In [ ]:

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv('netflix.csv')
df.head()
```

Out[1]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Do
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	T
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	T
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	

In [2]:
```
df.shape
```

Out[2]:
```
(8807, 12)
```

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
```

```
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

In [4]:
```
df.describe()
df.describe(include='object').T
```

Out[4]:

	count	unique	top	freq
show_id	8807	8807	s1	1
type	8807	2	Movie	6131
title	8807	8807	Dick Johnson Is Dead	1
director	6173	4528	Rajiv Chilaka	19
cast	7982	7692	David Attenborough	19
country	7976	748	United States	2818
date_added	8797	1767	January 1, 2020	109
rating	8803	17	TV-MA	3207
duration	8804	220	1 Season	1793
listed_in	8807	514	Dramas, International Movies	362
description	8807	8775	Paranormal activity at a lush, abandoned prope...	4

In [5]:
```
df.isnull().sum()/len(df)*100
```

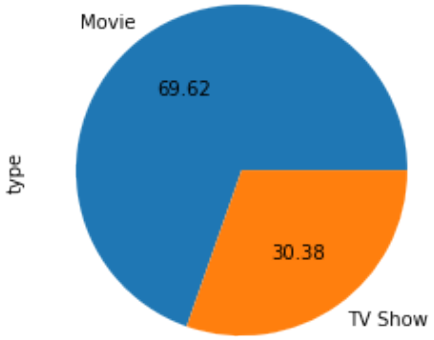
Out[5]:
```
show_id         0.000000
type            0.000000
title           0.000000
director       29.908028
cast            9.367549
country         9.435676
date_added      0.113546
release_year    0.000000
rating          0.045418
duration        0.034064
listed_in       0.000000
description     0.000000
dtype: float64
```

In [6]:
```
df['type'].value_counts(normalize=True)*100
```

Out[6]:
```
Movie      69.615079
TV Show    30.384921
Name: type, dtype: float64
```

In [7]:
```
df['type'].value_counts().plot(kind='pie',autopct='%.2f')
```

Out[7]:
```
<AxesSubplot:ylabel='type'>
```



In [29]:
```
df.head()
```

Out[29]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Dc
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	T
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	T
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	

In [9]:
```
df['release_year'].describe()
```

Out[9]:
```
count    8807.000000
mean     2014.180198
std         8.819312
min      1925.000000
25%      2013.000000
50%      2017.000000
75%      2019.000000
```

```
max      2021.000000
Name: release_year, dtype: float64
```

In [10]:
```
df['type'].describe()
```

Out[10]:
```
count      8807
unique        2
top       Movie
freq       6131
Name: type, dtype: object
```

In [11]:
```
df['country'].describe()
```

Out[11]:
```
count              7976
unique              748
top       United States
freq               2818
Name: country, dtype: object
```

In [12]:
```
df['rating'].describe()
```

Out[12]:
```
count      8803
unique       17
top       TV-MA
freq       3207
Name: rating, dtype: object
```

In [13]:
```
categorical_columns = ['type', 'country', 'rating']
df[categorical_columns] = df[categorical_columns].astype('category')
after_conversion_data_types = df.dtypes
df.isnull().sum()
```

Out[13]:
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

In [14]:
```
value_counts_type = df['type'].value_counts()
value_counts_country = df['country'].value_counts().head(10)  # Top 10 countries
value_counts_rating = df['rating'].value_counts()
value_counts_release_year = df['release_year'].value_counts().head(10)  # Top 10 release
```

In [15]:
```
unique_type = df['type'].unique()
unique_country = df['country'].unique()
unique_rating = df['rating'].unique()
unique_release_year = df['release_year'].unique()

value_counts_type, value_counts_country, value_counts_rating, value_counts_release_year,
```

Out[15]:
```
(Movie      6131
 TV Show    2676
 Name: type, dtype: int64,
 United States     2818
 India              972
```

```
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Egypt              106
Name: country, dtype: int64,
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
UR             3
NC-17          3
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64,
2018    1147
2017    1032
2019    1030
2020     953
2016     902
2021     592
2015     560
2014     352
2013     288
2012     237
Name: release_year, dtype: int64,
['Movie', 'TV Show']
Categories (2, object): ['Movie', 'TV Show'],
['United States', 'South Africa', NaN, 'India', 'United States, Ghana, Burkina Faso, Unit
ed Ki..., ..., 'Russia, Spain', 'Croatia, Slovenia, Serbia, Montenegro', 'Japan, Canada',
'United States, France, South Korea, Indonesia', 'United Arab Emirates, Jordan']
Length: 749
Categories (748, object): [', France, Algeria', ', South Korea', 'Argentina', 'Argentina,
Brazil, France, Poland, Germany, D..., ..., 'Venezuela, Colombia', 'Vietnam', 'West German
y', 'Zimbabwe'],
['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', ..., '66 min', 'NR', NaN, 'TV-Y7-FV', 'UR']
Length: 18
Categories (17, object): ['66 min', '74 min', '84 min', 'G', ..., 'TV-Y', 'TV-Y7', 'TV-Y7
-FV', 'UR'],
array([2020, 2021, 1993, 2018, 1996, 1998, 1997, 2010, 2013, 2017, 1975,
       1978, 1983, 1987, 2012, 2001, 2014, 2002, 2003, 2004, 2011, 2008,
       2009, 2007, 2005, 2006, 1994, 2015, 2019, 2016, 1982, 1989, 1990,
       1991, 1999, 1986, 1992, 1984, 1980, 1961, 2000, 1995, 1985, 1976,
       1959, 1988, 1981, 1972, 1964, 1945, 1954, 1979, 1958, 1956, 1963,
       1970, 1973, 1925, 1974, 1960, 1966, 1971, 1962, 1969, 1977, 1967,
       1968, 1965, 1946, 1942, 1955, 1944, 1947, 1943], dtype=int64))
```

In [16]:

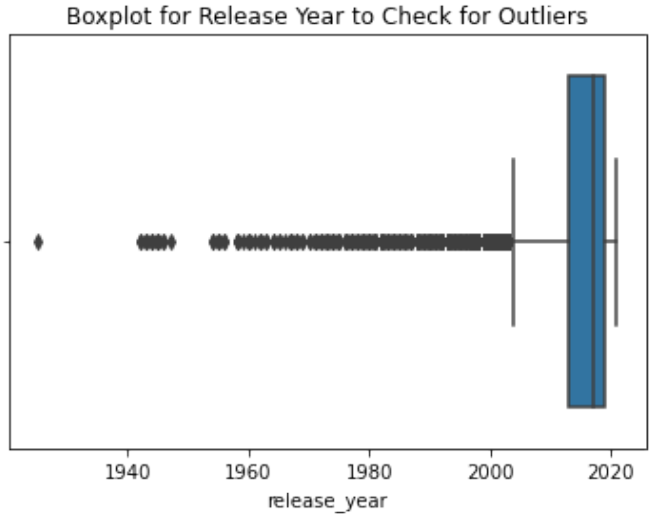
```
plt.figure(figsize=(10, 4))
```

Out[16]:

```
<Figure size 720x288 with 0 Axes>
```

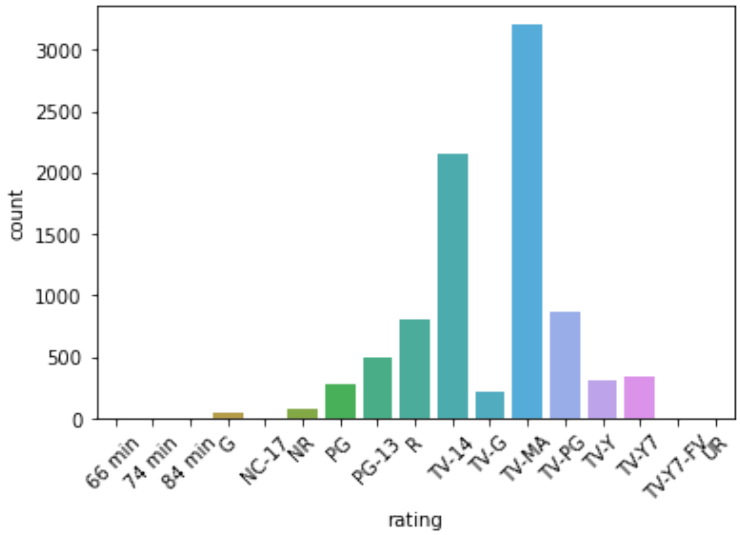
<Figure size 720x288 with 0 Axes>

In [17]:
```
sns.boxplot(x=df['release_year'])
plt.title('Boxplot for Release Year to Check for Outliers')
plt.show()
```



In [18]:
```
sns.countplot(data=df,x='rating')
plt.xticks(rotation=45)
```

Out[18]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16]),
 [Text(0, 0, '66 min'),
  Text(1, 0, '74 min'),
  Text(2, 0, '84 min'),
  Text(3, 0, 'G'),
  Text(4, 0, 'NC-17'),
  Text(5, 0, 'NR'),
  Text(6, 0, 'PG'),
  Text(7, 0, 'PG-13'),
  Text(8, 0, 'R'),
  Text(9, 0, 'TV-14'),
  Text(10, 0, 'TV-G'),
  Text(11, 0, 'TV-MA'),
  Text(12, 0, 'TV-PG'),
  Text(13, 0, 'TV-Y'),
  Text(14, 0, 'TV-Y7'),
  Text(15, 0, 'TV-Y7-FV'),
  Text(16, 0, 'UR')])
```

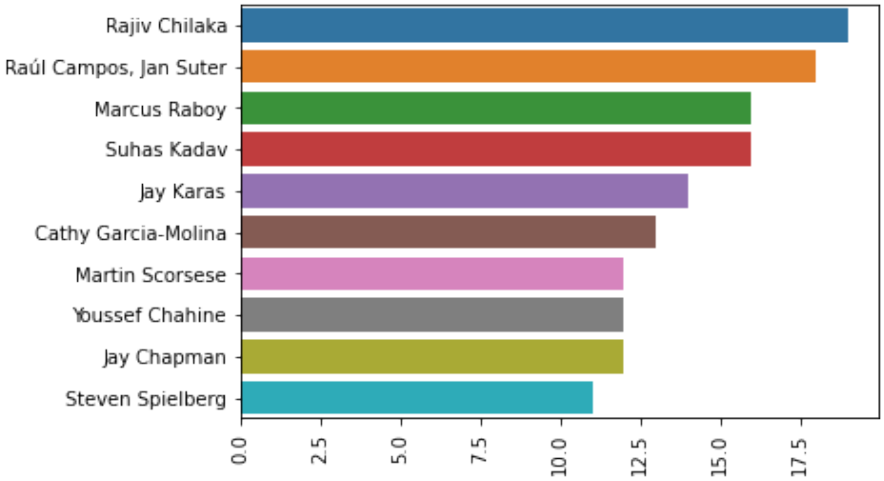


In [19]:
```
top_directors=df['director'].value_counts().head(10)
sns.barplot(y=top_directors.index,x=top_directors.values)
```

```
plt.xticks(rotation=90)
```

Out[19]:

```
(array([ 0. ,  2.5,  5. ,  7.5, 10. , 12.5, 15. , 17.5, 20. ]),
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```

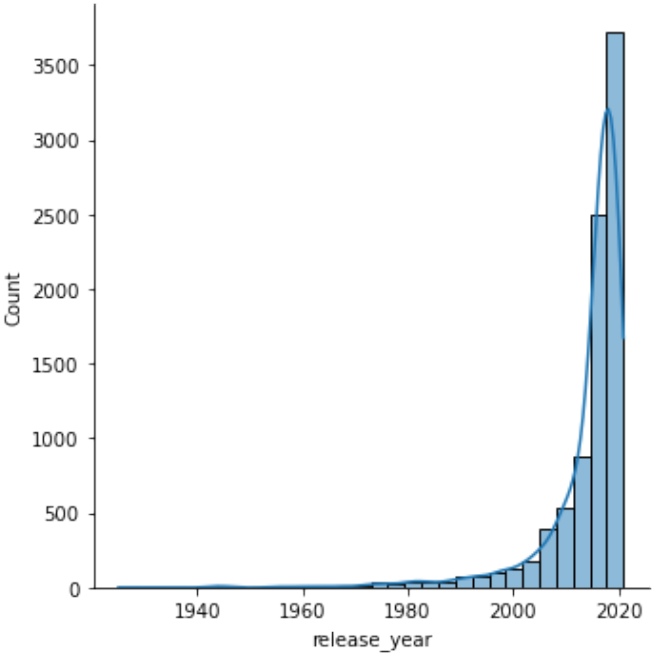


In [20]:

```
sns.displot(data=df,x='release_year',kde=True,bins=30)
```

Out[20]:

```
<seaborn.axisgrid.FacetGrid at 0x28eebcd82e0>
```

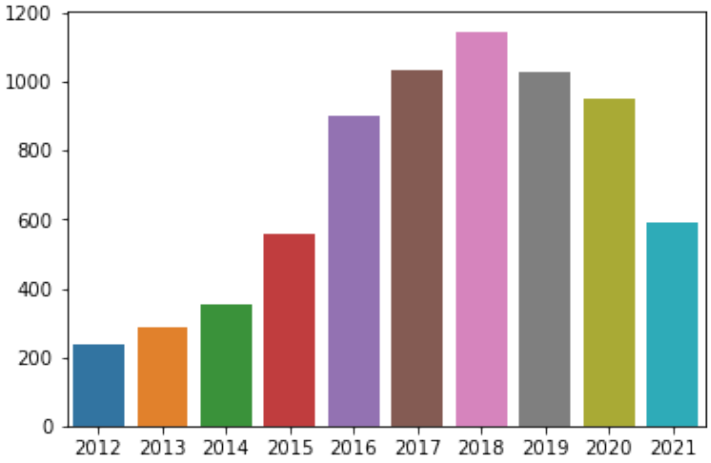


Countplot for Top 10 Most Frequent Release Years

In [28]:

```
top_release_year=df['release_year'].value_counts().head(10)
sns.barplot(x=top_release_year.index,y=top_release_year.values)
# The top 10 most frequent release years are all from the recent past, with the year 2018
```

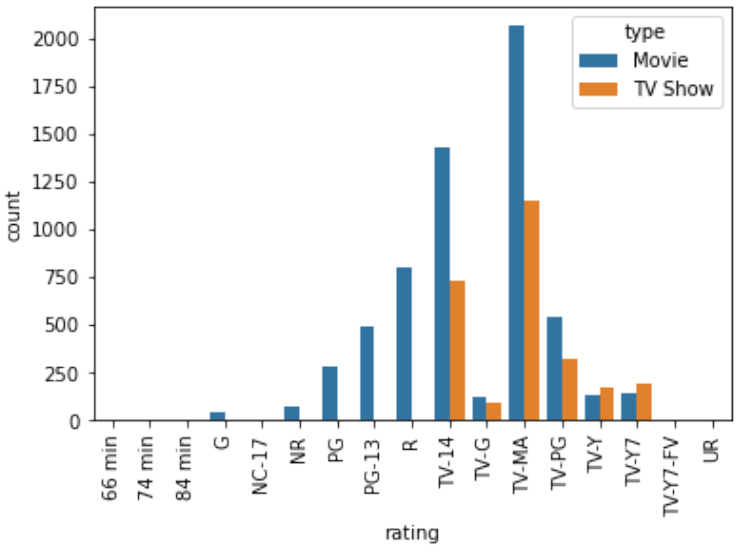
Out[28]: <AxesSubplot:>



Relationship Between Type and Rating

In [35]:
```
sns.countplot(data=df,x='rating',hue='type')
plt.xticks(rotation=90)
```

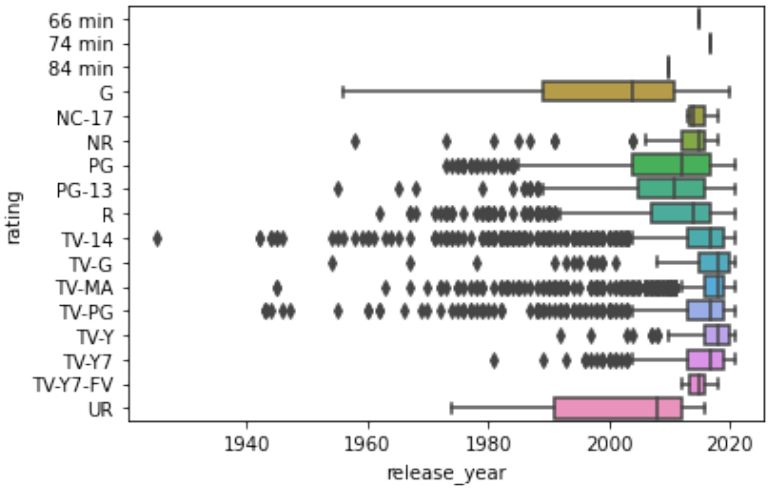
Out[35]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16]),
 [Text(0, 0, '66 min'),
  Text(1, 0, '74 min'),
  Text(2, 0, '84 min'),
  Text(3, 0, 'G'),
  Text(4, 0, 'NC-17'),
  Text(5, 0, 'NR'),
  Text(6, 0, 'PG'),
  Text(7, 0, 'PG-13'),
  Text(8, 0, 'R'),
  Text(9, 0, 'TV-14'),
  Text(10, 0, 'TV-G'),
  Text(11, 0, 'TV-MA'),
  Text(12, 0, 'TV-PG'),
  Text(13, 0, 'TV-Y'),
  Text(14, 0, 'TV-Y7'),
  Text(15, 0, 'TV-Y7-FV'),
  Text(16, 0, 'UR')])
```



Relationship between Rating and Release Year

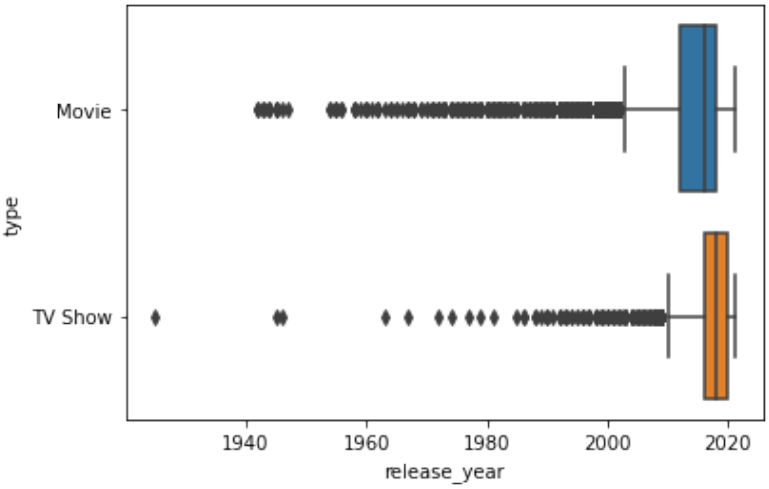
In [41]:
```
sns.boxplot(data=df,x='release_year',y='rating')
#The boxplot shows that the median release year for most ratings is relatively recent.
# Content with ratings “TV-Y” and “TV-Y7” tends to be older compared to other ratings.
```

Out[41]:
```
<AxesSubplot:xlabel='release_year', ylabel='rating'>
```



In [42]:
```
sns.boxplot(data=df,x='release_year',y='type')
```

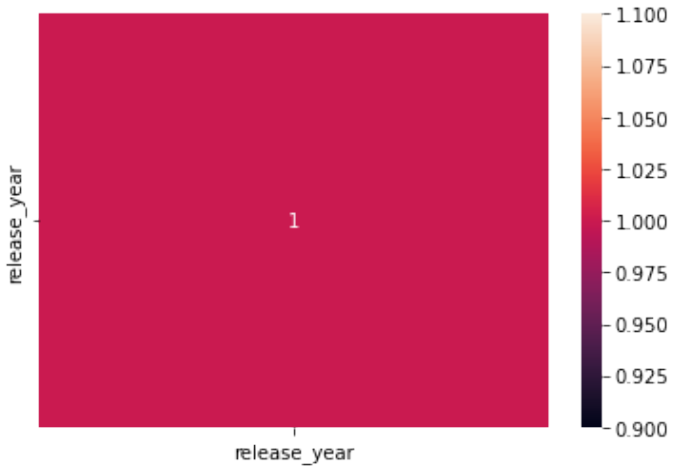
Out[42]:
```
<AxesSubplot:xlabel='release_year', ylabel='type'>
```



Correlation Analysis: Heatmaps and Pairplots

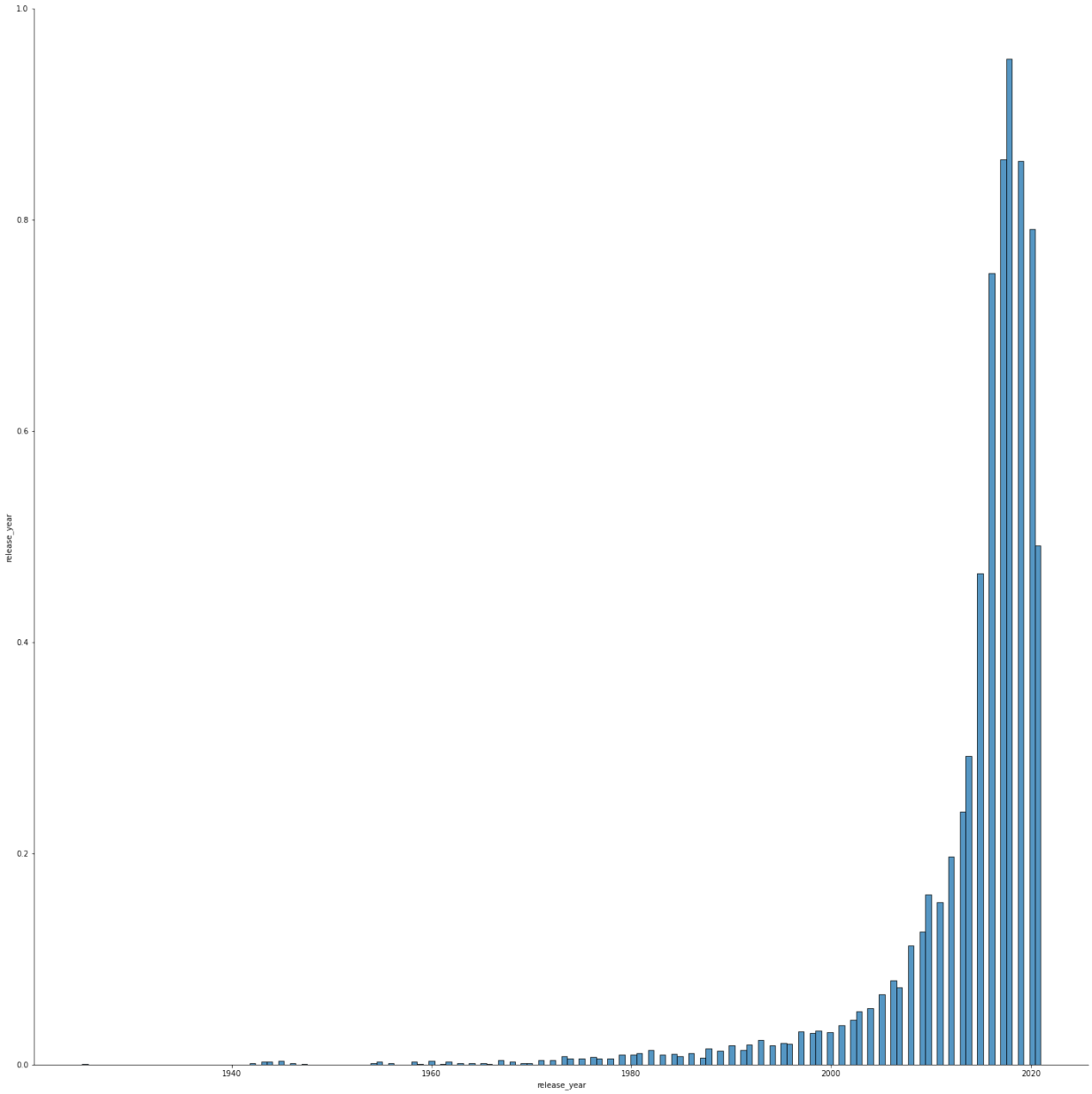
In [46]:
```
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True)
```

Out[46]:
```
<AxesSubplot:>
```



In [50]:
```
sns.pairplot(data=df,x_vars='release_year',height=20)
```

Out[50]:
```
<seaborn.axisgrid.PairGrid at 0x28eefa95640>
```



In [ ]:
```

```