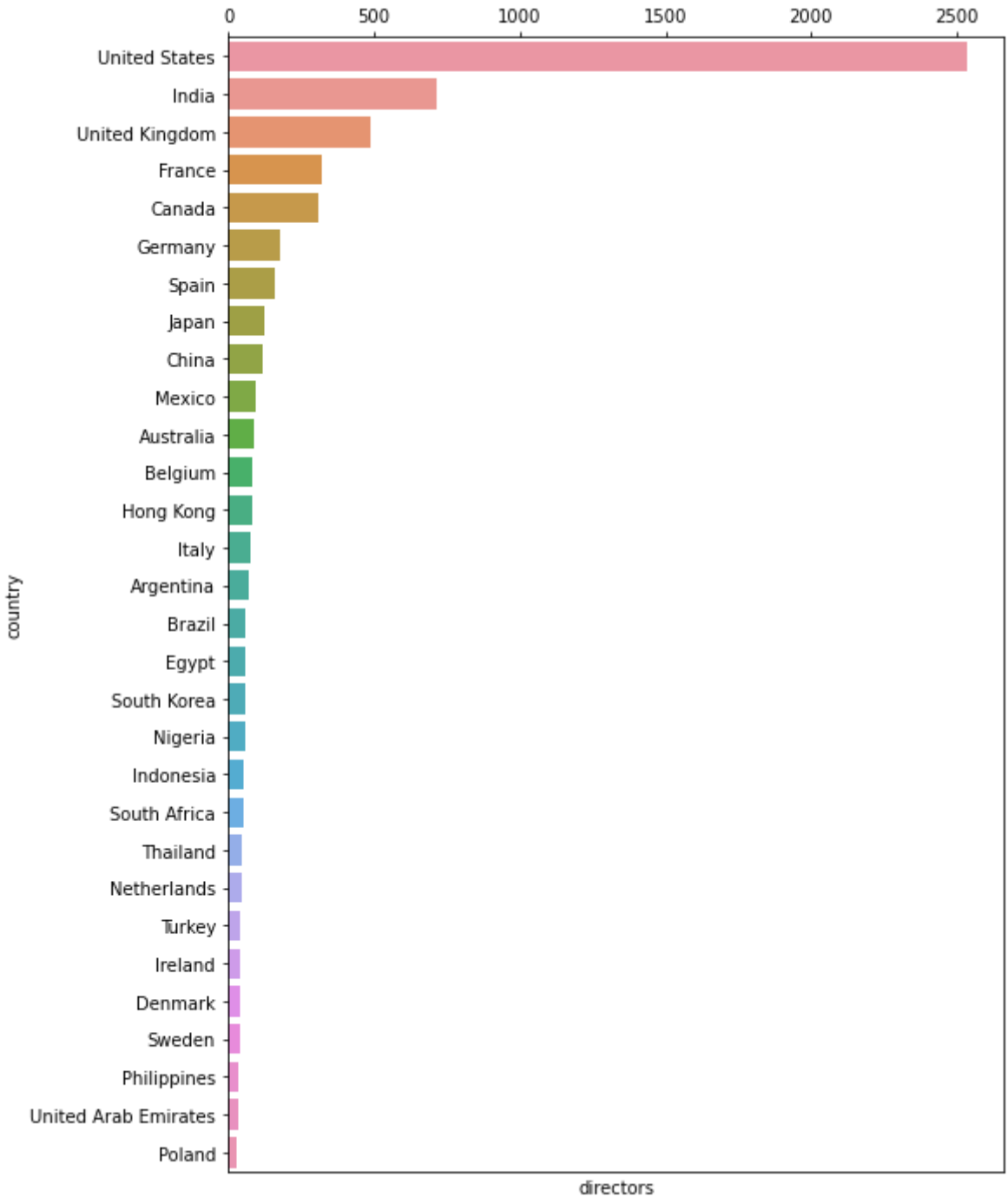
In [172]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["actors"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "actors")
7
8 ax.xaxis.tick_top()
```



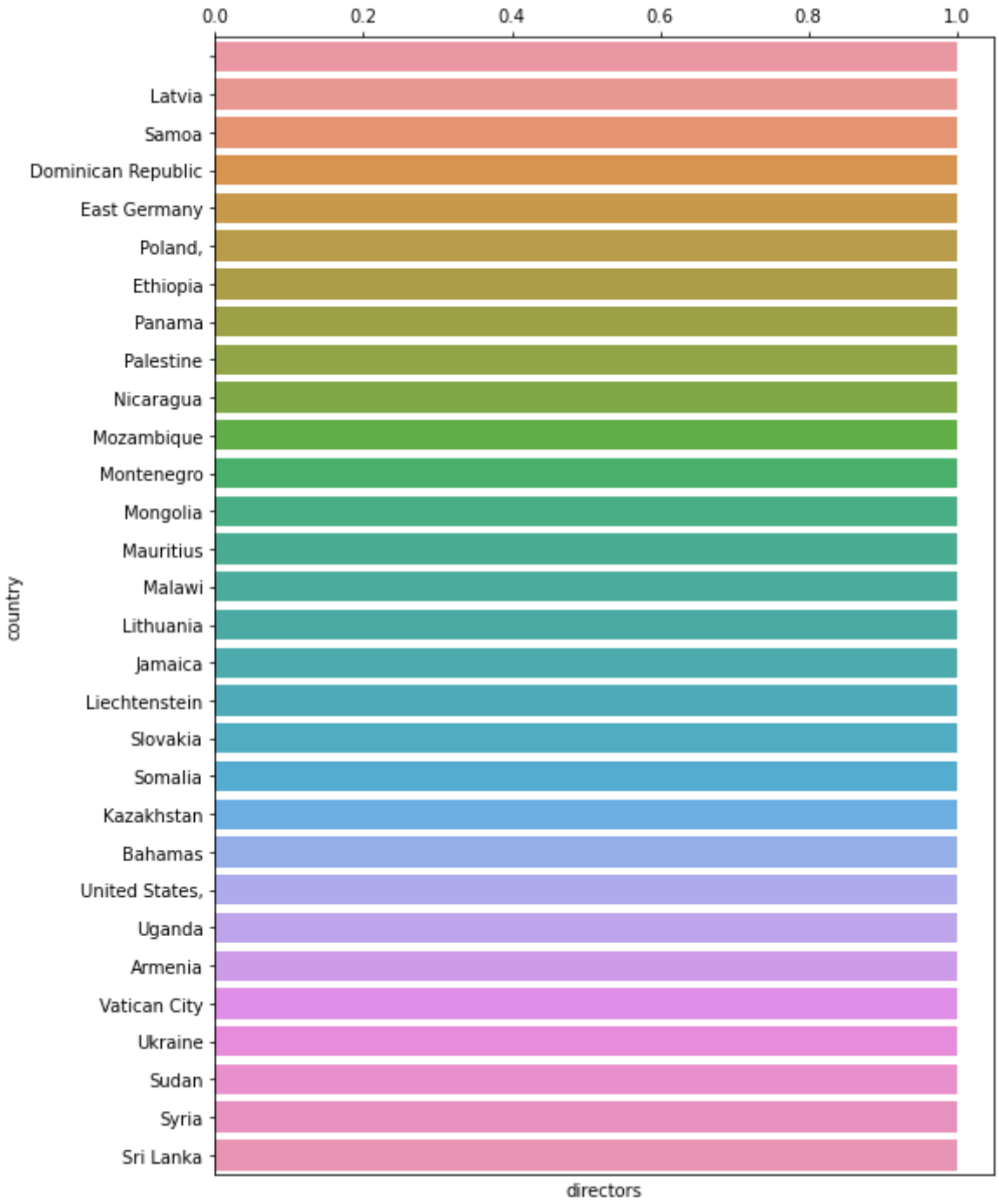
In [173]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["directors"], asc
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "directors")
7
8 ax.xaxis.tick_top()
```



In [174]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["directors"], asc
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "directors")
7
8 ax.xaxis.tick_top()
```



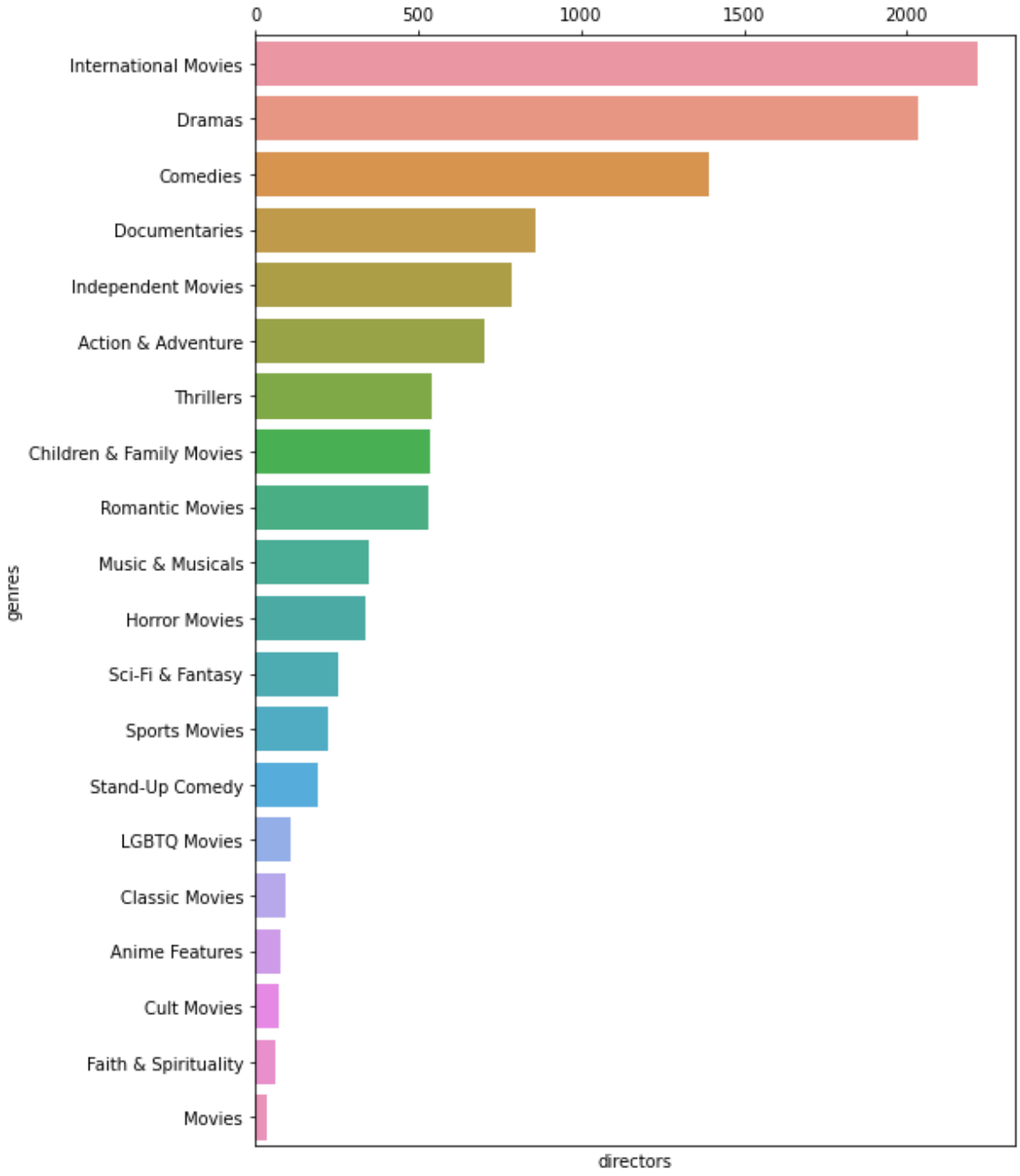
no. of actors/directors per genre

Observation:

- Most actors/directors are working on Dramas, Comedies, Independent Movies
- Very few directors on netflix are working on Classics, Cult, Anime Movies

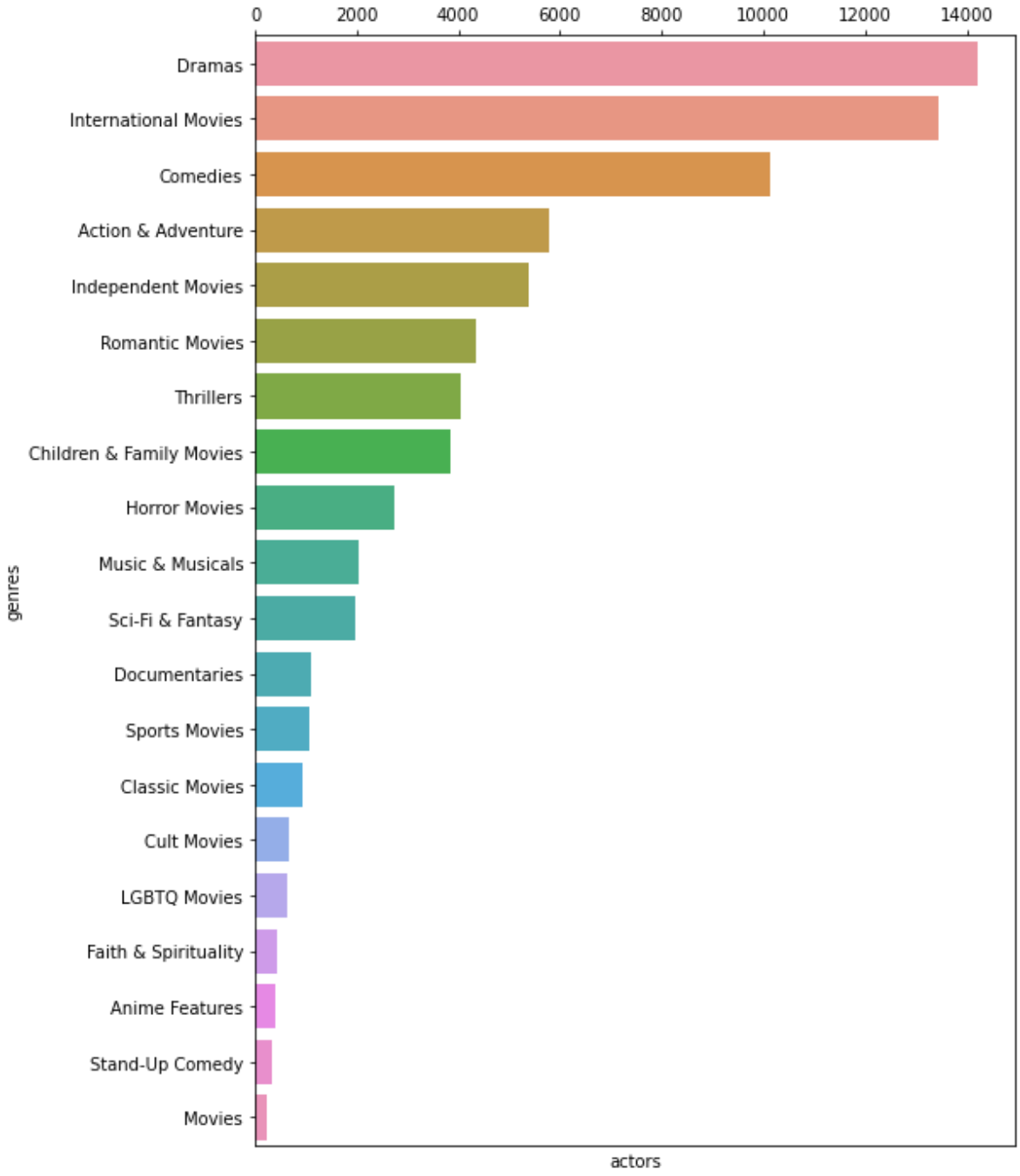
In [175]:

```
1
2 plt_df = df.groupby(["genres"]).nunique().reset_index().sort_values(["directors"], asce
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "genres", x= "directors")
7
8 ax.xaxis.tick_top()
```



In [176]:

```
plt_df = df.groupby(["genres"]).nunique().reset_index().sort_values(["actors"], ascendi

plt.figure(figsize= (8, 12))

ax = sns.barplot(data= plt_df, y= "genres", x= "actors")

ax.xaxis.tick_top()
```



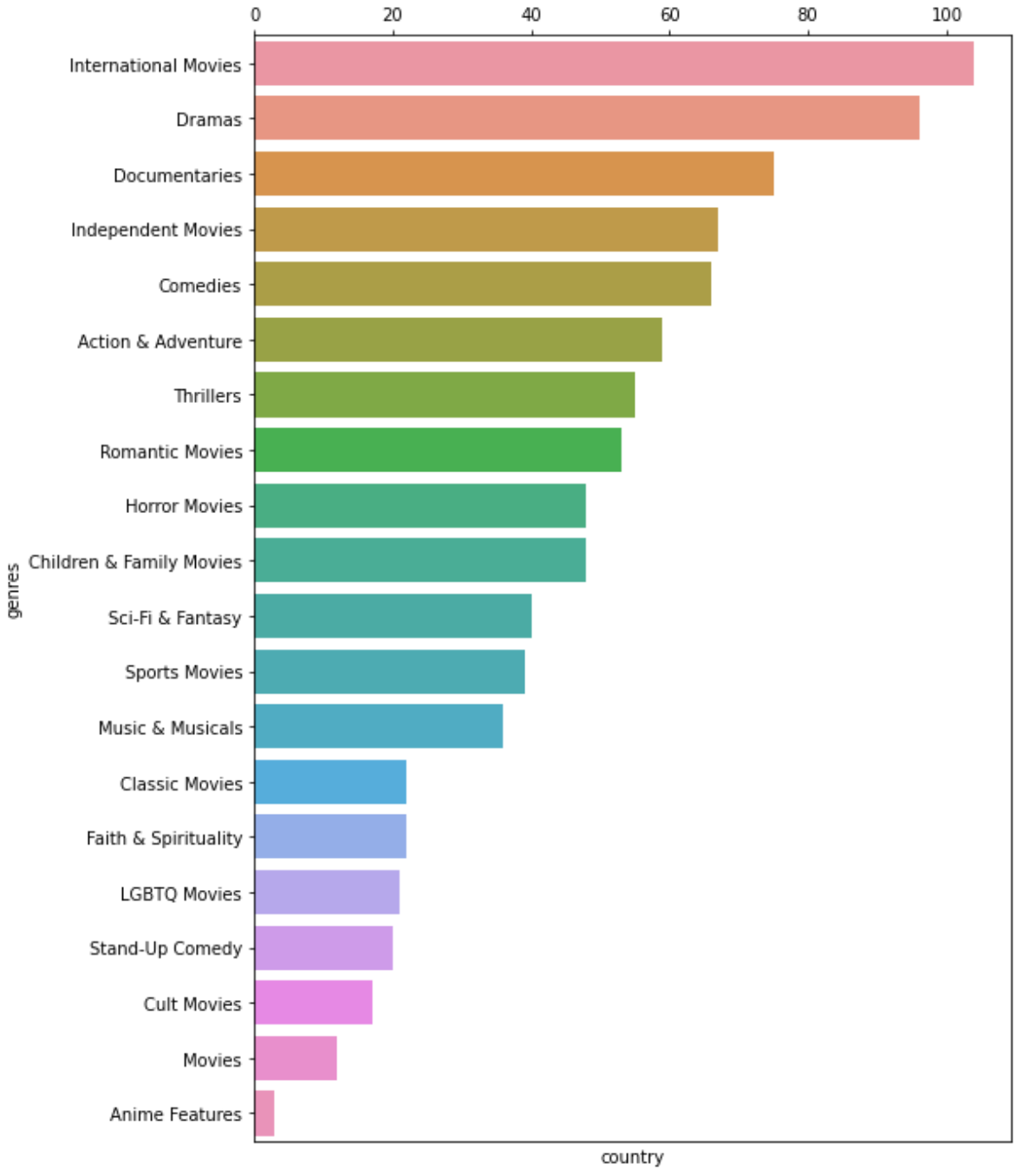
no. of countries per genre

Observation:

- Dramas, Comedies, Action, Thrillers are more popular among countries
- Classics, Cult and Anime Movies are least popular among countries

In [177]:

```
1
2 plt_df = df.groupby(["genres"]).nunique().reset_index().sort_values(["country"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "genres", x= "country")
7
8 ax.xaxis.tick_top()
```



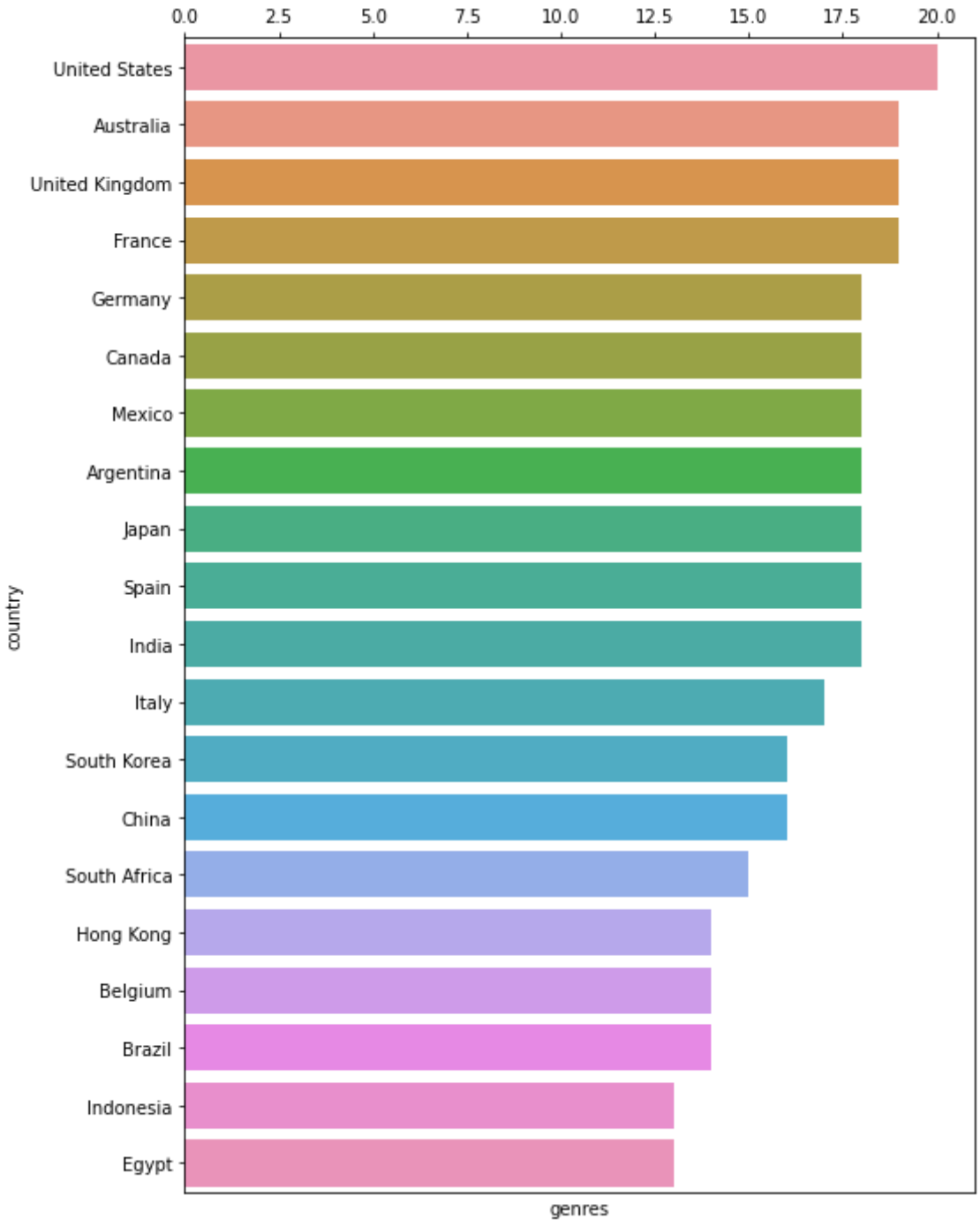
no. of genres per country

Observation:

- Only 10-20 countries (of 113) are streaming more than 12 (of 20) genres

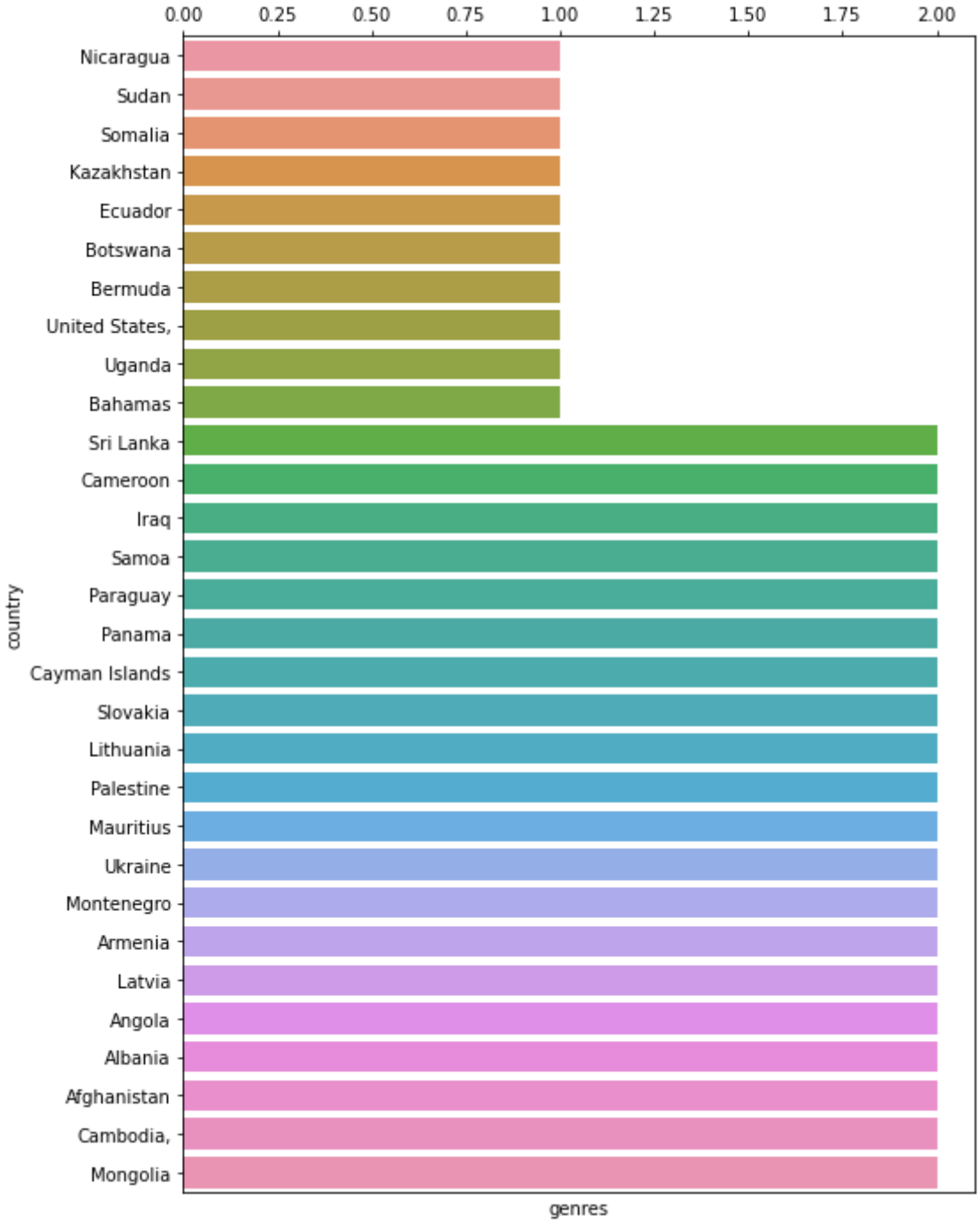
In [178]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["genres"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "genres")
7
8 ax.xaxis.tick_top()
```



In [179]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["genres"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "genres")
7
8 ax.xaxis.tick_top()
```



no. of movies/ directors/ actors vs added year

Observation:

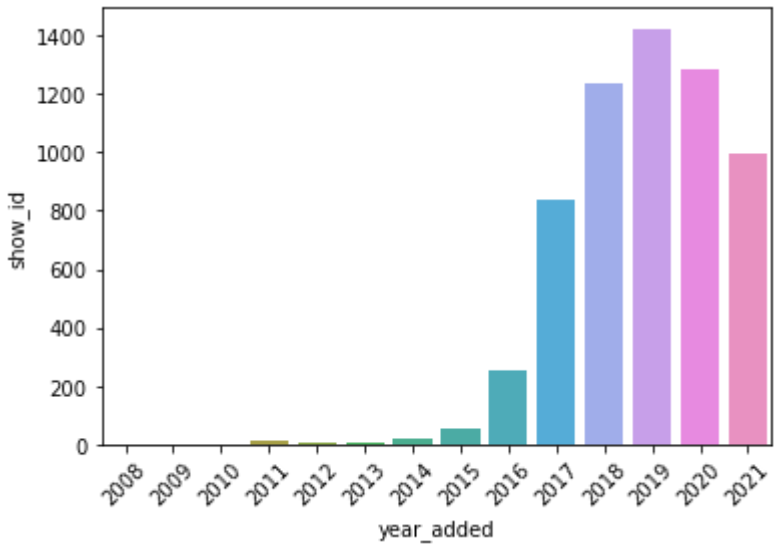
- The no. of movies/ directors/ actor peaked in 2019
- The no. of movies/ directors/ actors started growing since 2014

In [180]:

```
plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir

plt.xticks(rotation= 45)
sns.barplot(data= plt_df, y= "show_id", x= "year_added")
```

Out[180]:

```
<AxesSubplot:xlabel='year_added', ylabel='show_id'>
```

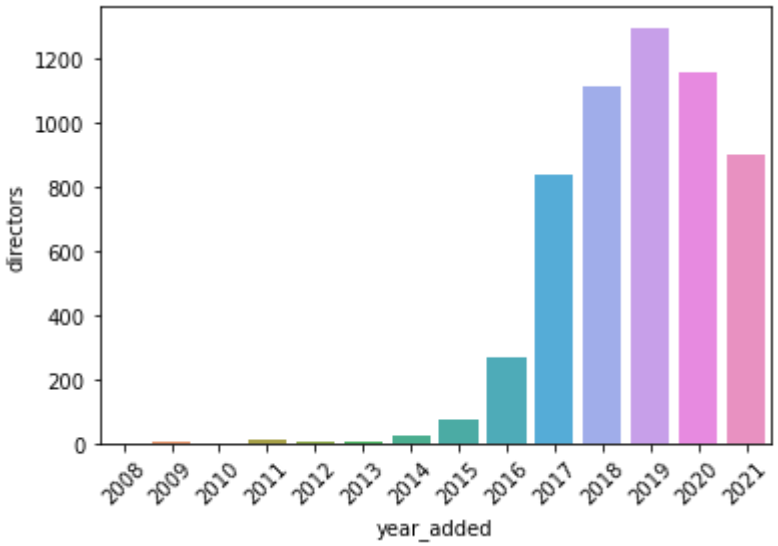


In [181]:

```
plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir

plt.xticks(rotation= 45)
sns.barplot(data= plt_df, y= "directors", x= "year_added")
```

Out[181]:

```
<AxesSubplot:xlabel='year_added', ylabel='directors'>
```

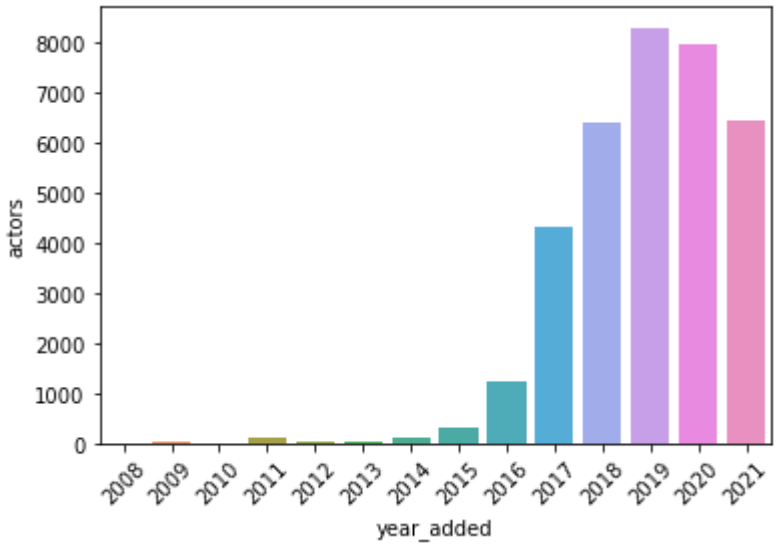


In [182]:

```
1
2 plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir
3
4 plt.xticks(rotation= 45)
5 sns.barplot(data= plt_df, y= "actors", x= "year_added")
```

Out[182]:

```
<AxesSubplot:xlabel='year_added', ylabel='actors'>
```



no. of countries/ genres streaming per aded year

Observation:

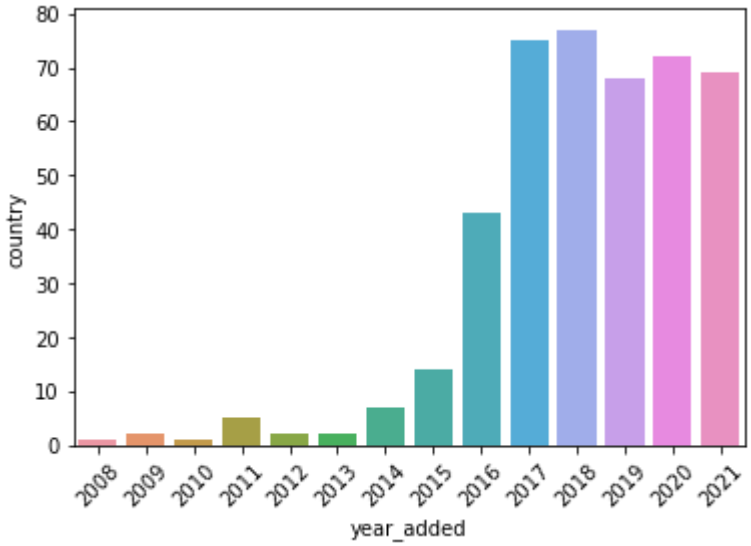
- The no. of countries/ genresstreaming grew b/w years 2014 - 17 and then stabilized around 110

In [183]:

```
plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir

plt.xticks(rotation= 45)
sns.barplot(data= plt_df, y= "country", x= "year_added")
```

Out[183]:

<AxesSubplot:xlabel='year_added', ylabel='country'>

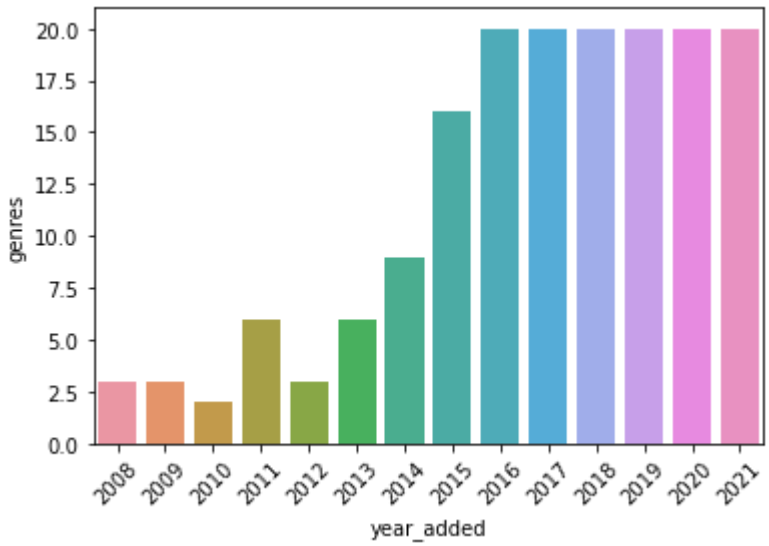


In [184]:

```
plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir

plt.xticks(rotation= 45)
sns.barplot(data= plt_df, y= "genres", x= "year_added")
```

Out[184]:

<AxesSubplot:xlabel='year_added', ylabel='genres'>



In [185]:

```
def new_nunique(x, df, col):
    year = x.date_added.dt.year.unique()[0]
    prev_col_vals = df.loc[df.date_added.dt.year < year][col].unique()

    result = pd.Series({ col: x.loc[~x[col].isin(prev_col_vals)].nunique()[col]})

    return result
```

no. of new directors/ actors added vs added year

Observation:

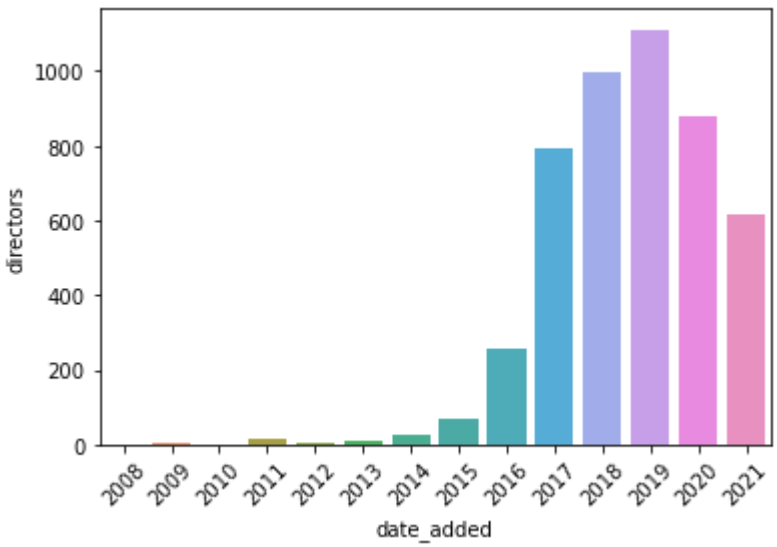
- The no. of new directors/ actor peaked in 2019
- The no. of new movies/ directors/ actors started growing since 2014

In [186]:

```
plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "director

plt.xticks(rotation= 45)

sns.barplot(x= plt_df.index, y= plt_df.directors)
```

Out[186]:

```
<AxesSubplot:xlabel='date_added', ylabel='directors'>
```

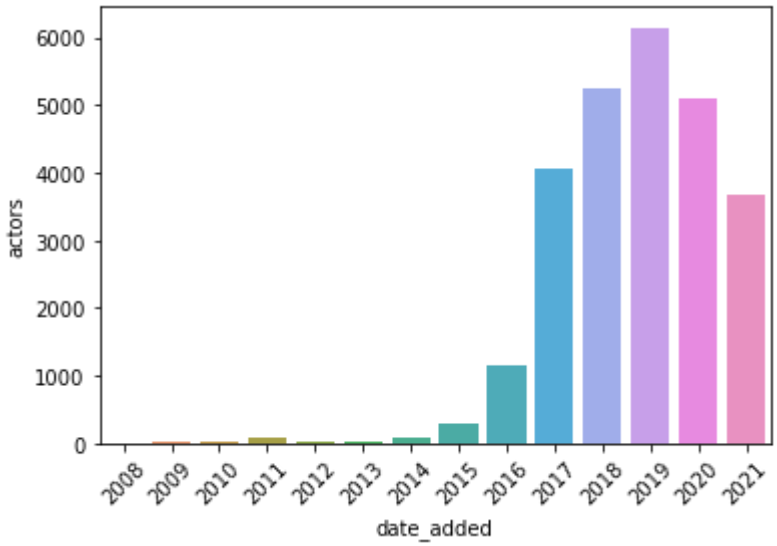


In [187]:

```
1
2 plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "actors"))
3
4 plt.xticks(rotation= 45)
5
6 sns.barplot(x= plt_df.index, y= plt_df.actors)
```

Out[187]:

```
<AxesSubplot:xlabel='date_added', ylabel='actors'>
```



no. of new genres added vs added year

Observation:

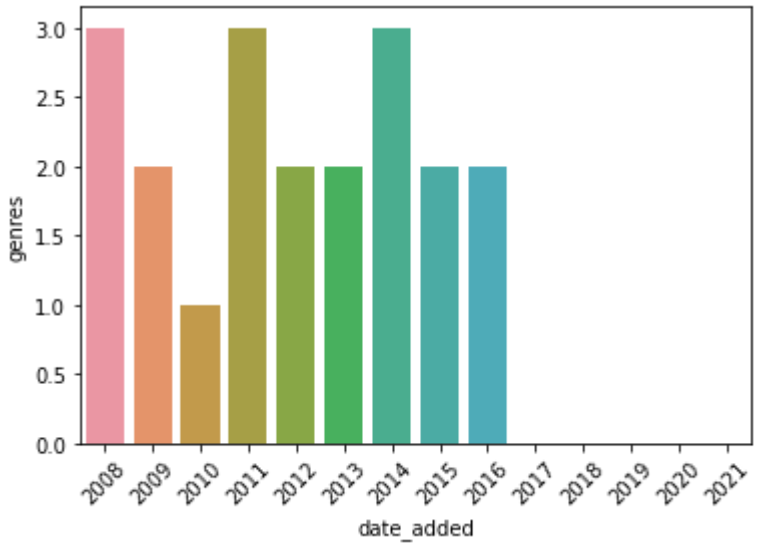
- New genres were added during the initial years and then kept at 20 since 2016

In [188]:

```
1
2 plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "genres"))
3
4 plt.xticks(rotation= 45)
5
6 sns.barplot(x= plt_df.index, y= plt_df.genres)
```

Out[188]:

```
<AxesSubplot:xlabel='date_added', ylabel='genres'>
```



no. of new countries added vs added year

Observation:

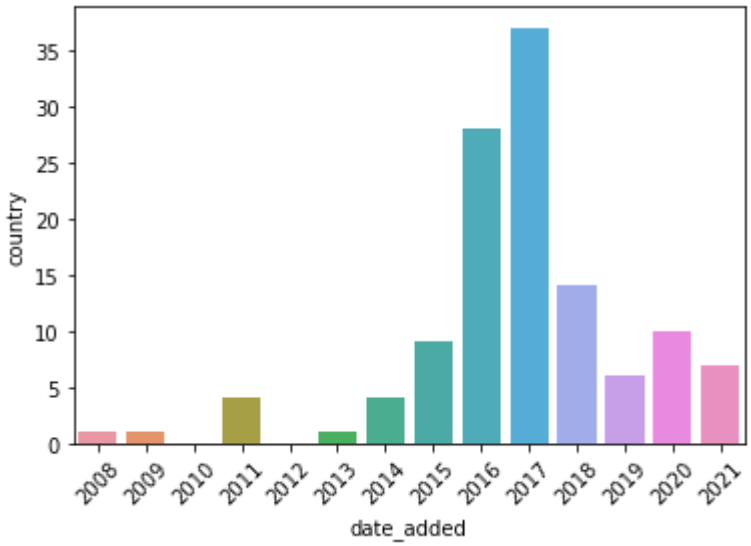
- No. of new countries added started growing from 2014 and peaked in 2017
- After 2017 the no. of new countries added slowed down to reach 113

In [189]:

```
1
2 plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "country"
3
4 plt.xticks(rotation= 45)
5
6 sns.barplot(x= plt_df.index, y= plt_df.country)
```

Out[189]:

```
<AxesSubplot:xlabel='date_added', ylabel='country'>
```



no. of new movies/ directors/ actors added per release year

Observation:

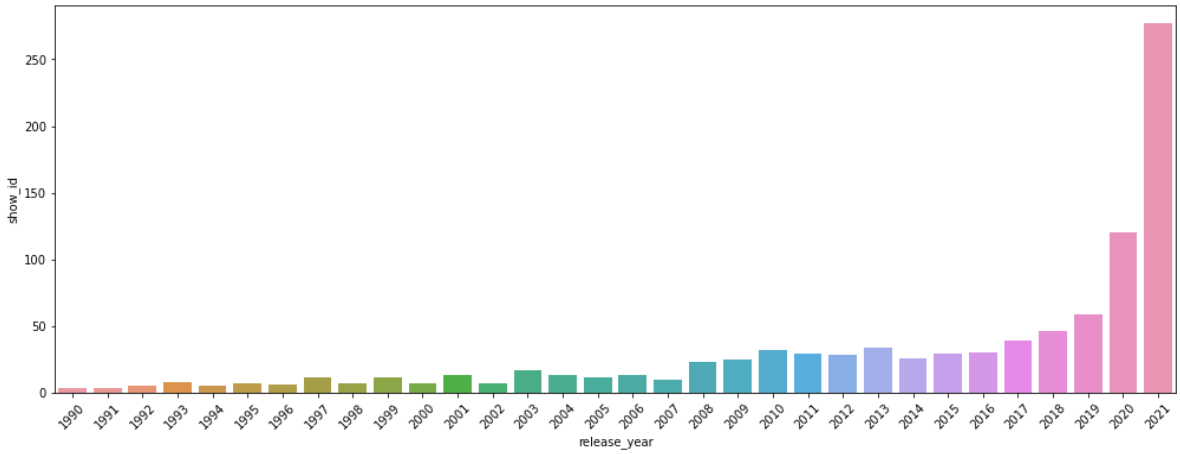
- The new directors/ actors in movies released every year that are added to netflix in peaked in 2017-18
- Post 2017-18 the directors/ actors in newly released movies strted to get repetative

In [190]:

```
1
2 plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "show_id")).res
3 plt_df = plt_df.loc[plt_df.release_year >= 1990]
4
5 plt.figure(figsize= (17, 6))
6 plt.xticks(rotation= 45)
7
8 sns.barplot(x= plt_df.release_year, y= plt_df.show_id)
```

Out[190]:

```
<AxesSubplot:xlabel='release_year', ylabel='show_id'>
```

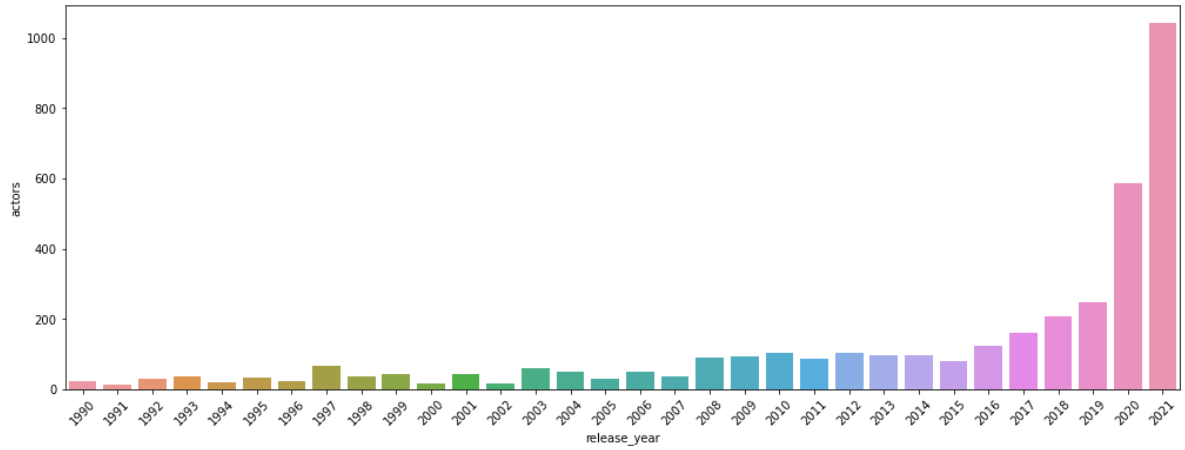


In [191]:

```
plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "actors")).rese
plt_df = plt_df.loc[plt_df.release_year >= 1990]

plt.figure(figsize= (17, 6))
plt.xticks(rotation= 45)

sns.barplot(x= plt_df.release_year, y= plt_df.actors)
```

Out[191]:

<AxesSubplot:xlabel='release_year', ylabel='actors'>

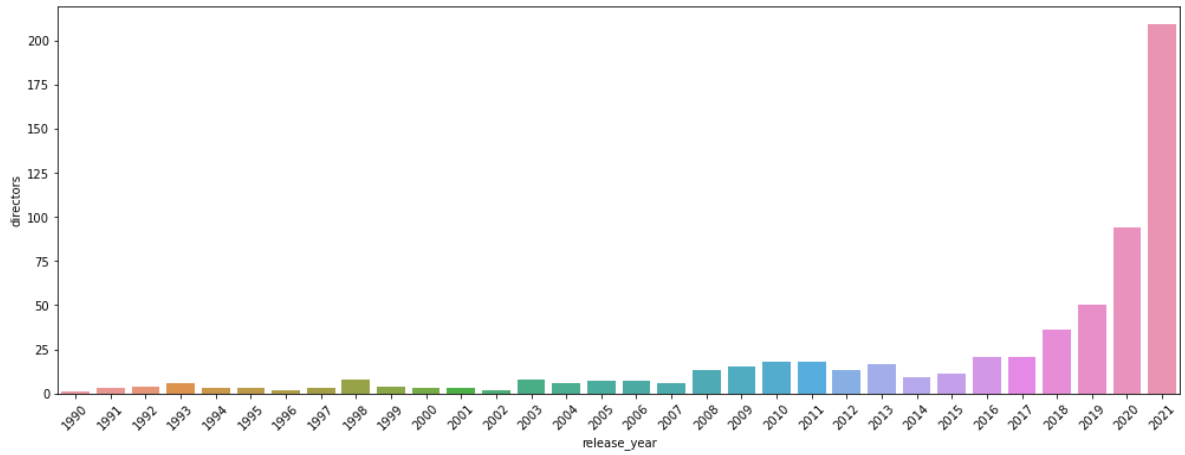


In [192]:

```
plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "directors")).r
plt_df = plt_df.loc[plt_df.release_year >= 1990]

plt.figure(figsize= (17, 6))
plt.xticks(rotation= 45)

sns.barplot(x= plt_df.release_year, y= plt_df.directors)
```

Out[192]:

<AxesSubplot:xlabel='release_year', ylabel='directors'>



no. of new countries added per release year

Observation:

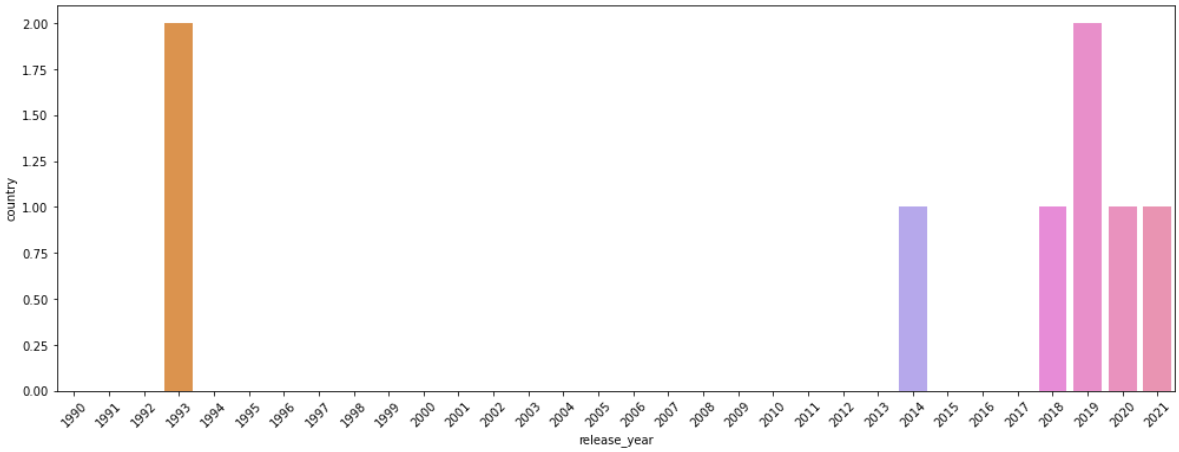
- The expansion to more countries happend through the movies released in 2014 and thn stated to stabilize

In [193]:

```
1
2 plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "country")).res
3 plt_df = plt_df.loc[plt_df.release_year >= 1990]
4
5 plt.figure(figsize= (17, 6))
6 plt.xticks(rotation= 45)
7
8 sns.barplot(x= plt_df.release_year, y= plt_df.country)
```

Out[193]:

```
<AxesSubplot:xlabel='release_year', ylabel='country'>
```



no. of new gernes added per release year

Observation:

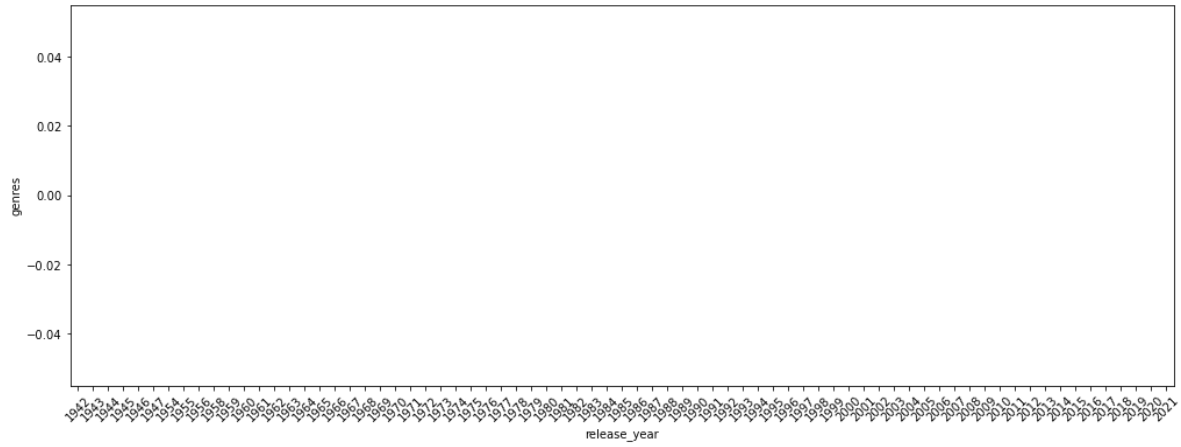
- The movie for the latest genre on netflix had released in 2000

In [194]:

```
plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "genres")).rese
# plt_df = plt_df.loc[plt_df.release_year >= 1990]

plt.figure(figsize= (17, 6))
plt.xticks(rotation= 45)

sns.barplot(x= plt_df.release_year, y= plt_df.genres)
```

Out[194]:

<AxesSubplot:xlabel='release_year', ylabel='genres'>



trend of movies added per release year and year added

Observation:

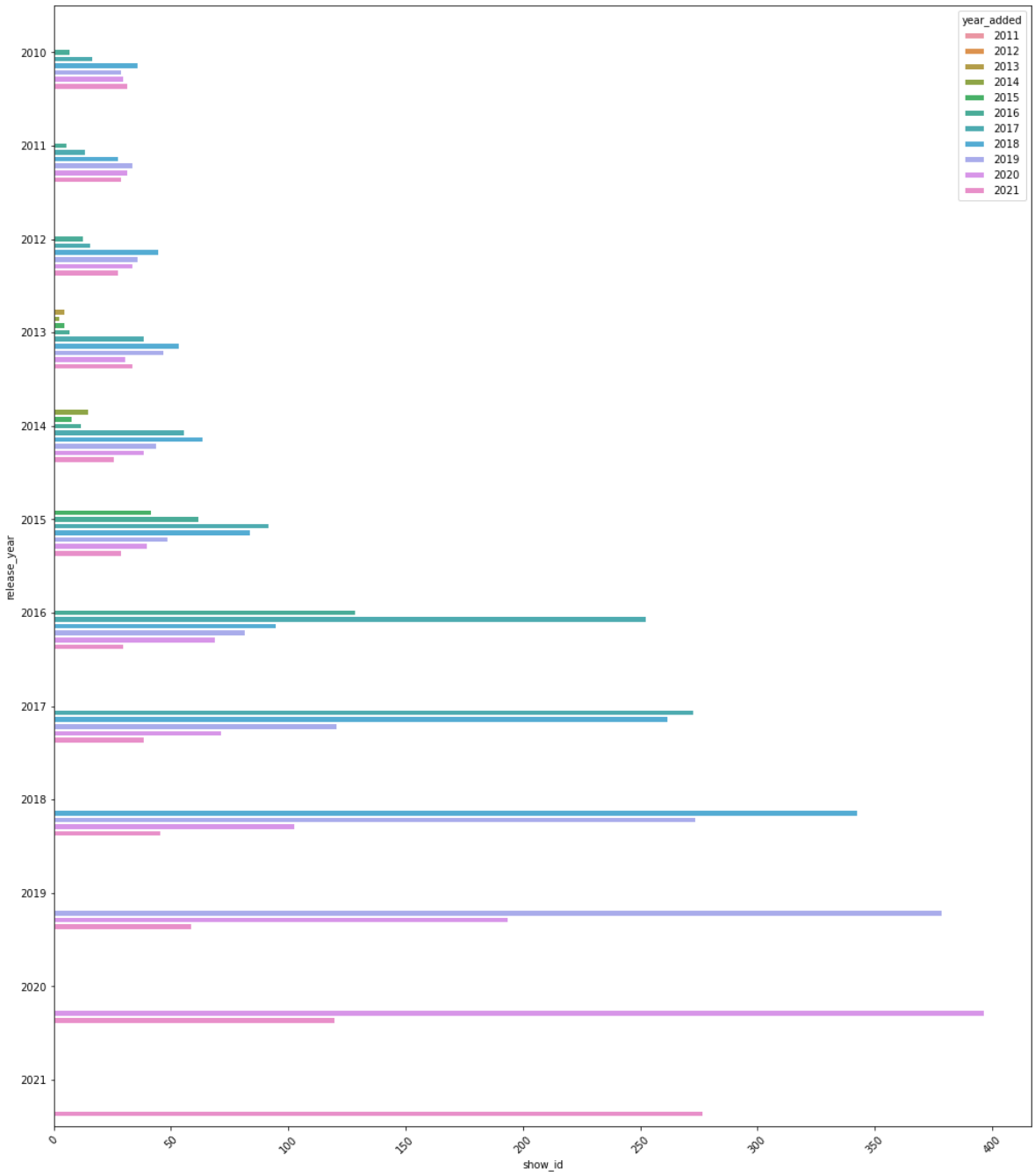
- Among the movies released in a given year the highest no. of movies are added to netflix in that year since 2017
- For the movies released before 2017, th ehighes no. of movies released in that year are added in 2017

In [195]:

```
1
2 plt_df = df.groupby([df.release_year, df.date_added.dt.year]).nunique().rename_axis(
3                                                                                 ["relea
4                                                                                 ).reset
5 plt_df = plt_df[plt_df.release_year >= 2010]
6 plt.figure(figsize= (17, 20))
7 plt.xticks(rotation= 45)
8
9 sns.barplot(y= plt_df.release_year, x= plt_df.show_id, hue= plt_df.year_added, orient=
```

Out[195]:

<AxesSubplot:xlabel='show_id', ylabel='release_year'>



trend of movies added per release year and year added

Observation:

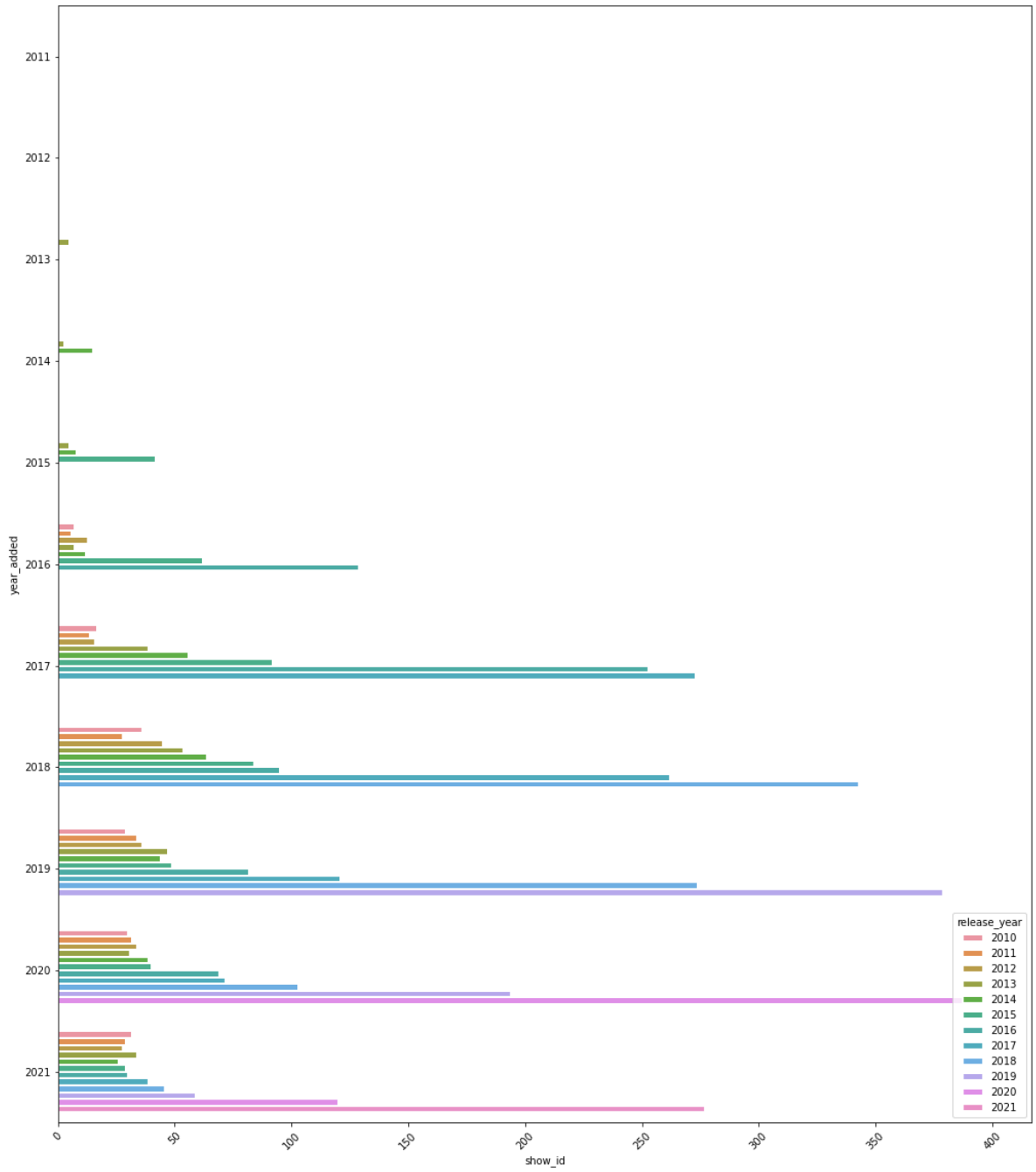
- Every year te highes t no. of movies added in that year belong to the movies released in that year

In [196]:

```
1
2 plt_df = df.groupby([df.release_year, df.date_added.dt.year]).nunique().rename_axis(
3                                                                                 ["relea
4                                                                                 ).reset
5 plt_df = plt_df[plt_df.release_year >= 2010]
6 plt.figure(figsize= (17, 20))
7 plt.xticks(rotation= 45)
8
9 sns.barplot(y= plt_df.year_added, x= plt_df.show_id, hue= plt_df.release_year, orient=
```

Out[196]:

<AxesSubplot:xlabel='show_id', ylabel='year_added'>



In [197]:

```
def new_nunique(x, df, col):
    year = x.date_added.dt.year.unique()[0]

    temp = df.loc[(df.date_added.dt.year >= year-4) & (df.date_added.dt.year <= year)]

    result = temp.groupby(temp.date_added.dt.year).nunique()

#     print("-"*50 + "\n", dict(zip(result.index.values, year - result.index.values)),

    new_index = dict(zip(result.index.values, year - result.index.values))

    result = result.rename(index= new_index)[col]

    return result
```

In [198]:

```
plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "show_id"

plt_df = plt_df.rename_axis(["year_added", "last_n_year_added"]).reset_index()
```

trend of movies added per release year and year added

Observation:

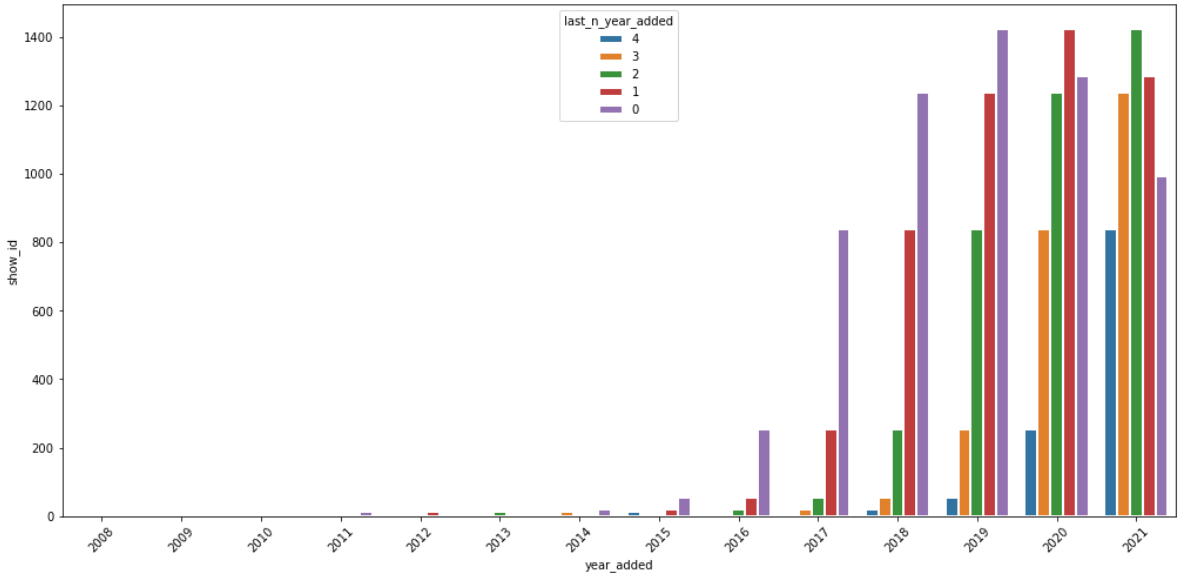
- Till 2019, most of the movies on netflix in a given year belong to the movies released that year
- Since 2019, most of the movies on Netflix by th eend of that year belong to th emovies released in 2019

In [199]:

```
1
2 plt.figure(figsize= (17, 8))
3 plt.xticks(rotation= 45)
4
5 sns.barplot(x= plt_df.year_added, y= plt_df.show_id, hue= plt_df.last_n_year_added,
6             hue_order = [4, 3, 2, 1, 0],
7             orient= 'v', edgecolor='white', linewidth=2)
```

Out[199]:

```
<AxesSubplot:xlabel='year_added', ylabel='show_id'>
```

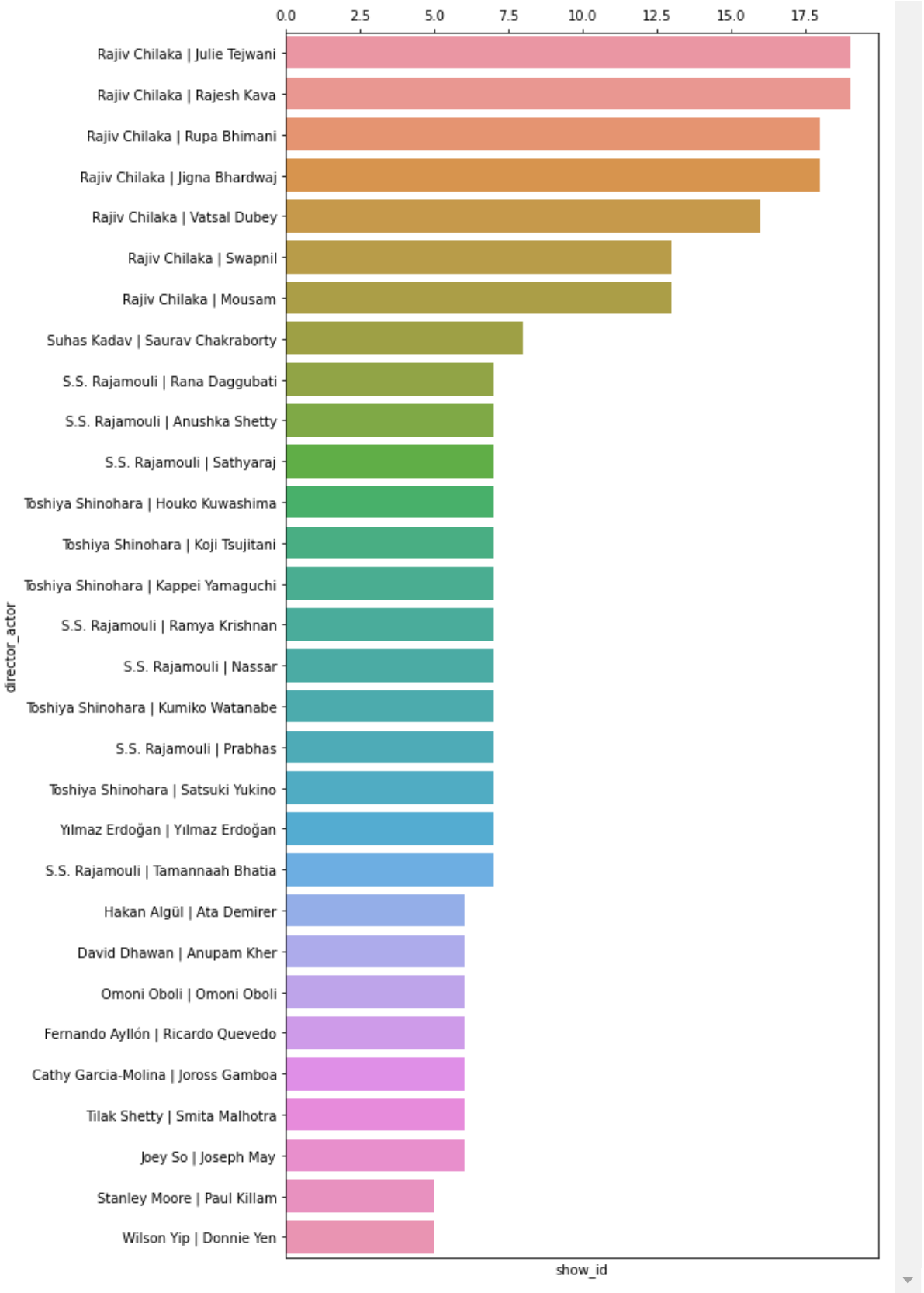


no. of movies per possible pair

- pairs between actors, directors, genres

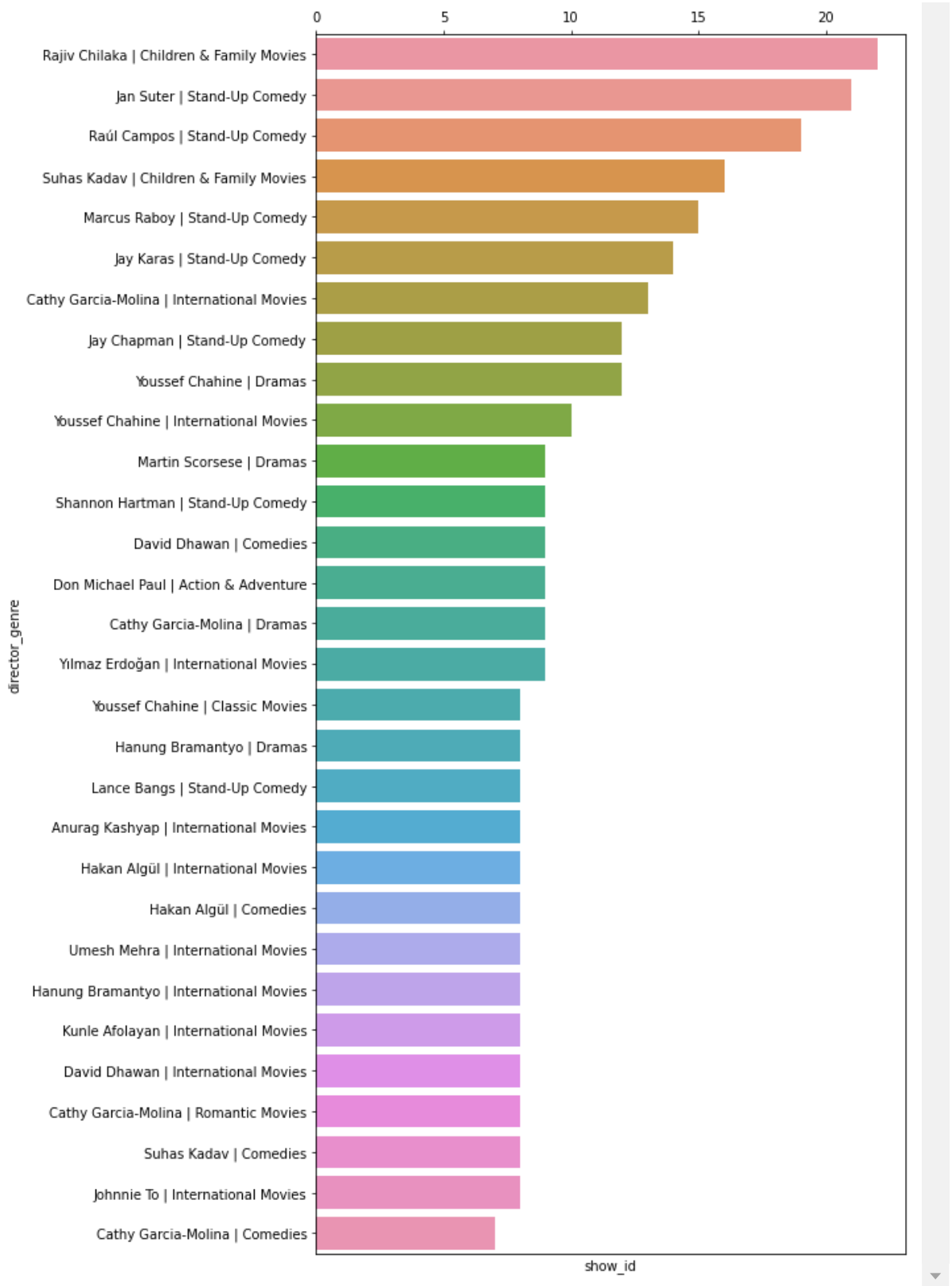
In [200]:

```
1
2  plt_df = df.loc[(df.directors != "Anonymous") & (df.actors != "Anonymous")].groupby(
3                                                                                ["directors
4                                                                                ).nunique()
5
6
7  plt.figure(figsize= (8, 17))
8
9  plt_df["director_actor"] = plt_df.apply(lambda x: x["directors"] + " | " + x["actors"],
10
11 ax = sns.barplot(y= plt_df.director_actor, x= plt_df.show_id)
12
13 ax.xaxis.tick_top()
```


0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Rajiv Chilaka | Julie Tejwani
Rajiv Chilaka | Rajesh Kava
Rajiv Chilaka | Rupa Bhimani
Rajiv Chilaka | Jigna Bhardwaj
Rajiv Chilaka | Vatsal Dubey
Rajiv Chilaka | Swapnil
Rajiv Chilaka | Mousam
Suhas Kadav | Saurav Chakraborty
S.S. Rajamouli | Rana Daggubati
S.S. Rajamouli | Anushka Shetty
S.S. Rajamouli | Sathyaraj
Toshiya Shinohara | Houko Kuwashima
Toshiya Shinohara | Koji Tsujitani
Toshiya Shinohara | Kappei Yamaguchi
S.S. Rajamouli | Ramya Krishnan
S.S. Rajamouli | Nassar
Toshiya Shinohara | Kumiko Watanabe
S.S. Rajamouli | Prabhas
Toshiya Shinohara | Satsuki Yukino
Yılmaz Erdoğan | Yılmaz Erdoğan
S.S. Rajamouli | Tamannaah Bhatia
Hakan Algül | Ata Demirer
David Dhawan | Anupam Kher
Omoni Oboli | Omoni Oboli
Fernando Ayllón | Ricardo Quevedo
Cathy Garcia-Molina | Joross Gamboa
Tilak Shetty | Smita Malhotra
Joey So | Joseph May
Stanley Moore | Paul Killam
Wilson Yip | Donnie Yen
director_actor
show_id

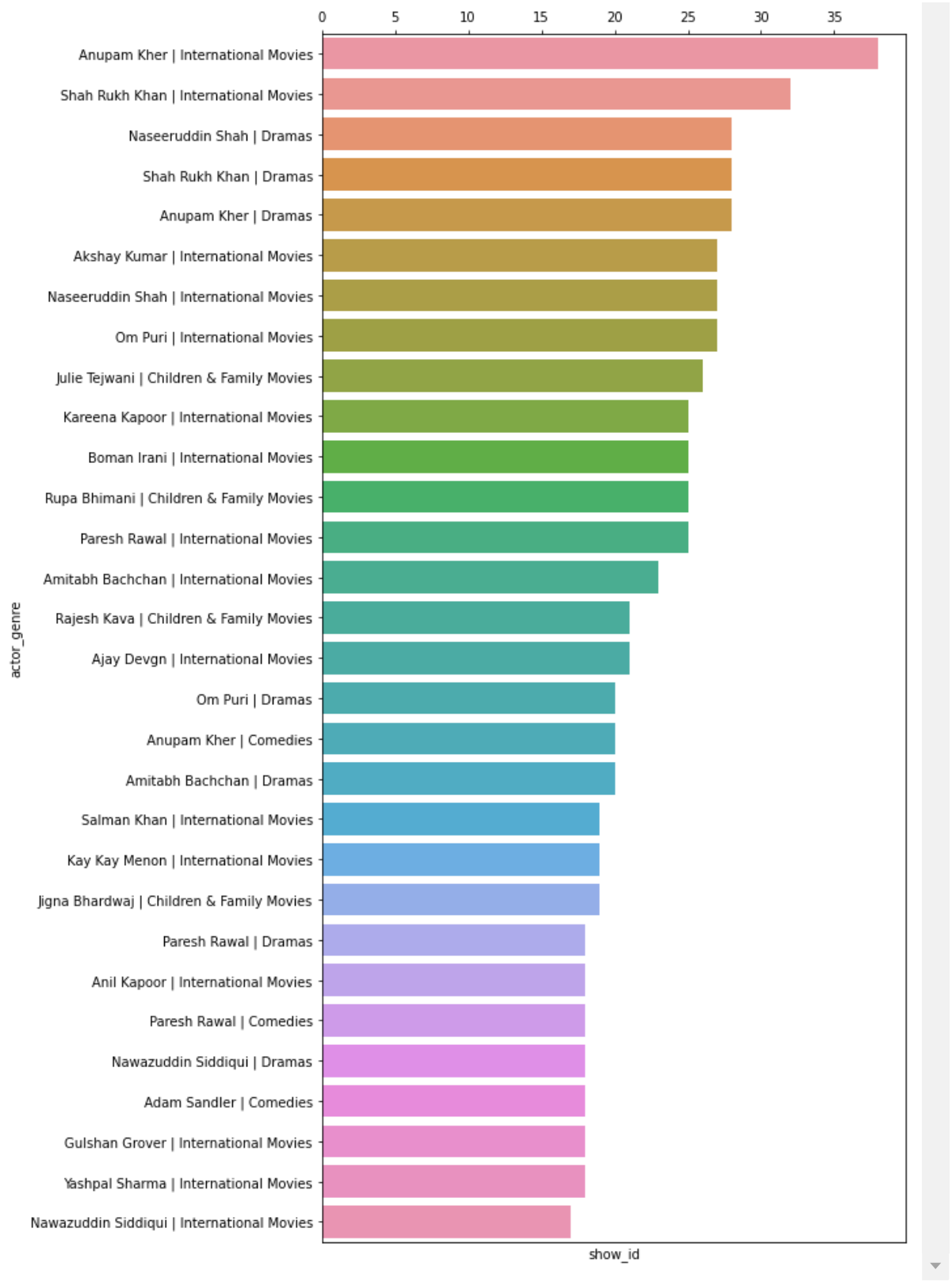
In [201]:

```
1
2  plt_df = df.loc[(df.directors != "Anonymous")].groupby(
3                                                  ["directors", "genres"]
4                                                  ).nunique().sort_values("show_id",
5                                                                          ascending= Fals
6
7  plt.figure(figsize= (8, 17))
8
9  plt_df["director_genre"] = plt_df.apply(lambda x: x["directors"] + " | " + x["genres"],
10
11 ax = sns.barplot(y= plt_df.director_genre, x= plt_df.show_id)
12
13 ax.xaxis.tick_top()
```


0
5
10
15
20
Rajiv Chilaka | Children & Family Movies
Jan Suter | Stand-Up Comedy
Raúl Campos | Stand-Up Comedy
Suhas Kadav | Children & Family Movies
Marcus Raboy | Stand-Up Comedy
Jay Karas | Stand-Up Comedy
Cathy Garcia-Molina | International Movies
Jay Chapman | Stand-Up Comedy
Youssef Chahine | Dramas
Youssef Chahine | International Movies
Martin Scorsese | Dramas
Shannon Hartman | Stand-Up Comedy
David Dhawan | Comedies
Don Michael Paul | Action & Adventure
Cathy Garcia-Molina | Dramas
Yılmaz Erdoğan | International Movies
Youssef Chahine | Classic Movies
Hanung Bramantyo | Dramas
Lance Bangs | Stand-Up Comedy
Anurag Kashyap | International Movies
Hakan Algül | International Movies
Hakan Algül | Comedies
Umesh Mehra | International Movies
Hanung Bramantyo | International Movies
Kunle Afolayan | International Movies
David Dhawan | International Movies
Cathy Garcia-Molina | Romantic Movies
Suhas Kadav | Comedies
Johnnie To | International Movies
Cathy Garcia-Molina | Comedies
director_genre
show_id

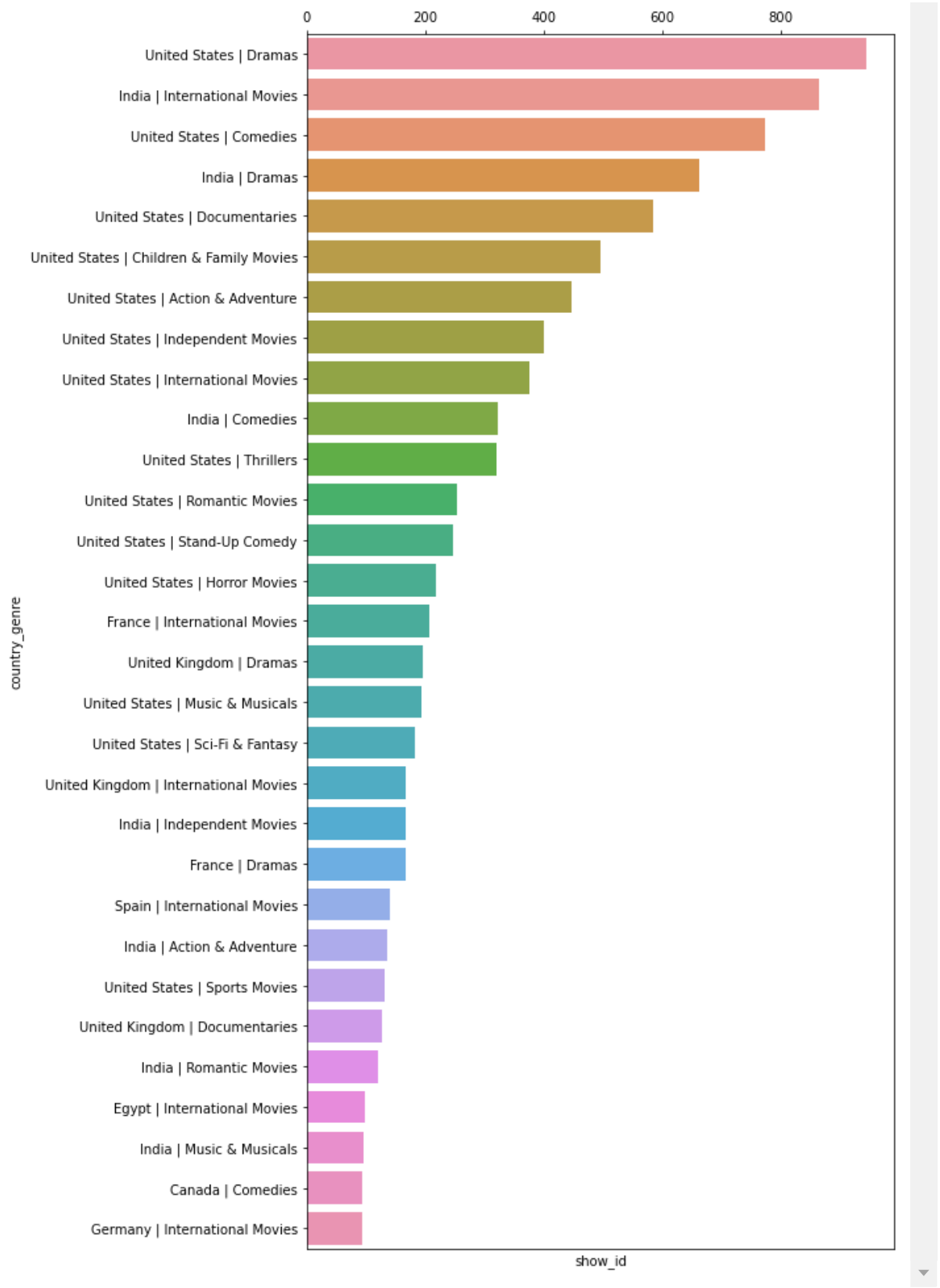
In [202]:

```
plt_df = df.loc[(df.actors != "Anonymous")].groupby(["actors", "genres"]).nunique().sor

plt.figure(figsize= (8, 17))

plt_df["actor_genre"] = plt_df.apply(lambda x: x["actors"] + " | " + x["genres"], axis=

ax = sns.barplot(y= plt_df.actor_genre, x= plt_df.show_id)

ax.xaxis.tick_top()
```


0
5
10
15
20
25
30
35
Anupam Kher | International Movies
Shah Rukh Khan | International Movies
Naseeruddin Shah | Dramas
Shah Rukh Khan | Dramas
Anupam Kher | Dramas
Akshay Kumar | International Movies
Naseeruddin Shah | International Movies
Om Puri | International Movies
Julie Tejwani | Children & Family Movies
Kareena Kapoor | International Movies
Boman Irani | International Movies
Rupa Bhimani | Children & Family Movies
Paresh Rawal | International Movies
Amitabh Bachchan | International Movies
Rajesh Kava | Children & Family Movies
Ajay Devgn | International Movies
Om Puri | Dramas
Anupam Kher | Comedies
Amitabh Bachchan | Dramas
Salman Khan | International Movies
Kay Kay Menon | International Movies
Jigna Bhardwaj | Children & Family Movies
Paresh Rawal | Dramas
Anil Kapoor | International Movies
Paresh Rawal | Comedies
Nawazuddin Siddiqui | Dramas
Adam Sandler | Comedies
Gulshan Grover | International Movies
Yashpal Sharma | International Movies
Nawazuddin Siddiqui | International Movies
actor_genre
show_id

In [203]:

```
1
2  plt_df = df.groupby(["country", "genres"]).nunique().sort_values("show_id",
3                                                                    ascending= False)["s
4
5  plt.figure(figsize= (8, 17))
6
7  plt_df["country_genre"] = plt_df.apply(lambda x: x["country"] + " | " + x["genres"], ax
8
9  ax = sns.barplot(y= plt_df.country_genre, x= plt_df.show_id)
10
11 ax.xaxis.tick_top()
```



no. of values per min_age

- values: directors, actors, country, genre

In [204]:

```
1
2 df.directors.nunique()
```

Out[204]:

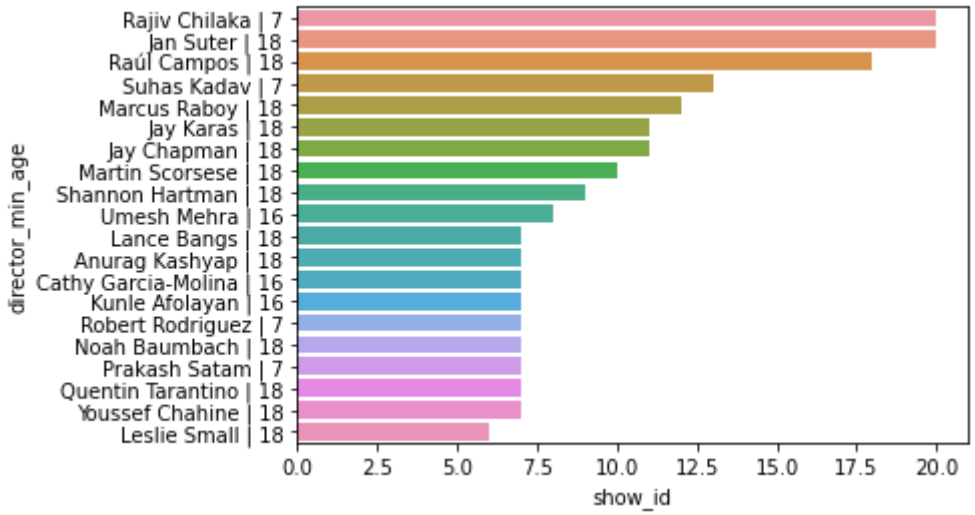
```
4778
```

In [205]:

```
1
2 plt_df = df.loc[(df.directors != "Anonymous")].groupby(["directors", "min_age"]).nuniqu
3                                                                                  ["s
4                                                                                  asc
5
6 plt_df = plt_df.groupby("directors").head(1).sort_values("show_id", ascending= False)
7
8 plt_df["director_min_age"] = plt_df.apply(lambda x: x["directors"] + " | " + str(x["mir
9
10 sns.barplot(data= plt_df.head(20), y= "director_min_age", x= "show_id", orient= 'h')
```

Out[205]:

```
<AxesSubplot:xlabel='show_id', ylabel='director_min_age'>
```

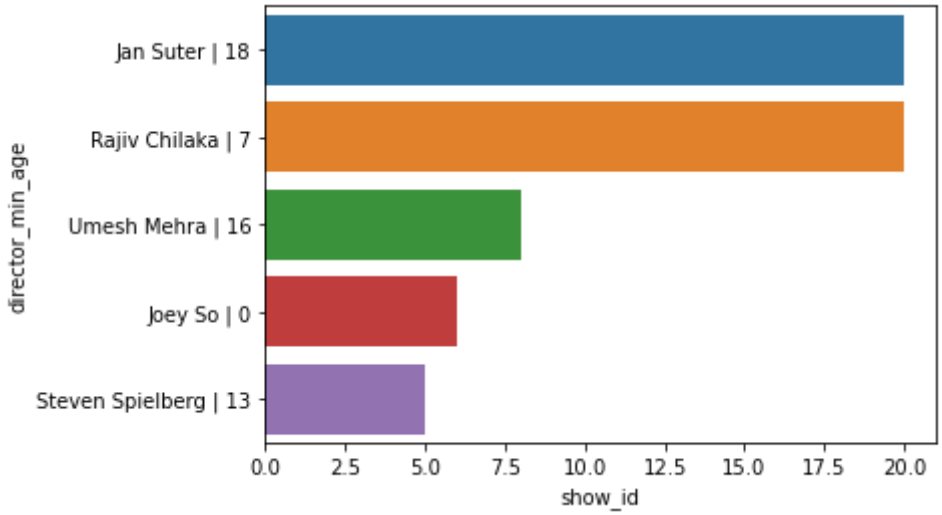


In [206]:

```
plt_df = df.loc[(df.directors != "Anonymous")].groupby(["directors", "min_age"]).nuniqu
                                                                                ["s
                                                                                asc

plt_df = plt_df.groupby("min_age").head(1).sort_values("show_id", ascending= False)

plt_df["director_min_age"] = plt_df.apply(lambda x: x["directors"] + " | " + str(x["mir

sns.barplot(data= plt_df.head(20), y= "director_min_age", x= "show_id", orient= 'h')
```

Out[206]:

<AxesSubplot:xlabel='show_id', ylabel='director_min_age'>

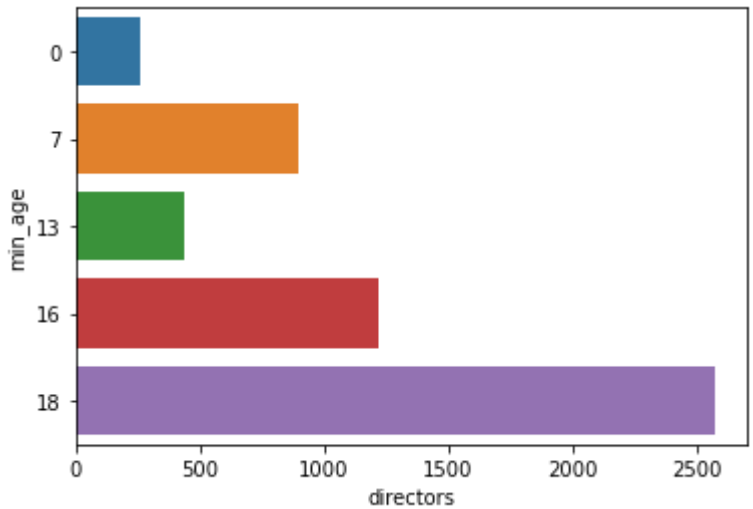


In [207]:

```
plt_df = df.groupby("min_age").nunique()["directors"].reset_index()

sns.barplot(y= plt_df.min_age, x= plt_df.directors, orient= 'h')
```

Out[207]:

<AxesSubplot:xlabel='directors', ylabel='min_age'>



TV Show Data Analysis: Unnested

The following is the analysis of movies data only for unnested columns

- "show_id", "type", "title", "date_added", "release_year", "rating", "duration", "min_age"

In [208]:

```
1
2 netflix_data_listed_tv = netflix_data_listed[netflix_data_listed.type == "TV Show"].cop
3
4 netflix_data_listed_tv_nonest = netflix_data_listed_tv[
5                                 ["show_id", "type", "title", "date_added", "release_yea
6                               ].copy()
7
8 netflix_data_listed_tv_nonest.head()
```

Out[208]:

	show_id	type	title	date_added	release_year	rating	duration	min_age
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	18
2	s3	TV Show	Ganglands	2021-09-24	2021	TV-MA	1	18
3	s4	TV Show	Jailbirds New Orleans	2021-09-24	2021	TV-MA	1	18
4	s5	TV Show	Kota Factory	2021-09-24	2021	TV-MA	2	18
5	s6	TV Show	Midnight Mass	2021-09-24	2021	TV-MA	1	18

Univariate Analysis

In [209]:

```
1
2 df = netflix_data_listed_tv_nonest
```

No of TV shows by rating

Observations:

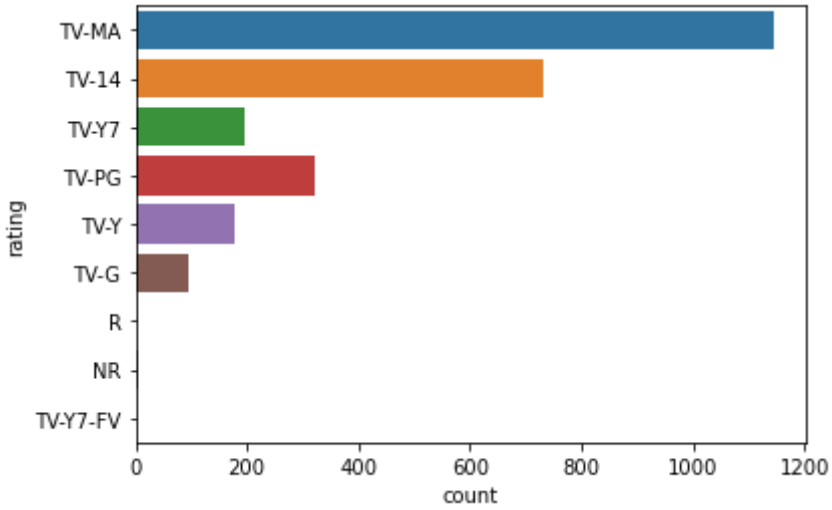
- The graph below shows that most no of TV Shows belong to 1. TV-MA, 2. TV-14, 3. TV-PG

In [210]:

```
1
2 sns.countplot(data= df, y= "rating")
```

Out[210]:

```
<AxesSubplot:xlabel='count', ylabel='rating'>
```



No of TV Shows by min_age

Observations:

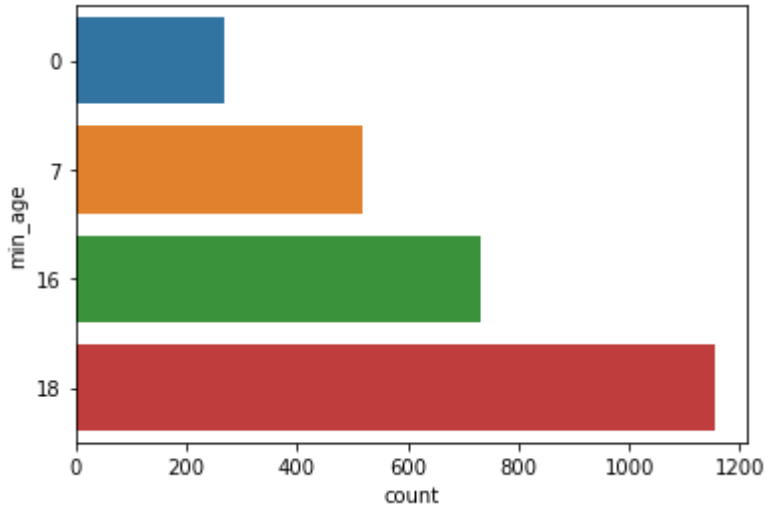
- Most of the TV Shows are for people of age 1. 18+, 2. 16+, 3. 7+
- The least no of shows are for 0+ (Generic)

In [211]:

```
1
2 sns.countplot(data= df, y= "min_age")
```

Out[211]:

```
<AxesSubplot:xlabel='count', ylabel='min_age'>
```



No of TV Shows by duration

Observation:

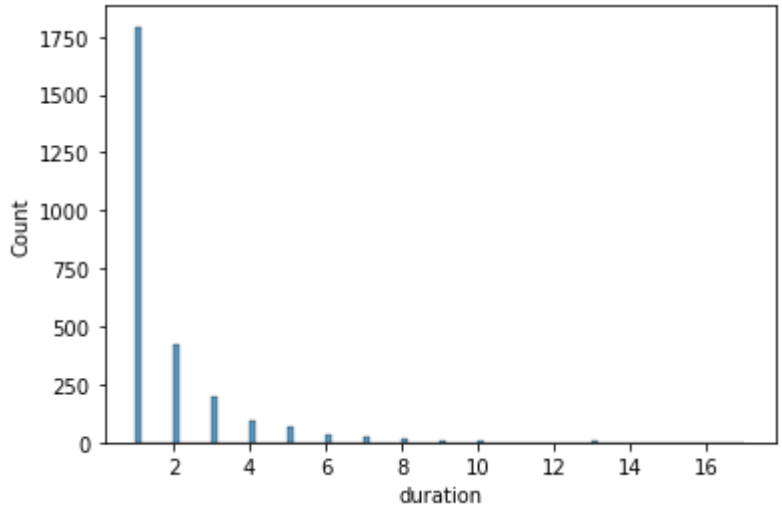
- most of the TV Shows are of 1 season duration

In [212]:

```
1
2 sns.histplot(data= df, x= "duration")
```

Out[212]:

```
<AxesSubplot:xlabel='duration', ylabel='Count'>
```

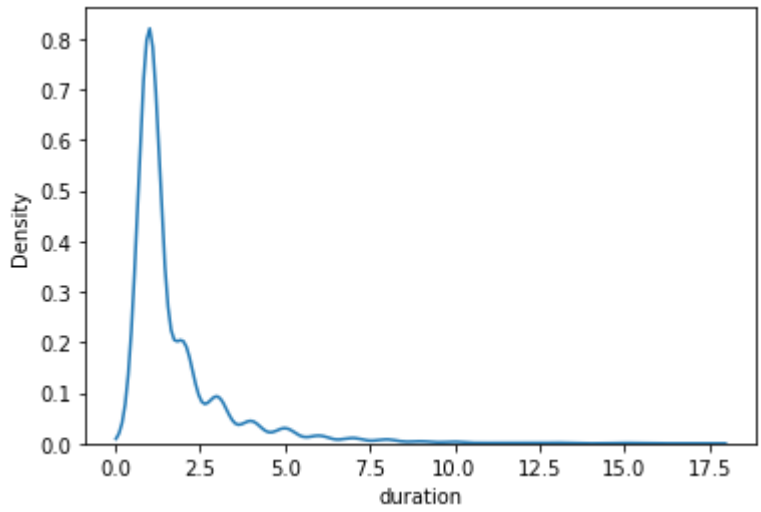


In [213]:

```
1
2 sns.kdeplot(data= df, x= "duration")
```

Out[213]:

```
<AxesSubplot:xlabel='duration', ylabel='Density'>
```



No of TV Shows by release year

Observation:

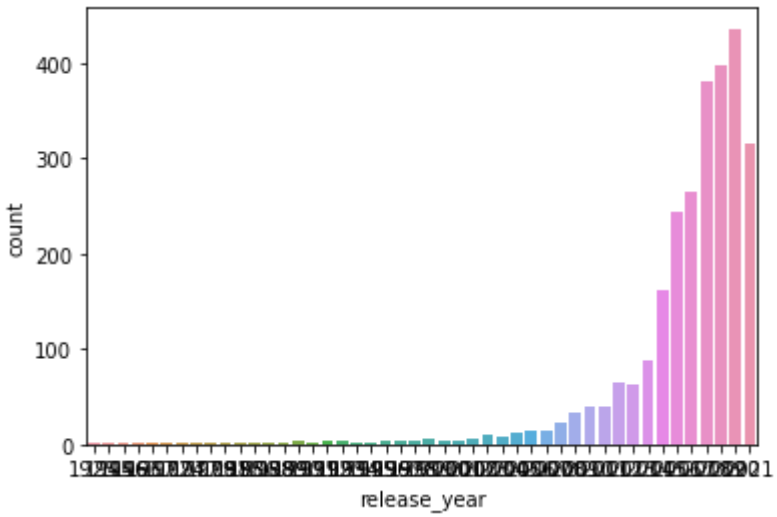
- most of the TV Shows on netflix are released in recent years i.e. very few old TV Shows

In [214]:

```
1
2 sns.countplot(data= df, x= "release_year")
```

Out[214]:

<AxesSubplot:xlabel='release_year', ylabel='count'>

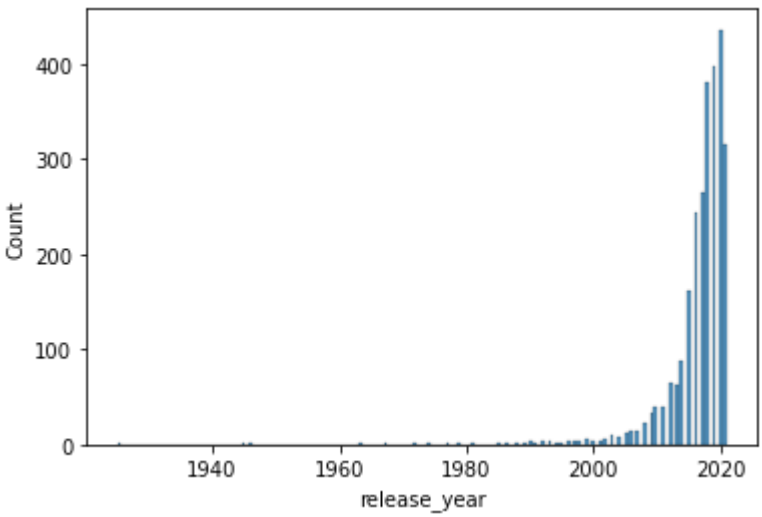


In [215]:

```
1
2 sns.histplot(data= df, x= "release_year")
```

Out[215]:

<AxesSubplot:xlabel='release_year', ylabel='Count'>

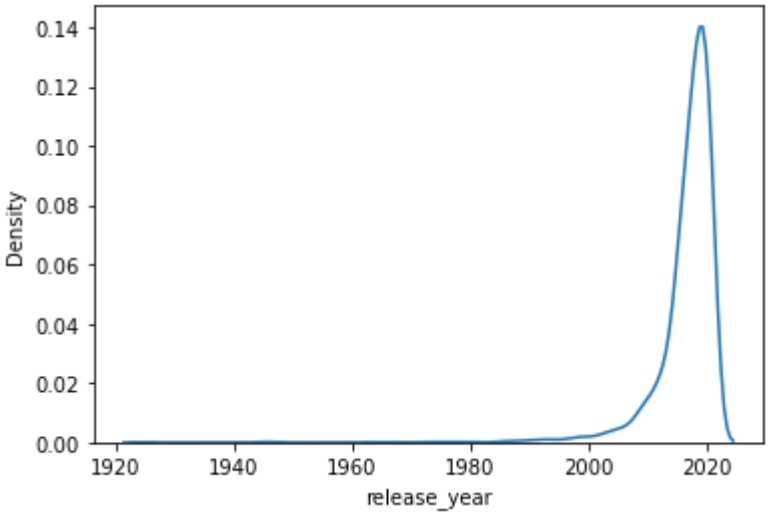


In [216]:

```
1
2 sns.kdeplot(data= df, x= "release_year")
```

Out[216]:

```
<AxesSubplot:xlabel='release_year', ylabel='Density'>
```



No of TV Shows by release year

Observation:

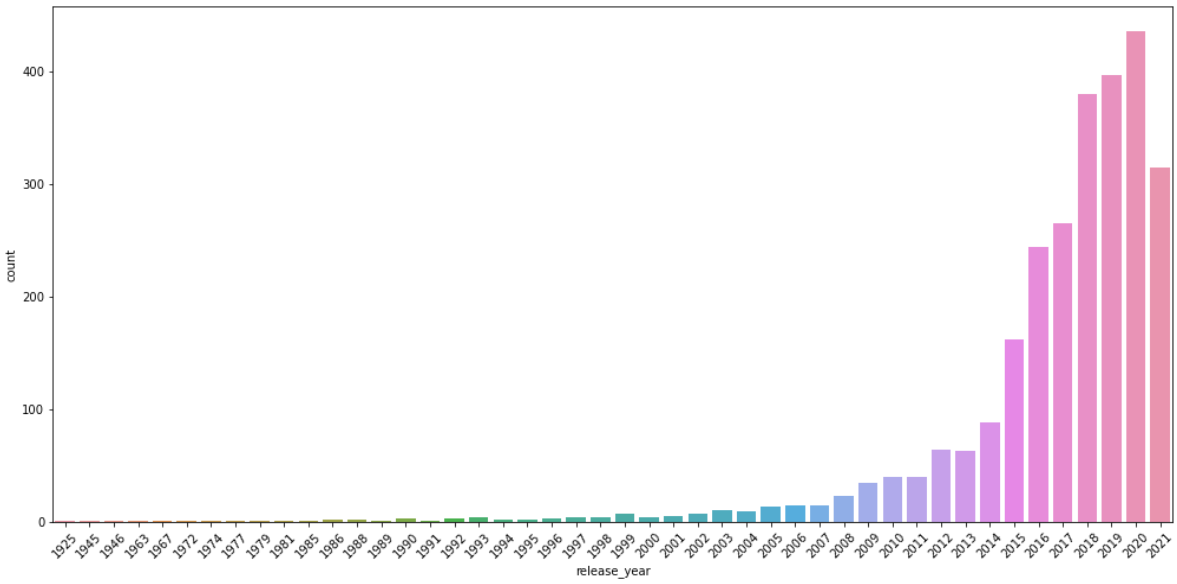
- The no of recently released TV Shows are lesser than the TV Shows released 2 to 3 years ago on Netflix

In [217]:

```
1
2 plt.figure(figsize=(17,8))
3 plt.xticks(rotation=45)
4 sns.countplot(data= df, x= "release_year")
```

Out[217]:

```
<AxesSubplot:xlabel='release_year', ylabel='count'>
```



No of TV Shows by added date

Observation:

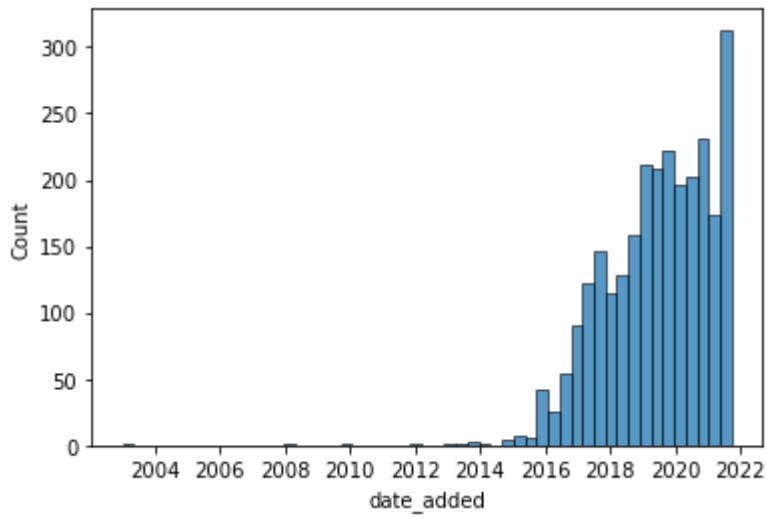
- The no. of TV Shows added increaed rapidly from 2014 to 2019 and then recently stabilized

In [218]:

```
1
2 sns.histplot(data= df, x= "date_added")
```

Out[218]:

```
<AxesSubplot:xlabel='date_added', ylabel='Count'>
```



No of movies by added date

Observation:

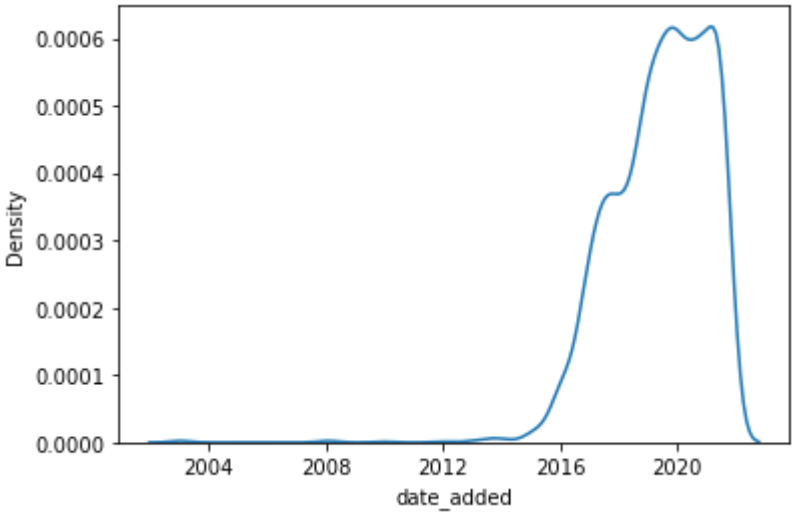
- The no. of movies added increaed rapidly from 2014 to 2016 and then recently stabilized

In [219]:

```
1
2 sns.kdeplot(data= df, x= "date_added")
```

Out[219]:

```
<AxesSubplot:xlabel='date_added', ylabel='Density'>
```



In [220]:

```
df["day_of_year_added"] = df["date_added"].dt.dayofyear
df["day_of_month_added"] = df["date_added"].dt.day
df["day_of_week_added"] = df["date_added"].dt.dayofweek
df["month_added"] = df["date_added"].dt.month
df["quarter_added"] = df["date_added"].dt.quarter
df["year_added"] = df["date_added"].dt.year
df["released_decade"] = df["release_year"].apply(lambda x: x - (x%10))

df.head()
```

Out[220]:

	show_id	type	title	date_added	release_year	rating	duration	min_age	day_of_yea
1	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2	18	
2	s3	TV Show	Ganglands	2021-09-24	2021	TV-MA	1	18	
3	s4	TV Show	Jailbirds New Orleans	2021-09-24	2021	TV-MA	1	18	
4	s5	TV Show	Kota Factory	2021-09-24	2021	TV-MA	2	18	
5	s6	TV Show	Midnight Mass	2021-09-24	2021	TV-MA	1	18	

No of TV Shows by added date

Observation:

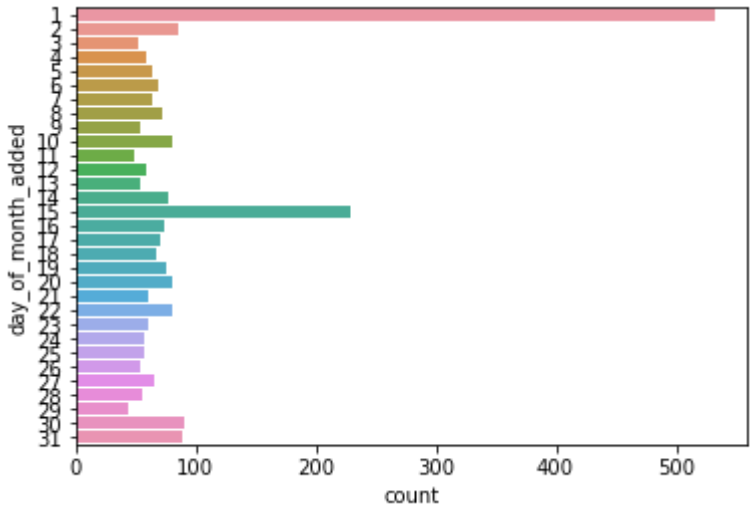
- Most TV Shows are added on the start of the month or the middle of the month
- no. of TV Shows added on the rest of the days of the month is similar

In [221]:

```
sns.countplot(data= df, y= "day_of_month_added")
```

Out[221]:

```
<AxesSubplot:xlabel='count', ylabel='day_of_month_added'>
```



No of TV Shows by added day of week

Observation:

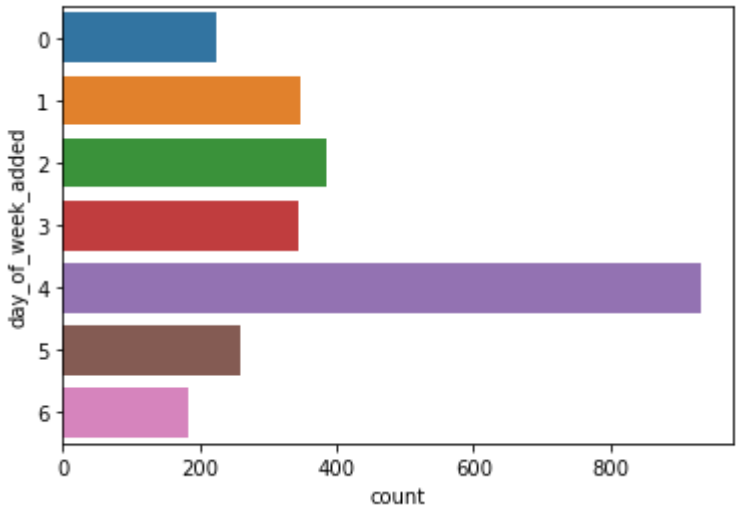
- Most TV Shows are added on the 5th day of the week (Friday) just before weekend

In [222]:

```
sns.countplot(data= df, y= "day_of_week_added")
```

Out[222]:

```
<AxesSubplot:xlabel='count', ylabel='day_of_week_added'>
```



No of TV Shows by added month of the year

Observation:

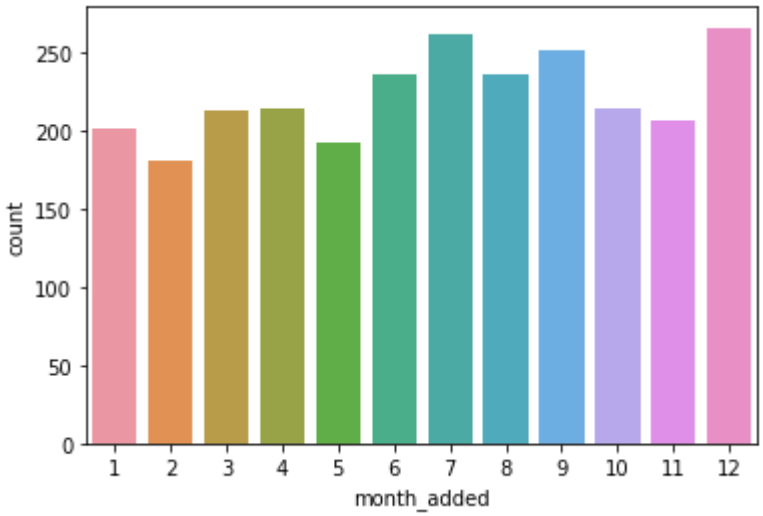
- the no. of shows added in the middle of th eyaer and end of the year tends to be higher and increasingly so every year

In [223]:

```
1
2 sns.countplot(data= df, x= "month_added")
```

Out[223]:

<AxesSubplot:xlabel='month_added', ylabel='count'>

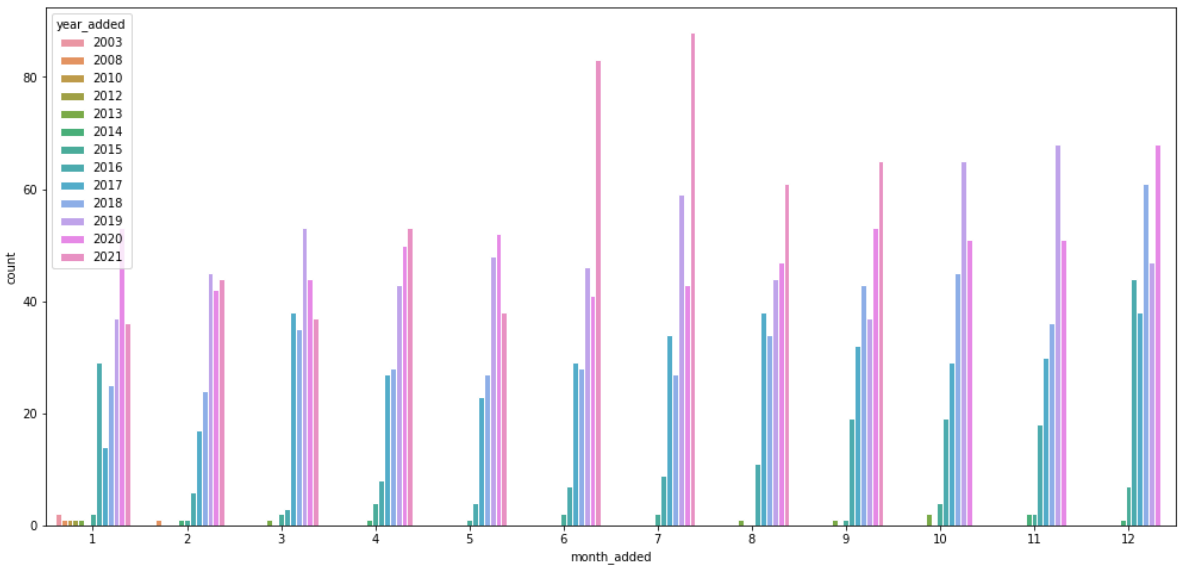


In [224]:

```
1
2 plt.figure(figsize= (17, 8))
3
4 sns.countplot(data= df, x= "month_added", hue= "year_added", edgecolor= "white")
```

Out[224]:

<AxesSubplot:xlabel='month_added', ylabel='count'>



Bivariate Analysis

No of movies by added quarter of the year

Observation:

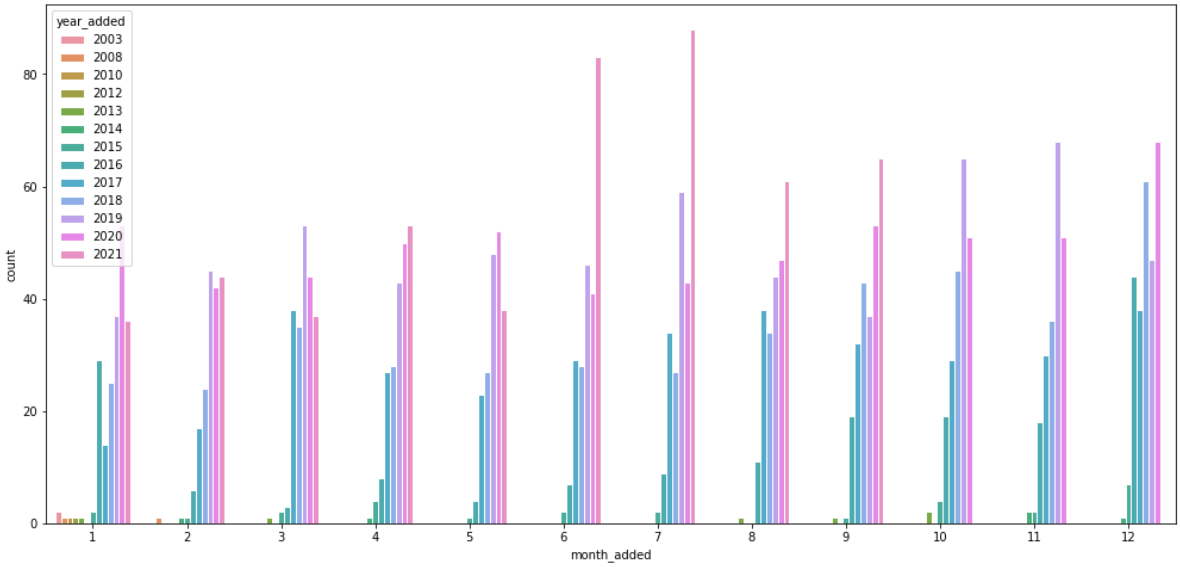
- The no. of movies added is moderately high in the 4th quarter
- the no. of movies added on other quarters are similar

In [225]:

```
1
2 plt.figure(figsize= (17, 8))
3
4 sns.countplot(data= df, x= "month_added", hue= "year_added", edgecolor= "white")
```

Out[225]:

```
<AxesSubplot:xlabel='month_added', ylabel='count'>
```

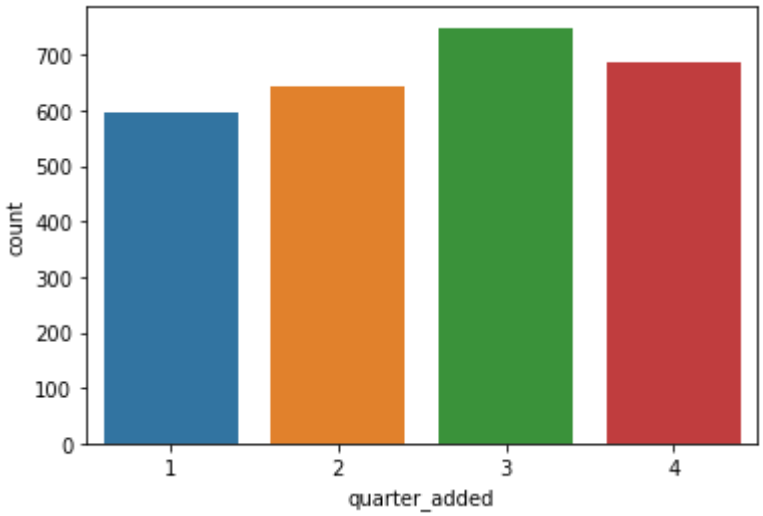


In [226]:

```
1
2 sns.countplot(data= df, x= "quarter_added")
```

Out[226]:

```
<AxesSubplot:xlabel='quarter_added', ylabel='count'>
```

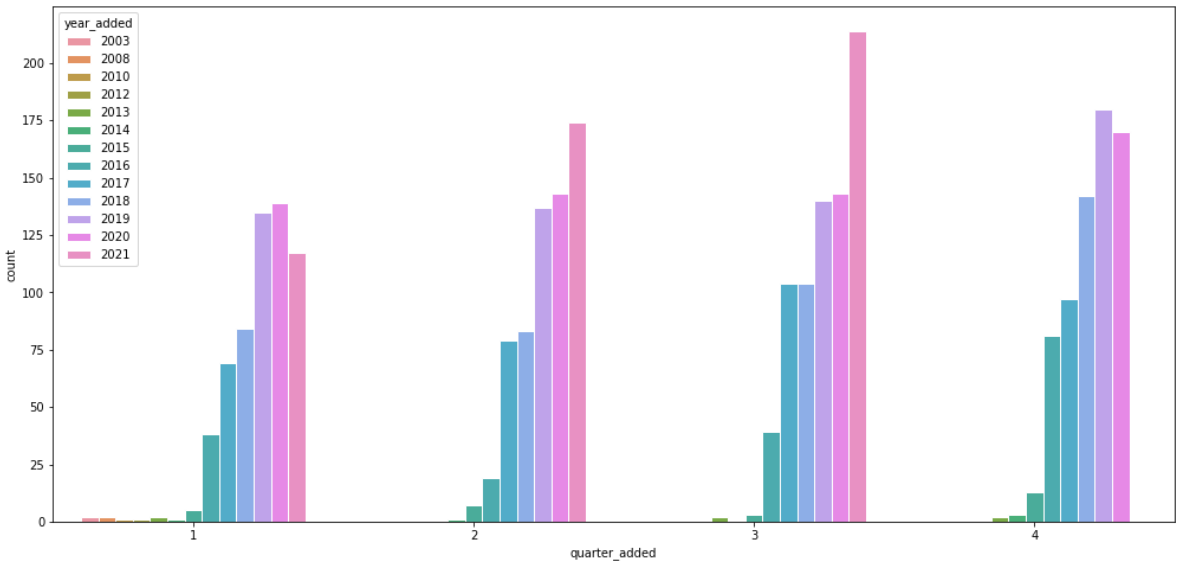


In [227]:

```
1
2 plt.figure(figsize= (17, 8))
3
4 sns.countplot(data= df, x= "quarter_added", hue= "year_added", edgecolor= "white")
```

Out[227]:

```
<AxesSubplot:xlabel='quarter_added', ylabel='count'>
```



distribution of duration of movies

Observation:

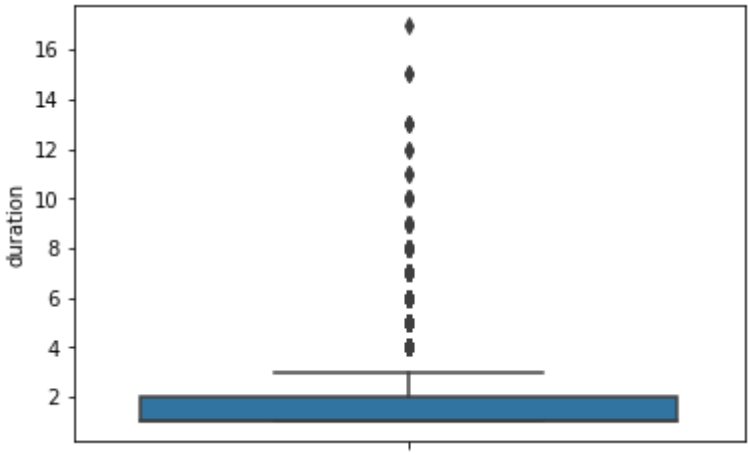
- most movies are around 1 season long
- There are only a few TV Shows that have more than 3 seasons

In [228]:

```
1
2 sns.boxplot(data= df, y= "duration")
```

Out[228]:

```
<AxesSubplot:ylabel='duration'>
```

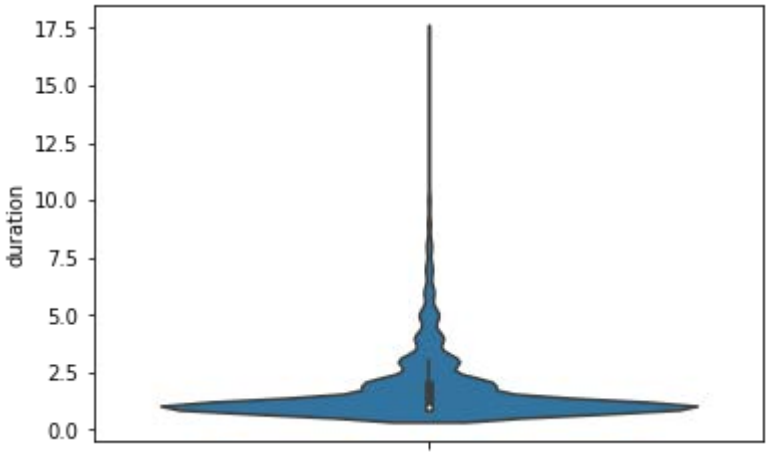


In [229]:

```
1
2 sns.violinplot(data= df, y= "duration")
```

Out[229]:

```
<AxesSubplot:ylabel='duration'>
```



distribution of release year of TV Shows

Observation:

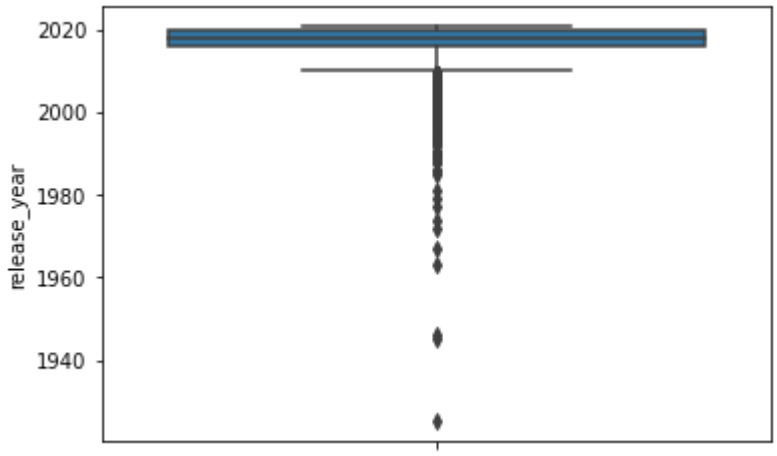
- most TV Shows are released during 2010 - 2020
- There are only a few old TV Shows

In [230]:

```
1
2 sns.boxplot(data= df, y= "release_year")
```

Out[230]:

```
<AxesSubplot:ylabel='release_year'>
```

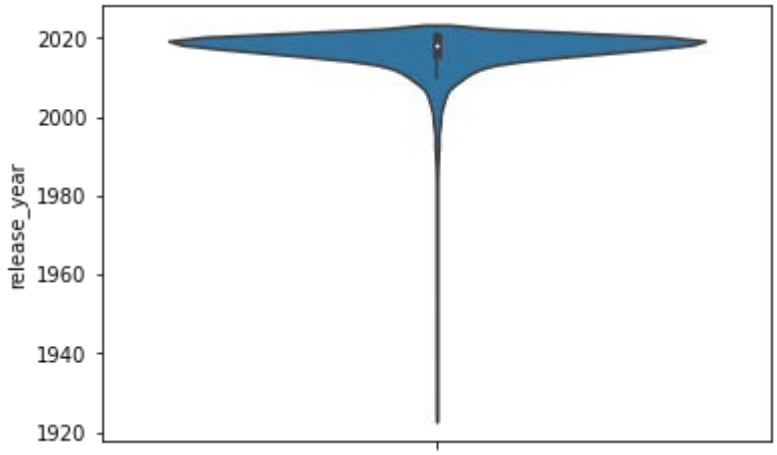


In [231]:

```
1
2 sns.violinplot(data= df, y= "release_year")
```

Out[231]:

```
<AxesSubplot:ylabel='release_year'>
```



distribution of day of month added of TV Shows

Observation:

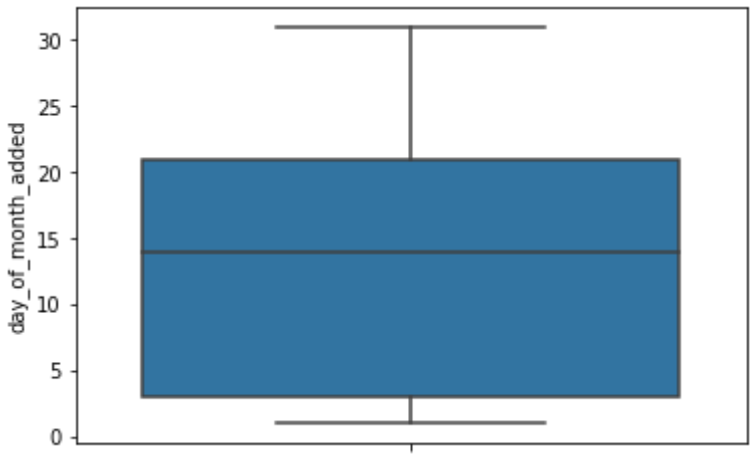
- most TV Sohws are added during the first half of th month

In [232]:

```
1
2 sns.boxplot(data= df, y= "day_of_month_added")
```

Out[232]:

```
<AxesSubplot:ylabel='day_of_month_added'>
```

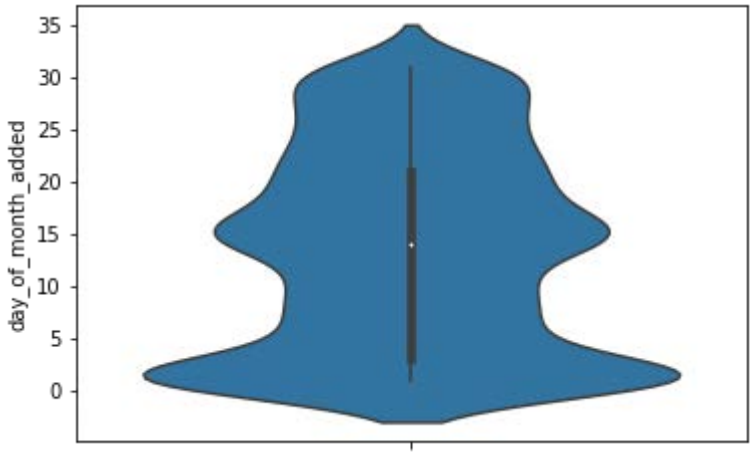


In [233]:

```
1
2 sns.violinplot(data= df, y= "day_of_month_added")
```

Out[233]:

```
<AxesSubplot:ylabel='day_of_month_added'>
```



distribution of added year of TV Shows

Observation:

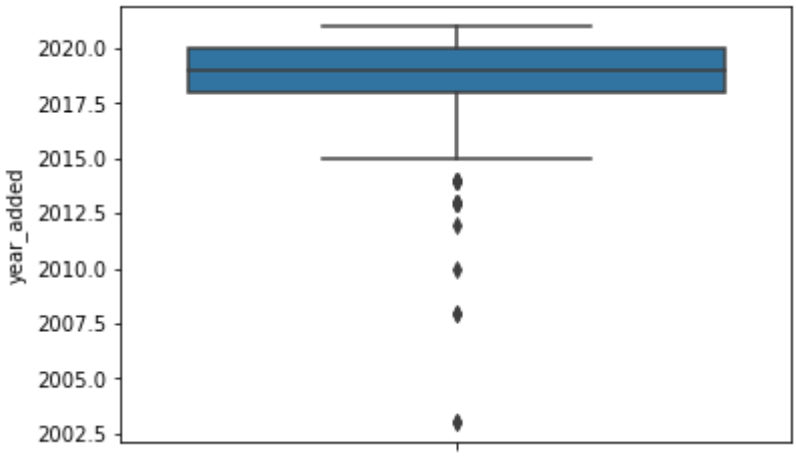
- most TV Shows are added during the last couple years (after 2018)
- the no. of TV Shows on netflix before 2015 were very low

In [234]:

```
1
2 sns.boxplot(data= df, y= "year_added")
```

Out[234]:

```
<AxesSubplot:ylabel='year_added'>
```

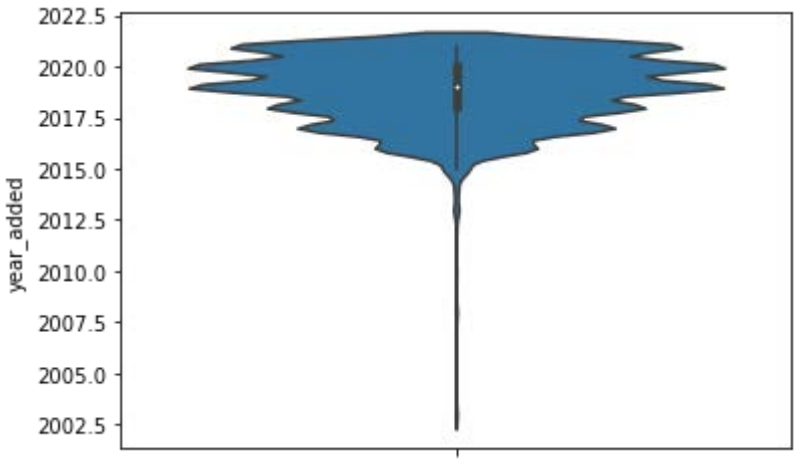


In [235]:

```
1
2 sns.violinplot(data= df, y= "year_added")
```

Out[235]:

<AxesSubplot:ylabel='year_added'>



distribution of added day of week for TV Shows

Observation:

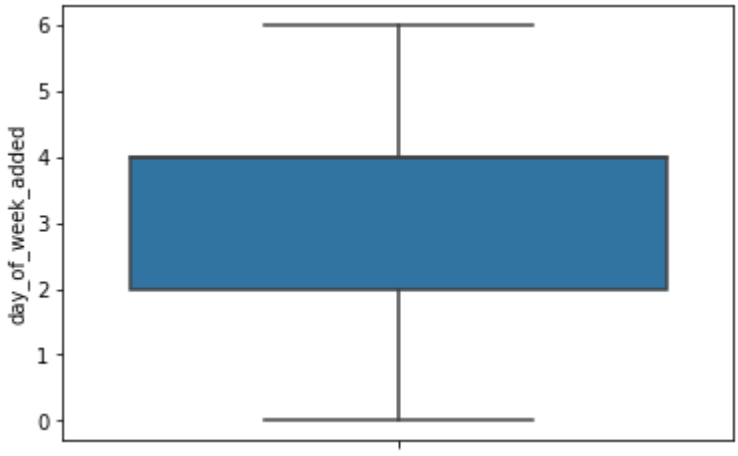
- most TV Shows are added in the middle of the week

In [236]:

```
1
2 sns.boxplot(data= df, y= "day_of_week_added")
```

Out[236]:

<AxesSubplot:ylabel='day_of_week_added'>



no. of TV Shows by year added

Observation:

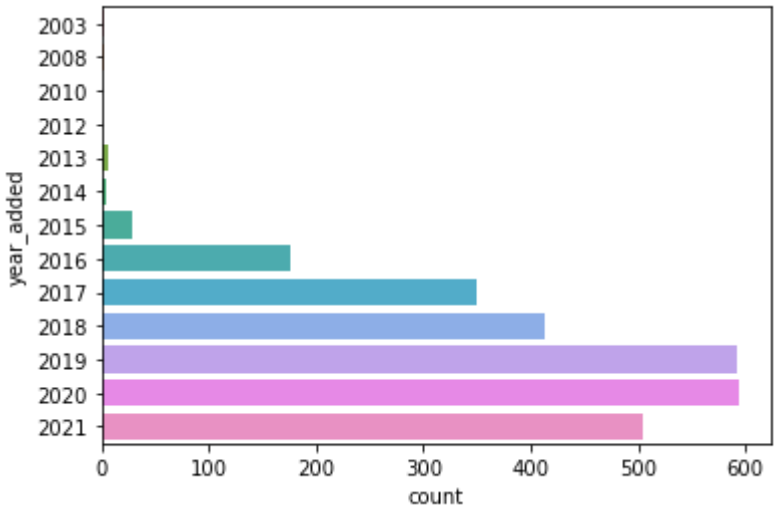
- most TV Shows are added in the last couple years (after 2018)
- the no of TV Shows aded per yaer is decreasing over the last few years (since 2020)
- the no of TV Shows added was increasing from 2014 - 2019

In [237]:

```
1
2 sns.countplot(data= df, y= "year_added")
```

Out[237]:

```
<AxesSubplot:xlabel='count', ylabel='year_added'>
```

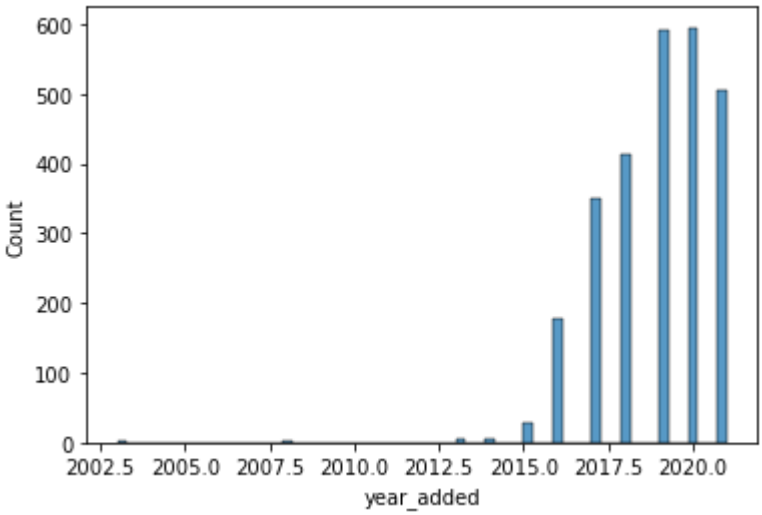


In [238]:

```
1
2 sns.histplot(data= df, x= "year_added")
```

Out[238]:

```
<AxesSubplot:xlabel='year_added', ylabel='Count'>
```



no. of TV Shows by year added and month of year

Observation:

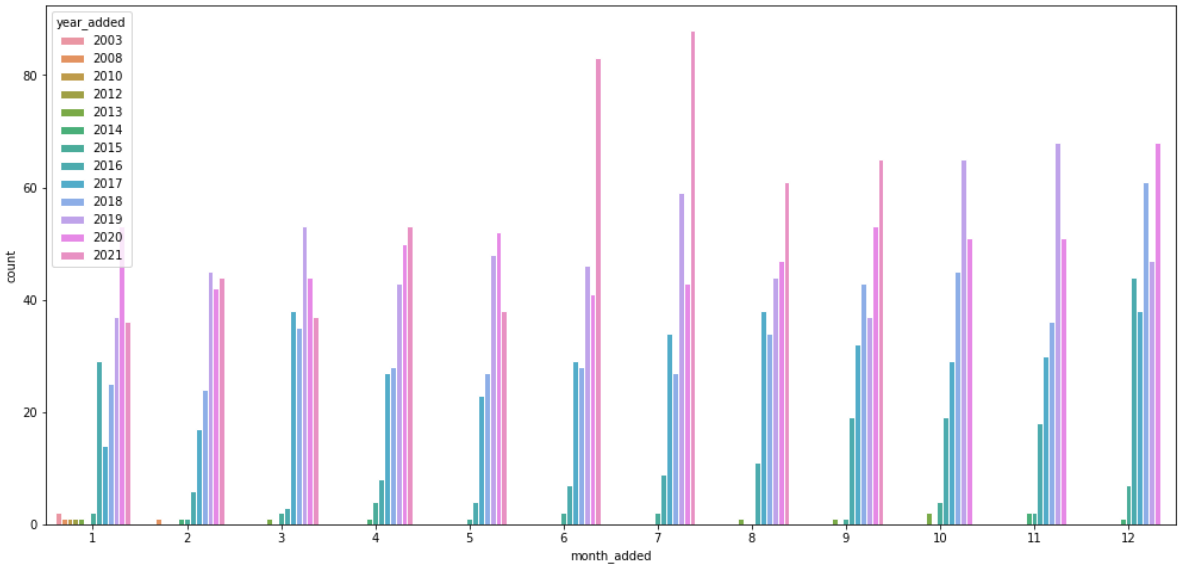
- the no of TV Shows added each month is increasing every year

In [239]:

```
plt.figure(figsize= (17, 8))

sns.countplot(data= df, x= "month_added", hue= "year_added", edgecolor= "white")
```

Out[239]:

```
<AxesSubplot:xlabel='month_added', ylabel='count'>
```



no. of TV Shows by release decade

Observation:

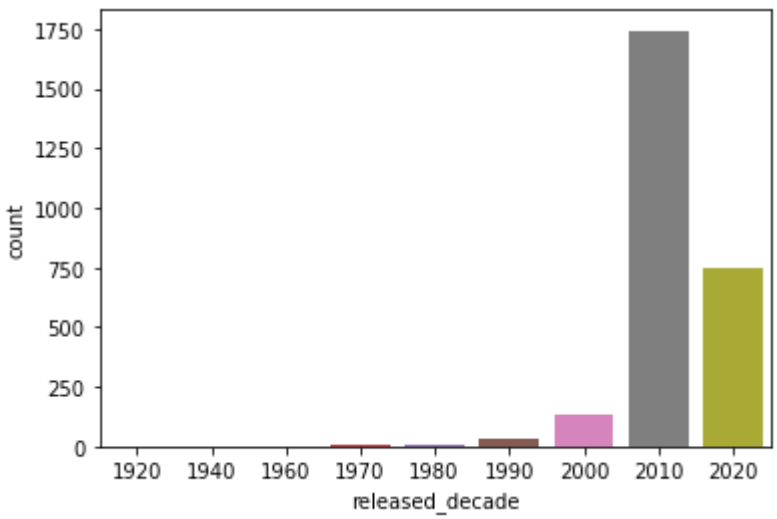
- Most TV Shows on netflix are from the 2010's

In [240]:

```
sns.countplot(data= df, x= "released_decade")
```

Out[240]:

```
<AxesSubplot:xlabel='released_decade', ylabel='count'>
```

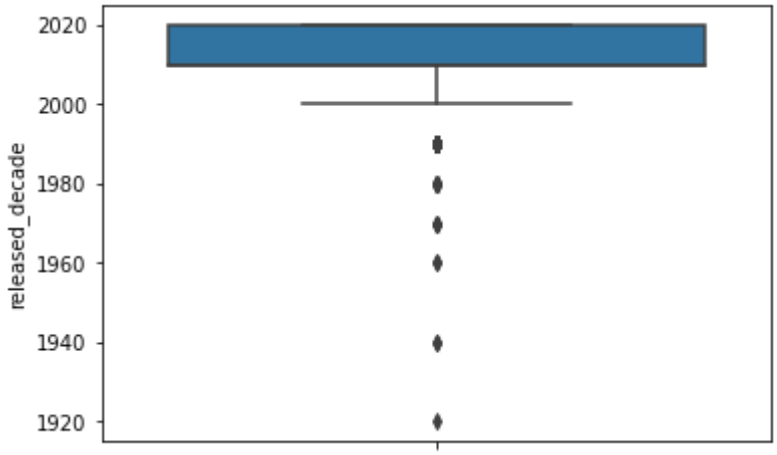


In [241]:

```
1
2 sns.boxplot(data= df, y= "released_decade")
```

Out[241]:

<AxesSubplot:ylabel='released_decade'>



duration of TV Shows by release_year

Observation:

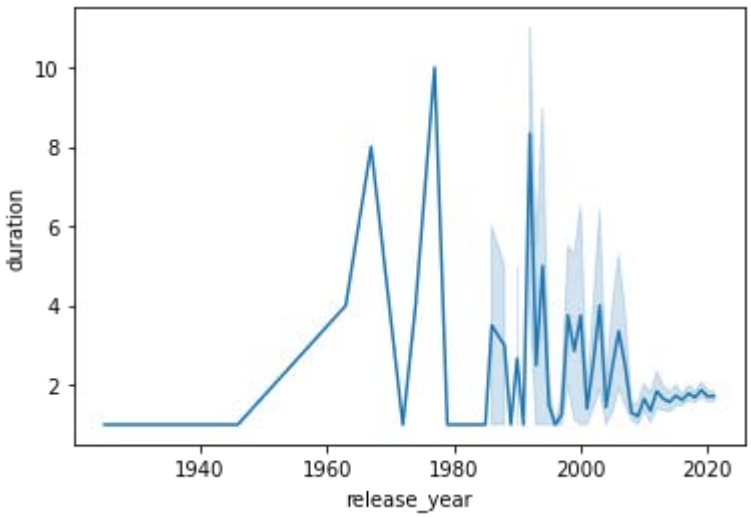
- The few TV Shows (on Netflix) that released during the 90's and 2000's were longer in season duration
- Rest of the TV Shows were around 1 season long irrespective their release yaer

In [242]:

```
1
2 sns.lineplot(data= df, x= "release_year", y= "duration")
```

Out[242]:

<AxesSubplot:xlabel='release_year', ylabel='duration'>

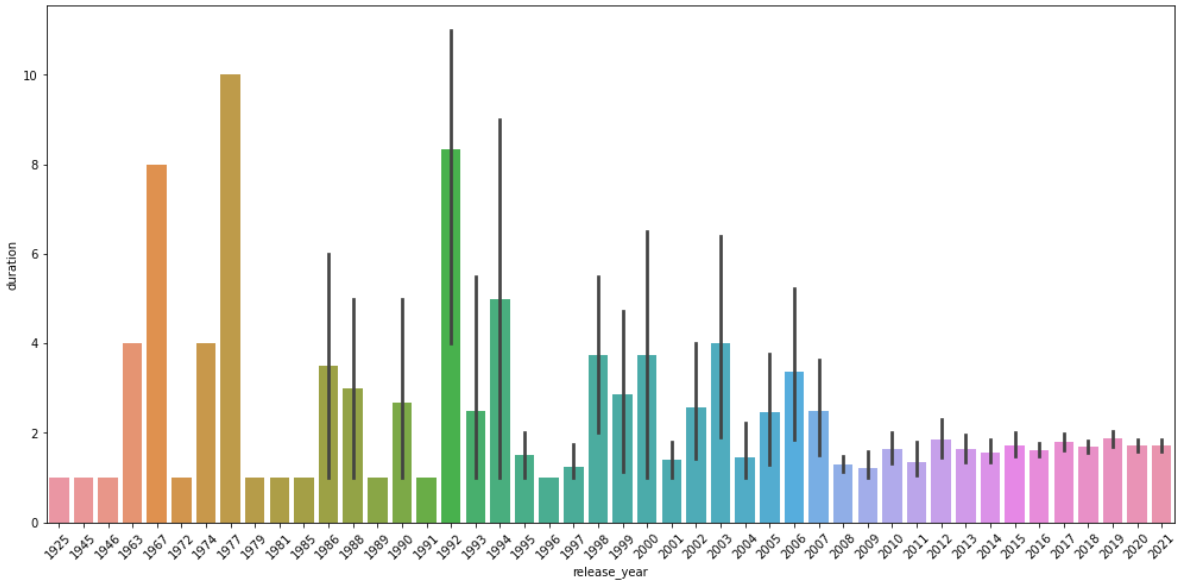


In [243]:

```
1
2 plt.figure(figsize=(17,8))
3 plt.xticks(rotation=45)
4 sns.barplot(data= df, x= "release_year", y= "duration", estimator= np.mean)
```

Out[243]:

<AxesSubplot:xlabel='release_year', ylabel='duration'>

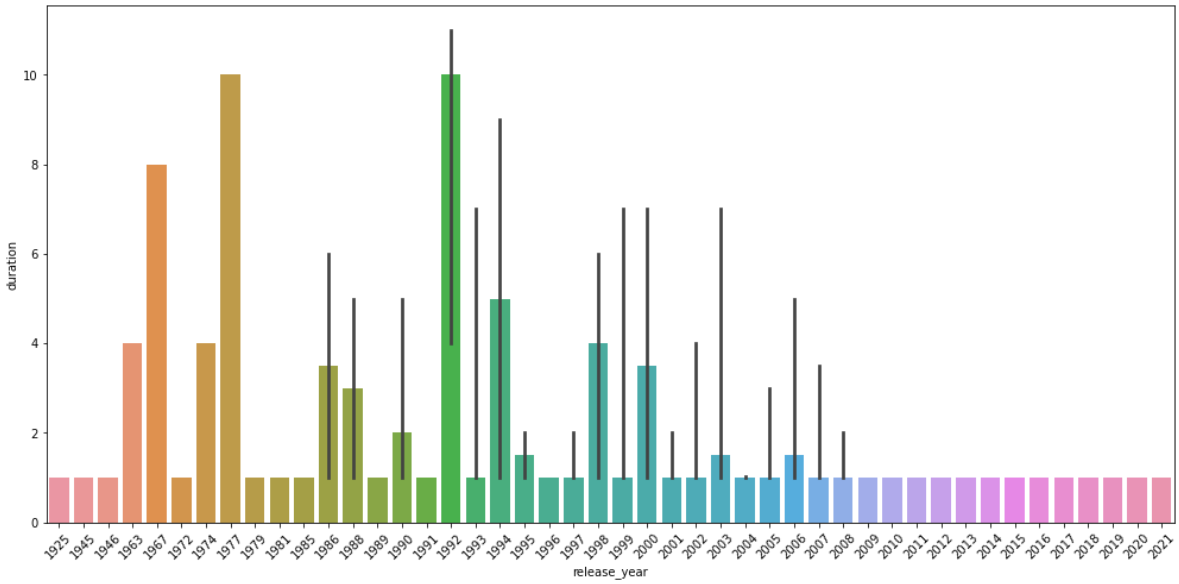


In [244]:

```
1
2 plt.figure(figsize=(17,8))
3 plt.xticks(rotation=45)
4 sns.barplot(data= df, x= "release_year", y= "duration", estimator= np.median)
```

Out[244]:

<AxesSubplot:xlabel='release_year', ylabel='duration'>



duration of TV Shows by release_decade

Observation:

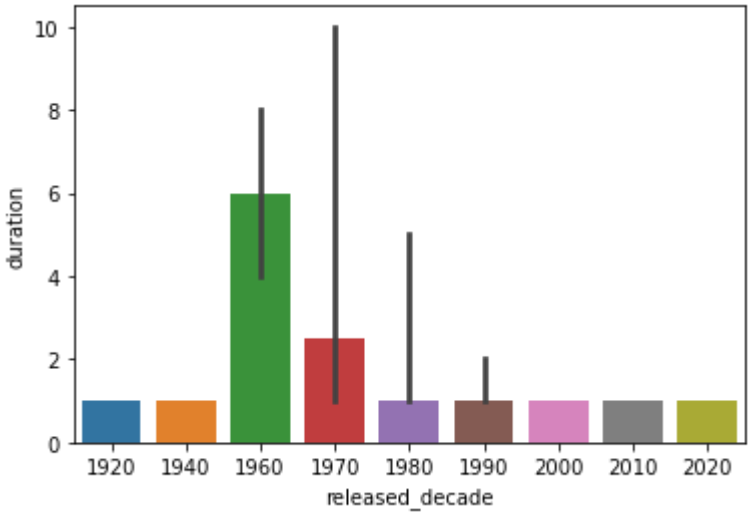
- the few TV Shows (on Netflix) that released before 2000 were longer in duration

In [245]:

```
1
2 sns.barplot(data= df, x= "released_decade", y= "duration", estimator= np.median)
```

Out[245]:

```
<AxesSubplot:xlabel='released_decade', ylabel='duration'>
```

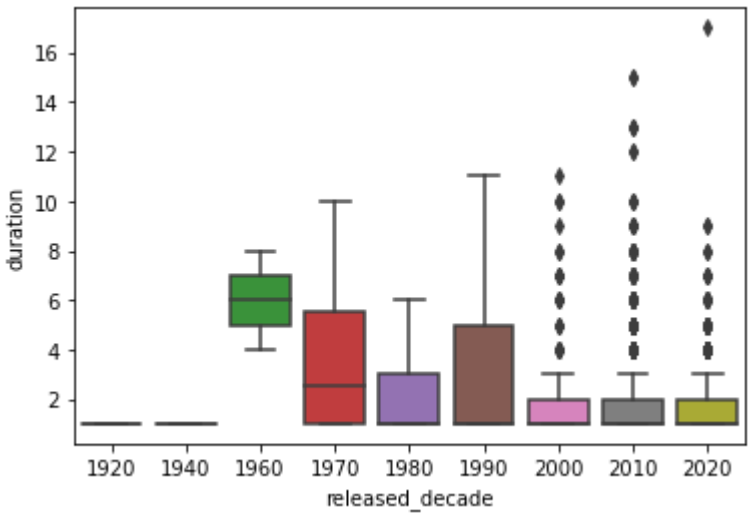


In [246]:

```
1
2 sns.boxplot(data= df, x= "released_decade", y= "duration")
```

Out[246]:

```
<AxesSubplot:xlabel='released_decade', ylabel='duration'>
```

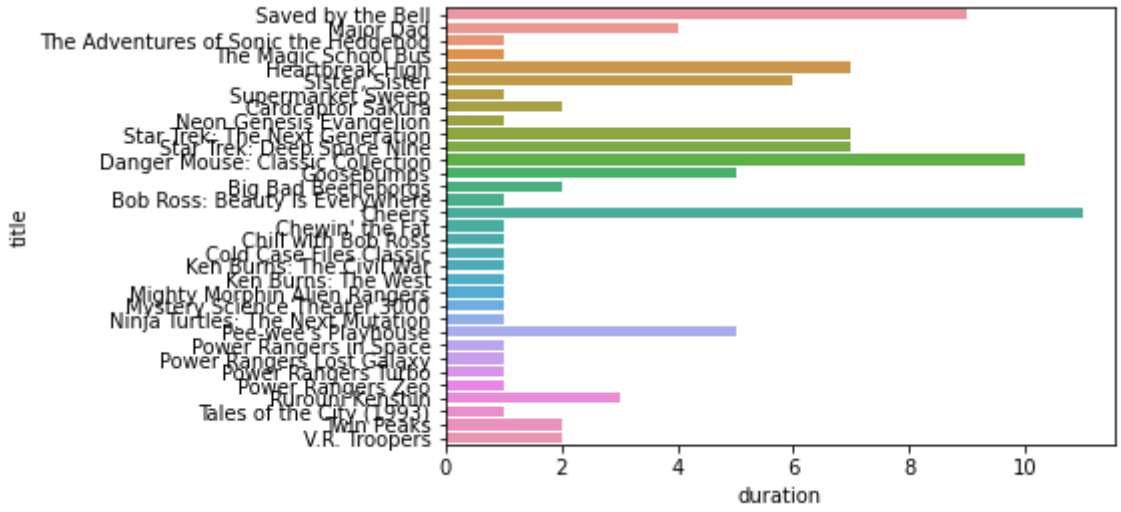


In [247]:

```
1
2 sns.barplot(data= df[df.released_decade == 1990], y= "title", x= "duration")
```

Out[247]:

```
<AxesSubplot:xlabel='duration', ylabel='title'>
```



duration of movies by rating

Observation:

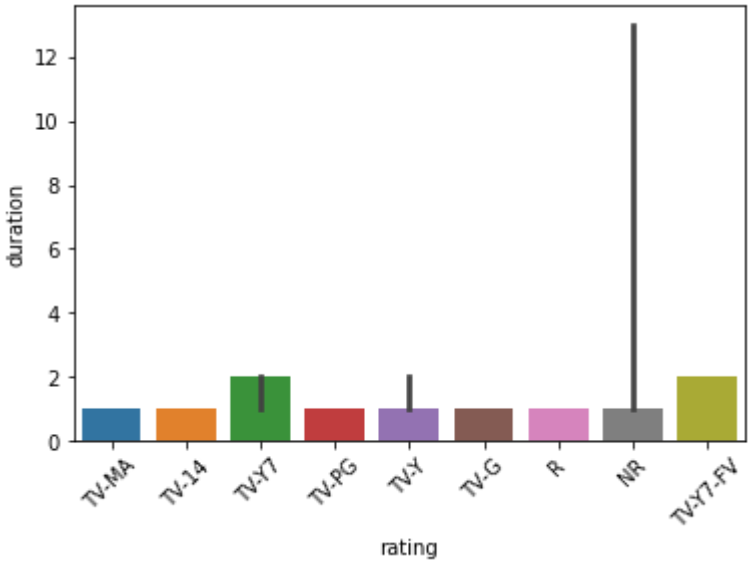
- the movies of rating TV-Y, TV-Y7, TV-Y7-FY are of longer season duration (for kids maybe)

In [248]:

```
1
2 plt.xticks(rotation = 45)
3 sns.barplot(data= df, x= "rating", y= "duration", estimator= np.median)
```

Out[248]:

<AxesSubplot:xlabel='rating', ylabel='duration'>

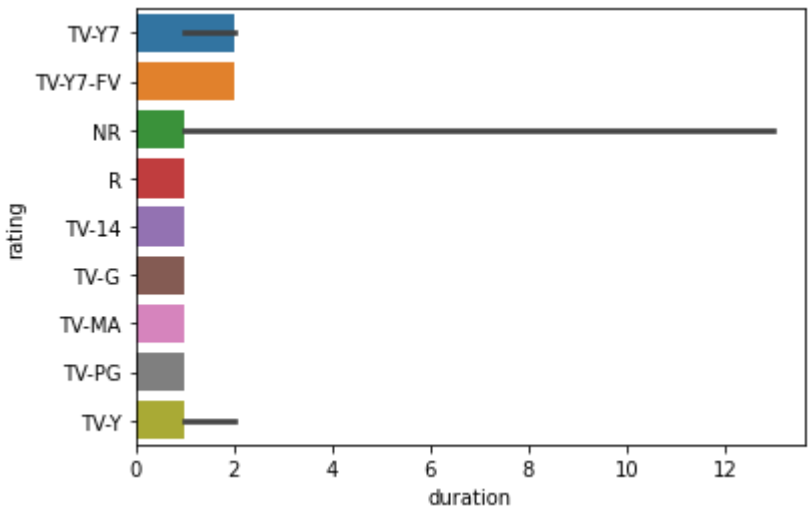


In [249]:

```
1
2 sns.barplot(data= df, y= "rating", x= "duration", estimator= np.median,
3             order = df.groupby(["rating"]).median().sort_values(["duration"], ascending
```

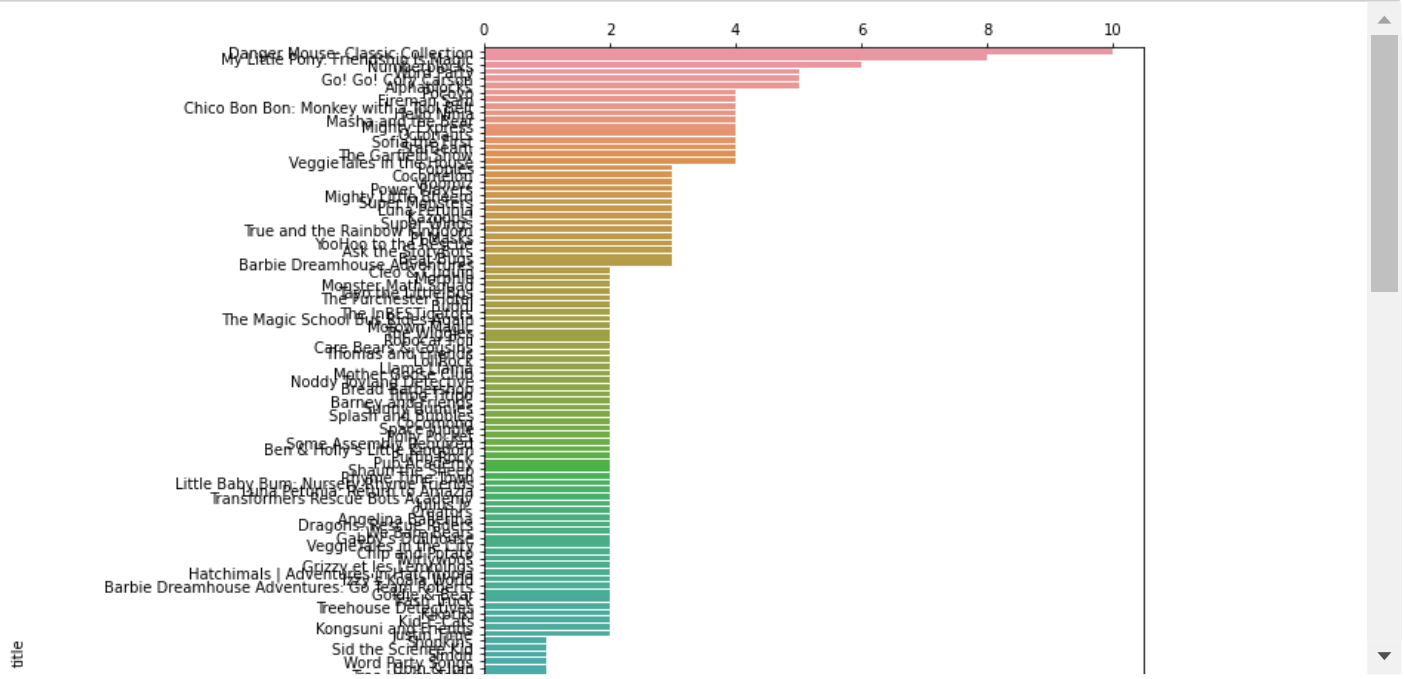
Out[249]:

<AxesSubplot:xlabel='duration', ylabel='rating'>



In [250]:

```
1
2 plt.figure(figsize= (8, 15))
3
4 ax = sns.barplot(data= df[df.rating == "TV-Y"], y= "title", x= "duration", estimator= r
5             order = df[df.rating == "TV-Y"].groupby(["title"]).median().sort_values(["d
6 ax.xaxis.tick_top()
```



year added by release yaer

Observation:

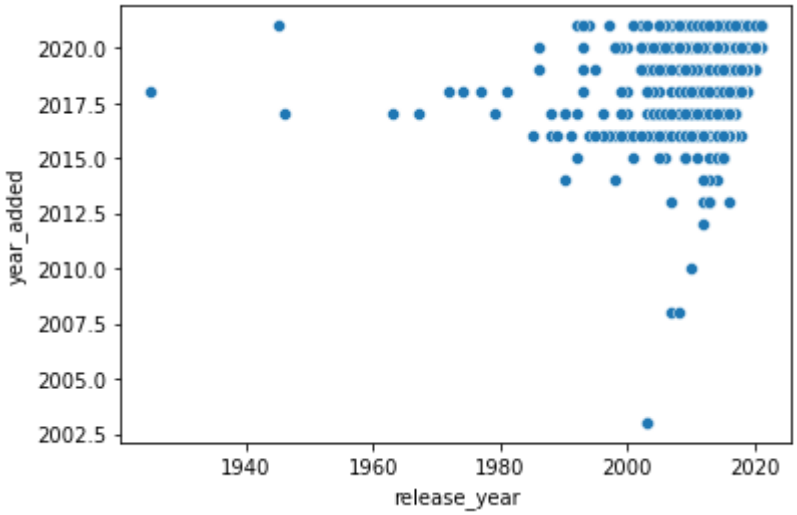
- most TV Shows are added in last few years (after 2018)
- most of the TV Shows that are added released in 2000's and 2010's

In [251]:

```
sns.scatterplot(data= df, x= "release_year", y= "year_added")
```

Out[251]:

```
<AxesSubplot:xlabel='release_year', ylabel='year_added'>
```



total duration of content added by added day of week

Observation:

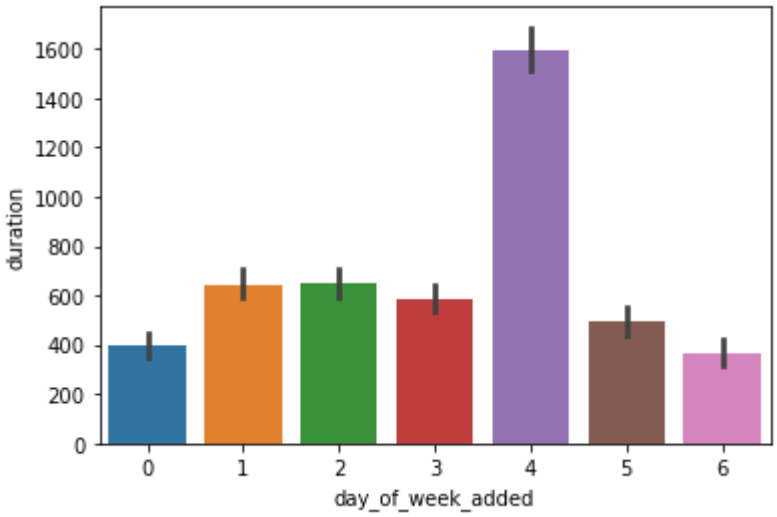
- the total amount of seasons of TV Shows added is more on Fridays

In [252]:

```
sns.barplot(data= df, x= "day_of_week_added", y= "duration", estimator= np.sum)
```

Out[252]:

```
<AxesSubplot:xlabel='day_of_week_added', ylabel='duration'>
```



total duration of content added by added day of week and year

Observation:

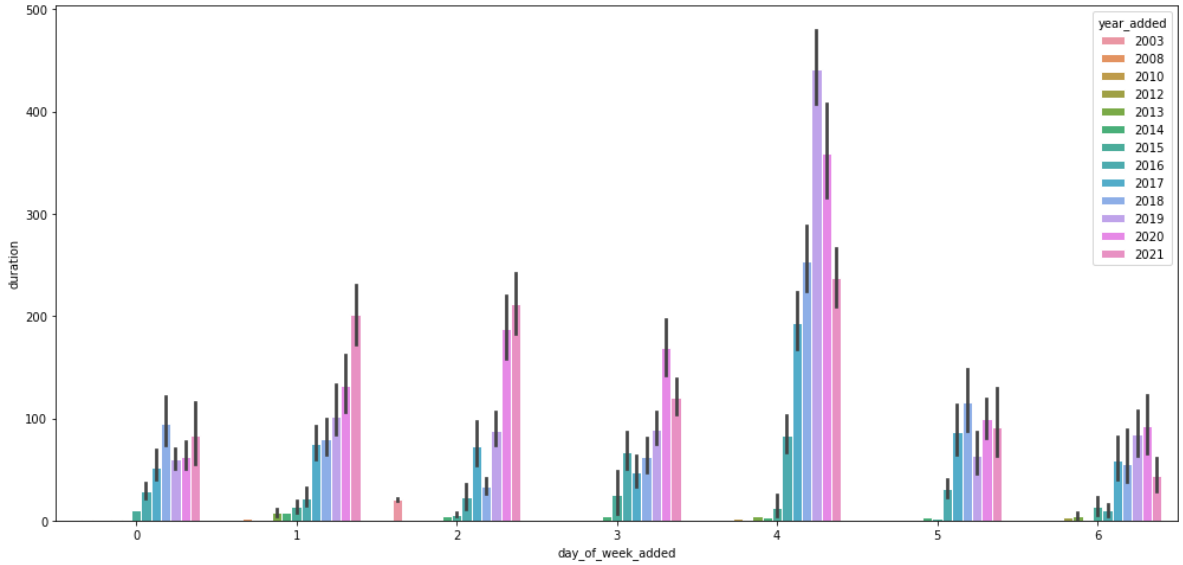
- the total amount of seasons of TV Shows added on weekdays had been increasing every year
- The total amount of seasons of TV shows added on weekend has been around the same for last few years

In [253]:

```
1
2 plt.figure(figsize= (17, 8))
3
4 sns.barplot(data= df, x= "day_of_week_added", y= "duration", hue= "year_added", estimat
```

Out[253]:

```
<AxesSubplot:xlabel='day_of_week_added', ylabel='duration'>
```



no. of TV shows greater than 4 seasons long vs release_decade

Observation:

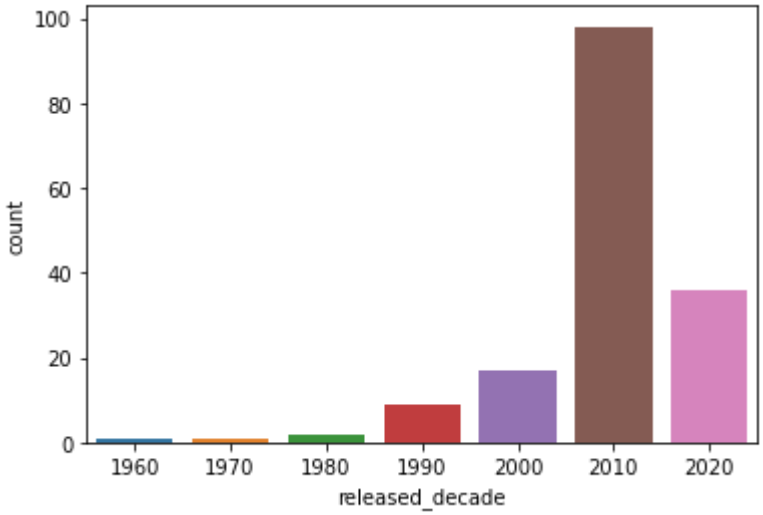
- The no. of TV Shows that are greater than 4 seasons are also belong to the 2010's

In [254]:

```
1
2 sns.countplot(data= df[df.duration > 4], x= "released_decade")
```

Out[254]:

```
<AxesSubplot:xlabel='released_decade', ylabel='count'>
```



total seasons vs added day of month

Observation:

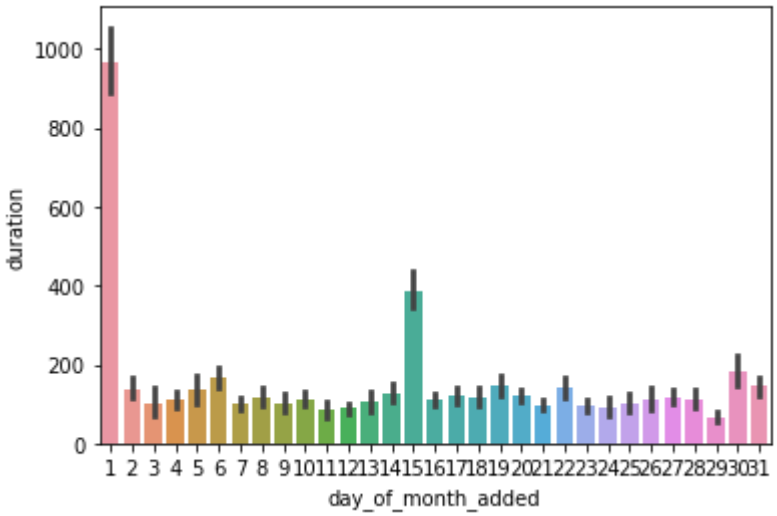
- The total seasons of content that was added on start/middle of a given month is significantly higher than that of any day of month

In [255]:

```
1
2 sns.barplot(data= df, x= "day_of_month_added", y= "duration", estimator= np.sum)
```

Out[255]:

```
<AxesSubplot:xlabel='day_of_month_added', ylabel='duration'>
```



release_year vs rating

Observation:

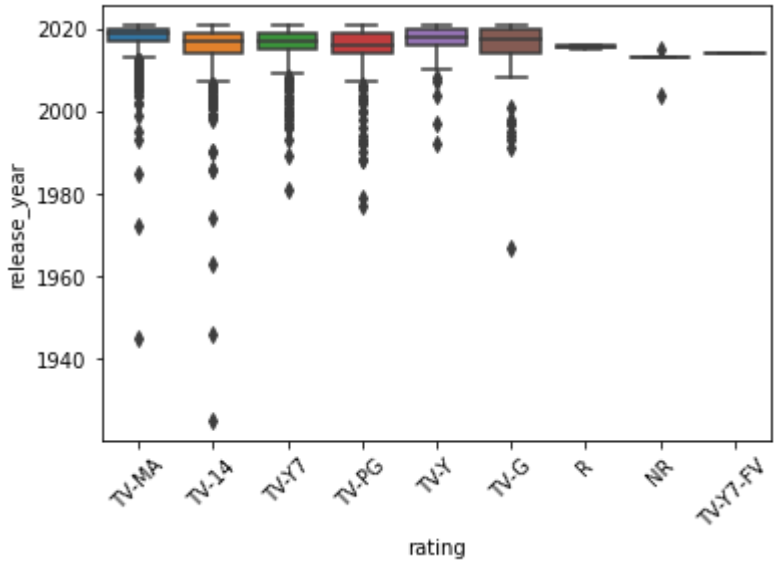
- The relese_year is independent of the rating of the TV Show

In [256]:

```
1
2 plt.xticks(rotation= 45)
3 sns.boxplot(data= df, x= "rating", y= "release_year")
```

Out[256]:

```
<AxesSubplot:xlabel='rating', ylabel='release_year'>
```

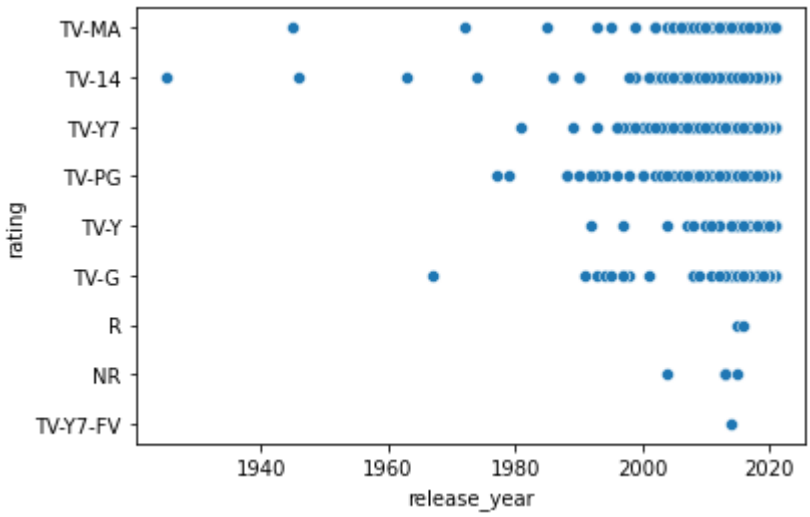


In [257]:

```
1
2 sns.scatterplot(data= df, y= "rating", x= "release_year")
```

Out[257]:

```
<AxesSubplot:xlabel='release_year', ylabel='rating'>
```



added day of month/ week vs rating

Observation:

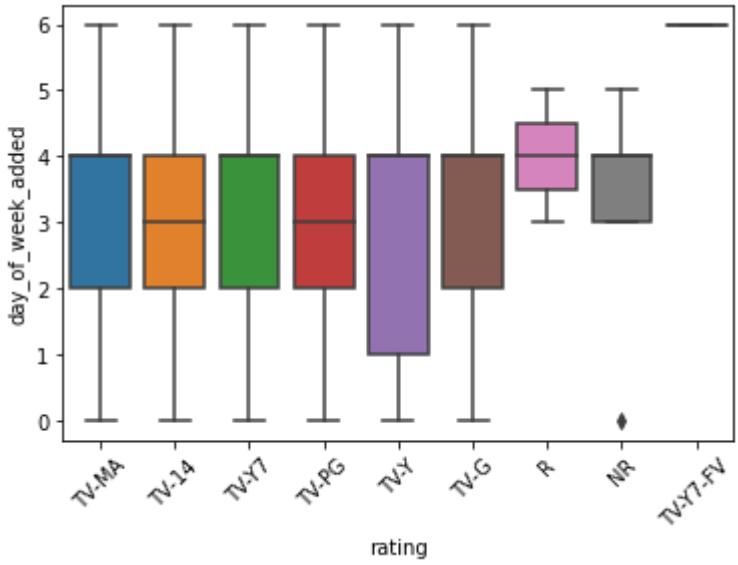
- the added day of week or month is independent of the rating of the TV Show

In [258]:

```
plt.xticks(rotation= 45)
sns.boxplot(data= df, x= "rating", y= "day_of_week_added")
```

Out[258]:

```
<AxesSubplot:xlabel='rating', ylabel='day_of_week_added'>
```

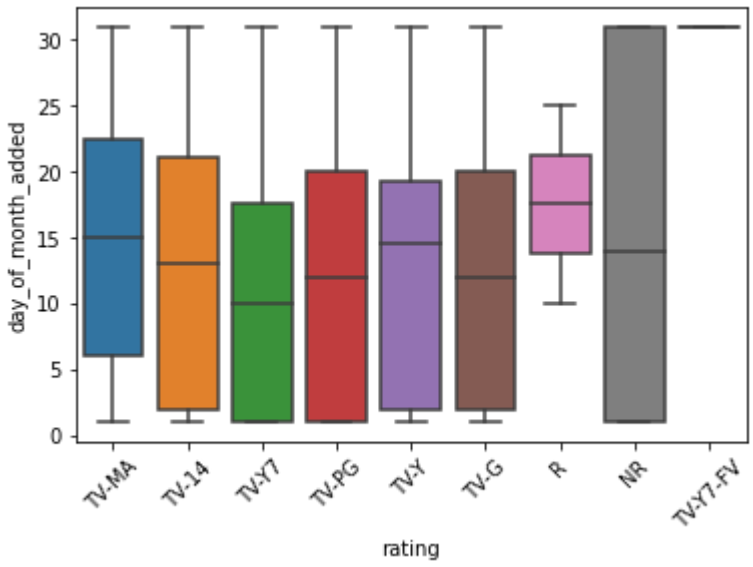


In [259]:

```
plt.xticks(rotation= 45)
sns.boxplot(data= df, x= "rating", y= "day_of_month_added")
```

Out[259]:

```
<AxesSubplot:xlabel='rating', ylabel='day_of_month_added'>
```



distribution of duration vs min_age

Observation:

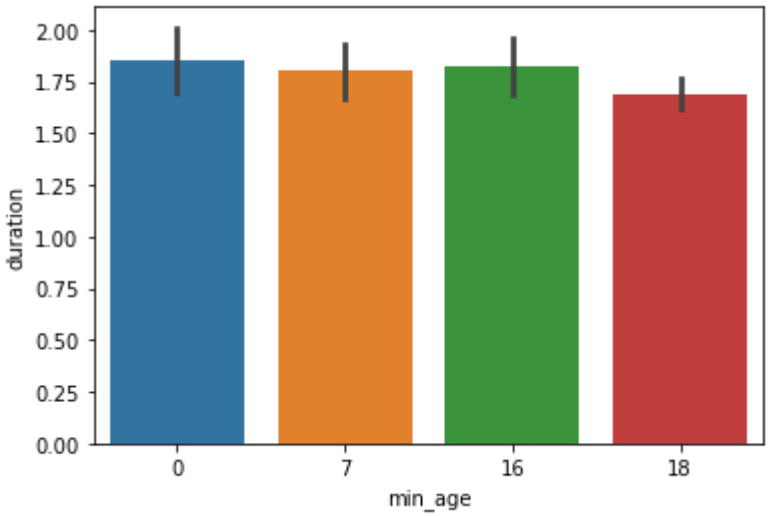
- the mean duration of the movies for 16+, 13+, 18+ is longer than for other age groups
- the duration for the genreic age group (0+) is lower than for any other age group
- the total duration of content is the highest for 18+, 16+ and 7+ age groups

In [260]:

```
1
2 sns.barplot(data= df, x= "min_age", y= "duration", estimator= np.mean)
```

Out[260]:

```
<AxesSubplot:xlabel='min_age', ylabel='duration'>
```

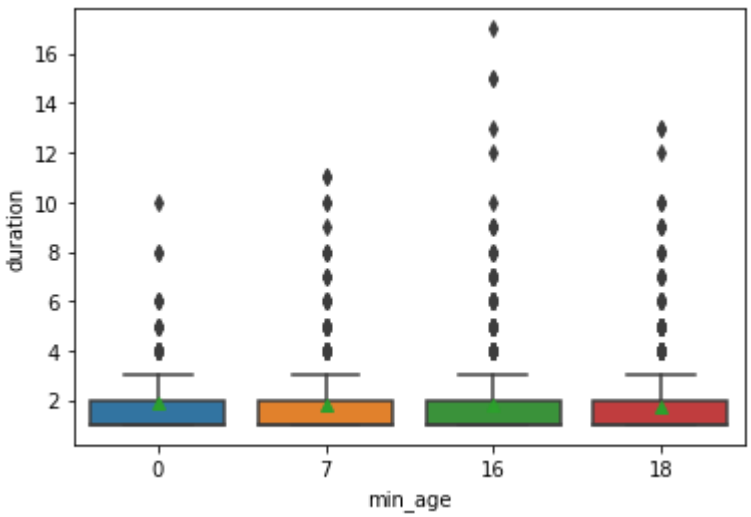


In [261]:

```
1
2 sns.boxplot(data= df, x= "min_age", y= "duration", showmeans= True)
```

Out[261]:

```
<AxesSubplot:xlabel='min_age', ylabel='duration'>
```

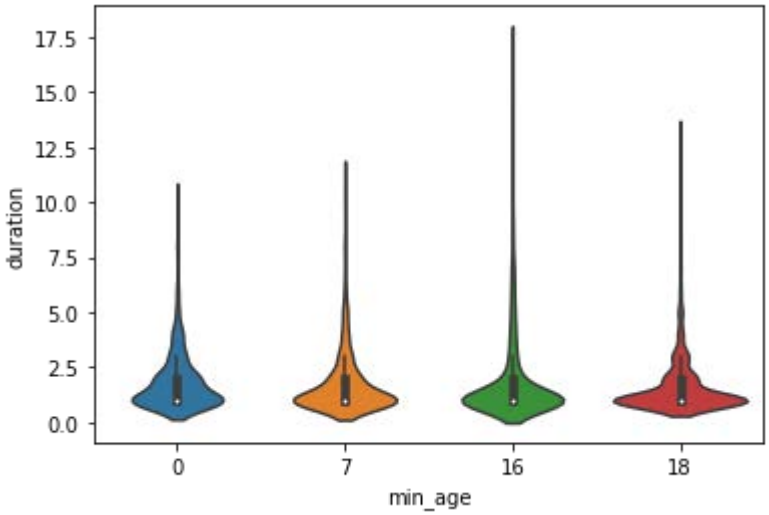


In [262]:

```
1
2 sns.violinplot(data= df, x= "min_age", y= "duration", showmeans= True)
```

Out[262]:

<AxesSubplot:xlabel='min_age', ylabel='duration'>

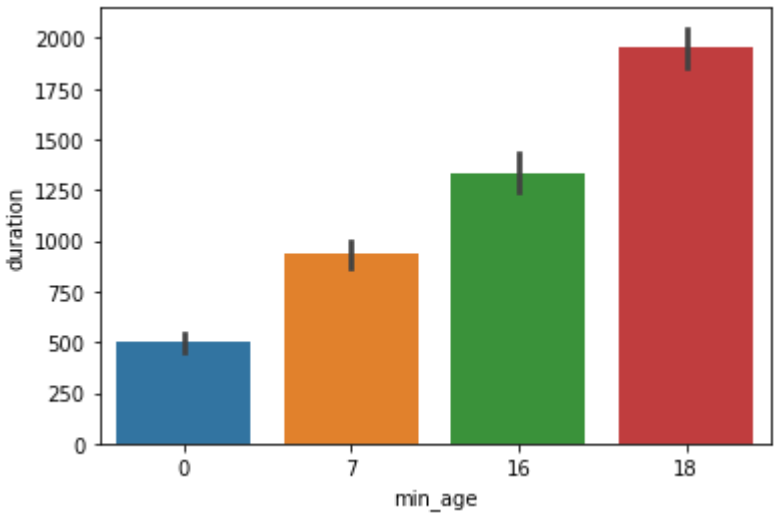


In [263]:

```
1
2 sns.barplot(data= df, x= "min_age", y= "duration", estimator= np.sum)
```

Out[263]:

<AxesSubplot:xlabel='min_age', ylabel='duration'>



distribution of release_year/ added year/ day of week/ month vs min_age

Observation:

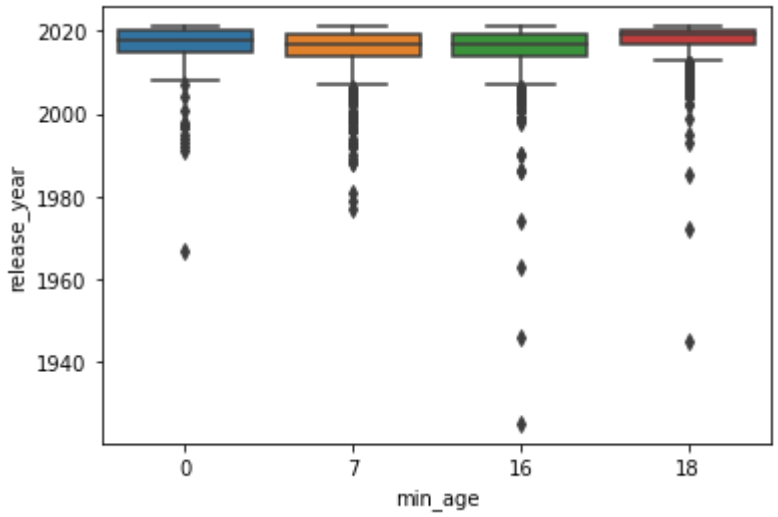
- The release_year/ added year are independent of min_age
- The day of week added tends to be on Thur, Friday for all age groups

In [264]:

```
1
2 sns.boxplot(data= df, x= "min_age", y= "release_year")
```

Out[264]:

```
<AxesSubplot:xlabel='min_age', ylabel='release_year'>
```

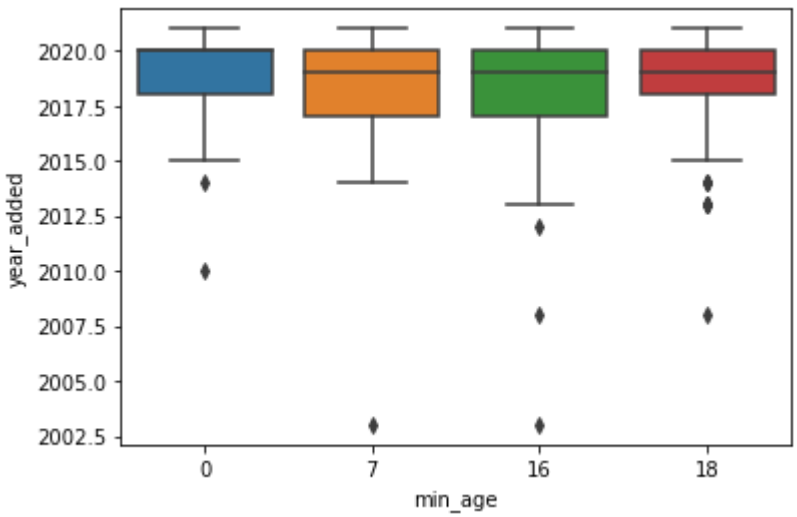


In [265]:

```
1
2 sns.boxplot(data= df, x= "min_age", y= "year_added")
```

Out[265]:

```
<AxesSubplot:xlabel='min_age', ylabel='year_added'>
```

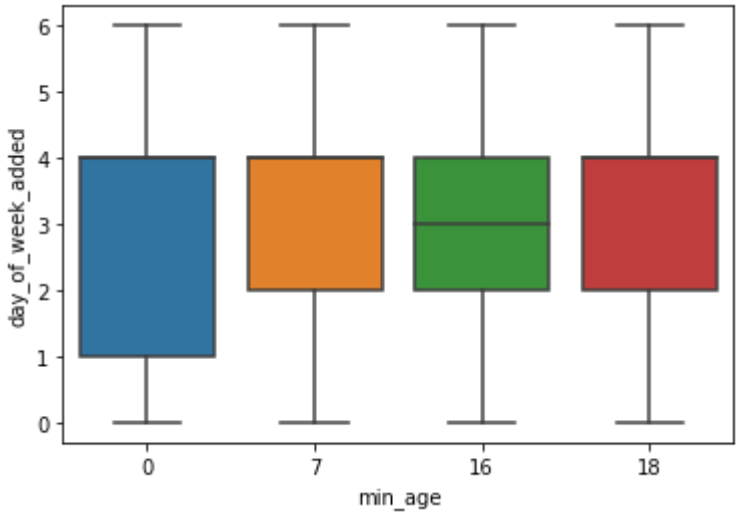


In [266]:

```
1
2 sns.boxplot(data= df, x= "min_age", y= "day_of_week_added")
```

Out[266]:

```
<AxesSubplot:xlabel='min_age', ylabel='day_of_week_added'>
```

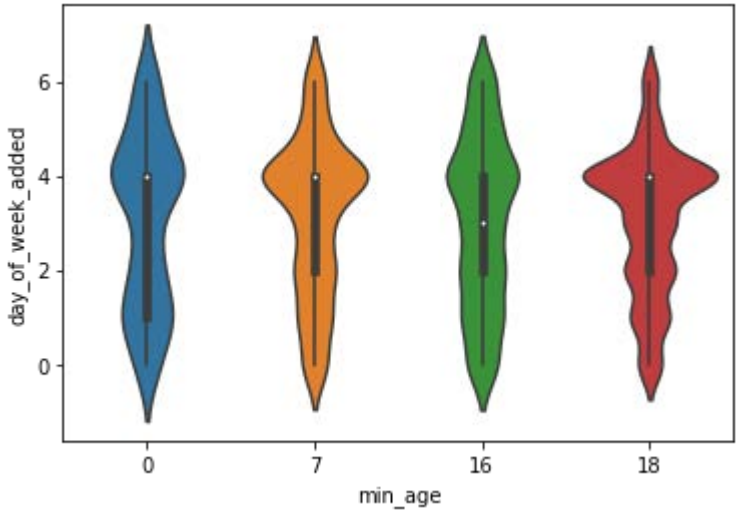


In [267]:

```
1
2 sns.violinplot(data= df, x= "min_age", y= "day_of_week_added")
```

Out[267]:

```
<AxesSubplot:xlabel='min_age', ylabel='day_of_week_added'>
```

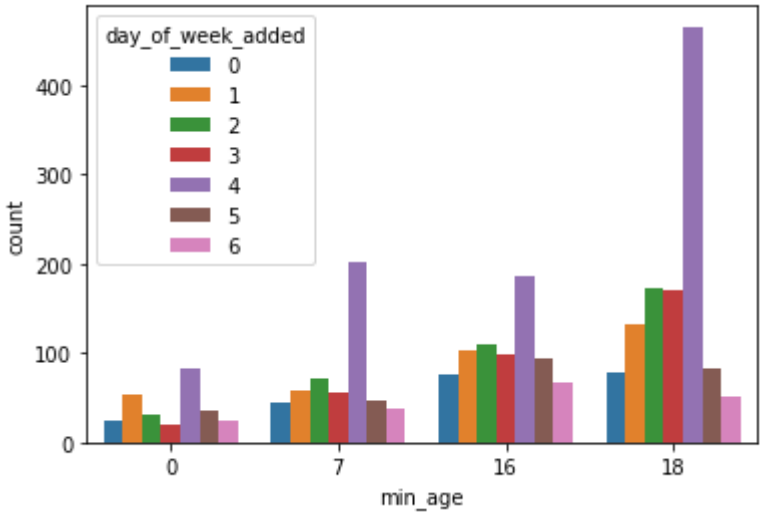


In [268]:

```
1
2 sns.countplot(data= df, x= "min_age", hue= "day_of_week_added")
```

Out[268]:

```
<AxesSubplot:xlabel='min_age', ylabel='count'>
```

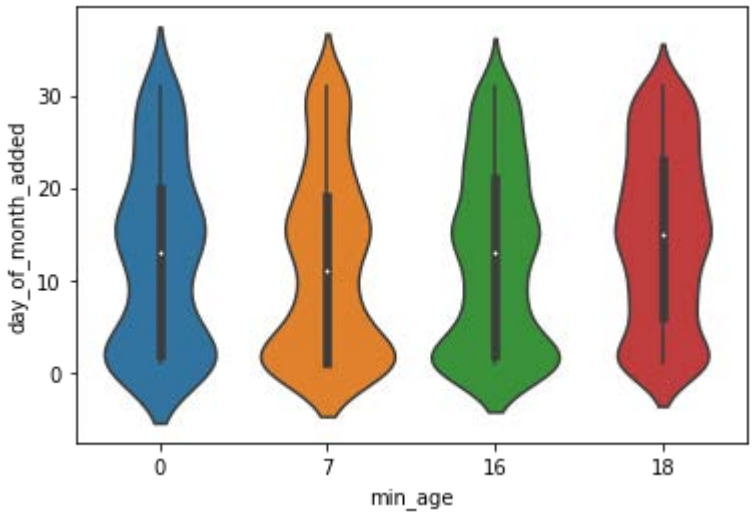


In [269]:

```
1
2 sns.violinplot(data= df, x= "min_age", y= "day_of_month_added")
```

Out[269]:

```
<AxesSubplot:xlabel='min_age', ylabel='day_of_month_added'>
```



TV Show Data Analysis: nested included

In [270]:

```
1
2 netflix_data_full_tv = netflix_data_full[netflix_data_full.type == "TV Show"].copy()
3
4 netflix_data_full_tv.head()
```

Out[270]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
1	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	
2	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	
3	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	
4	s2	TV Show	Blood & Water	Anonymous	Khosi Ngema	South Africa	2021-09-24	2021	TV-MA	
5	s2	TV Show	Blood & Water	Anonymous	Khosi Ngema	South Africa	2021-09-24	2021	TV-MA	

In [271]:

```
1
2 df = netflix_data_full_tv.copy()
```

In [272]:

```
df["release_decade"] = df["release_year"].apply(lambda x: x - (x%10))

df.head()
```

Out[272]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
1	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	
2	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	
3	s2	TV Show	Blood & Water	Anonymous	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	
4	s2	TV Show	Blood & Water	Anonymous	Khosi Ngema	South Africa	2021-09-24	2021	TV-MA	
5	s2	TV Show	Blood & Water	Anonymous	Khosi Ngema	South Africa	2021-09-24	2021	TV-MA	

In [273]:

```
df.columns = ['show_id', 'type', 'title', 'directors', 'actors', 'country', 'date_added
              'release_year', 'rating', 'duration', 'genres', 'min_age', 'release_decad

df.columns
```

Out[273]:

```
Index(['show_id', 'type', 'title', 'directors', 'actors', 'country',
       'date_added', 'release_year', 'rating', 'duration', 'genres', 'min_ag
e',
       'release_decade'],
      dtype='object')
```

Univariate analysis

In [274]:

```
test_df = df.copy(); test_df["count"] = 1

test_df.groupby("count").nunique()[["show_id", "title", "directors", "actors", "country
                                    "release_year", "genres", "release_decade"]]
```

Out[274]:

count	show_id	title	directors	actors	country	release_year	genres	release_decade
1	2676	2676	300	14864	66	46	22	9

Bivariate analysis

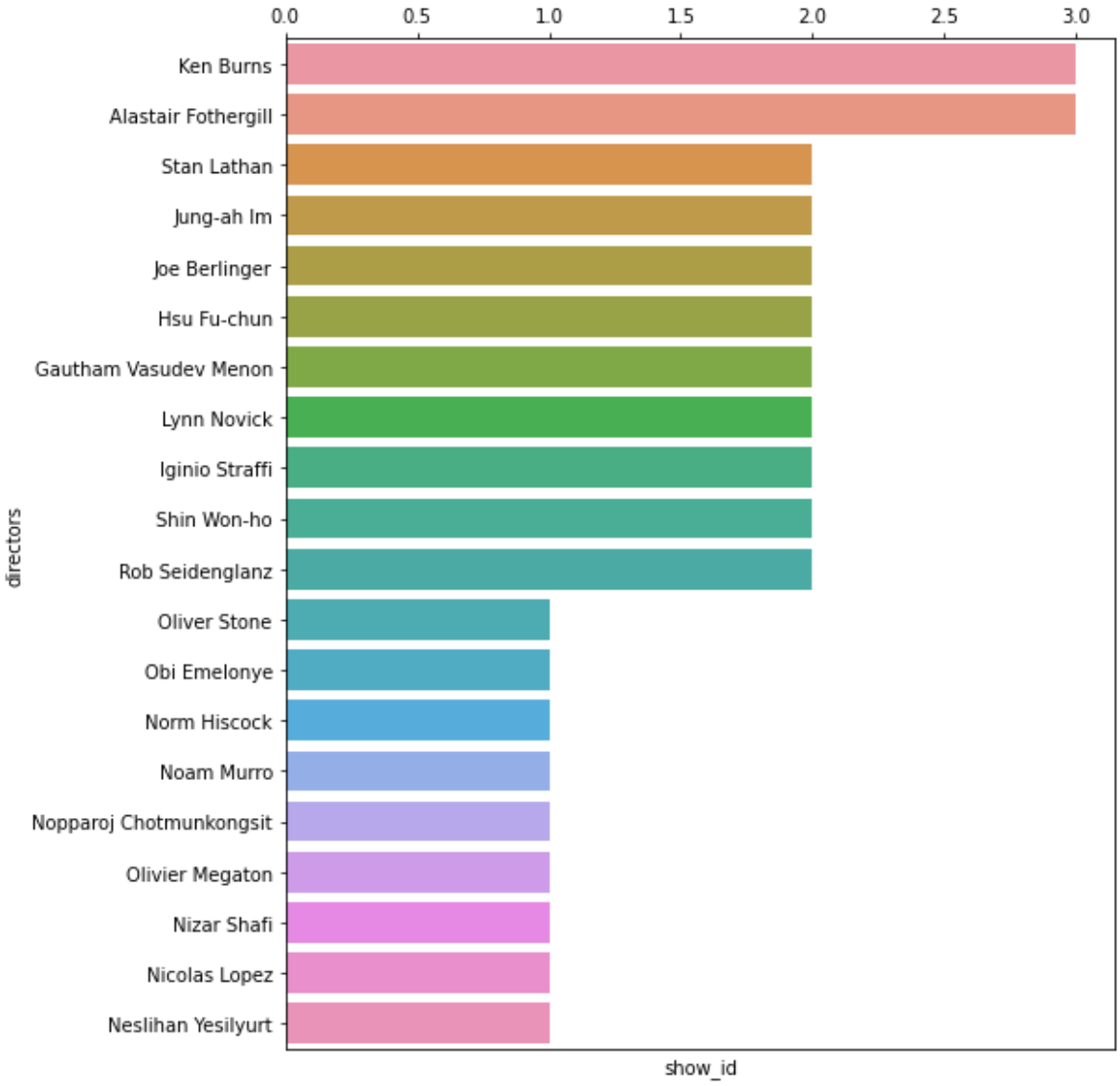
no. of TV Shows vs directors

Observation:

- There are only a handful of directors (among 789) who have done more than 10 TV Shows

In [275]:

```
1
2 plt_df = df.loc[(df.directors != "Anonymous")].groupby(["directors"]).nunique().reset_i
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "directors", x= "show_id")
7
8 ax.xaxis.tick_top()
```



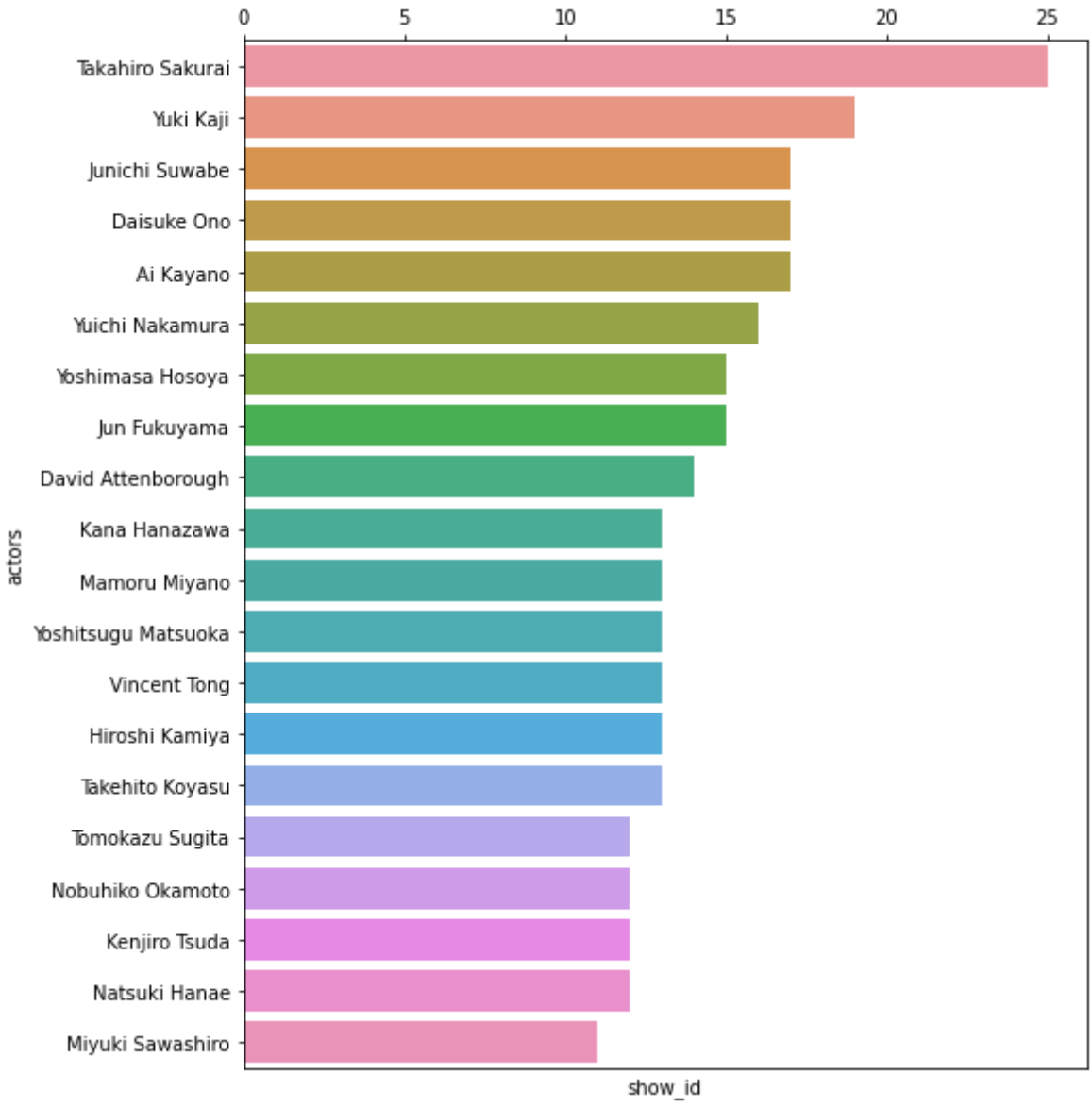
no. of TV Shows per actor

Observation:

- There are only a handful of actors (among 11472) who have done more than 10 TV Shows

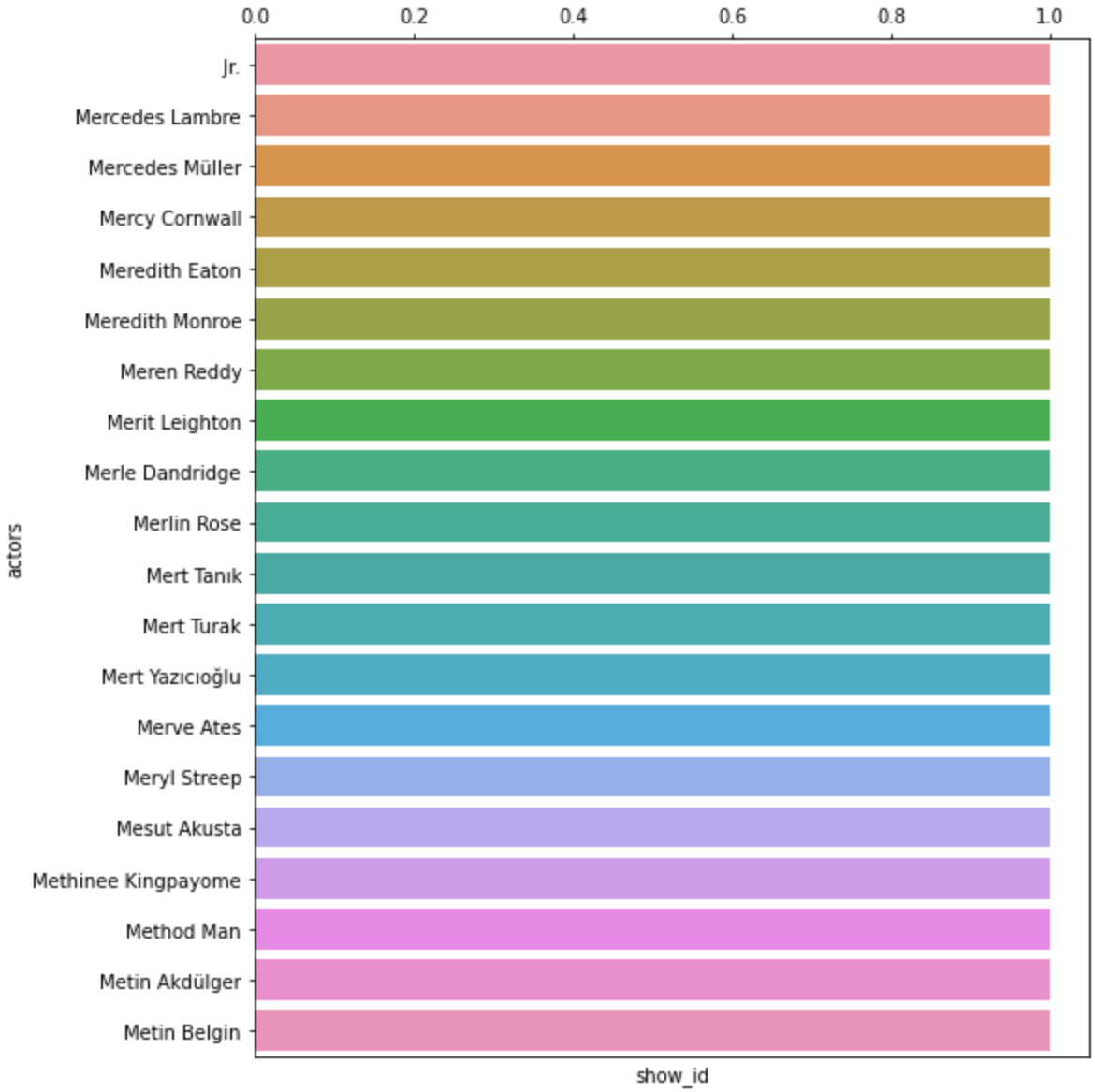
In [276]:

```
1
2 plt_df = df.loc[(df.actors != "Anonymous")].groupby(["actors"]).nunique().reset_index()
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "actors", x= "show_id")
7
8 ax.xaxis.tick_top()
```



In [277]:

```
1
2 plt_df = df.loc[(df.actors != "Anonymous")].groupby(["actors"]).nunique().reset_index()
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "actors", x= "show_id")
7
8 ax.xaxis.tick_top()
```



no. of TV Shows per country

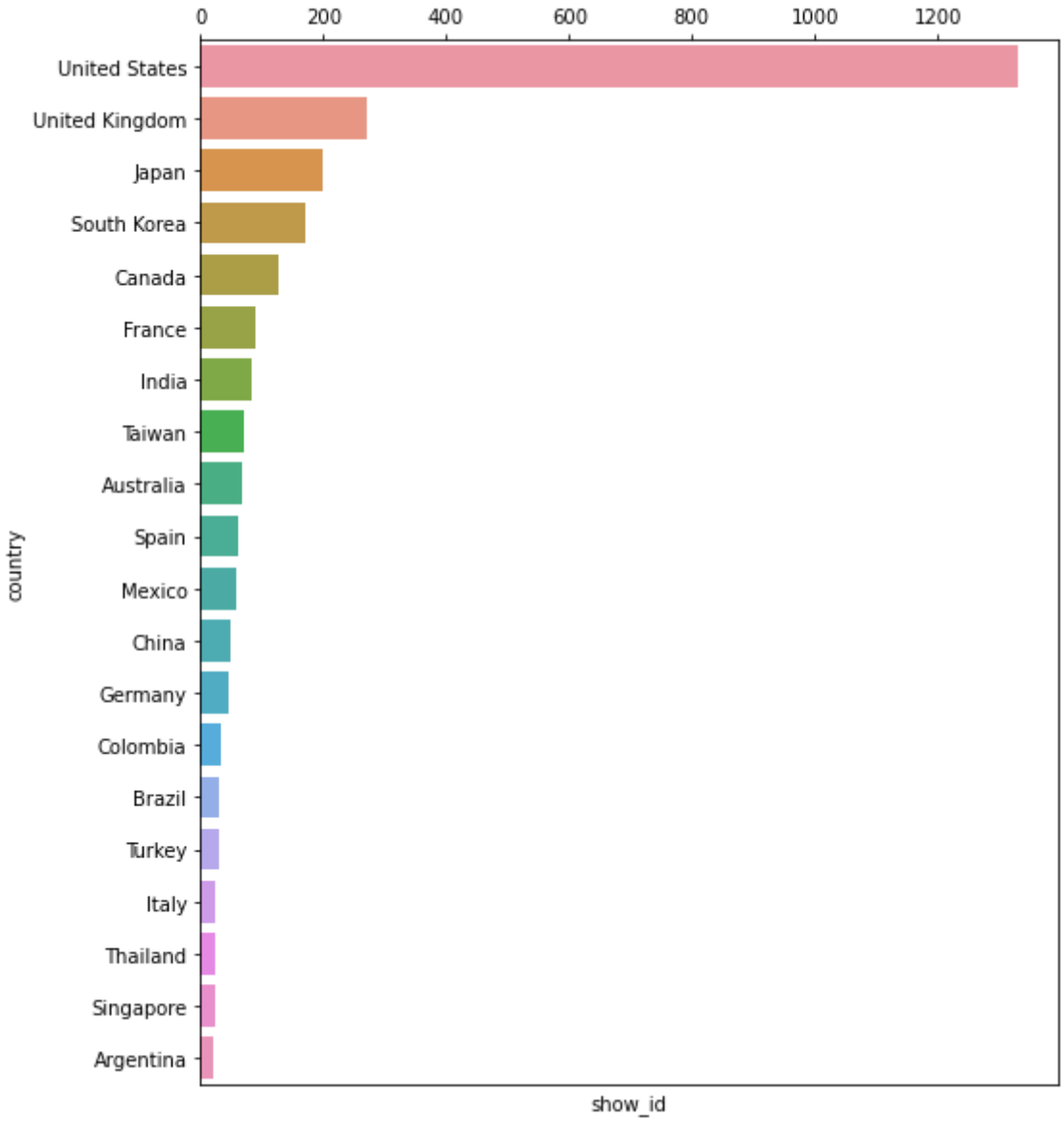
Observation:

- Most TV Shows are availabe only for US, Japan, UK, Canada

- UAE, Ukraine, Austri and few other countries have only one TV Show streaming

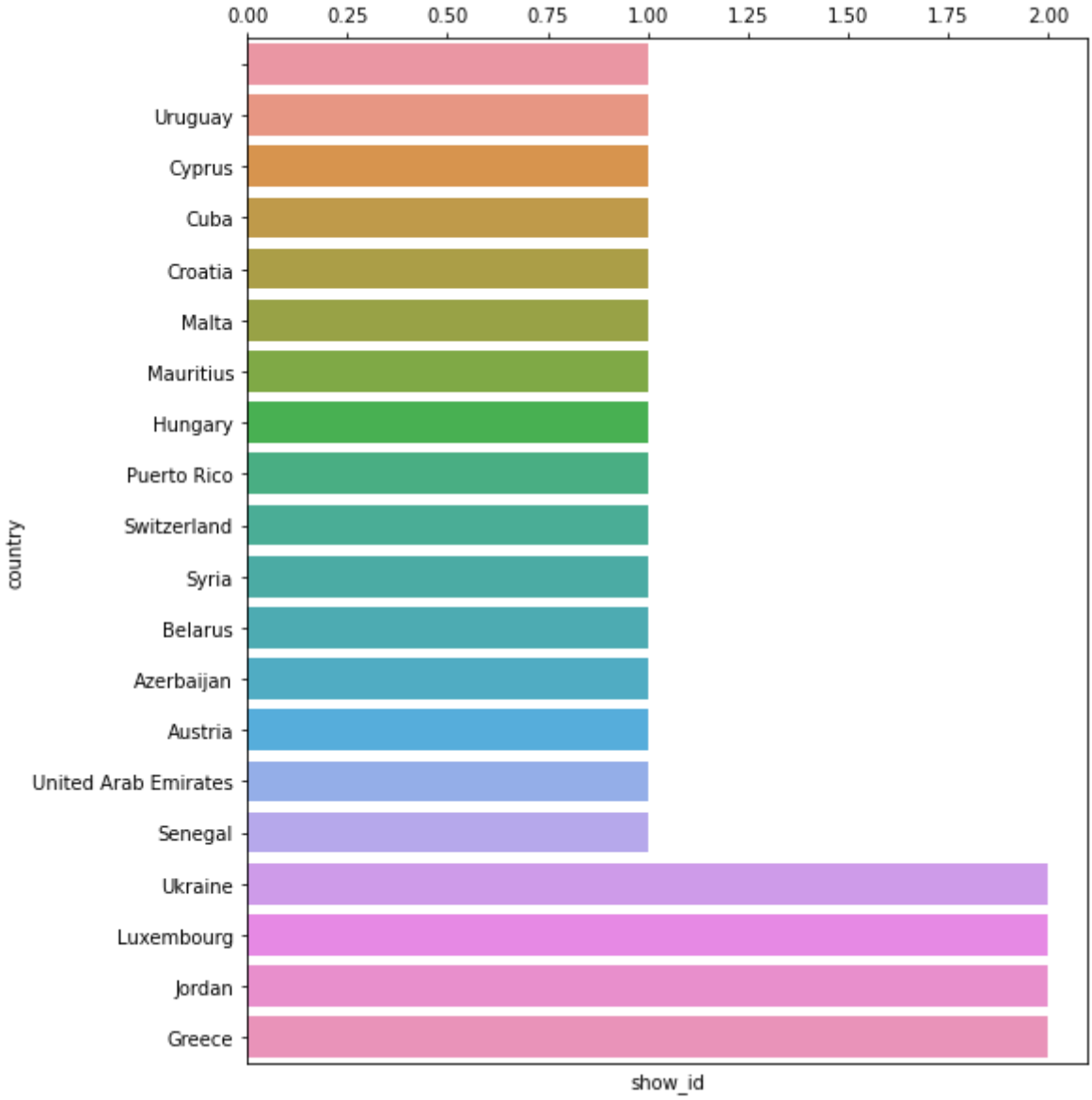
In [278]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["show_id"], ascer
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "show_id")
7
8 ax.xaxis.tick_top()
```



In [279]:

```
plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["show_id"], ascer

plt.figure(figsize= (8, 10))

ax = sns.barplot(data= plt_df, y= "country", x= "show_id")

ax.xaxis.tick_top()
```



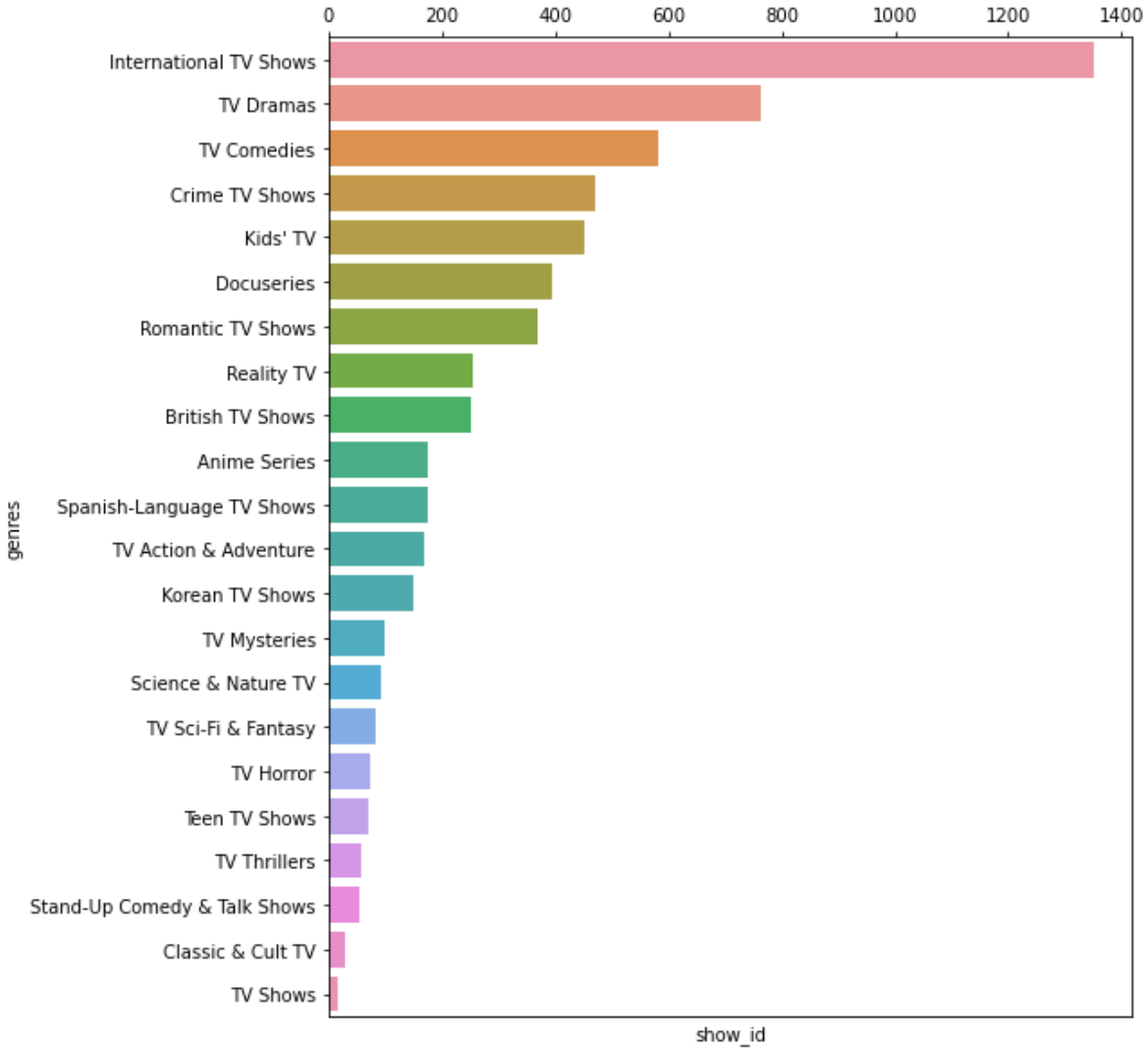
no. of TV Shows per genre

Observation:

- Most TV Shows belong to the genre of International, Dramas, Comedies
- Very few TV Shows belong to the genre of Classic, Cult, Reality, Thriller, Teen, Science & Nature

In [280]:

```
1
2 plt_df = df.groupby(["genres"]).nunique().reset_index().sort_values(["show_id"], ascend
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "genres", x= "show_id")
7
8 ax.xaxis.tick_top()
```



In [281]:

```
1
2 df.genres.unique()
```

Out[281]:

```
array(['International TV Shows', 'TV Dramas', 'TV Mysteries',
       'Crime TV Shows', 'TV Action & Adventure', 'Docuseries',
       'Reality TV', 'Romantic TV Shows', 'TV Comedies', 'TV Horror',
       'British TV Shows', 'Spanish-Language TV Shows', 'TV Thrillers',
       "Kids' TV", 'TV Sci-Fi & Fantasy', 'Anime Series',
       'Korean TV Shows', 'Science & Nature TV', 'Teen TV Shows',
       'TV Shows', 'Stand-Up Comedy & Talk Shows', 'Classic & Cult TV'],
      dtype=object)
```

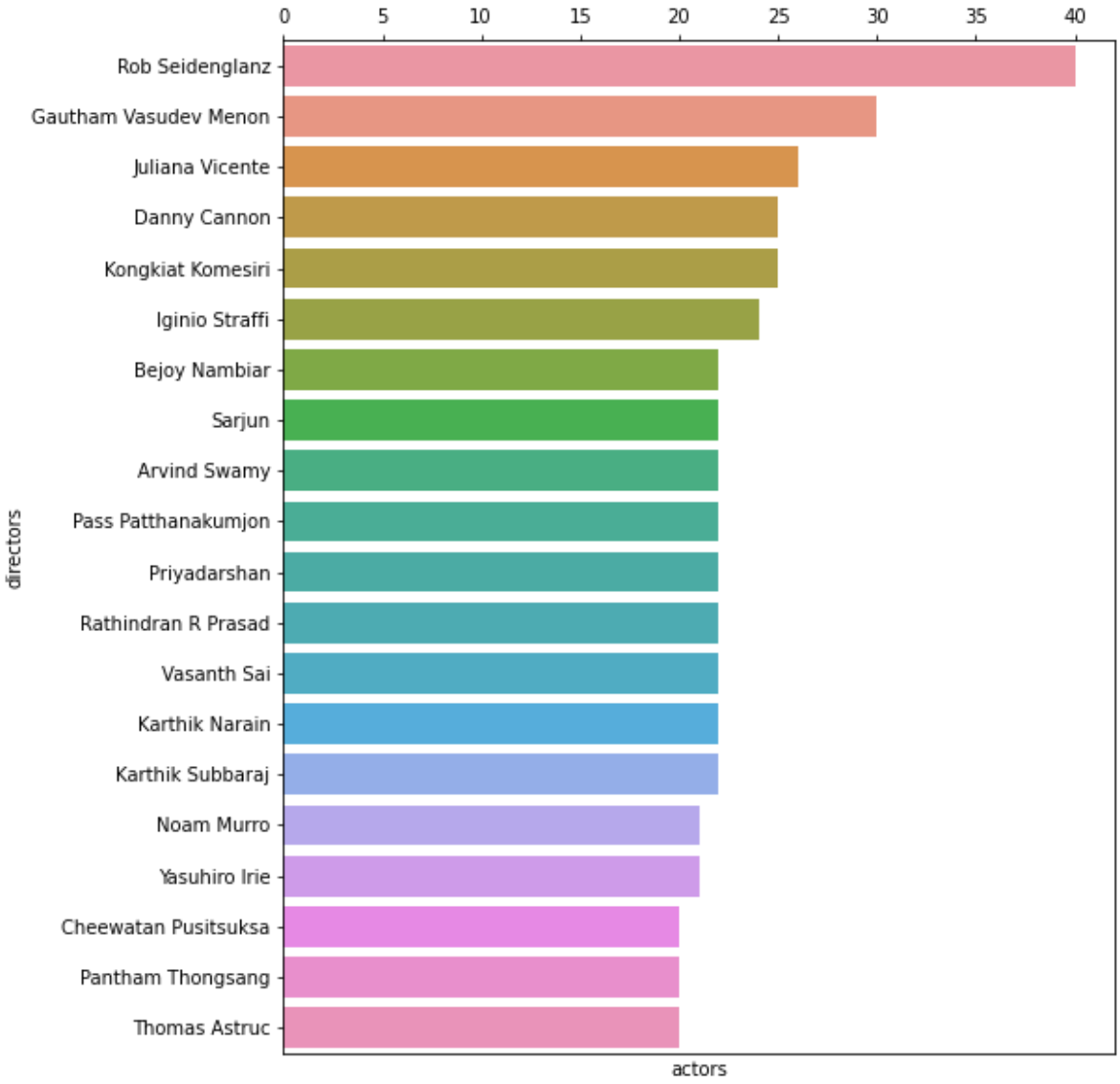
no. of actors worked with per director

Observation:

- Around 10-20 directors (among 790) worked with more than 100 actors (among 12000)

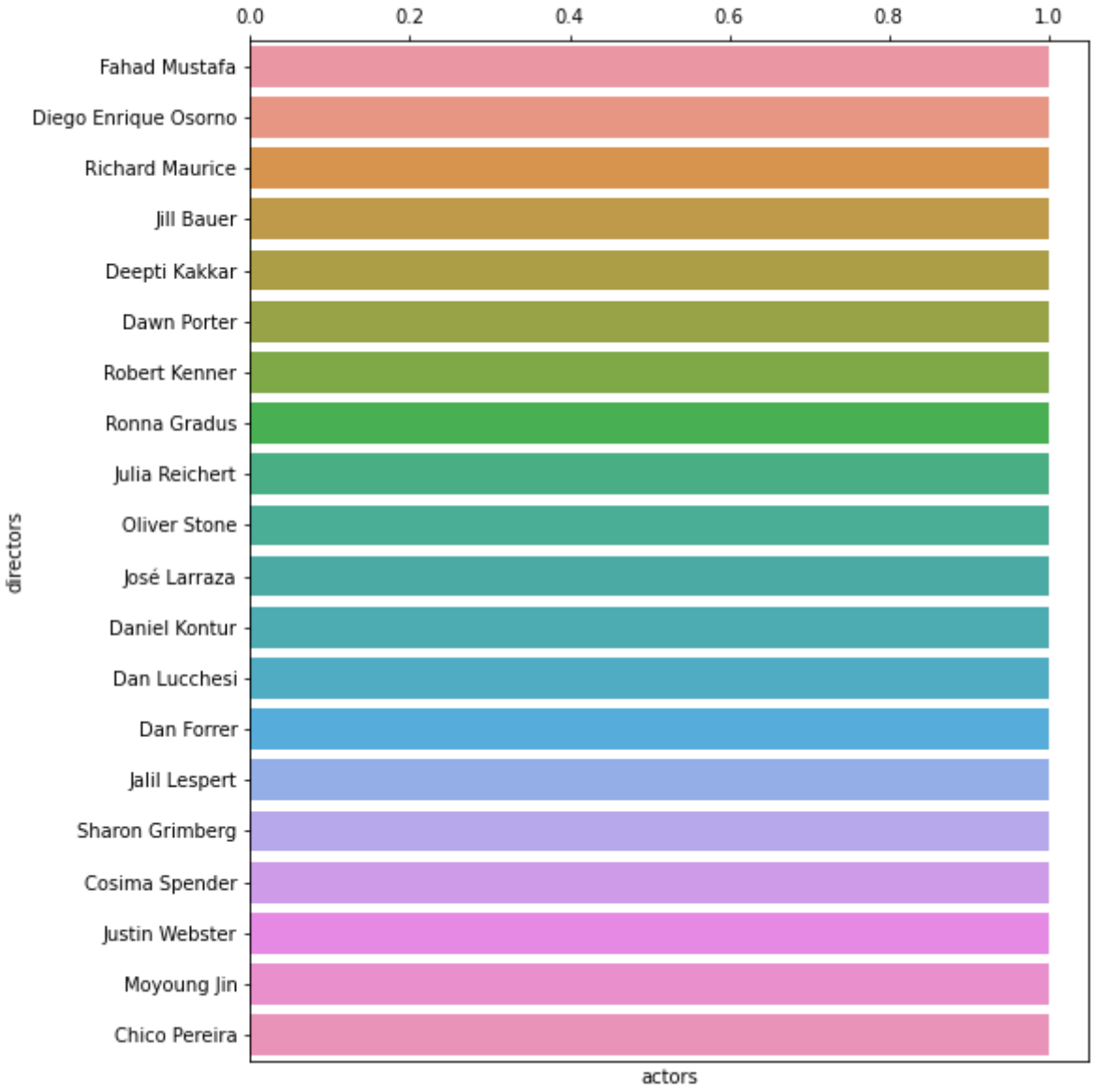
In [282]:

```
1
2 plt_df = df.loc[(df.directors != "Anonymous")].groupby(["directors"]).nunique().reset_i
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "directors", x= "actors")
7
8 ax.xaxis.tick_top()
```



In [283]:

```
1
2 plt_df = df.loc[(df.directors != "Anonymous")].groupby(["directors"]).nunique().reset_i
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "directors", x= "actors")
7
8 ax.xaxis.tick_top()
```



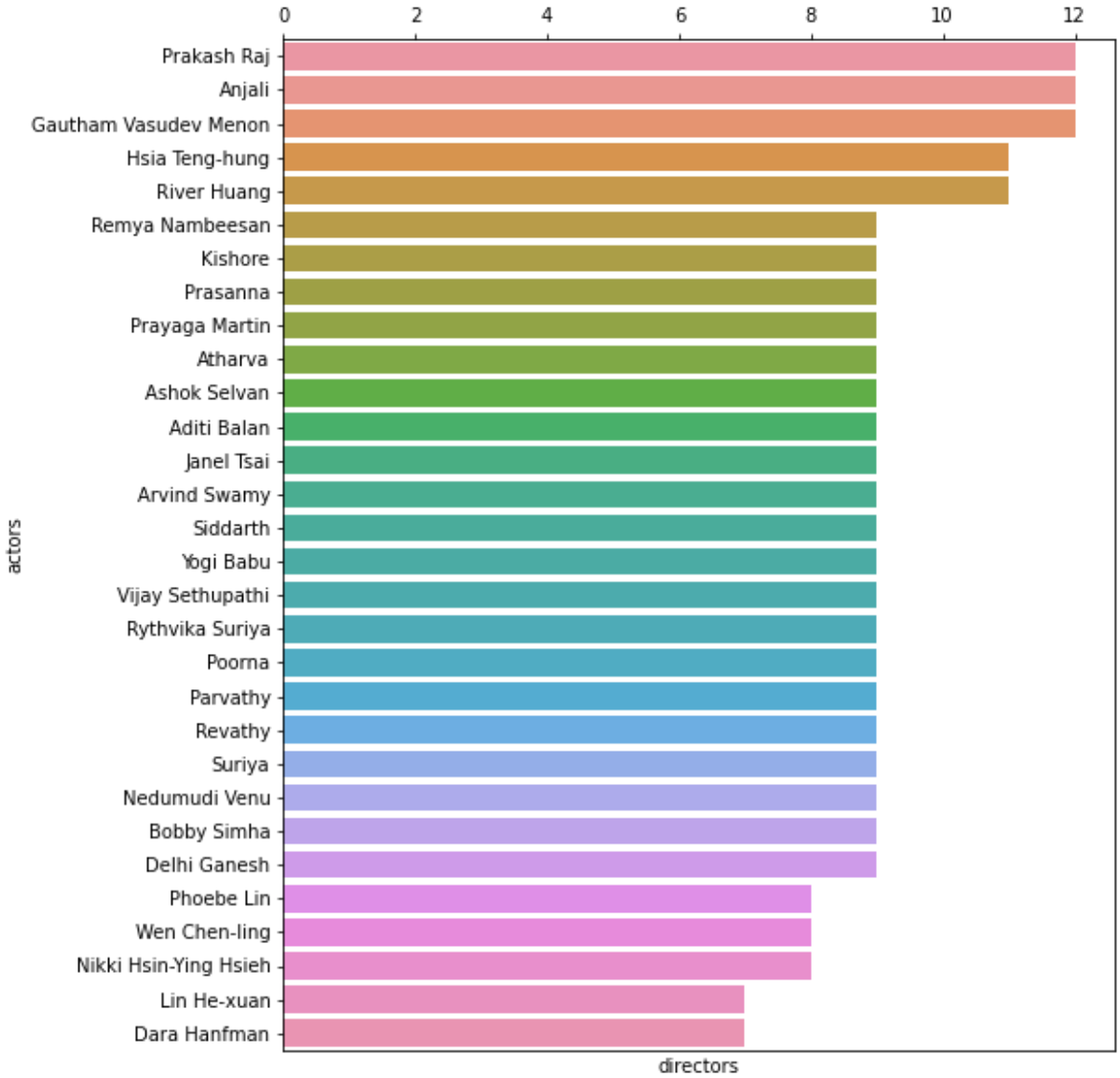
no. of directors worked with per actor

Observation:

- Around 20-30 actors (among 12000) worked with more than 10 directors (among 790)

In [284]:

```
1
2 plt_df = df.loc[(df.actors != "Anonymous")].groupby(["actors"]).nunique().reset_index()
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "actors", x= "directors")
7
8 ax.xaxis.tick_top()
```



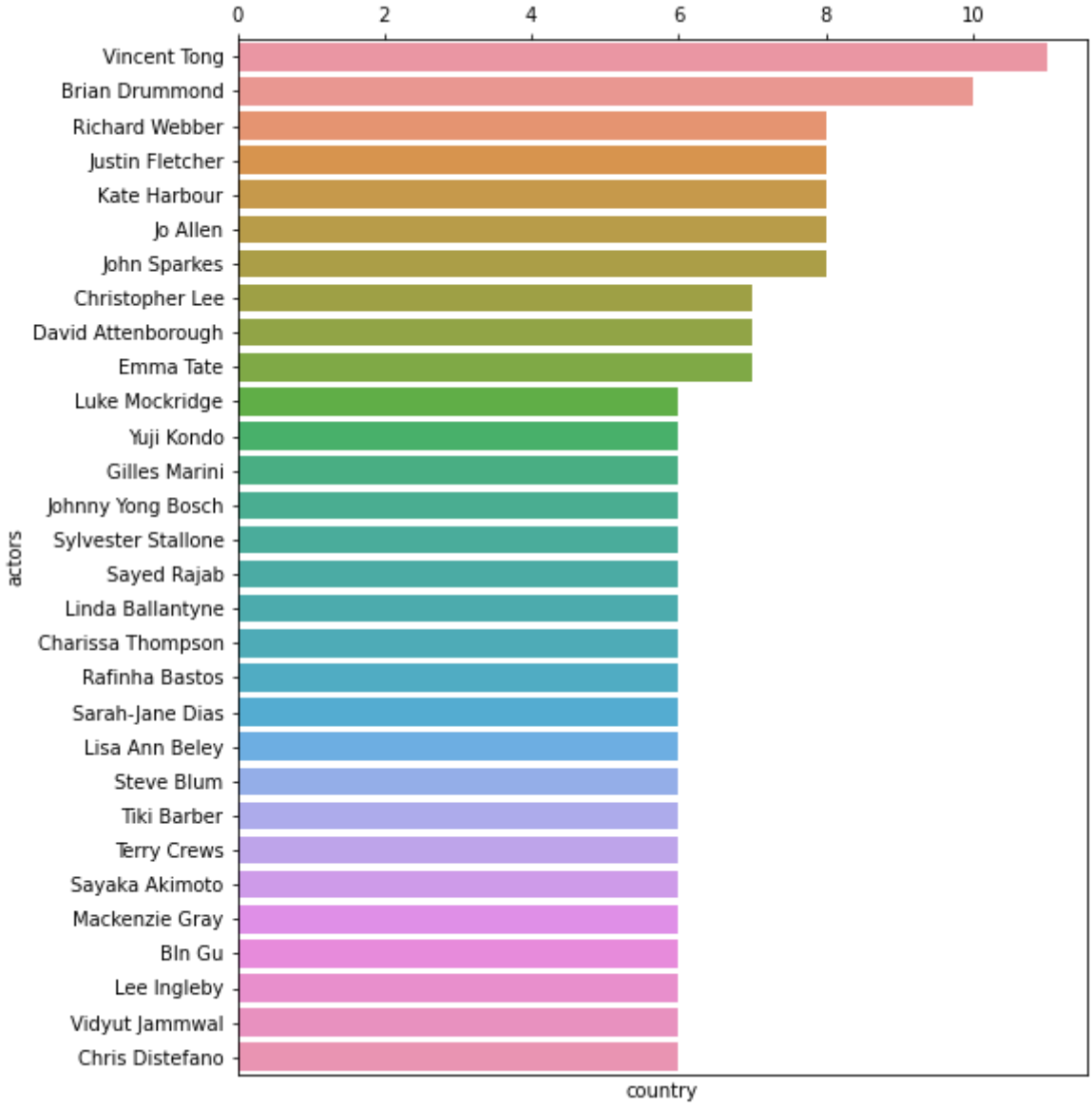
no. of countries streaming per actor

Observation:

- Around 10 actors (among 12000) are streaming (popular) in more than 6 countries

In [285]:

```
1
2 plt_df = df.loc[(df.actors != "Anonymous")].groupby(["actors"]).nunique().reset_index()
3
4 plt.figure(figsize= (8, 10))
5
6 ax = sns.barplot(data= plt_df, y= "actors", x= "country")
7
8 ax.xaxis.tick_top()
```



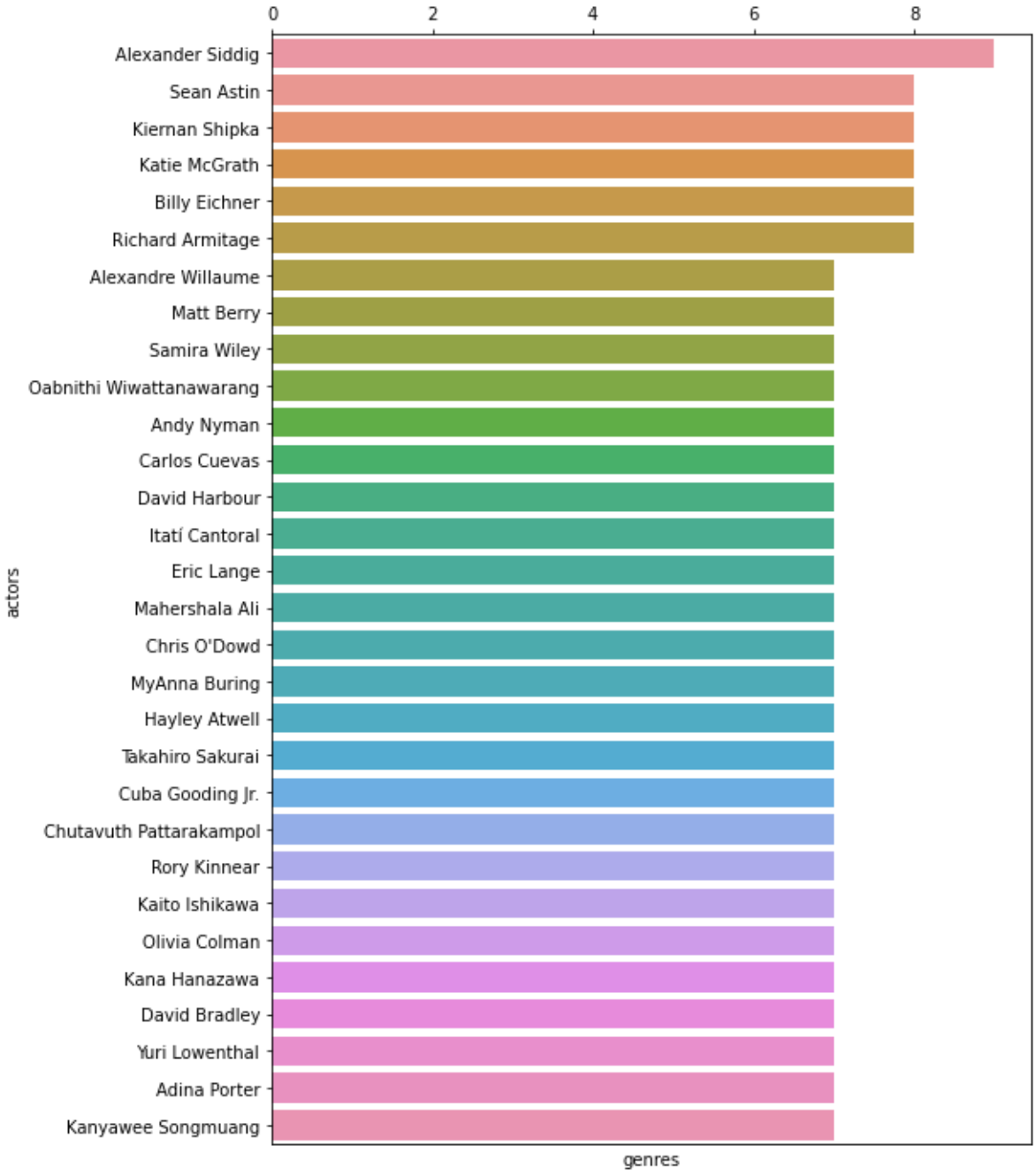
no. of genres per actor

Observation:

- Around 30 actors (among 12000) are (versatile) in more than 6 genres

In [286]:

```
1
2 plt_df = df.loc[(df.actors != "Anonymous")].groupby(["actors"]).nunique().reset_index()
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "actors", x= "genres")
7
8 ax.xaxis.tick_top()
```



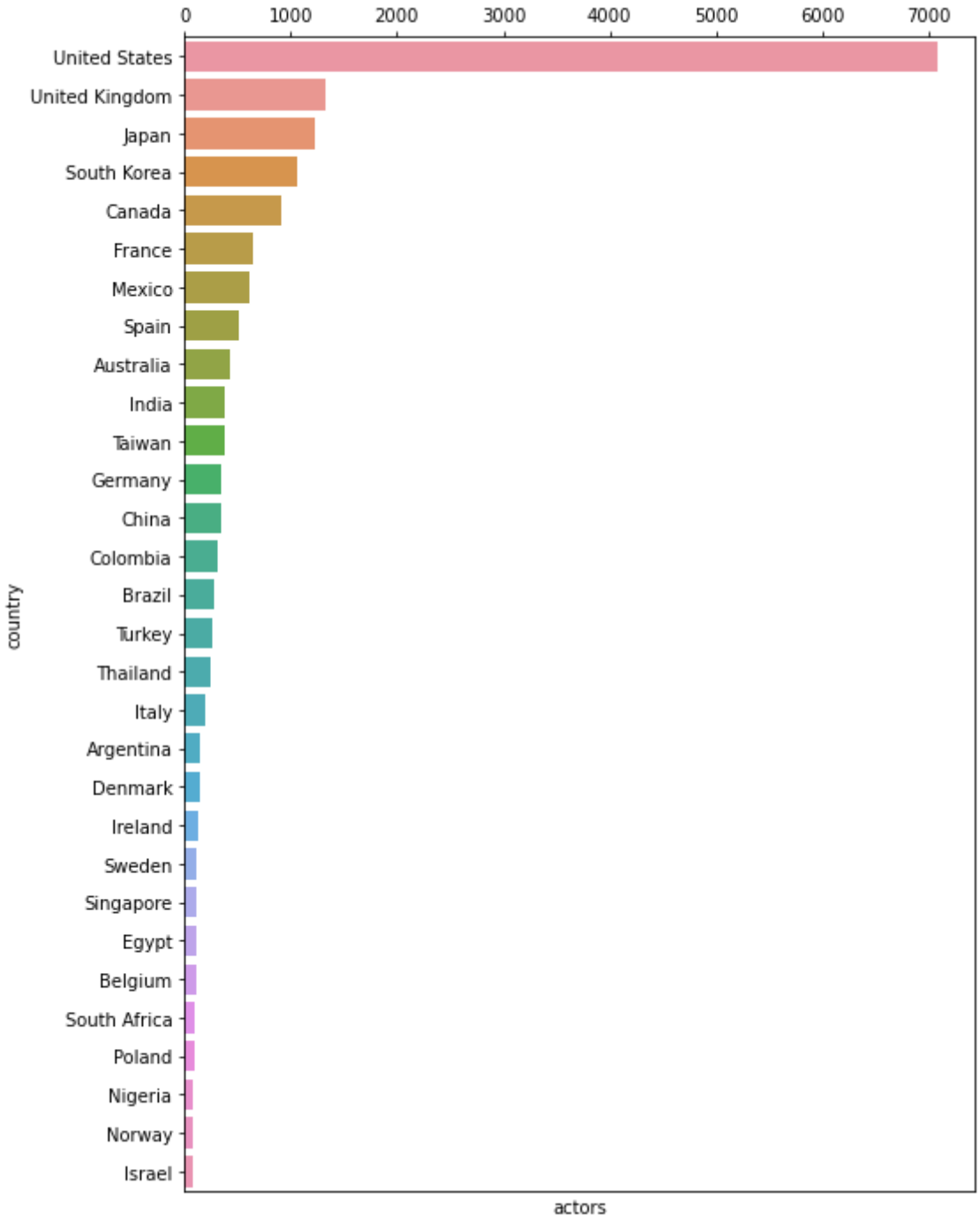
no. of actors/directors per country

Observation:

- Most actors/directors are streaming on US, UK, Japan, Canada, South Korea
- The countries such as Greece, Malta, Cuba, Austria and few more have only 1 actor/director streaming

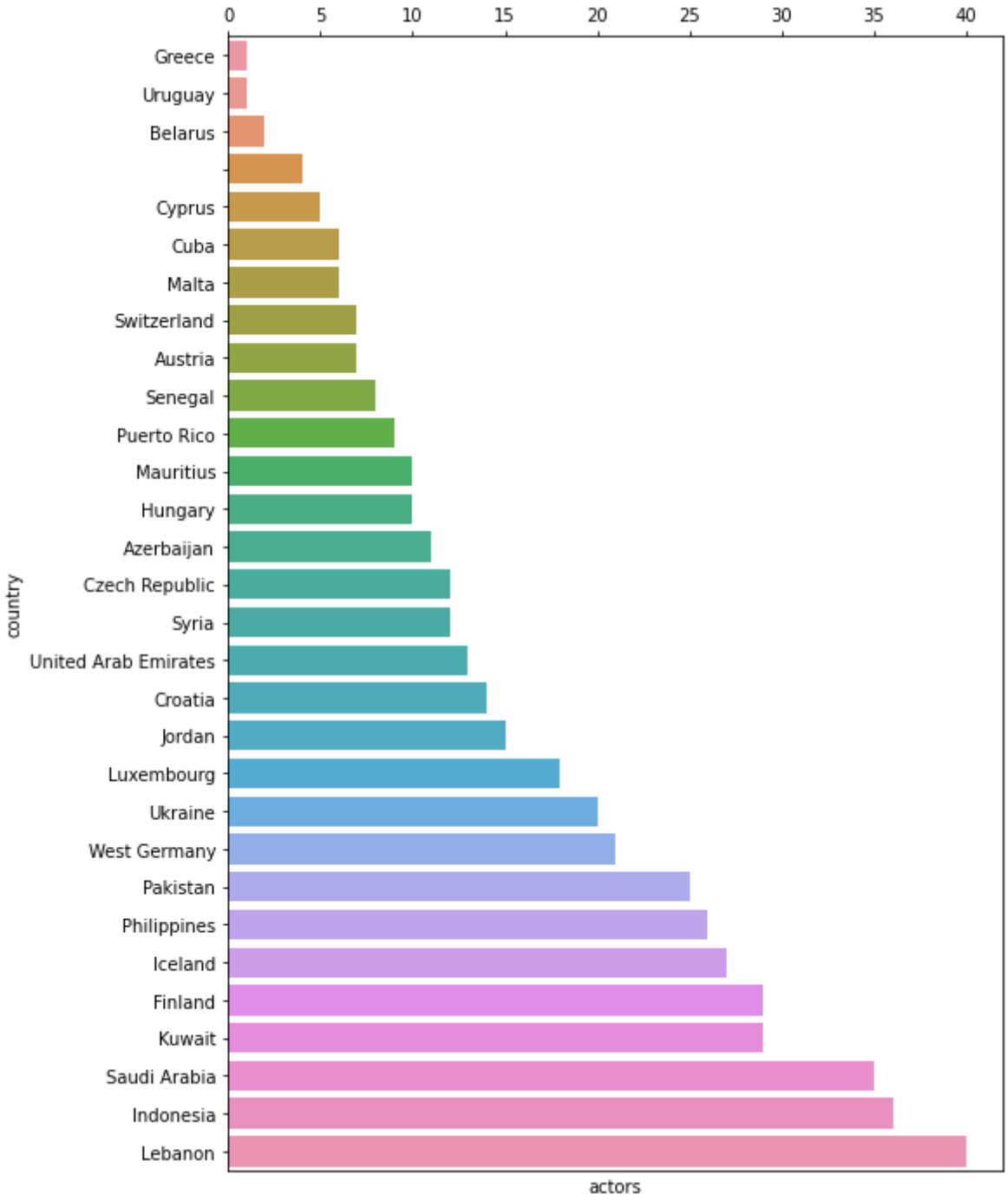
In [287]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["actors"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "actors")
7
8 ax.xaxis.tick_top()
```



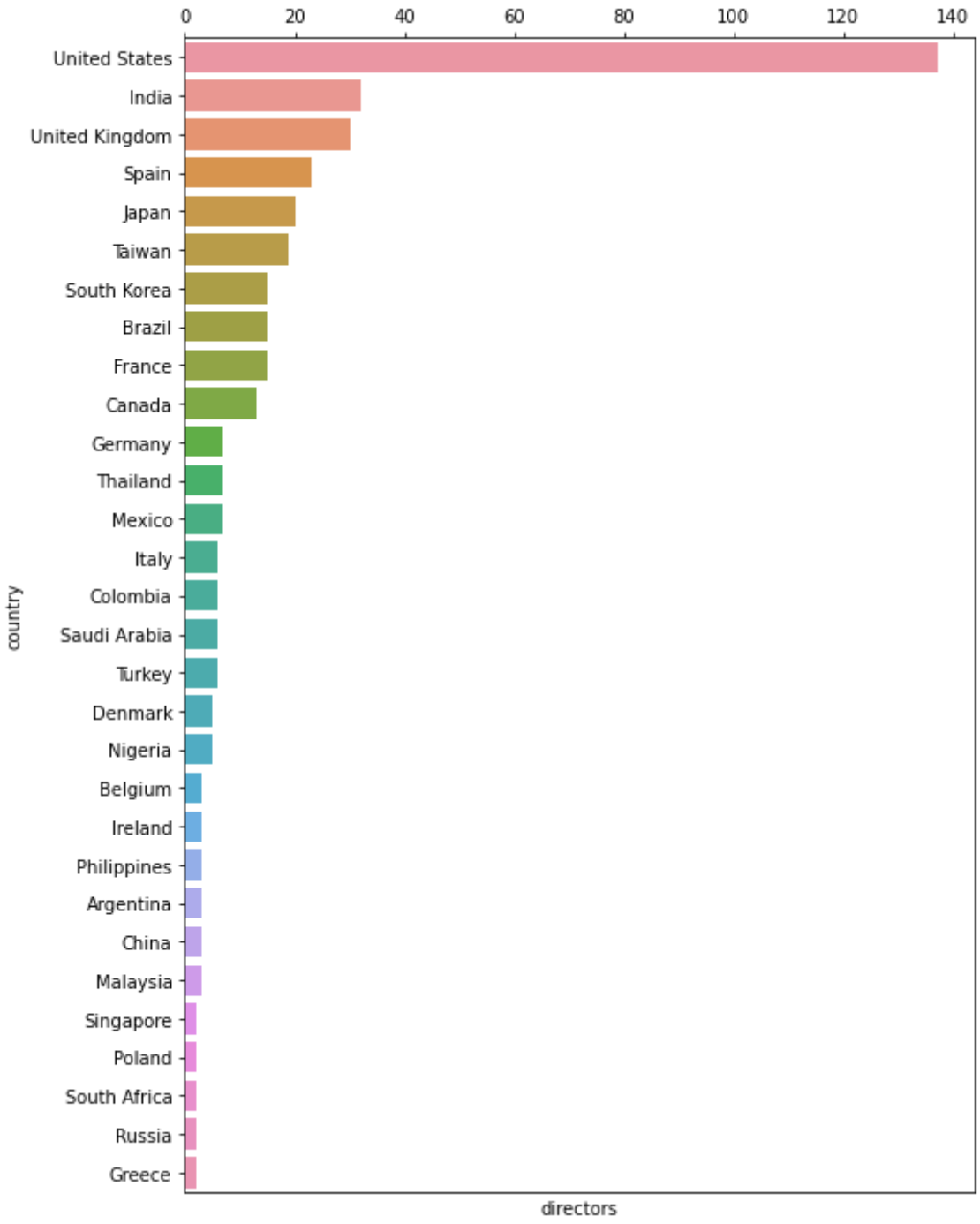
In [288]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["actors"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "actors")
7
8 ax.xaxis.tick_top()
```



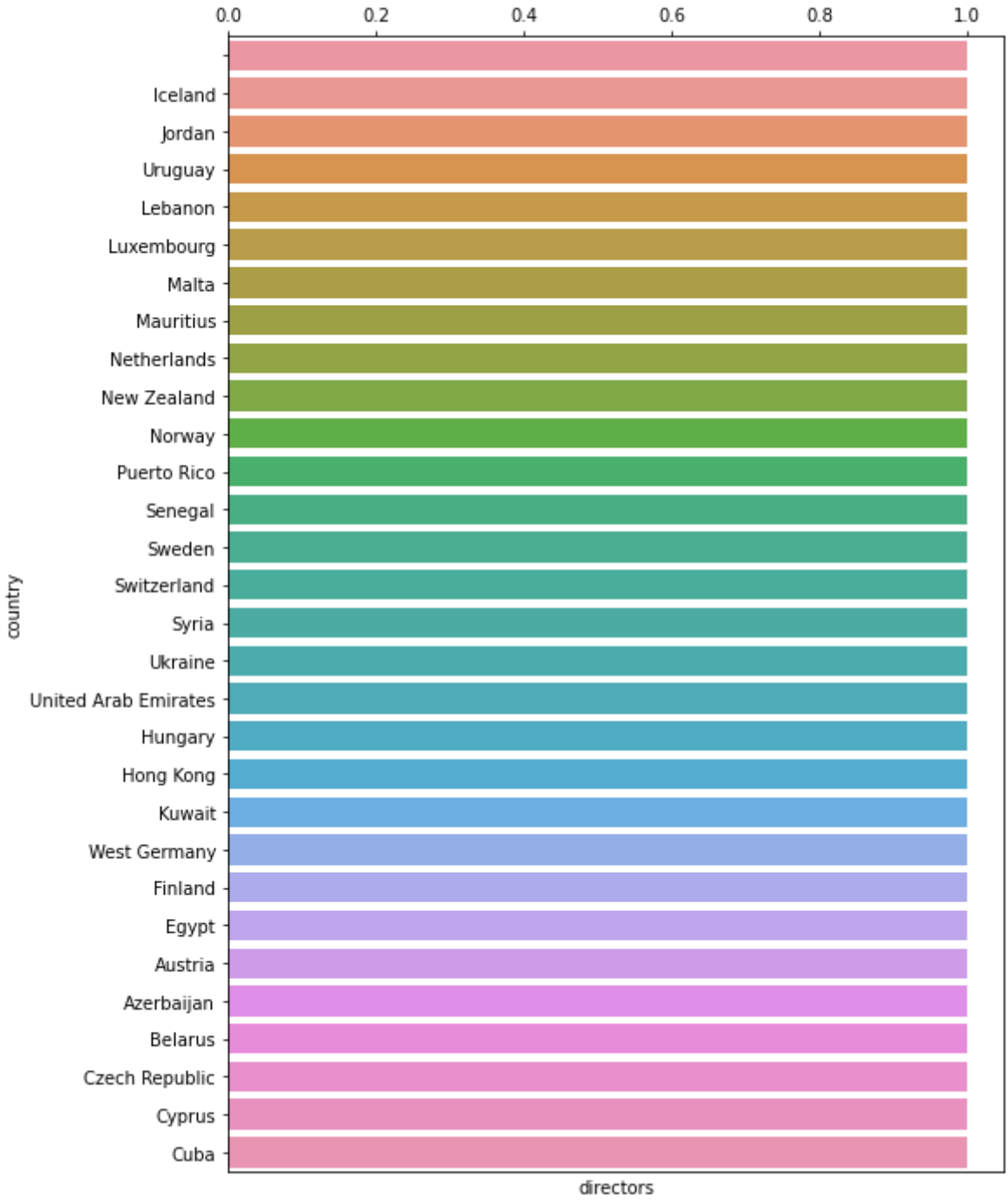
In [289]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["directors"], asc
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "directors")
7
8 ax.xaxis.tick_top()
```



In [290]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["directors"], asc
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "directors")
7
8 ax.xaxis.tick_top()
```



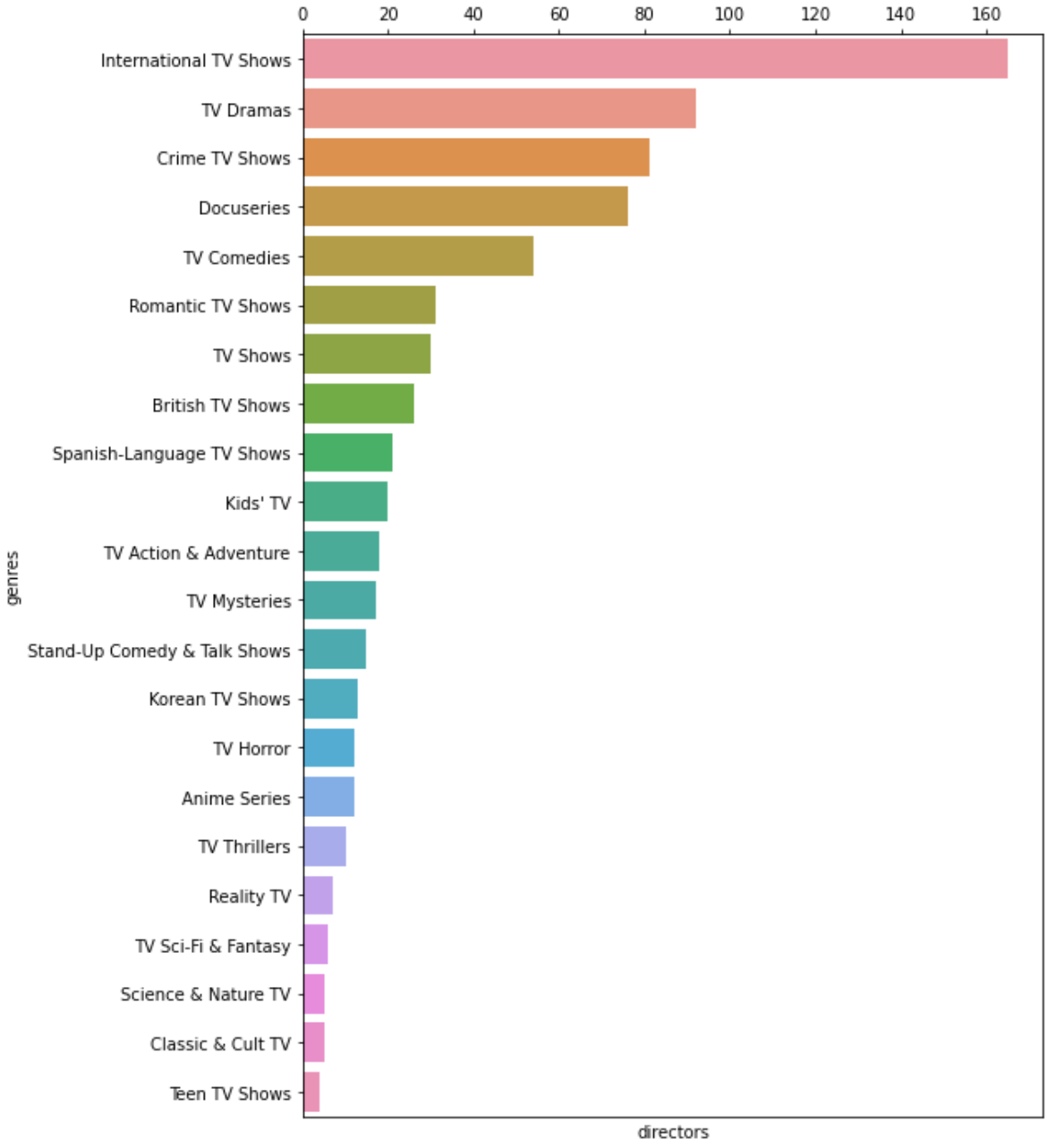
no. of actors/directors per genre

Observation:

- Most actors/directors are working on Dramas, Comedies, Crime TV Shows
- Very few directors on netflix are working on Classics, Cult, Science, Nature, Reality and Talk Shows

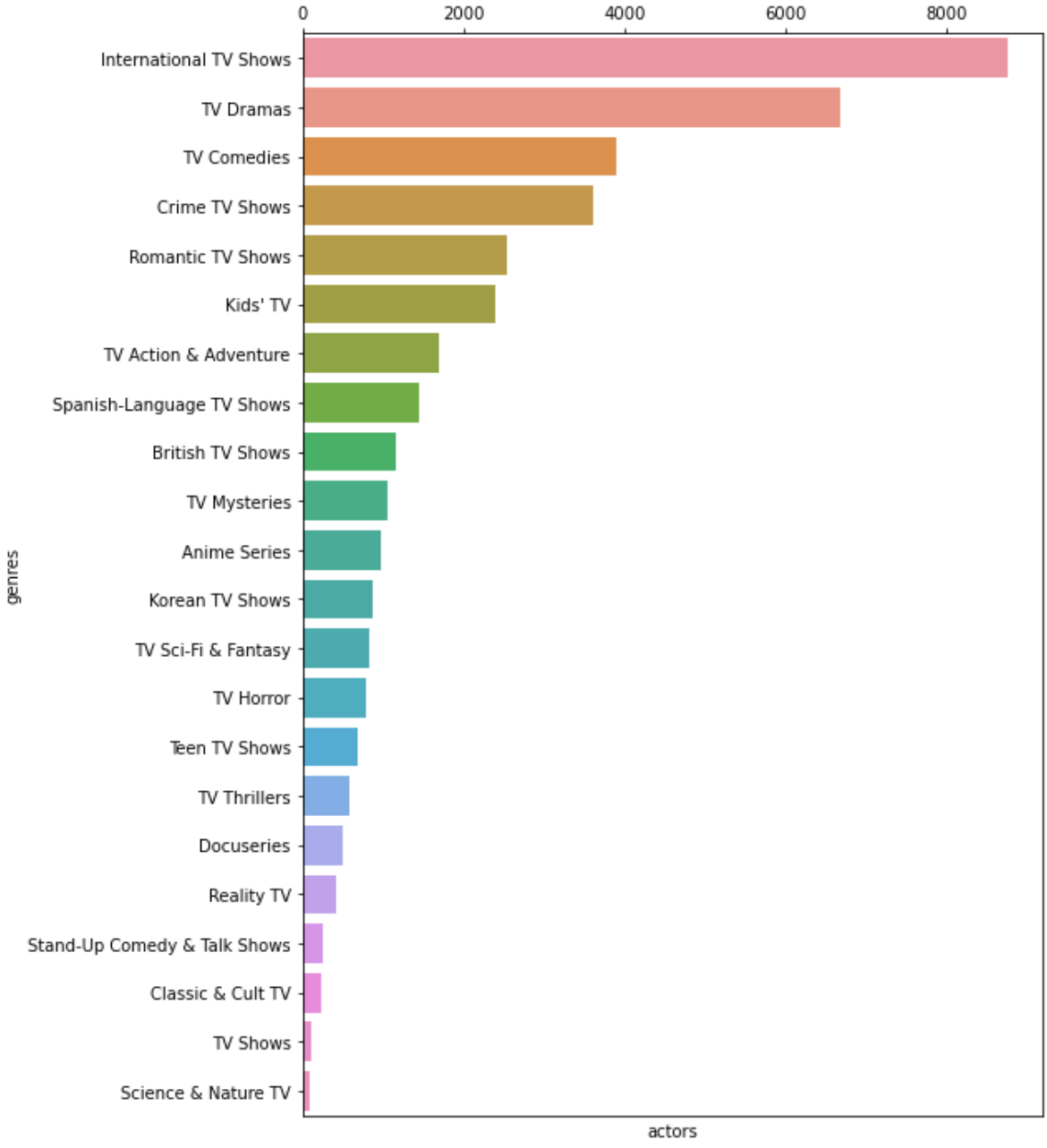
In [291]:

```
1
2 plt_df = df.groupby(["genres"]).nunique().reset_index().sort_values(["directors"], asce
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "genres", x= "directors")
7
8 ax.xaxis.tick_top()
```



In [292]:

```
1
2 plt_df = df.groupby(["genres"]).nunique().reset_index().sort_values(["actors"], ascendi
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "genres", x= "actors")
7
8 ax.xaxis.tick_top()
```



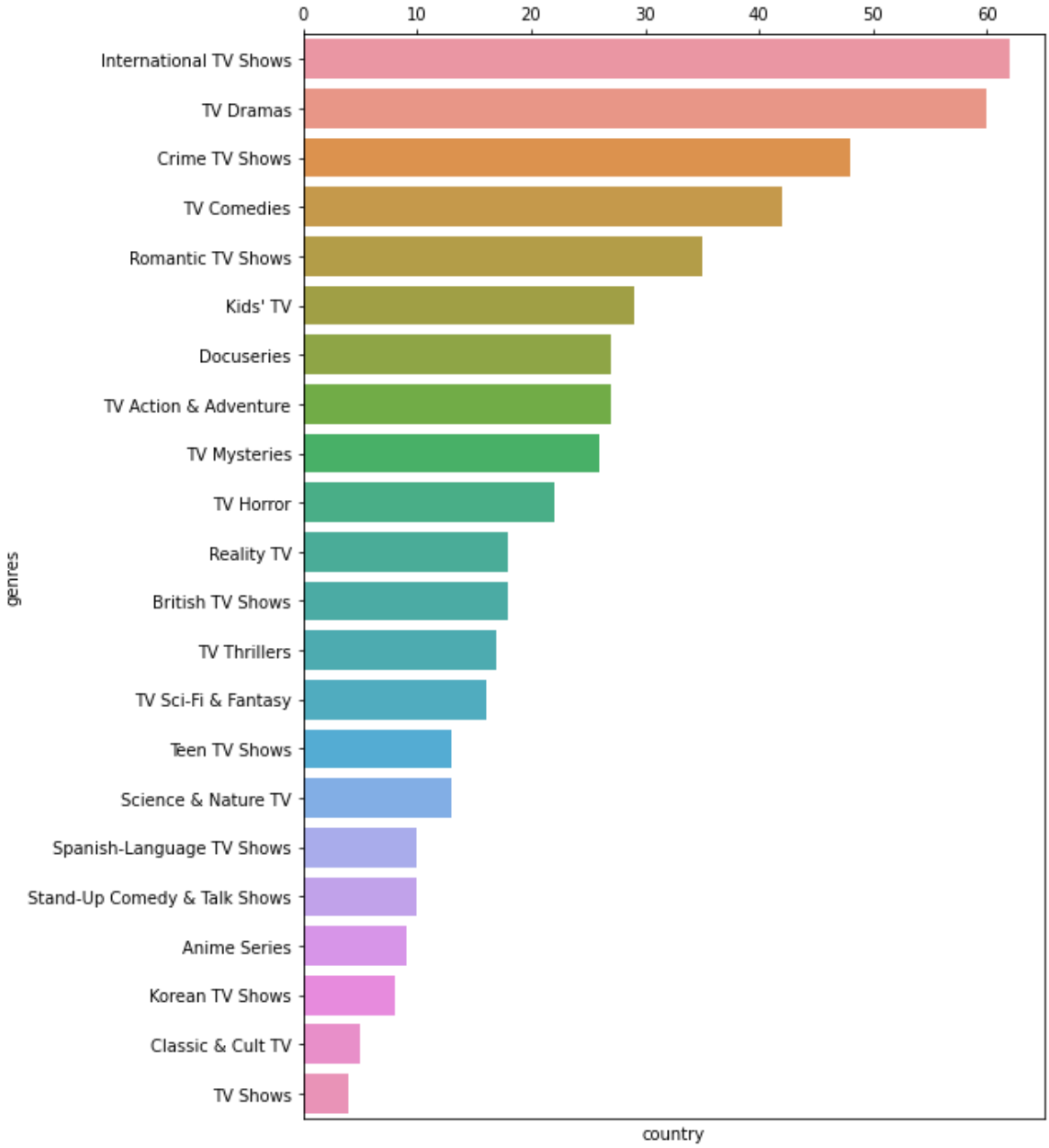
no. of countries per genre

Observation:

- Dramas, Comedies, Crime, Romantic are more popular among countries
- Classic, Science, Nature, Korean, Anime are least popular among countries

In [293]:

```
1
2 plt_df = df.groupby(["genres"]).nunique().reset_index().sort_values(["country"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "genres", x= "country")
7
8 ax.xaxis.tick_top()
```



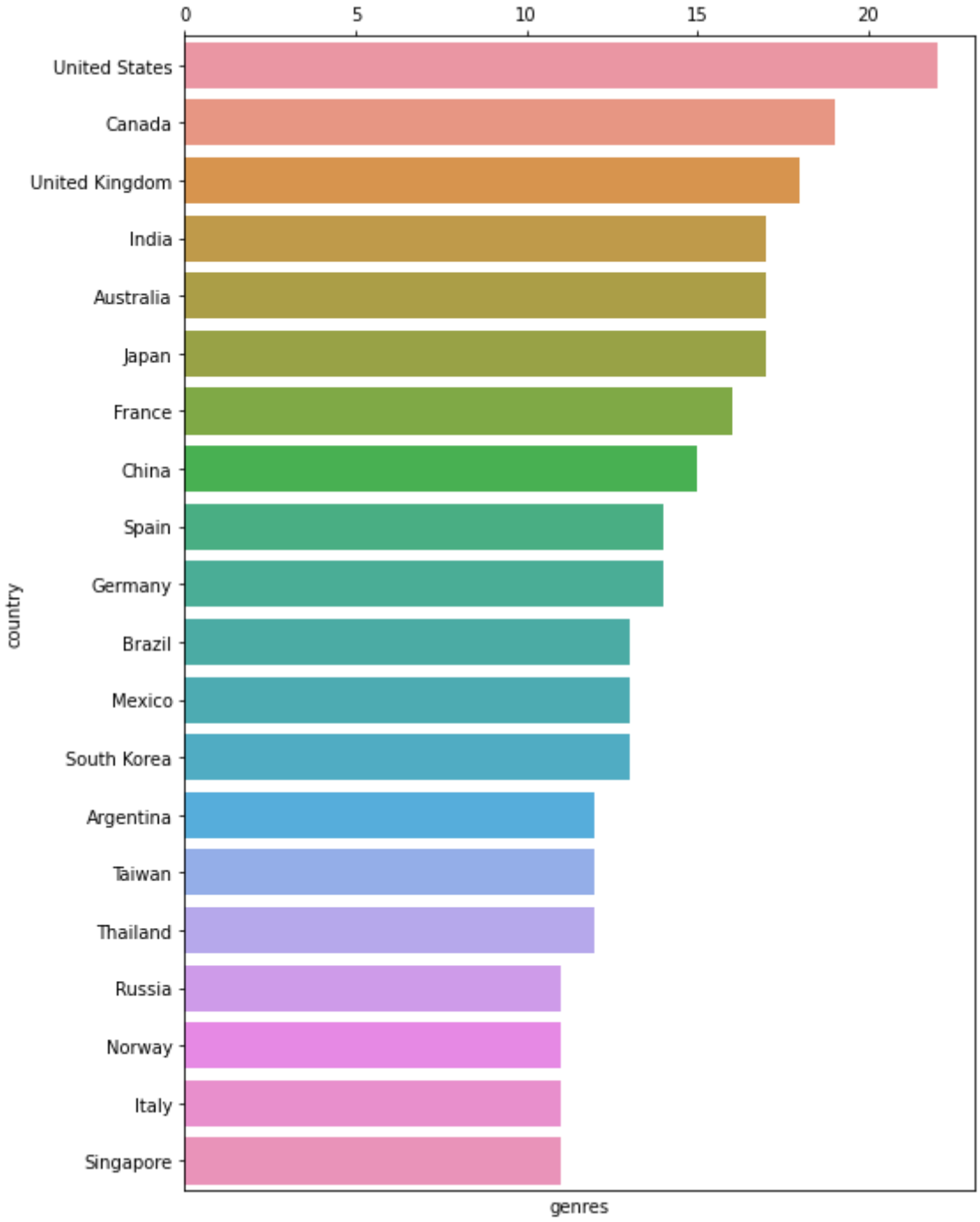
no. of genres per country

Observation:

- Only 10-20 countries (of 113) are streaming more than 10 (of 22) genres

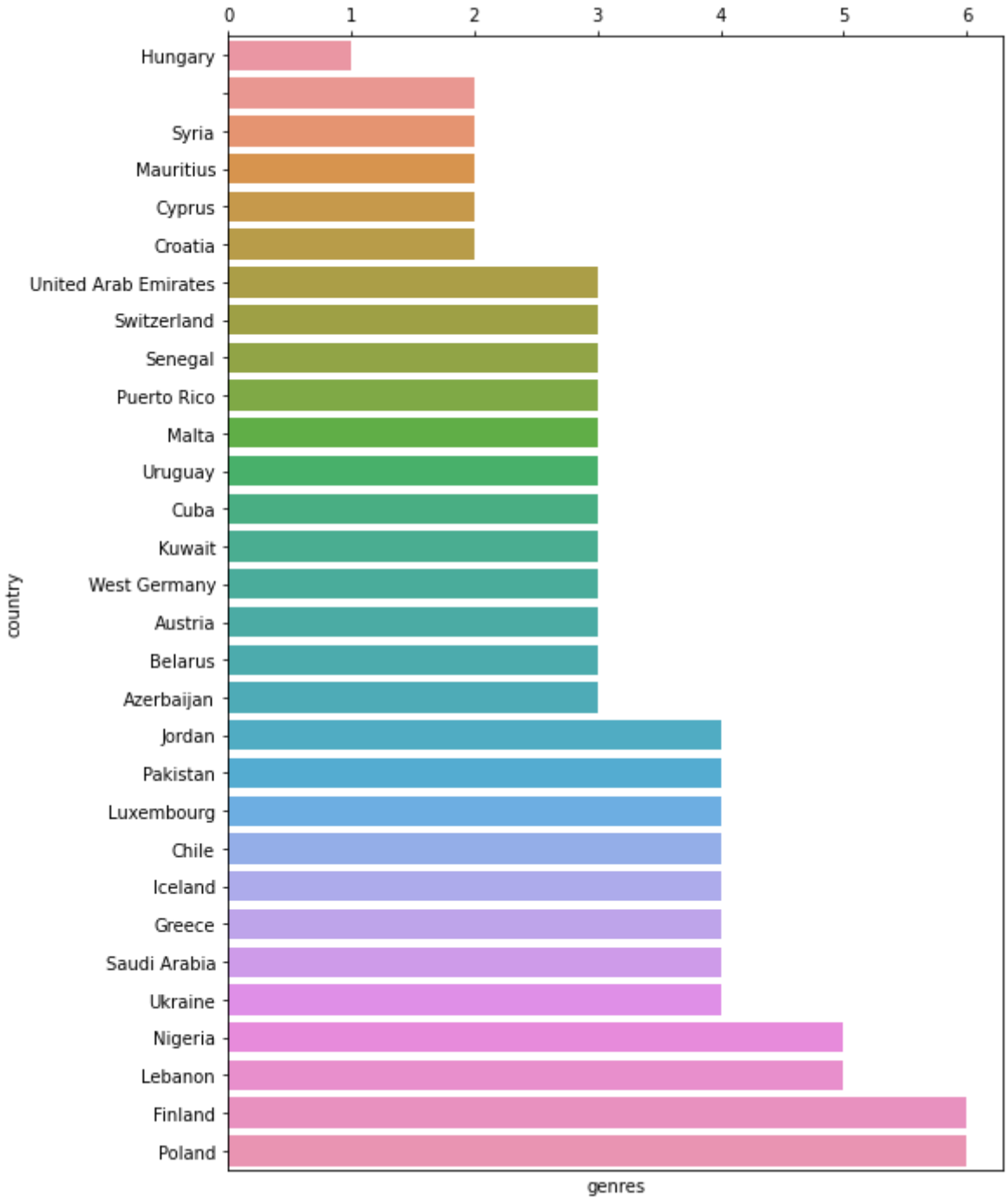
In [294]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["genres"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "genres")
7
8 ax.xaxis.tick_top()
```



In [295]:

```
1
2 plt_df = df.groupby(["country"]).nunique().reset_index().sort_values(["genres"], ascend
3
4 plt.figure(figsize= (8, 12))
5
6 ax = sns.barplot(data= plt_df, y= "country", x= "genres")
7
8 ax.xaxis.tick_top()
```



no. of Shows/ directors/ actors vs added year

Observation:

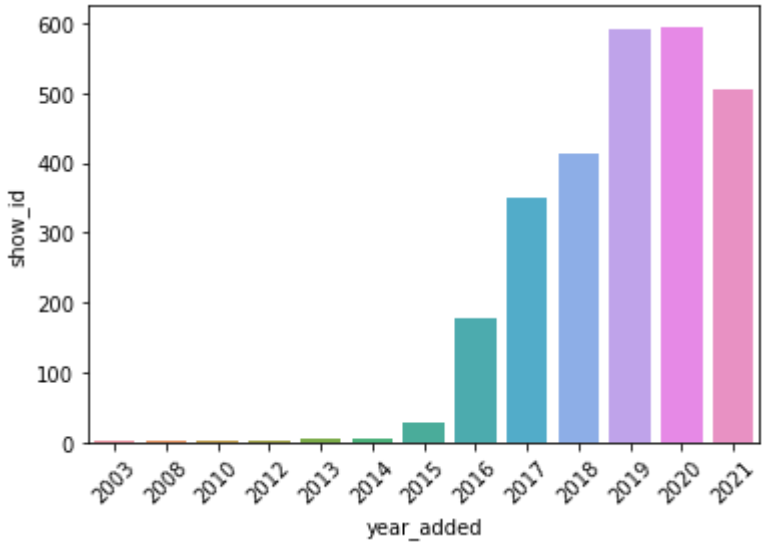
- The no. of Shows/ directors/ actor peaked in 2020
- The no. of Shows/ directors/ actors started growing since 2015

In [296]:

```
1
2 plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir
3
4 plt.xticks(rotation= 45)
5 sns.barplot(data= plt_df, y= "show_id", x= "year_added")
```

Out[296]:

```
<AxesSubplot:xlabel='year_added', ylabel='show_id'>
```

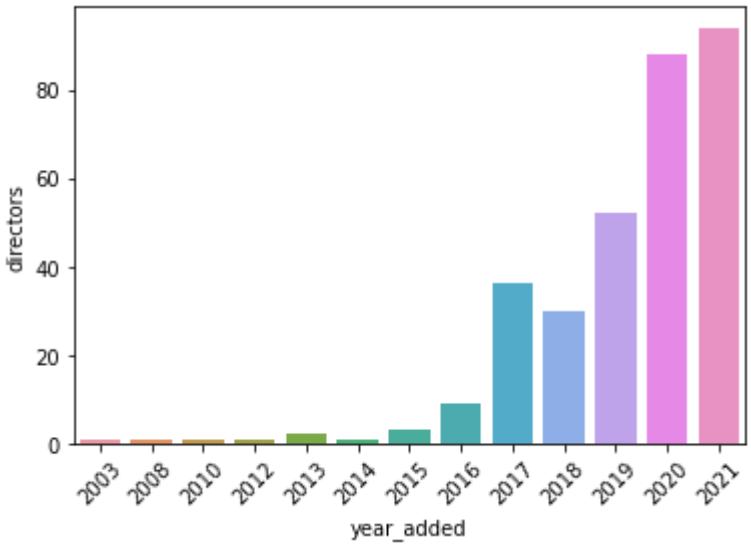


In [297]:

```
plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir

plt.xticks(rotation= 45)
sns.barplot(data= plt_df, y= "directors", x= "year_added")
```

Out[297]:

<AxesSubplot:xlabel='year_added', ylabel='directors'>

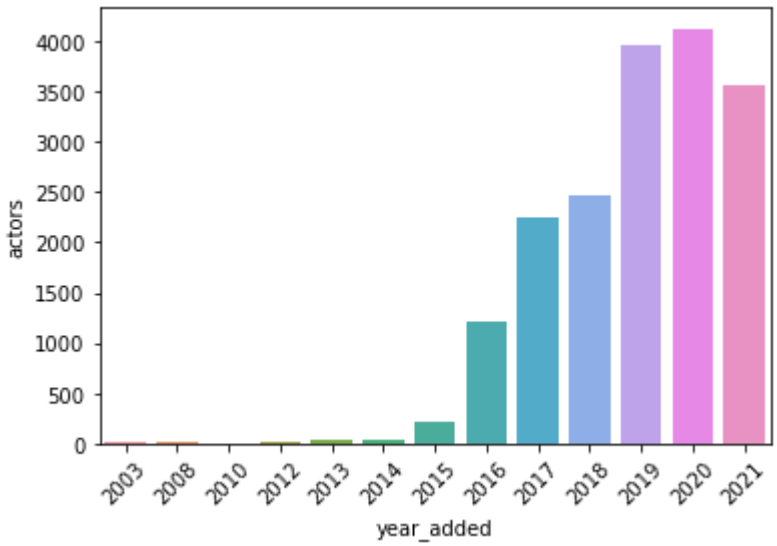


In [298]:

```
plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir

plt.xticks(rotation= 45)
sns.barplot(data= plt_df, y= "actors", x= "year_added")
```

Out[298]:

<AxesSubplot:xlabel='year_added', ylabel='actors'>



no. of countries/ genres streaming per aded year

Observation:

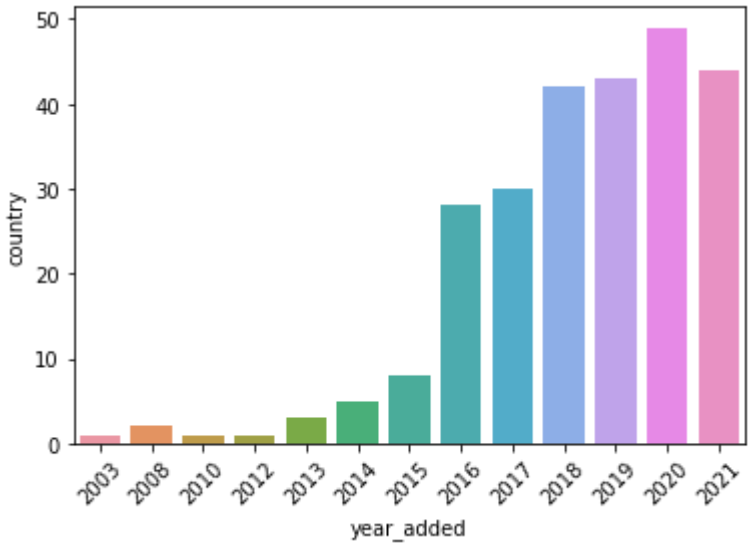
- The no. of countries/ genresstreaming grew b/w years 2015 - 18 and then stabilized around 110

In [299]:

```
1
2 plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir
3
4 plt.xticks(rotation= 45)
5 sns.barplot(data= plt_df, y= "country", x= "year_added")
```

Out[299]:

```
<AxesSubplot:xlabel='year_added', ylabel='country'>
```

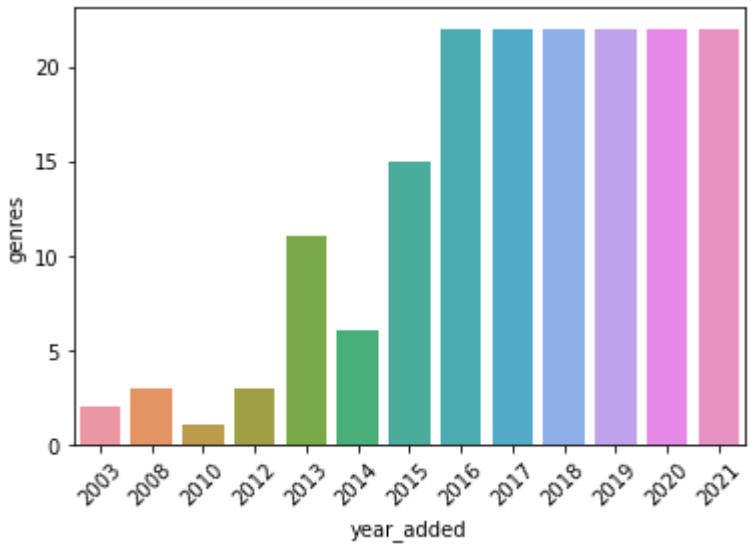


In [300]:

```
1
2 plt_df = df.groupby(df.date_added.dt.year).nunique().rename_axis("year_added").reset_ir
3
4 plt.xticks(rotation= 45)
5 sns.barplot(data= plt_df, y= "genres", x= "year_added")
```

Out[300]:

```
<AxesSubplot:xlabel='year_added', ylabel='genres'>
```



In [301]:

```
def new_nunique(x, df, col):
    year = x.date_added.dt.year.unique()[0]
    prev_col_vals = df.loc[df.date_added.dt.year < year][col].unique()

    result = pd.Series({ col: x.loc[~x[col].isin(prev_col_vals)].nunique()[col]})

    return result
```

no. of new directors/ actors added vs added year

Observation:

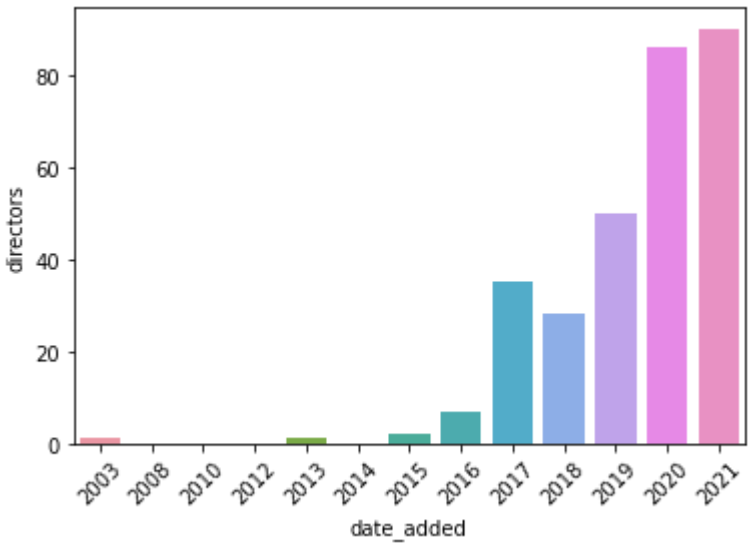
- The no. of new directors/ actor peaked in 2019-20
- The no. of new Shows/ directors/ actors started growing since 2015

In [302]:

```
plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "director

plt.xticks(rotation= 45)

sns.barplot(x= plt_df.index, y= plt_df.directors)
```

Out[302]:

```
<AxesSubplot:xlabel='date_added', ylabel='directors'>
```

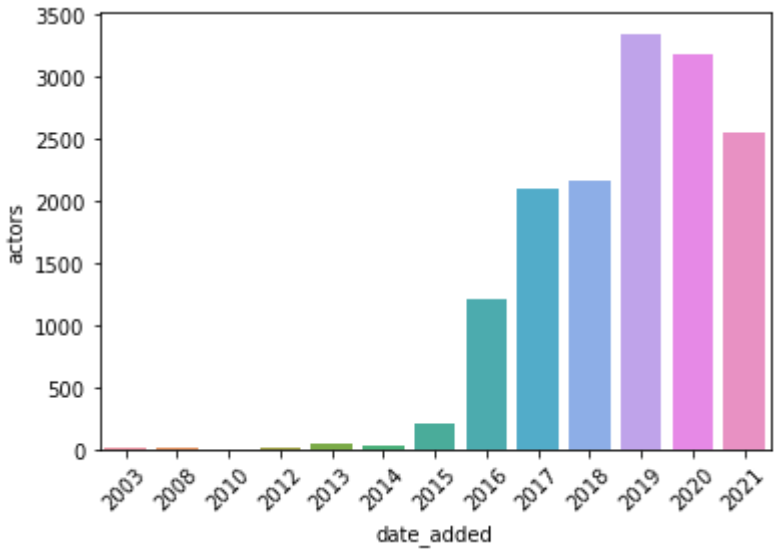


In [303]:

```
1
2 plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "actors"))
3
4 plt.xticks(rotation= 45)
5
6 sns.barplot(x= plt_df.index, y= plt_df.actors)
```

Out[303]:

```
<AxesSubplot:xlabel='date_added', ylabel='actors'>
```



no. of new genres added vs added year

Observation:

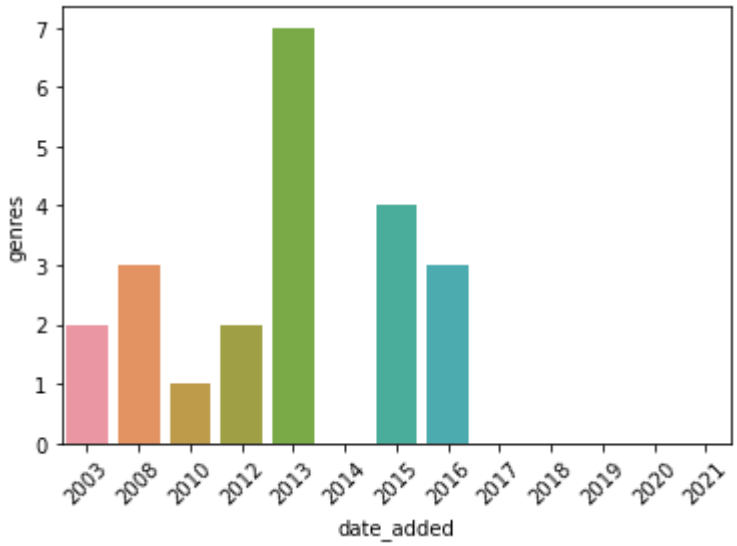
- New genres were added during the initial years and then kept at 20 since 2016

In [304]:

```
1
2 plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "genres"))
3
4 plt.xticks(rotation= 45)
5
6 sns.barplot(x= plt_df.index, y= plt_df.genres)
```

Out[304]:

```
<AxesSubplot:xlabel='date_added', ylabel='genres'>
```



no. of new countries added vs added year

Observation:

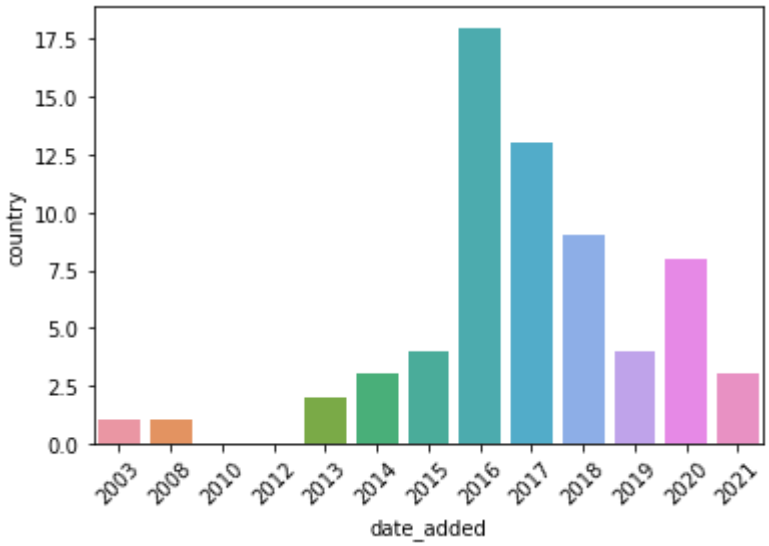
- No. of new countries added started growing from 2014 and peaked in 2017
- After 2017 the no. of new countries added slowed down to reach 113

In [305]:

```
plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "country"

plt.xticks(rotation= 45)

sns.barplot(x= plt_df.index, y= plt_df.country)
```

Out[305]:

<AxesSubplot:xlabel='date_added', ylabel='country'>



In [306]:

```
def new_nunique(x, df, col):
    year = x.release_year.unique()[0]
    prev_col_vals = df.loc[df.release_year < year][col].unique()

    result = pd.Series({ col: x.loc[~x[col].isin(prev_col_vals)].nunique()[col]})

    return result
```

no. of new TV Shows/ directors/ actors added per release year

Observation:

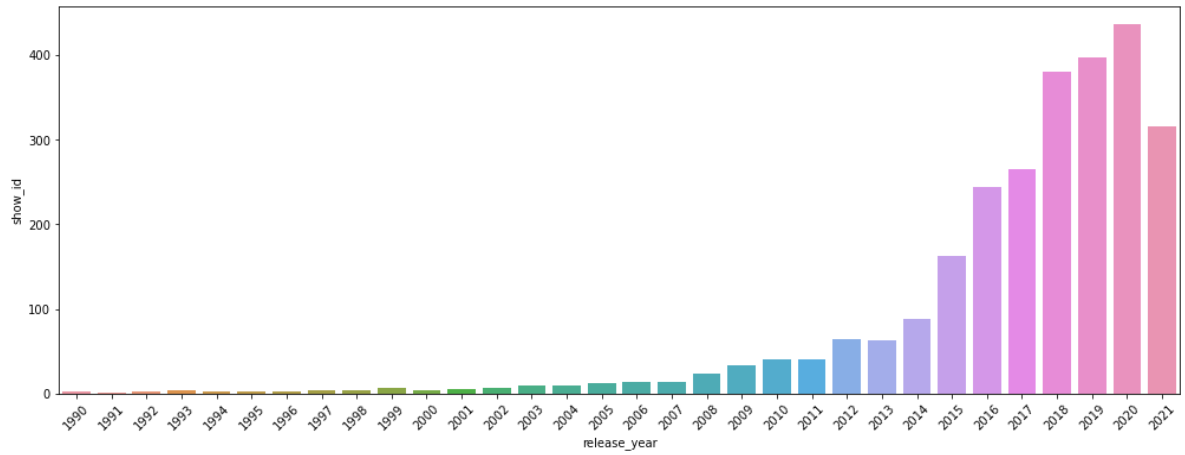
- The new directors/ actors in TV Shows released every year that are added to netflix in peaked in 2019-20
- Post 2019-20 the directors/ actors in newly released TV Shows strted to get repetative

In [307]:

```
plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "show_id")).res
plt_df = plt_df.loc[plt_df.release_year >= 1990]

plt.figure(figsize= (17, 6))
plt.xticks(rotation= 45)

sns.barplot(x= plt_df.release_year, y= plt_df.show_id)
```

Out[307]:

<AxesSubplot:xlabel='release_year', ylabel='show_id'>

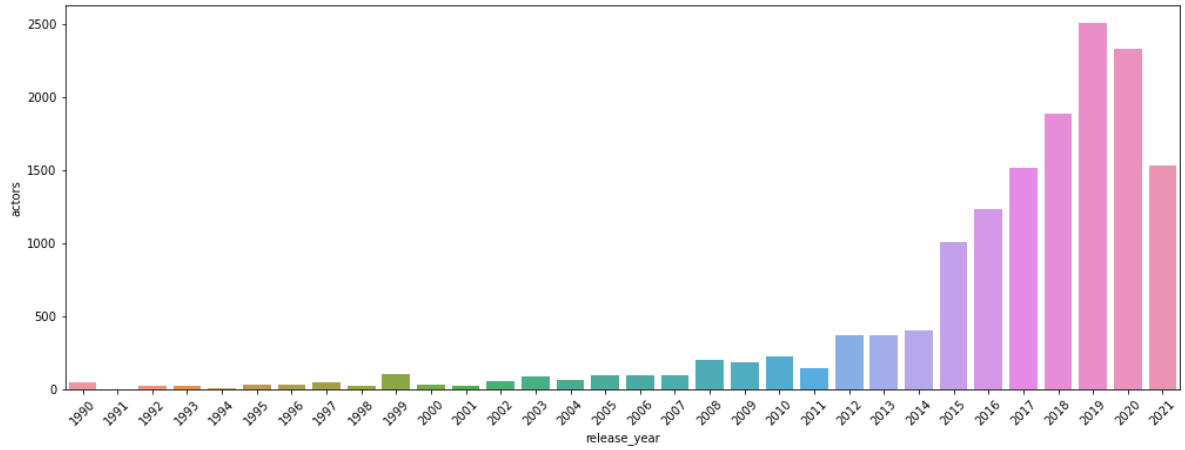


In [308]:

```
plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "actors")).rese
plt_df = plt_df.loc[plt_df.release_year >= 1990]

plt.figure(figsize= (17, 6))
plt.xticks(rotation= 45)

sns.barplot(x= plt_df.release_year, y= plt_df.actors)
```

Out[308]:

<AxesSubplot:xlabel='release_year', ylabel='actors'>

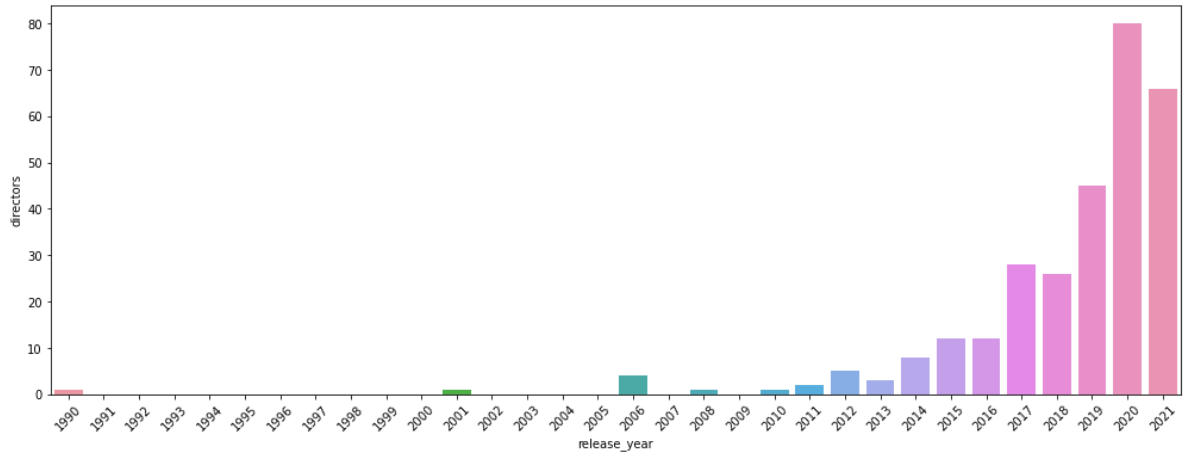


In [309]:

```
1
2 plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "directors")).r
3 plt_df = plt_df.loc[plt_df.release_year >= 1990]
4
5 plt.figure(figsize= (17, 6))
6 plt.xticks(rotation= 45)
7
8 sns.barplot(x= plt_df.release_year, y= plt_df.directors)
```

Out[309]:

```
<AxesSubplot:xlabel='release_year', ylabel='directors'>
```



no. of new countries added per release year

Observation:

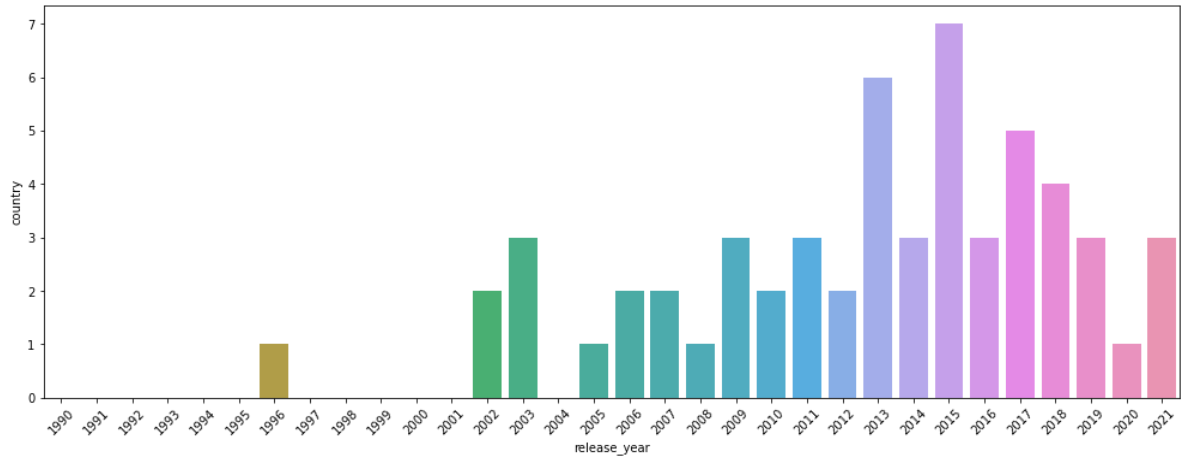
- The expansion to most countries happend through the TV Sohws released in 2013, 2015, 2017 and then stated to stabilize

In [310]:

```
1
2 plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "country")).res
3 plt_df = plt_df.loc[plt_df.release_year >= 1990]
4
5 plt.figure(figsize= (17, 6))
6 plt.xticks(rotation= 45)
7
8 sns.barplot(x= plt_df.release_year, y= plt_df.country)
```

Out[310]:

```
<AxesSubplot:xlabel='release_year', ylabel='country'>
```



no. of new gernes added per release year

Observation:

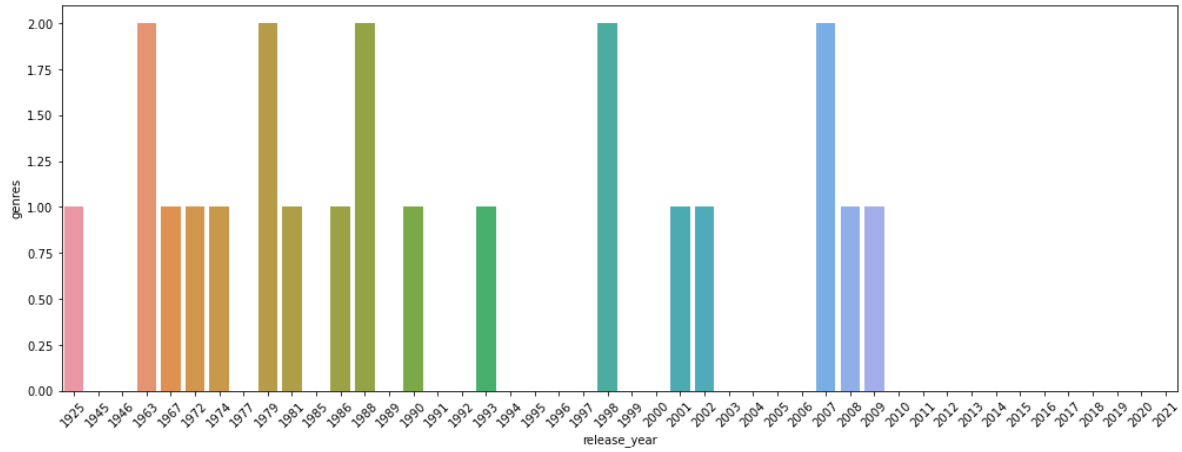
- The TV Show for the latest genre on netflix had released in 2014

In [311]:

```
plt_df = df.groupby(df.release_year).apply(lambda x: new_nunique(x, df, "genres")).rese
# plt_df = plt_df.loc[plt_df.release_year >= 1990]

plt.figure(figsize= (17, 6))
plt.xticks(rotation= 45)

sns.barplot(x= plt_df.release_year, y= plt_df.genres)
```

Out[311]:

```
<AxesSubplot:xlabel='release_year', ylabel='genres'>
```



trend of TV Shows added per release year and year added

Observation:

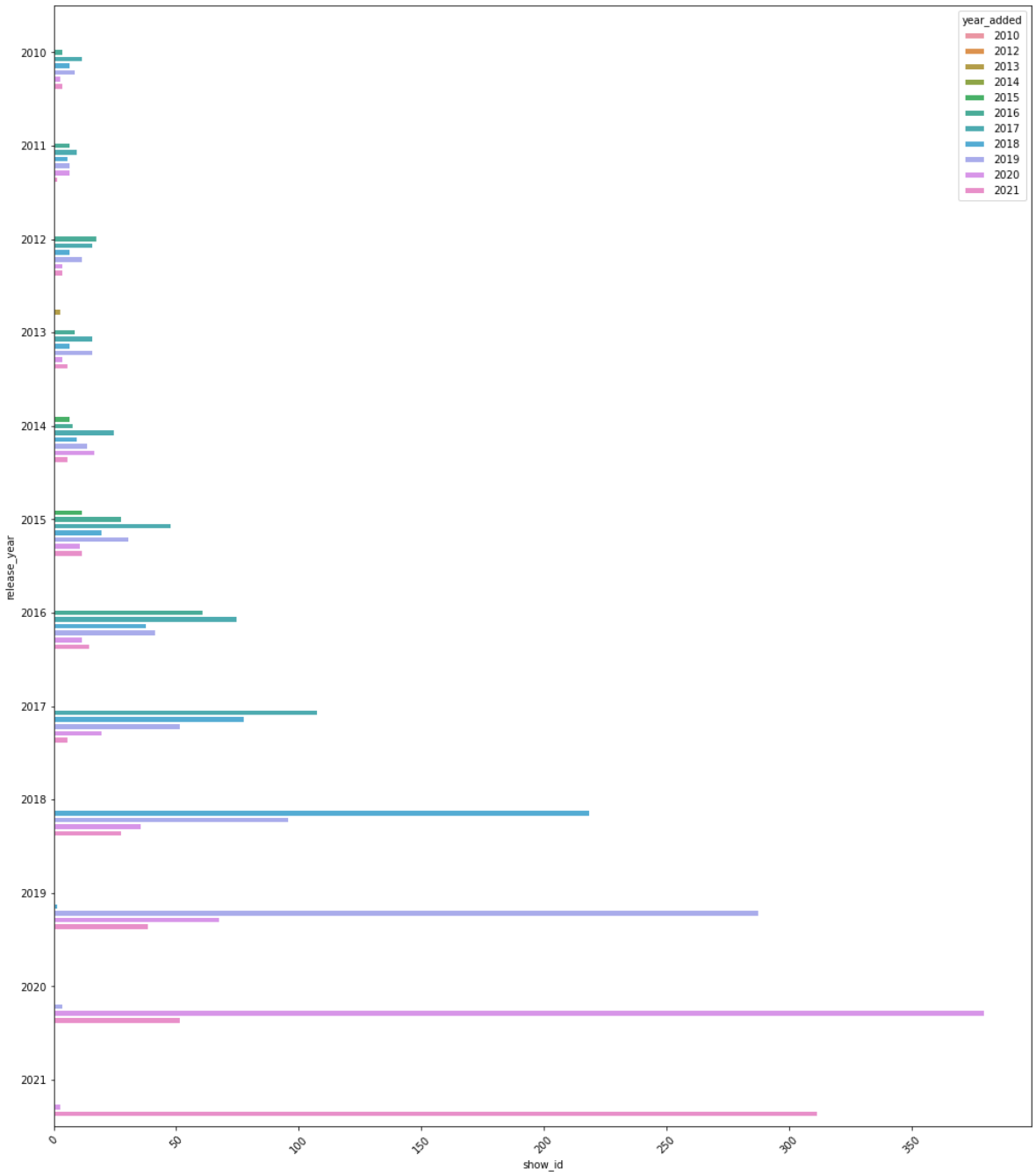
- Among the TV Shows released in a given year the highest no. of TV Shows are added to netflix in that year since 2016
- For the TV Shows released before 2017, the highest no. of TV Shows released in that year are added in 2016-17

In [312]:

```
1
2 plt_df = df.groupby([df.release_year, df.date_added.dt.year]).nunique().rename_axis(
3                                                                                 ["relea
4                                                                                 ).reset
5 plt_df = plt_df[plt_df.release_year >= 2010]
6 plt.figure(figsize= (17, 20))
7 plt.xticks(rotation= 45)
8
9 sns.barplot(y= plt_df.release_year, x= plt_df.show_id, hue= plt_df.year_added, orient=
```

Out[312]:

<AxesSubplot:xlabel='show_id', ylabel='release_year'>



trend of TVShows added per release year and year added

Observation:

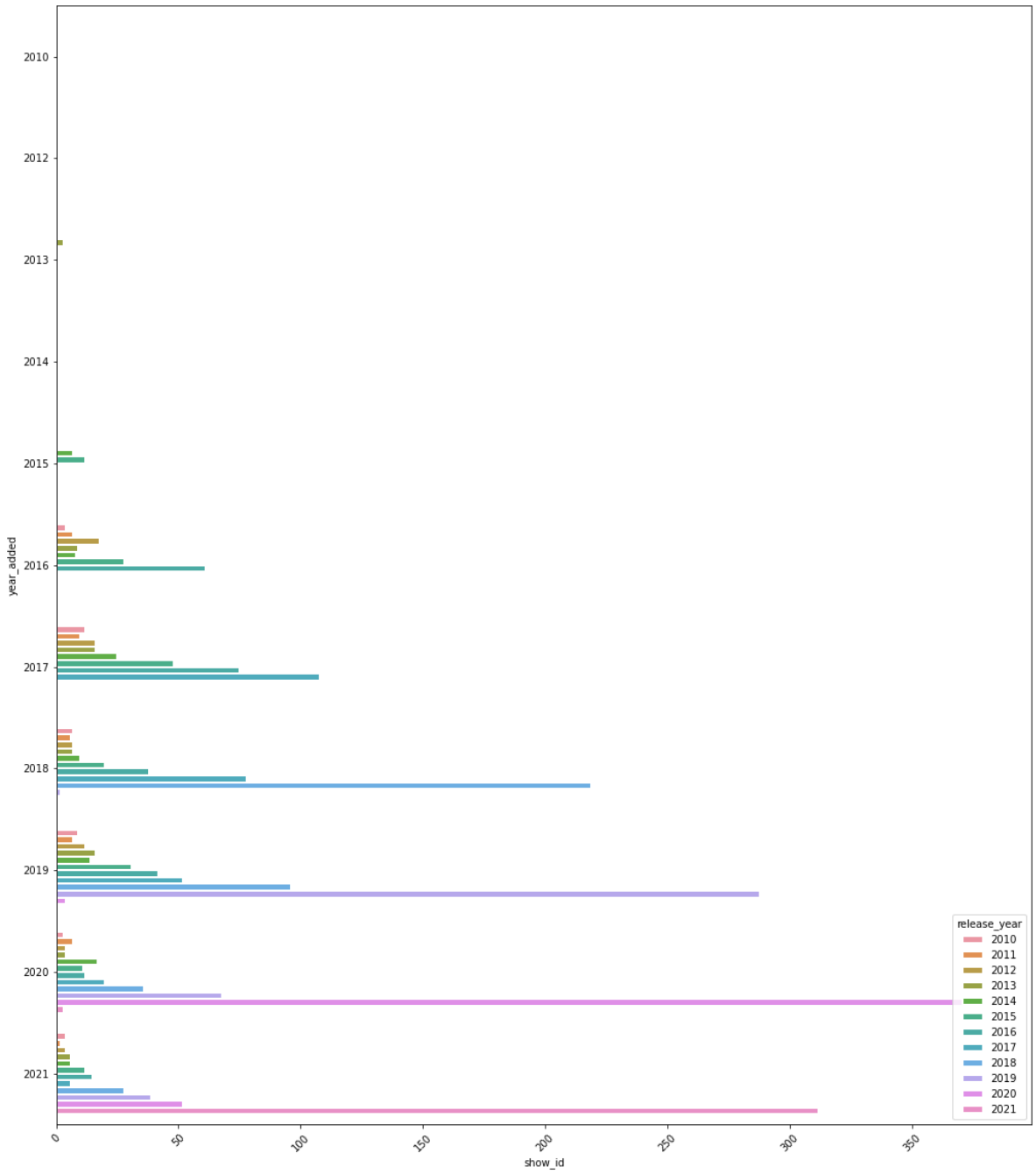
- Every year te highes t no. of TV Shows added in that year belong to the movies released in that year

In [313]:

```
1
2 plt_df = df.groupby([df.release_year, df.date_added.dt.year]).nunique().rename_axis(
3                                                                                 ["relea
4                                                                                 ).reset
5 plt_df = plt_df[plt_df.release_year >= 2010]
6 plt.figure(figsize= (17, 20))
7 plt.xticks(rotation= 45)
8
9 sns.barplot(y= plt_df.year_added, x= plt_df.show_id, hue= plt_df.release_year, orient=
```

Out[313]:

<AxesSubplot:xlabel='show_id', ylabel='year_added'>



In [314]:

```
def new_nunique(x, df, col):
    year = x.date_added.dt.year.unique()[0]

    temp = df.loc[(df.date_added.dt.year >= year-4) & (df.date_added.dt.year <= year)].

    result = temp.groupby(temp.date_added.dt.year).nunique()

#     print("-"*50 + "\n", dict(zip(result.index.values, year - result.index.values)),

    new_index = dict(zip(result.index.values, year - result.index.values))

    result = result.rename(index= new_index)[col]

    return result
```

In [315]:

```
plt_df = df.groupby(df.date_added.dt.year).apply(lambda x: new_nunique(x, df, "show_id"

plt_df = plt_df.rename_axis(["year_added", "last_n_year_added"]).reset_index()
```

trend of TV Shows added per release year and year added

Observation:

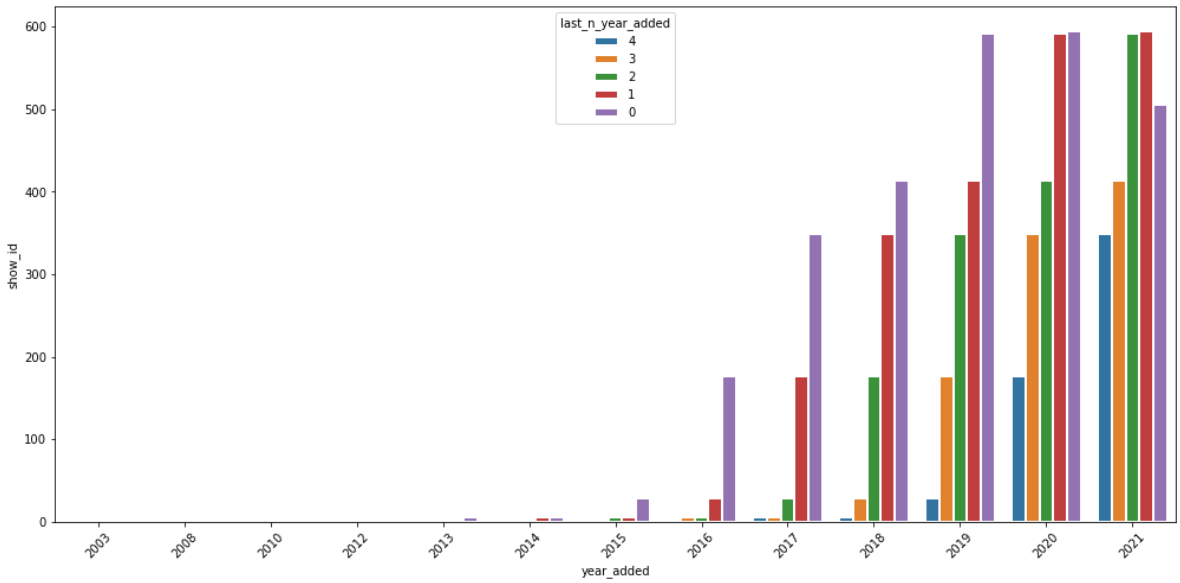
- Till 2020, most of the TV Shows on netflix in a given year belong to the TV Shows released that year
- Since 2020, most of the movies on Netflix by the end of that year belong to the movies released in previous year

In [316]:

```
1
2 plt.figure(figsize= (17, 8))
3 plt.xticks(rotation= 45)
4
5 sns.barplot(x= plt_df.year_added, y= plt_df.show_id, hue= plt_df.last_n_year_added,
6             hue_order = [4, 3, 2, 1, 0],
7             orient= 'v', edgecolor='white', linewidth=2)
```

Out[316]:

```
<AxesSubplot:xlabel='year_added', ylabel='show_id'>
```

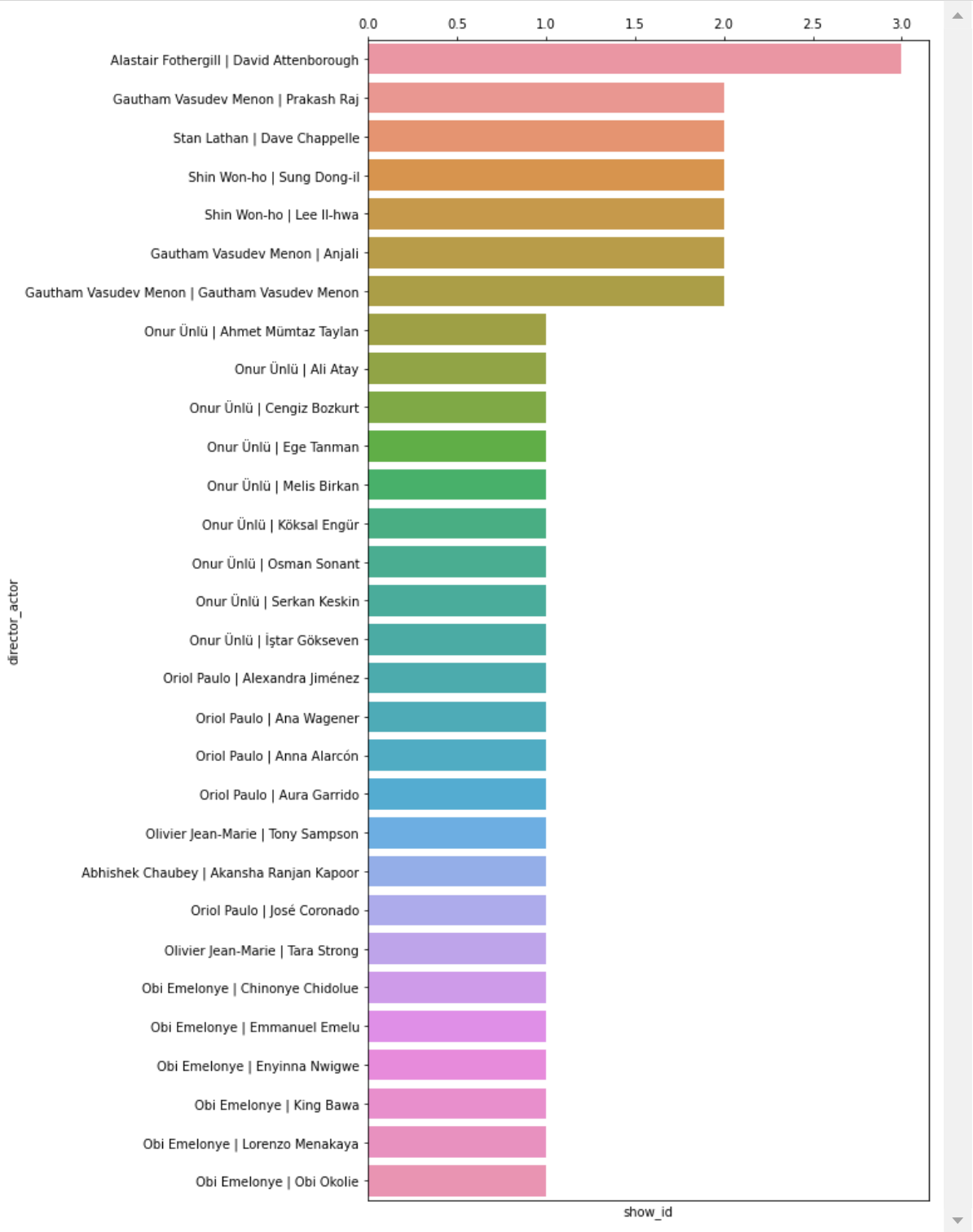


no. of TV Shows per possible pair

- pairs between actors, directors, genres

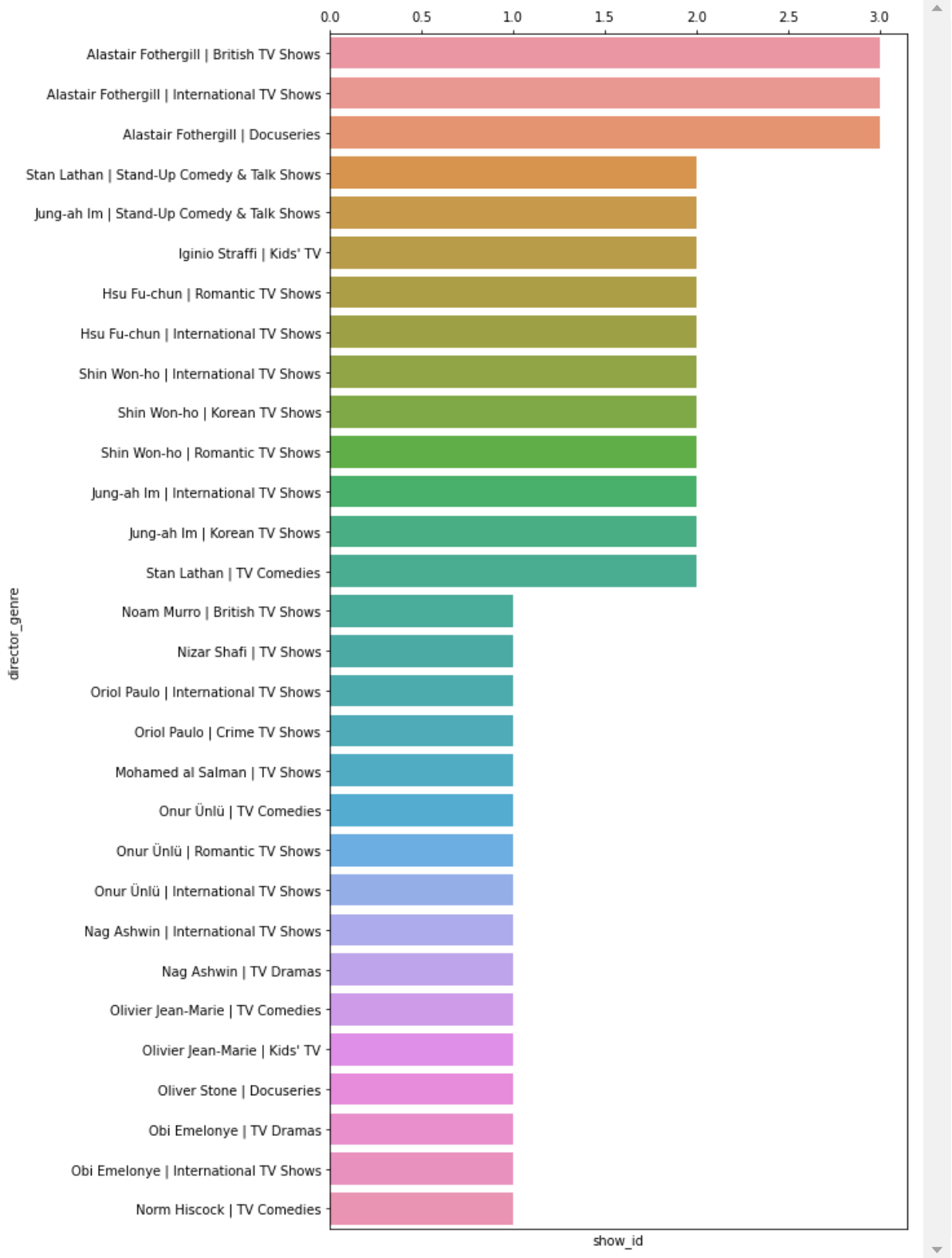
In [317]:

```
1
2  plt_df = df.loc[(df.directors != "Anonymous") & (df.actors != "Anonymous")].groupby(
3                                                                                  ["directors
4                                                                                  ).nunique()
5
6
7  plt.figure(figsize= (8, 17))
8
9  plt_df["director_actor"] = plt_df.apply(lambda x: x["directors"] + " | " + x["actors"],
10
11 ax = sns.barplot(y= plt_df.director_actor, x= plt_df.show_id)
12
13 ax.xaxis.tick_top()
```



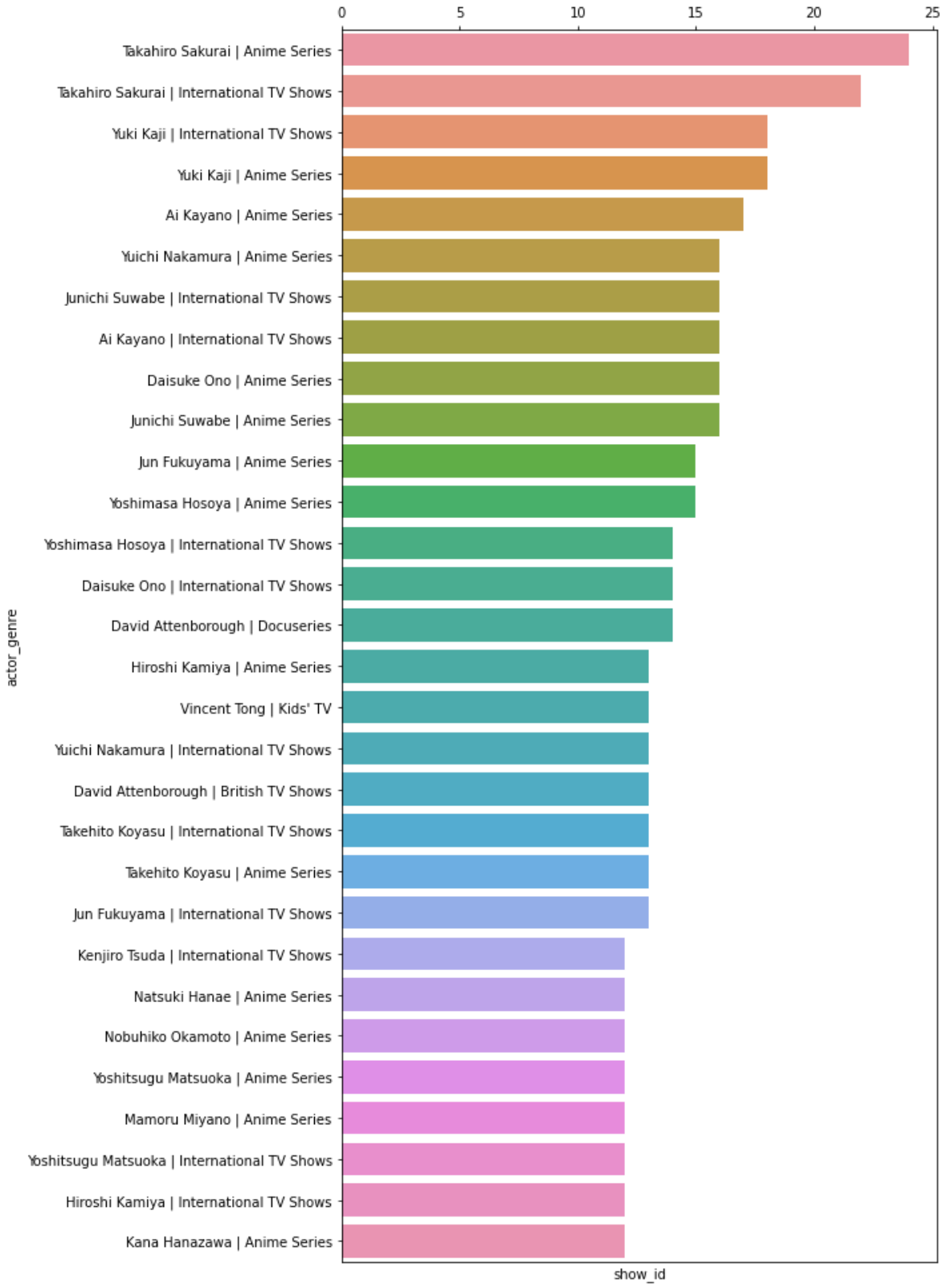
In [318]:

```
1
2  plt_df = df.loc[(df.directors != "Anonymous") & (df.actors != "Anonymous")].groupby(["
3                                                                    ascending= False)["
4
5  plt.figure(figsize= (8, 17))
6
7  plt_df["director_genre"] = plt_df.apply(lambda x: x["directors"] + " | " + x["genres"],
8
9  ax = sns.barplot(y= plt_df.director_genre, x= plt_df.show_id)
10
11 ax.xaxis.tick_top()
```



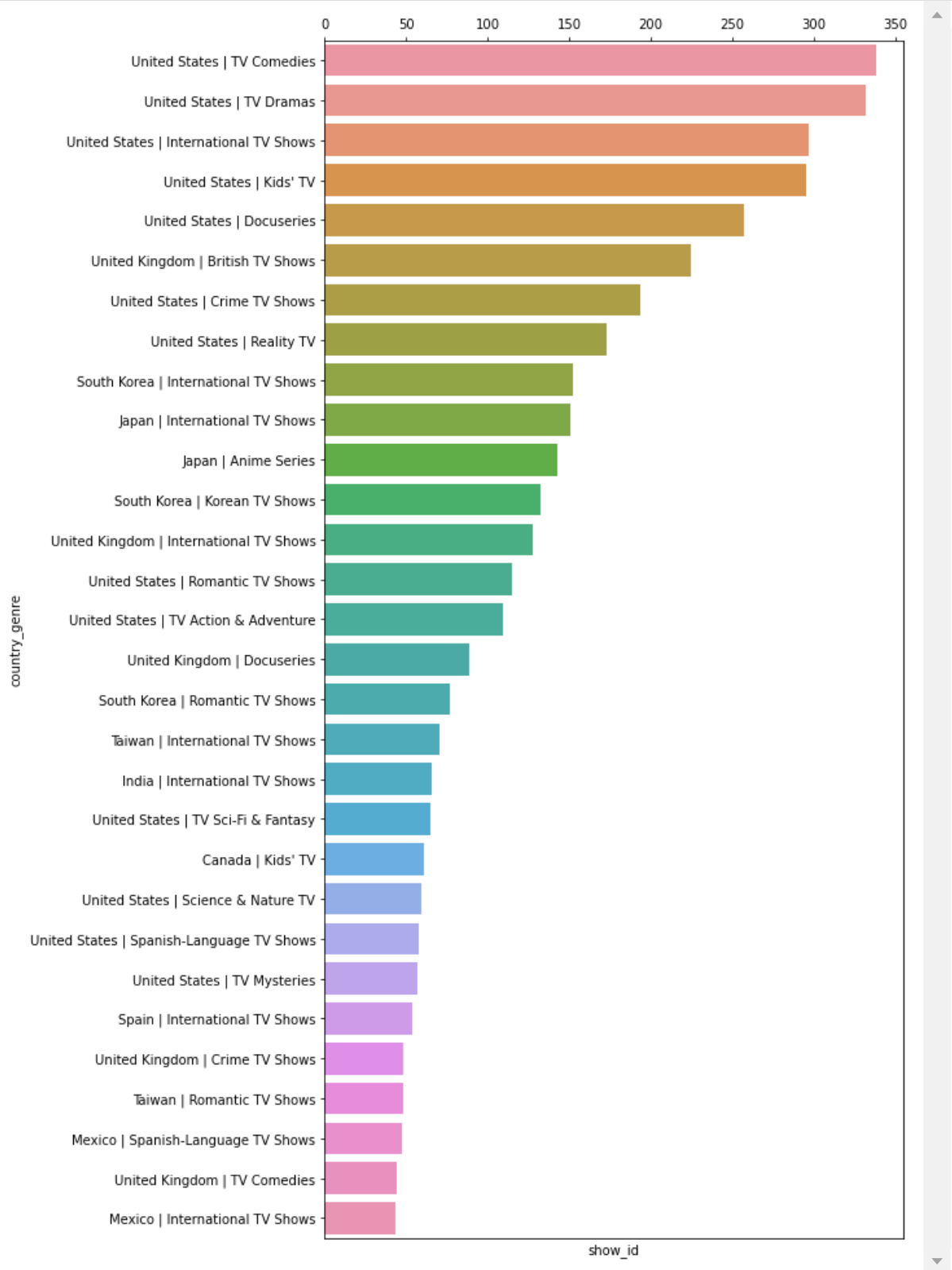
In [319]:

```
1
2  plt_df = df.loc[(df.actors != "Anonymous")].groupby(["actors", "genres"]).nunique().sor
3                                                                    ascending= False)["s
4
5  plt.figure(figsize= (8, 17))
6
7  plt_df["actor_genre"] = plt_df.apply(lambda x: x["actors"] + " | " + x["genres"], axis=
8
9  ax = sns.barplot(y= plt_df.actor_genre, x= plt_df.show_id)
10
11 ax.xaxis.tick_top()
```



In [320]:

```
1
2  plt_df = df.groupby(["country", "genres"]).nunique().sort_values("show_id",
3                                                                    ascending= False)["s
4
5  plt.figure(figsize= (8, 17))
6
7  plt_df["country_genre"] = plt_df.apply(lambda x: x["country"] + " | " + x["genres"], ax
8
9  ax = sns.barplot(y= plt_df.country_genre, x= plt_df.show_id)
10
11 ax.xaxis.tick_top()
```



no. of values per min_age

- values: directors, actors, country, genre

In [321]:

```
1
2 df.directors.nunique()
```

Out[321]:

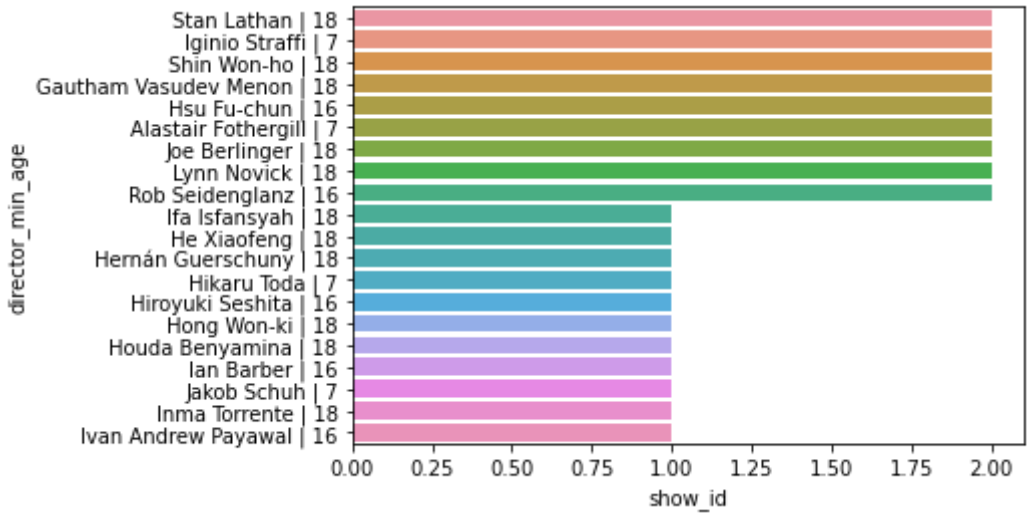
```
300
```

In [322]:

```
plt_df = df.loc[(df.directors != "Anonymous")].groupby(["directors", "min_age"]).nuniqu
                                                                                ["s
                                                                                asc

plt_df = plt_df.groupby("directors").head(1).sort_values("show_id", ascending= False)

plt_df["director_min_age"] = plt_df.apply(lambda x: x["directors"] + " | " + str(x["mir

sns.barplot(data= plt_df.head(20), y= "director_min_age", x= "show_id", orient= 'h')
```

Out[322]:

<AxesSubplot:xlabel='show_id', ylabel='director_min_age'>

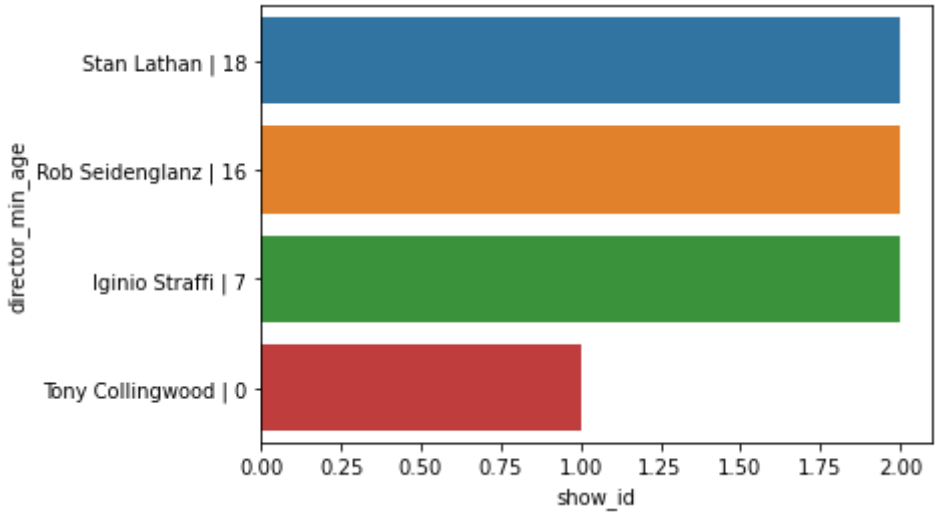


In [323]:

```
plt_df = df.loc[(df.directors != "Anonymous")].groupby(["directors", "min_age"]).nuniqu
                                                                                ["s
                                                                                asc

plt_df = plt_df.groupby("min_age").head(1).sort_values("show_id", ascending= False)

plt_df["director_min_age"] = plt_df.apply(lambda x: x["directors"] + " | " + str(x["mir

sns.barplot(data= plt_df.head(20), y= "director_min_age", x= "show_id", orient= 'h')
```

Out[323]:

<AxesSubplot:xlabel='show_id', ylabel='director_min_age'>

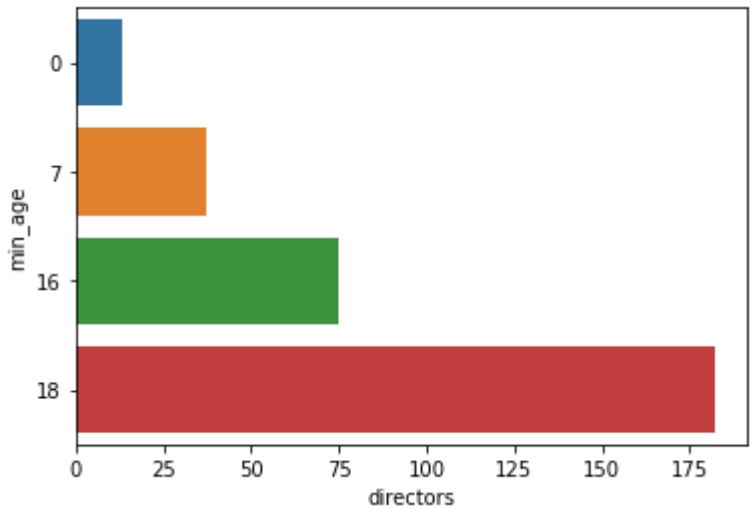


In [324]:

```
plt_df = df.groupby("min_age").nunique()["directors"].reset_index()

sns.barplot(y= plt_df.min_age, x= plt_df.directors, orient= 'h')
```

Out[324]:

<AxesSubplot:xlabel='directors', ylabel='min_age'>



Correlation Analysis

Movies Data Correlation

Observation:

- Ther isn't much correlation between any numerical data:
  - duration, min_age, release_year

In [325]:

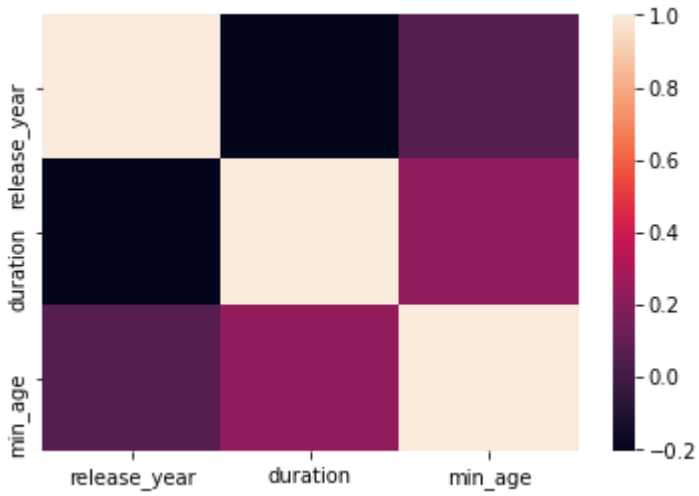
```
df = netflix_data_listed.loc[netflix_data_listed.type == "Movie"].copy()
```

In [326]:

```
sns.heatmap(df.corr())
```

Out[326]:

<AxesSubplot:>

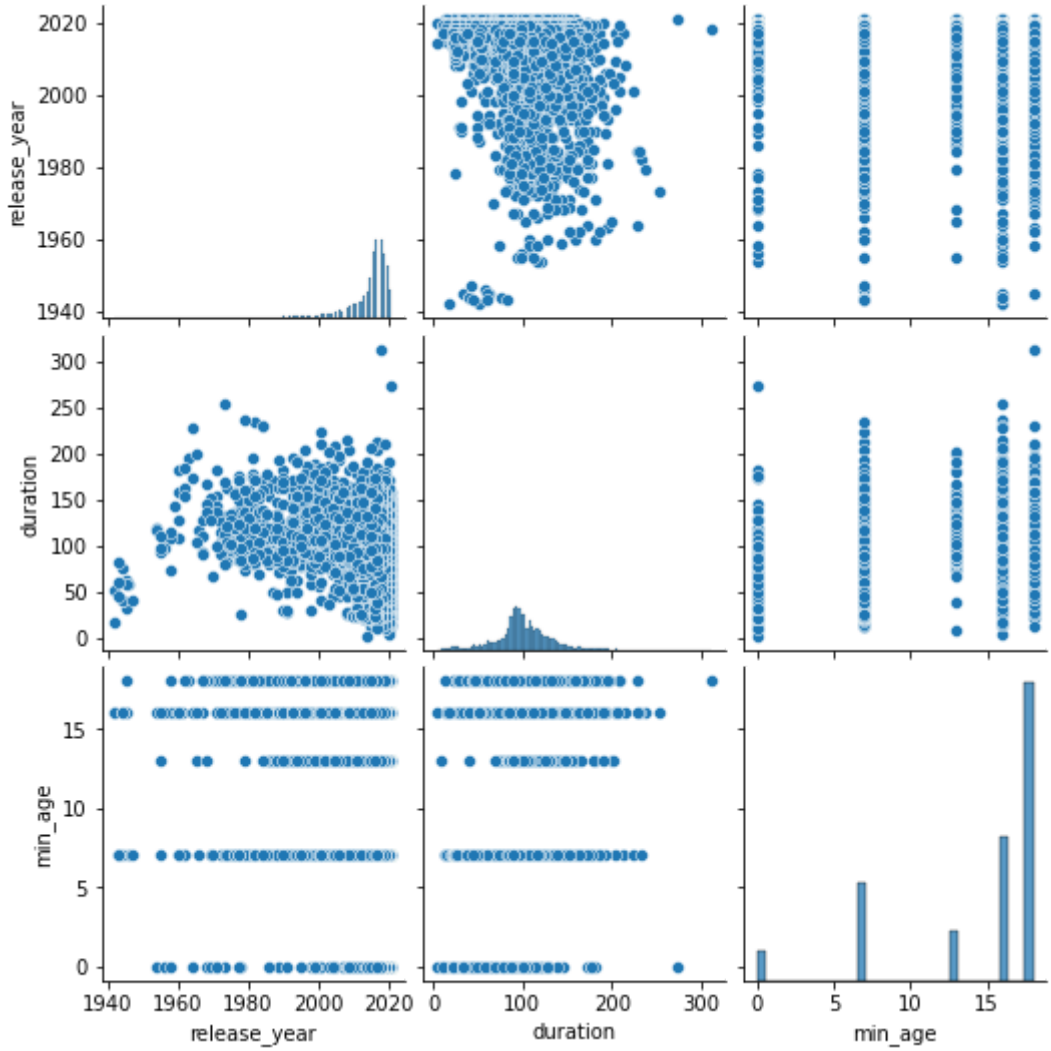


In [327]:

```
1
2 sns.pairplot(df)
```

Out[327]:

```
<seaborn.axisgrid.PairGrid at 0x1ddb4a43948>
```



TV Show Data Correlation

Observation:

- Ther isn't much correlation between any numerical data:
  - duration, min_age, release_year

In [328]:

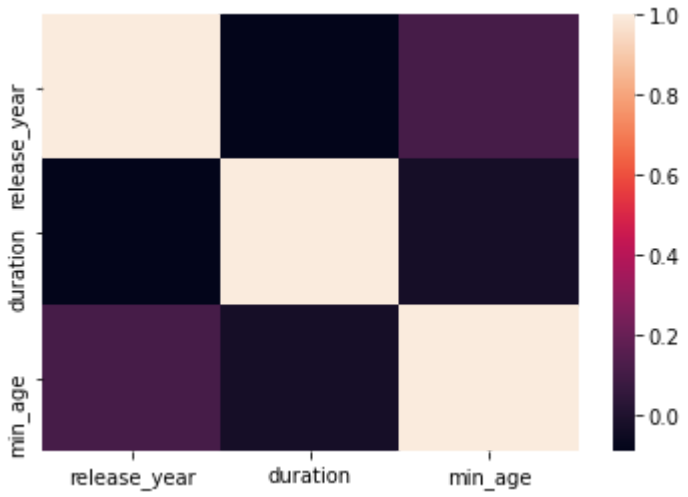
```
df = netflix_data_listed.loc[netflix_data_listed.type == "TV Show"].copy()
```

In [329]:

```
sns.heatmap(df.corr())
```

Out[329]:

<AxesSubplot:>

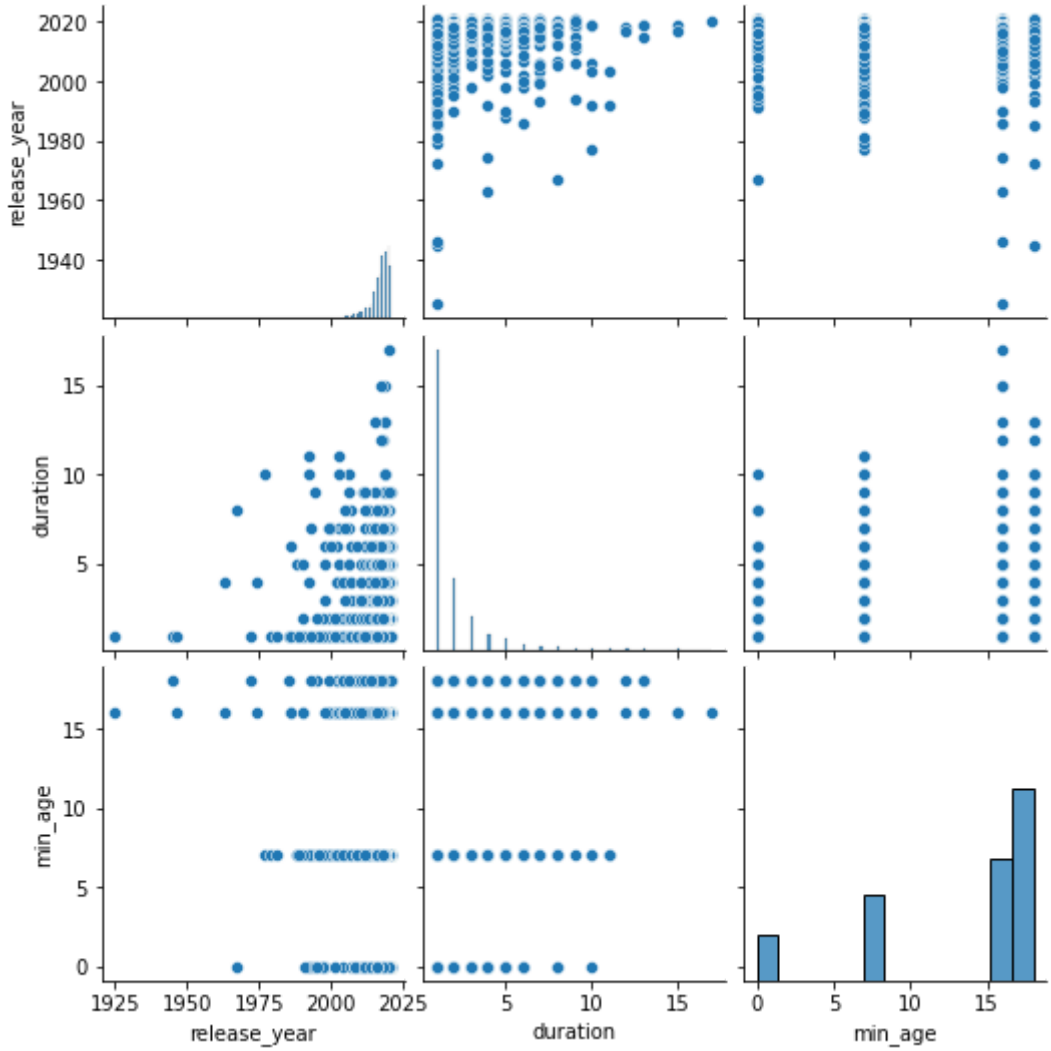


In [330]:

```
1
2 sns.pairplot(df)
```

Out[330]:

```
<seaborn.axisgrid.PairGrid at 0x1ddbee5f0c8>
```



Business Insights:

- There are very few Movies and TV Shows on Netflix that were released before 2000
- There are very few Movies that are shorter than an hour in duration and there are very very few short films
- There are very few TV Shows that are more than 4 Seasons and there are very very few long-form shows
- Most Movies/ TV Shows are added on Friday or Thursday, very few are added on Weekends
- Most of the Movies/ TV Shows are streaming for very few countries (10)
- There are very few Classic, Cult, Anime, Sci-Fi Movies on Netflix
- There are very few Classic, Cult, Reality, Thriller, Teen, Science & Nature TV Shows on Netflix
- There are very few Movies/ TV Shows that are targeted toward Generic Age Group (0+)
- Among the new Movies/ TV Shows, up and coming Actors/ Directors are very few
- Most Movies/ TV Shows are released at the start of the month and few TV Shows and Movies in the middle of the month

Recommendations:

- Limited Period Streaming of Old Movies/ TV Shows and permanently adding the successful ones
- Introducing Short Films or Movies with a shorter duration
- Introducing recurring Live Streams or TV Shows of longer duration
- In-app Movie/ TV Shows release Events to keep every day/ week/ month interesting, such as Horror Day/ Thriller Week, and Sci-Fi month.
- Expanding the streaming availability to more countries
- Producing or Adding more Movies/ TV Shows with up-and-coming Artists
- Adding more movies for the generic (0+) age group
- Adding more Movies/ TV Shows of niche genres such as Sci-FI, Thrillers, Anime, Science & nature etc

In [ ]:

```
1
```