Netflix:

Netflix is an American subscription video-on-demand over-the-top streaming service. The service primarily distributes original and acquired films and television shows from various genres, and it is available internationally in multiple languages. Launched on January 16, 2007, nearly a decade after Netflix, Inc. began its DVD-by-mail service, Netflix is the most-subscribed video-on-demand streaming media service, with 238.39 million paid memberships in more than 190 countries. By 2022, original productions accounted for half of its library in the United States and the namesake company had ventured into other categories, such as video game publishing via its eponymous service. company headquartered in Los Gatos, California. Netflix was founded in 1997 by Reed Hastings and Marc Randolph in Scotts Valley, California. The company's primary business is its subscript on-based streaming service, which offers online streaming of a library of films and television series, including those produced in-house.

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

```
df=pd.read_csv('netflix.csv')
```

```
df.head(5)
```

	show_id	type	title	director	cast	country	date_added	release_year	rat
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy	NaN	September 24, 2021	2021	TV-

```
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
```

```
df.describe()
```

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
max	2021.000000

```
df.shape
```

```
(8807, 12)
```

There are 88807 rows and 12 columns in the netflix file

```
df.isnull().sum()
```

```
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
```

```
df.isnull().sum()/len(df)*100
```

```
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
```

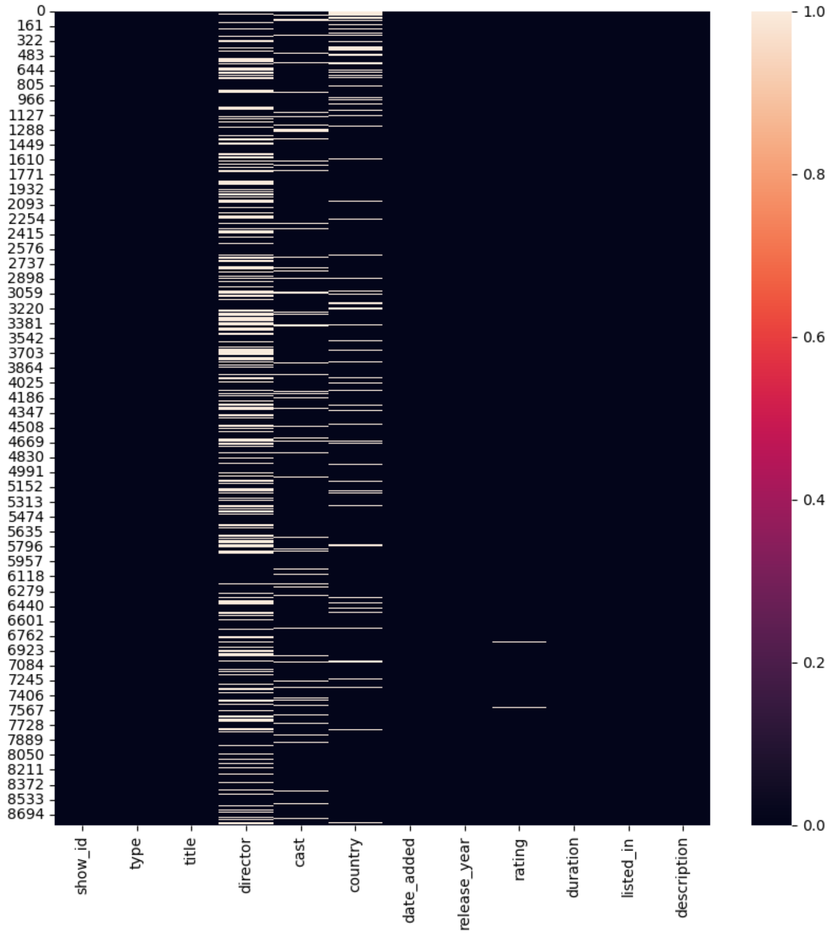
```
df.isnull().sum().plot(kind='bar')
```

```
<Axes: >
```

As we can see there are more null values in director column, followed by cast and country columns. There are a few null values in date_added, rating, duration.

```
plt.figure(figsize=[10,10])
sns.heatmap(df.isnull())
```

```
<Axes: >
```



The same can be seen in the heatmap.

```
df['Date']=pd.to_datetime(df['date_added'])#converting date_added to datetime type to do the anaylsis
df.info()
```

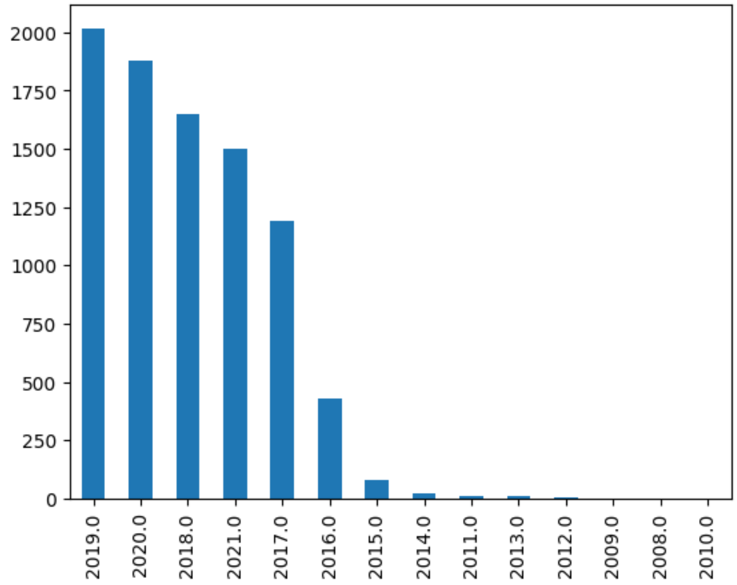
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 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
 12  Date          8797 non-null   datetime64[ns]
dtypes: datetime64[ns](1), int64(1), object(11)
memory usage: 894.6+ KB
```

```
df['Date'].dt.year.value_counts().astype('int64')
```

```
2019.0    2016
2020.0    1879
2018.0    1649
2021.0    1498
2017.0    1188
2016.0     429
2015.0      82
2014.0      24
2011.0      13
2013.0      11
2012.0       3
2009.0       2
2008.0       2
2010.0       1
Name: Date, dtype: int64
```

```
df['Date'].dt.year.value_counts().plot(kind='bar')
```

```
<Axes: >
```



1. The highest number of contents were added to Netflix in the year 2019.
2. More contents are added from 2016-2021 when caompared to 2008_2015.

```
df['Date'].dt.year.aggregate(['max', 'min'])
```

```
max    2021.0
min    2008.0
Name: Date, dtype: float64
```

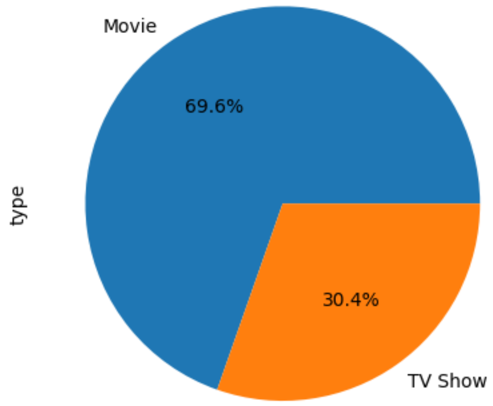
We have the data for movies and tv shows added to netflix from the year 2008 to the year 2021

```
df['type'].value_counts()
```

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

```
df['type'].value_counts().plot(kind='pie', autopct='%1.1f%%')
```

```
<Axes: ylabel='type'>
```



There are more movies than tv shows.

Steps to be performed

1. fill nulls
2. unnest director, cast, country, listed_in.
3. set duration, date

```
data = df
```

1.Filling nulls with unk(unknown) Or -1

```
data['director'].fillna('unk_dir', inplace = True)
data['cast'].fillna('unk_cast', inplace = True)
data['country'].fillna('unk_country', inplace = True)
data['rating'].fillna('unk_rating', inplace = True)
data['duration'].fillna('-1 min', inplace = True)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 13 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    8797 non-null   object
 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
 11  description   8807 non-null   object
 12  Date          8797 non-null   datetime64[ns]
dtypes: datetime64[ns](1), int64(1), object(11)
memory usage: 894.6+ KB
```

The movies and tv shows released from the year 1925 to 2021 are there in the data given.

```
data.describe(include='object')
```

	show_id	type	title	director	cast	country	date_added	rating	du
count	8807	8807	8807	8807	8807	8807	8797	8807	
unique	8807	2	8807	4529	7693	749	1767	18	
top	s1	Movie	Dick Johnson	unk_dir	unk_cast	United States	January 1, 2020	TV-MA	1

```
data['type'].value_counts(normalize=True)*100
```

```
Movie      69.615079
TV Show    30.384921
Name: type, dtype: float64
```

```
data.head()
```

	show_id	type	title	director	cast	country	date_added	release_
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	unk_cast	United States	September 25, 2021	:
1	s2	TV Show	Blood & Water	unk_dir	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	:
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	unk_country	September 24, 2021	:

Unnesting columns

Unnesting director

```
constraint1=data['director'].apply(lambda x: str(x).split(', ')).tolist()
```

```
data_director=pd.DataFrame(constraint1,index=data['show_id'])
```

```
data_director=data_director.stack()
```

```
data_director=pd.DataFrame(data_director)
```

```
data_director.reset_index(inplace=True)
```

```
data_director=data_director[['show_id',0]]
```

```
data_director.columns=['show_id','director1']
data_director.head()
```

	show_id	director1
0	s1	Kirsten Johnson
1	s2	unk_dir
2	s3	Julien Leclercq
3	s4	unk_dir
4	s5	unk_dir

Unnesting cast

```
constraint2=data['cast'].apply(lambda x: str(x).split(', ')).tolist()
```

```
data_cast=pd.DataFrame(constraint2,index=data['show_id'])
```

```
data_cast=data_cast.stack()
```

```
data_cast=pd.DataFrame(data_cast)
```

```
data_cast.reset_index(inplace=True)
```

```
data_cast=data_cast[['show_id',0]]
```

```
data_cast.columns=['show_id','cast1']
```

```
data_cast.head()
```

	show_id	cast1
0	s1	unk_cast
1	s2	Ama Qamata
2	s2	Khosi Ngema
3	s2	Gail Mabalane
4	s2	Thabang Molaba

Unnesting country

```
constraint3=data['country'].apply(lambda x: str(x).split(', ')).tolist()
```

```
data_country=pd.DataFrame(constraint3,index=data['show_id'])
```

```
data_country=data_country.stack()
```

```
data_country=pd.DataFrame(data_country)
```

```
data_country.reset_index(inplace=True)
```

```
data_country=data_country[['show_id',0]]
```

```
data_country.columns=['show_id','country1']
```

```
data_country.head()
```

	show_id	country1
0	s1	United States
1	s2	South Africa
2	s3	unk_country
4	s5	India

Unnesting listed_in

```
constraint4=data['listed_in'].apply(lambda x: str(x).split(', ')).tolist()

data_listed_in=pd.DataFrame(constraint4,index=data['show_id'])

data_listed_in=data_listed_in.stack()

data_listed_in=pd.DataFrame(data_listed_in)

data_listed_in.reset_index(inplace=True)

data_listed_in=data_listed_in[['show_id',0]]

data_listed_in.columns=['show_id','listed_in1']

data_listed_in.head()
```

	show_id	listed_in1
0	s1	Documentaries
1	s2	International TV Shows
2	s2	TV Dramas
3	s2	TV Mysteries
4	s3	Crime TV Shows

Setting Duration into numericals

```
constraint5=data['duration'].apply(lambda x: str(x).split(' ')).tolist()

duration1 = [i[0] for i in constraint5]

data_duration=pd.DataFrame(duration1,index=data['show_id'])

data_duration.reset_index(inplace=True)

data_duration=data_duration[['show_id',0]]

data_duration.columns=['show_id','duration1']

data_duration.head()
```

	show_id	duration1
0	s1	90
1	s2	2
2	s3	1
3	s4	1
4	s5	2

Merging all the above columns with complete data

```
merge1 = data_director.merge(data_cast.merge(data_country.merge(data_listed_in.merge(data_duration, on = 'show_id'), on = 'show_id'), on='sh
```

```
data = data.merge(merge1, on = 'show_id')
```

```
data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	unk_cast	United States	September 25, 2021	2020
1	s2	TV Show	Blood & Water	unk_dir	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
2	s2	TV Show	Blood & Water	unk_dir	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
3	s2	TV Show	Blood & Water	unk_dir	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
4	s2	TV Show	Blood & Water	unk_dir	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021

Setting date column

```
data['date_add']=pd.to_datetime(data['date_added'], errors='coerce')
```

```
data['month']=data['date_add'].dt.month.fillna(-1)
```

```
data['month']=data['month'].astype('int64')
```

```
data['year']=data['date_add'].dt.year.fillna(-1)
```

```
data['year']=data['year'].astype('int64')
```

```
day_name = data['date_add'].dt.day_name()
day_name
```

```
0         Saturday
1           Friday
2           Friday
3           Friday
4           Friday
            ...
201986    Saturday
201987    Saturday
```

```
201988    Saturday
201989    Saturday
201990    Saturday
Name: date_add, Length: 201991, dtype: object
```

```
data['day_name']=data['date_add'].dt.day_name().fillna('unk_day')
```

```
data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	unk_cast	United States	September 25, 2021	2020
1	s2	TV Show	Blood & Water	unk_dir	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
2	s2	TV Show	Blood & Water	unk_dir	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
3	s2	TV Show	Blood & Water	unk_dir	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
4	s2	TV Show	Blood & Water	unk_dir	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021

5 rows × 22 columns

Removing original director, cast, country, date_added, listed_in and duration

```
df_1 = data
```

```
df_1.drop(columns= ['director','cast','country','date_added','duration','listed_in', 'Date'], inplace = True)
```

```
df_1.head(5)
```

show_id	type	title	release_year	rating	description	director1	cast1

```
df_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201991 entries, 0 to 201990
Data columns (total 15 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   release_year  201991 non-null  int64         
 4   rating        201991 non-null  object        
 5   description   201991 non-null  object        
 6   director1     201991 non-null  object        
 7   cast1         201991 non-null  object        
 8   country1      201991 non-null  object        
 9   listed_in1    201991 non-null  object        
 10  duration1     201991 non-null  object        
 11  date_add      201833 non-null  datetime64[ns]
 12  month         201991 non-null  int64         
 13  year          201991 non-null  int64         
 14  day_name      201991 non-null  object        
dtypes: datetime64[ns](1), int64(3), object(11)
memory usage: 24.7+ MB
```

Converting duration1 column to int64 type.

```
df_1['duration']=df_1['duration1'].astype('int64')
```

```
df_1.drop(columns= ['duration1'], inplace = True)#dropping duration1 column
```

```
df_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201991 entries, 0 to 201990
Data columns (total 15 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   release_year  201991 non-null  int64         
 4   rating        201991 non-null  object        
 5   description   201991 non-null  object        
 6   director1     201991 non-null  object        
 7   cast1         201991 non-null  object        
 8   country1      201991 non-null  object        
 9   listed_in1    201991 non-null  object        
 10  date_add      201833 non-null  datetime64[ns]
 11  month         201991 non-null  int64         
 12  year          201991 non-null  int64         
 13  day_name      201991 non-null  object        
 14  duration      201991 non-null  int64         
dtypes: datetime64[ns](1), int64(4), object(10)
memory usage: 24.7+ MB
```

```
movie_dur=df_1[df_1['type']=='Movie'] #data of the movie duration
movie_dur.head()
```

	show_id	type	title	release_year	rating	description	director1	ca
0	s1	Movie	Dick Johnson Is Dead	2020	PG-13	As her father nears the end of his life, filmm...	Kirsten Johnson	unk_(
159	s7	Movie	My Little Pony: A New Generation	2021	PG	Equestria's divided. But a bright-eyed hero be...	Robert Cullen	Vane Hudg
	s7	Movie	My Little New	2021	PG	Equestria's a bright-eyed	Cullen	Mars

```
movie_dur[ movie_dur['duration']!=-1]['duration'].aggregate(['max', 'min'])
```

```
max    312
min      3
Name: duration, dtype: int64
```

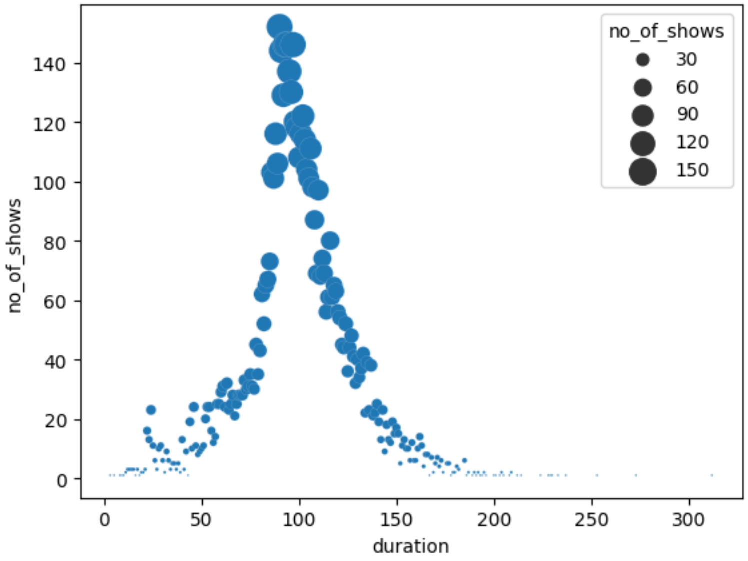
```
mv_dur=movie_dur[movie_dur['duration']!=-1].groupby('duration')['show_id'].nunique().reset_index()
mv_dur.columns=['duration','no_of_shows']
mv_dur
```

	duration	no_of_shows
0	3	1
1	5	1
2	8	1
3	9	1
4	10	1
...	...	...
200	233	1
201	237	1
202	253	1
203	273	1
204	312	1

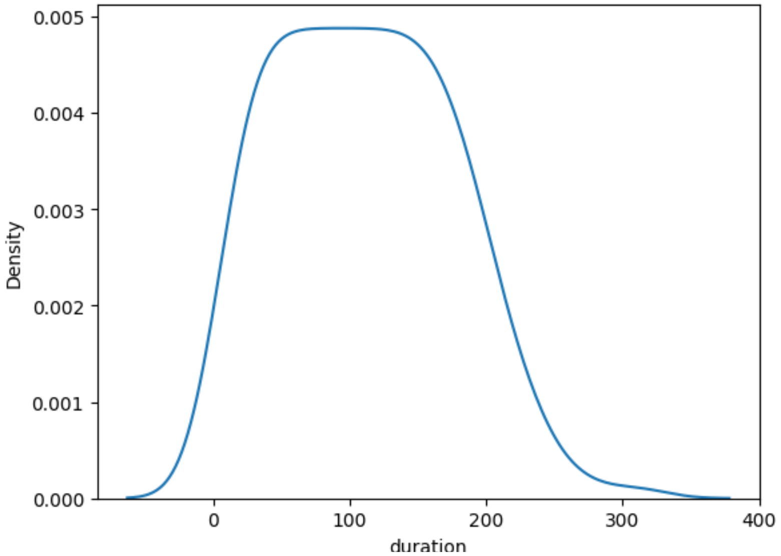
205 rows × 2 columns

```
sns.scatterplot(data = mv_dur, x="duration", y='no_of_shows', size=('no_of_shows'), sizes=(1,200))
```

```
<Axes: xlabel='duration', ylabel='no_of_shows'>
```



```
sns.kdeplot(data = mv_dur['duration'])
```



- The maximum duration of movies are 312 minutes and the minimum duration is 3 minutes and these are the outliers
- More movies are in the range of around 50 minutes to 160 minutes

```
tv=df_1[df_1['type']=='TV Show']
tv.head()
```

	show_id	type	title	release_year	rating	description	director1	cast1	cou
1	s2	TV Show	Blood & Water	2021	TV-MA	After crossing paths at a party, a Cape Town t...	unk_dir	Ama Qamata	
2	s2	TV Show	Blood & Water	2021	TV-MA	After crossing paths at a party, a Cape Town t...	unk_dir	Ama Qamata	
			Blood			After crossing			

```
tv[ tv['duration']!=-1]['duration'].aggregate(['max', 'min'])
```

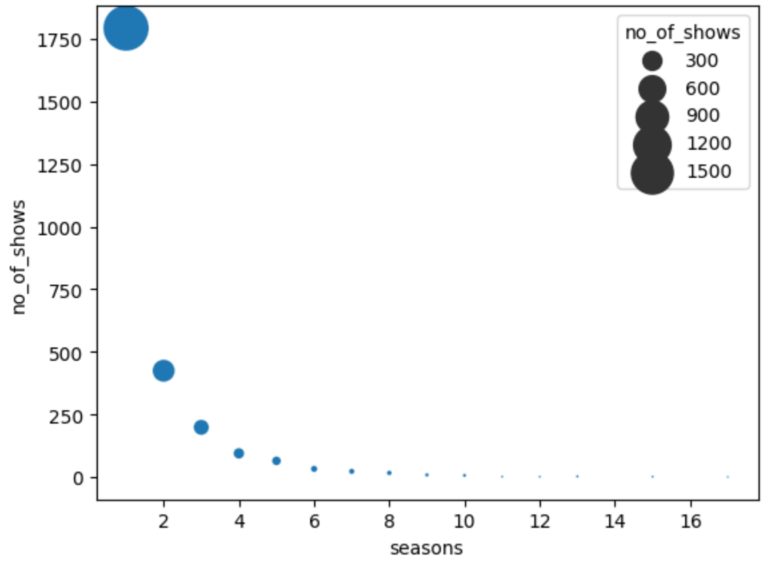
```
max    17
min     1
Name: duration, dtype: int64
```

```
TVshow_dur=df_1[(df_1['type']=='TV Show') & (df_1['duration']!=-1) ].groupby(['duration'])['show_id'].nunique().sort_values(ascending=False
TVshow_dur.columns=['seasons','no_of_shows']
TVshow_dur
```

	seasons	no_of_shows
0	1	1793
1	2	425
2	3	199

```
sns.scatterplot(data = TVshow_dur, x="seasons", y='no_of_shows', size=('no_of_shows'), sizes=(1,600))
```

```
<Axes: xlabel='seasons', ylabel='no_of_shows'>
```



- There are more tv shows which have only one season.
- There is only on tv_show with 17 seasons. This is an outlier.

Analysis on number of directors, cast, countries, and release year.

```
df_1['director1'].nunique()
```

```
4994
```

```
df_1['cast1'].nunique()
```

```
36440
```

```
df_1['country1'].nunique()
```

```
128
```

```
df_1['release_year'].nunique()
```

```
74
```

- There are 4994 unique directors, 36440 actors in the data given.
- The shows release in 128 countries are present in this given data.
- There are shows released in 74 years.

```
rating=df_1.groupby('rating')['show_id'].nunique().sort_values(ascending=False).reset_index()
rating.columns=['rating', 'no_of_shows']
rating
```

	rating	no_of_shows
0	TV-MA	3207
1	TV-14	2160
2	TV-PG	863
3	R	799
4	PG-13	490
5	TV-Y7	334
6	TV-Y	307
7	PG	287
8	TV-G	220
9	NR	80
10	G	41
11	TV-Y7-FV	6
12	unk_rating	4
13	NC-17	3
14	UR	3

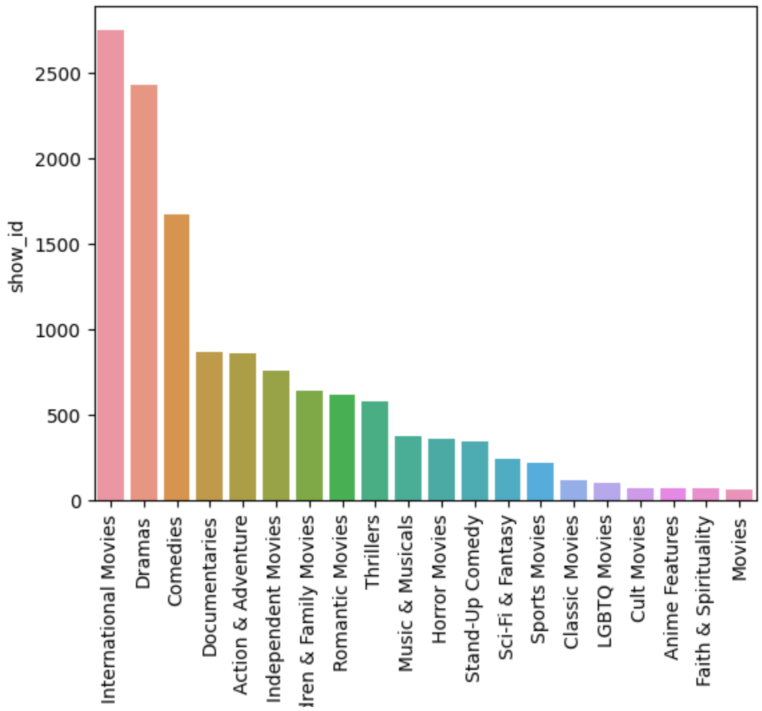
Which genre shows are released more on Netflix?

```
genreM=df_1[df_1['type']=='Movie'].groupby(['listed_in1'])['show_id'].nunique().sort_values(ascending=False).reset_index()
genreM
```

	listed_in1	show_id
0	International Movies	2752
1	Dramas	2427
2	Comedies	1674
3	Documentaries	869
4	Action & Adventure	859
5	Independent Movies	756
6	Children & Family Movies	641
7	Romantic Movies	616
8	Thrillers	577
9	Music & Musicals	375
10	Horror Movies	357
11	Stand-Up Comedy	343
12	Sci-Fi & Fantasy	243
13	Sports Movies	219
14	Classic Movies	116
15	LGBTQ Movies	102
16	Cult Movies	71
17	Anime Features	71
18	Faith & Spirituality	65
19	Movies	57

```
sns.barplot(x='listed_in1', y='show_id', data=genreM)
plt.xticks(rotation=90)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19]),
 [Text(0, 0, 'International Movies'),
  Text(1, 0, 'Dramas'),
  Text(2, 0, 'Comedies'),
  Text(3, 0, 'Documentaries'),
  Text(4, 0, 'Action & Adventure'),
  Text(5, 0, 'Independent Movies'),
  Text(6, 0, 'Children & Family Movies'),
  Text(7, 0, 'Romantic Movies'),
  Text(8, 0, 'Thrillers'),
  Text(9, 0, 'Music & Musicals'),
  Text(10, 0, 'Horror Movies'),
  Text(11, 0, 'Stand-Up Comedy'),
  Text(12, 0, 'Sci-Fi & Fantasy'),
  Text(13, 0, 'Sports Movies'),
  Text(14, 0, 'Classic Movies'),
  Text(15, 0, 'LGBTQ Movies'),
  Text(16, 0, 'Cult Movies'),
  Text(17, 0, 'Anime Features'),
  Text(18, 0, 'Faith & Spirituality'),
  Text(19, 0, 'Movies')])
```



- There are more number of International Movies added on Netflix followed by Dramas, Comedies
- There are very few movies belonging to genre Cult, Anime, Faith & Spirituality

```
genreTV=df_1[df_1['type']=='TV Show'].groupby(['listed_in1'])['show_id'].nunique().sort_values(ascending=False).reset_index()
genreTV.columns=['listed_in1','no_of_tvshows']
genreTV
```

	listed_in1	no_of_tvshows
0	International TV Shows	1351
1	TV Dramas	763
2	TV Comedies	581
3	Crime TV Shows	470
4	Kids' TV	451
5	Docuseries	395
6	Romantic TV Shows	370
7	Reality TV	255
8	British TV Shows	253
9	Anime Series	176

```
sns.barplot(x='listed_in1', y='no_of_tvshows', data=genreTV)
plt.xticks(rotation=90)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21]),
```

- There are more number of International TV Shows added on Netflix followed by TV Dramas
- There are very few TV Shows belonging to genre Stand-Up Comedy & Talk Shows and Classic & Cult

```
Text(4, 0, "Kids' TV")
```

The number of movies produced in each country and pick the top 10 countries

```
Text(3, 0, 'British TV Shows'),
```

```
MPC=df_1[df_1['type']=='Movie'].groupby('country1')['show_id'].nunique().sort_values(ascending=False)
MPC=MPC.reset_index()
MPC    #Netfilx has released movies in approximately 123 countries.
```

	country1	show_id
0	United States	2751
1	India	962
2	United Kingdom	532
3	unk_country	440
4	Canada	319
...	...	...
118	Nicaragua	1
119	Palestine	1
120	Panama	1
121	Paraguay	1
122	Malawi	1

123 rows × 2 columns

```
MPC[MPC['country1']!='unk_country'].head(10)
 #top 10 countries which produce the more number of movies.
 #according to the data US has produced the most number of movies.
```

	country1	show_id
0	United States	2751
1	India	962
2	United Kingdom	532
4	Canada	319
5	France	303
6	Germany	182
7	Spain	171
8	Japan	119
9	China	114
10	Mexico	111

- The number of movies added to Netflix from the USA is the highest and it is followed by India and UK
- The number of movies added to Netflix from the Mexico is the least and it is followed by China and Japan
- More number of movies from the region Mexico, Japan and China can be added to attract viewers from that region.

The number of TV Shows produced in each country and pick the top 10 countries

```
TPC=df_1[df_1['type']=='TV Show'].groupby('country1')['show_id'].nunique().sort_values(ascending=False)
TPC=TPC.reset_index()
TPC
```

	country1	show_id
0	United States	938
1	unk_country	391
2	United Kingdom	272
3	Japan	199
4	South Korea	170
...	...	...
62	Mauritius	1
63	Senegal	1
64	Puerto Rico	1
65	Hungary	1
66		1

67 rows × 2 columns

```
TPC[TPC['country1']!='unk_country'].head(10)
```

	country1	show_id
0	United States	938
2	United Kingdom	272
3	Japan	199
4	South Korea	170
5	Canada	126
6	France	90
7	India	84
8	Taiwan	70
9	Australia	66
10	Spain	61

Double-click (or enter) to edit

- The number of TV Shows added to Netflix from the USA is the highest and it is followed by UK and Japan
- The number of tv shows added to Netflix from Spain is the least and it is followed by Australia, Taiwan and India.
- More number of TV Shows from the countries Spain, Australia, Taiwan and India can be added to attract viewers from that region.

The best time to launch a TV show?

the best week to release the Tv-show or the movie

```
BWM=df_1[df_1['type']=='Movie'].groupby('day_name')['show_id'].nunique().sort_values(ascending=False).reset_index()
BWM.columns=['day_name', 'no_of_shows']
BWM
```

	day_name	no_of_shows

```
BWM.head(1)
```

	day_name	no_of_shows
0	Friday	1566

```
BWT=df_1[df_1['type']=='TV Show'].groupby('day_name')['show_id'].nunique().sort_values(ascending=False).reset_index()
BWT.columns=['day_name', 'no_of_shows']
BWT
```

	day_name	no_of_shows
0	Friday	932
1	Wednesday	382
2	Tuesday	345
3	Thursday	343
4	Saturday	259
5	Monday	223
6	Sunday	182
7	unk_day	10

```
BWT.head(1)
```

	day_name	no_of_shows
0	Friday	932

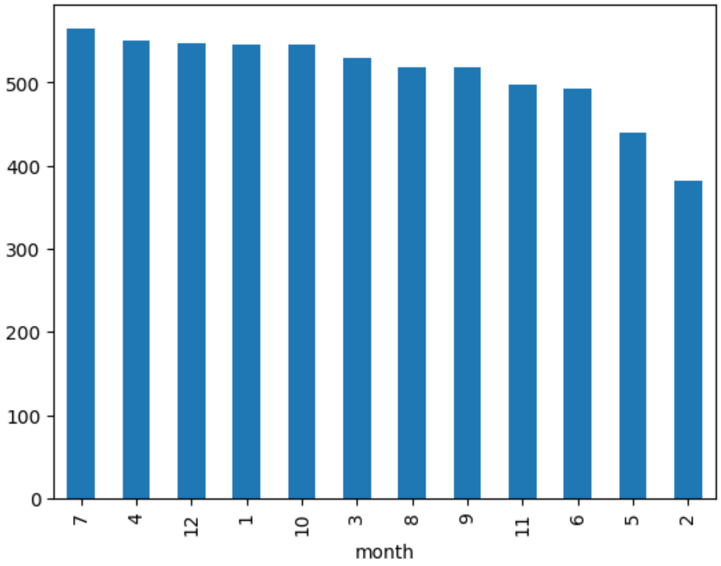
- There are two things which we can notice. Considering the weekends, more movies are added on Friday and Thursday. That might be good as more viewers are free then, there is high chance of them watching the movies
- The other thing is since a very few movies are released on Saturday and Sunday, releasing more movies on these days could also increase the chances of viewers watching the movies
- More number of TV Shows are added on Friday and very few are added on Sunday.
- More number of TV shows can be added on Sunday because there are few shows added, the chances of watching the show added on that day will be high.

The best month to release the TV-show or the movie

```
BMM=df_1[df_1['type']=='Movie'].groupby('month')['show_id'].nunique().sort_values(ascending=False).reset_index()
BMM.columns=['month', 'no_of_shows']
BMM
```

	month	no_of_shows
0	7	565

```
BMMG=df_1[df_1['type']=='Movie'].groupby('month')['show_id'].nunique().sort_values(ascending=False).plot(kind='bar')
```



- There are very few movies added in February so adding more movies in that month increase the chance of watching the movies added on that month.

```
BMM.head(1)
```

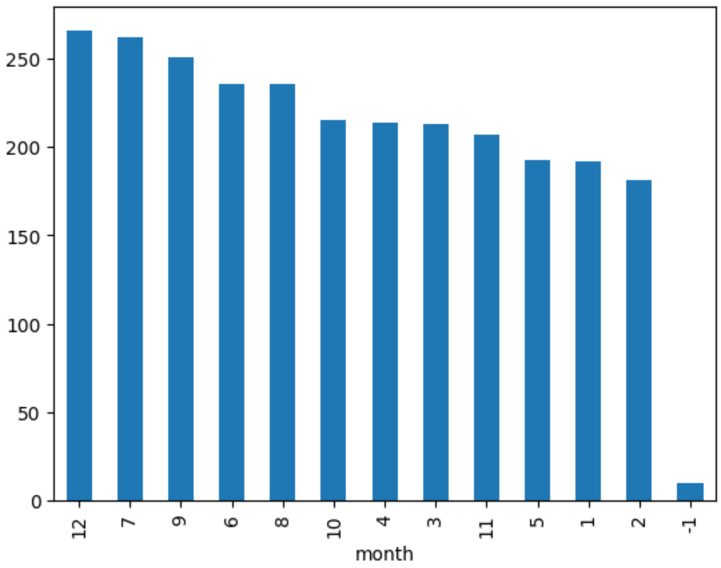
	month	no_of_shows
0	7	565

The highest number of movies are added in the month of July.

```
BMT=df_1[df_1['type']=='TV Show'].groupby('month')['show_id'].nunique().sort_values(ascending=False).reset_index()
BMT.columns=['month', 'no_of_shows']
BMT
```

	month	no_of_shows
0	12	266
1	7	262
2	9	251
3	6	236
4	8	236
5	10	215
6	4	214
7	3	213
8	11	207
9	5	193
10	1	192
11	2	181
12	-1	10

```
BMTG=df_1[df_1['type']=='TV Show'].groupby('month')['show_id'].nunique().sort_values(ascending=False).plot(kind='bar')
```



- The highest number of Tv shows are added in December but we can also see that in July also almost the same number of TV shows are addded.
- Since a few TV shows are added in February, January and May, we can add more TV sows in these months as there is a high chances of viewers watching these shows

```
BMT.head(1)
```

	month	no_of_shows
0	12	266

The highest number of Tv shows are added in December

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

The top 10 directors who have appeared in most movies

```
BD=df_1.groupby('director1')['show_id'].nunique().sort_values(ascending=False)
BD=BD.reset_index()
BD.columns=['director1', 'no_of_shows']
BD
```

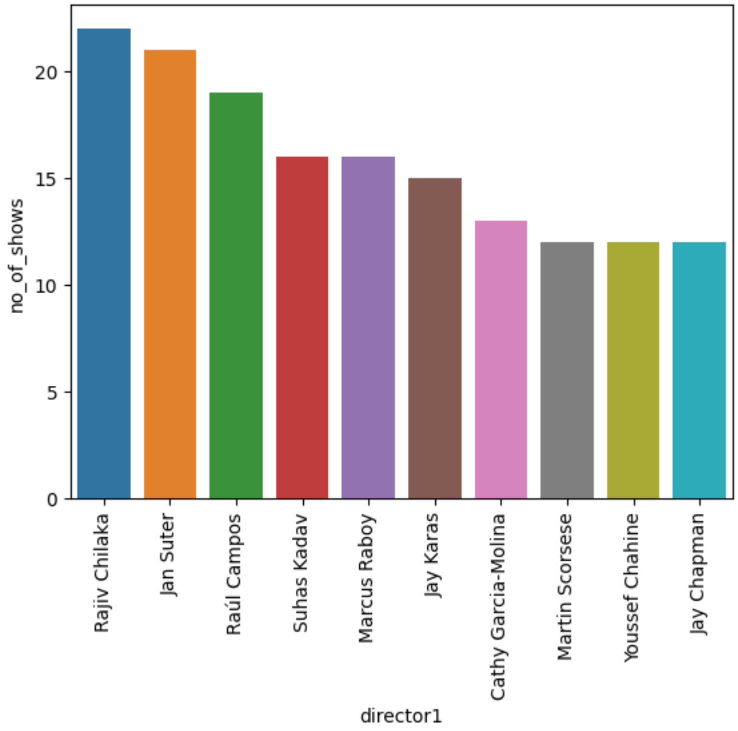
	director1	no_of_shows
0	unk_dir	2634
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Suhas Kadav	16
...	...	...
4989	Brandon Camp	1
4990	Juan Antin	1
4991	Juan Antonio de la Riva	1
4992	Juan Camilo Pinzon	1
4993	María Jose Cuevas	1

4994 rows × 2 columns

```
BD10=BD[BD['director1']!='unk_dir'].head(10)
```

```
sns.barplot(data=BD10, x='director1', y='no_of_shows')
plt.xticks(rotation=90)
```

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'Rajiv Chilaka'),
  Text(1, 0, 'Jan Suter'),
  Text(2, 0, 'Raúl Campos'),
  Text(3, 0, 'Suhas Kadav'),
  Text(4, 0, 'Marcus Raboy'),
  Text(5, 0, 'Jay Karas'),
  Text(6, 0, 'Cathy Garcia-Molina'),
  Text(7, 0, 'Martin Scorsese'),
  Text(8, 0, 'Youssef Chahine'),
  Text(9, 0, 'Jay Chapman')])
```



The shows directed by Rajiv Chilaka is the highest followed by Jan Suter.

```
BDM=df_1[df_1['type']=='Movie'].groupby('director1')['show_id'].nunique().sort_values(ascending=False).reset_index()
BDM.columns=['director', 'no_of_shows']
BDM
```

	director	no_of_shows
0	unk_dir	188
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Suhas Kadav	16
...	...	...
4773	José Ortuño	1
4774	Bob Persichetti	1
4775	Jovanka Vuckovic	1
4776	Bob Odenkirk	1
4777	Mary Lambert	1

4778 rows × 2 columns

```
BDM[BDM['director']!='unk_dir'].head(10)
```

	director	no_of_shows
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Suhas Kadav	16
5	Marcus Raboy	15
6	Jay Karas	15
7	Cathy Garcia-Molina	13
8	Youssef Chahine	12
9	Martin Scorsese	12
10	Jay Chapman	12

The movies directed by Rajiv Chilaka is the highest followed by Jan Suter on Netflix.

```
BDT=df_1[df_1['type']=='TV Show'].groupby('director1')['show_id'].nunique().sort_values(ascending=False).reset_index()
BDT.columns=['director', 'no_of_shows']
BDT
```

	director	no_of_shows
0	unk_dir	2446
1	Ken Burns	3
2	Alastair Fothergill	3
3	Jung-ah Im	2
4	Joe Berlinger	2
...	...	...
295	Houda Benyamina	1
296	Hong Won-ki	1
297	Hiroyuki Seshita	1
298	Hikaru Toda	1
299	Kim Seong-hun	1

300 rows × 2 columns

```
BDT[BDT['director']!='unk_dir'].head(10)
```

	director	no_of_shows
1	Ken Burns	3
2	Alastair Fothergill	3
3	Jung-ah Im	2
4	Joe Berlinger	2
5	Hsu Fu-chun	2
6	Stan Lathan	2
7	Gautham Vasudev Menon	2
8	Lynn Novick	2
9	Shin Won-ho	2
10	Iginio Straffi	2

Ken Burns and Alastair Fothergill have directed the highest number of TV Shows on netflix.

```
BA=df_1.groupby('cast1')['show_id'].nunique().sort_values(ascending=False).reset_index()
BA.columns=['cast', 'no_of_shows']
BA
```

	cast	no_of_shows
0	unk_cast	825
1	Anupam Kher	43
2	Shah Rukh Khan	35
3	Julie Tejwani	33
4	Naseeruddin Shah	32
...	...	...
36435	Jamie Lee	1
36436	Jamie Kenna	1
36437	Jamie Kaler	1
36438	Jamie Johnston	1
36439	Ṣọpẹ́ Dìrísù	1

36440 rows × 2 columns

```
BA[BA['cast']!='unk_cast'].head(10)
```

	cast	no_of_shows
1	Anupam Kher	43
2	Shah Rukh Khan	35
3	Julie Tejwani	33
4	Naseeruddin Shah	32
5	Takahiro Sakurai	32
6	Rupa Bhimani	31
7	Om Puri	30
8	Akshay Kumar	30
9	Yuki Kaji	29
10	Paresh Rawal	28

```
BAM=df_1[df_1['type']=='Movie'].groupby('cast1')['show_id'].nunique().sort_values(ascending=False).reset_index()
BAM.columns=['cast', 'no_of_shows']
BAM
```

	cast	no_of_shows
0	unk_cast	475
1	Anupam Kher	42
2	Shah Rukh Khan	35
3	Naseeruddin Shah	32
4	Akshay Kumar	30
...	...	...
25947	Jacob Blair	1
25948	Jacob Bertrand	1
25949	Jacob Batalon	1
25950	Jacob Artist	1
25951	Ṣọpẹ́ Dìrísù	1

25952 rows × 2 columns

```
BAM[BAM['cast']!='unk_cast'].head(10)
```

	cast	no_of_shows
1	Anupam Kher	42
2	Shah Rukh Khan	35
3	Naseeruddin Shah	32
4	Akshay Kumar	30
5	Om Puri	30
6	Amitabh Bachchan	28
7	Paresh Rawal	28
8	Julie Tejwani	28
9	Rupa Bhimani	27
10	Boman Irani	27

Anupam Kher has acted in most number of movies followed by Shah Rukh Khan.

```
BAT=df_1[df_1['type']=='TV Show'].groupby('cast1')['show_id'].nunique().sort_values(ascending=False).reset_index()
BAT.columns=['cast', 'no_of_shows']
BAT
```

	cast	no_of_shows
0	unk_cast	350
1	Takahiro Sakurai	25
2	Yuki Kaji	19
3	Daisuke Ono	17
4	Ai Kayano	17
...	...	...
14859	Ivy Yin	1
14860	Iván Pellicer	1
14861	Iván Álvarez de Araya	1
14862	Iza Moreira	1
14863	Şükrü Özyıldız	1

14864 rows × 2 columns

```
BAT[BAT['cast']!='unk_cast'].head(10)
```

	cast	no_of_shows
1	Takahiro Sakurai	25
2	Yuki Kaji	19
3	Daisuke Ono	17
4	Ai Kayano	17
5	Junichi Suwabe	17
6	Yuichi Nakamura	16
7	Yoshimasa Hosoya	15
8	Jun Fukuyama	15
9	David Attenborough	14
10	Vincent Tong	13

Takahiro Sakurai has acted in most number of TV Shows.

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

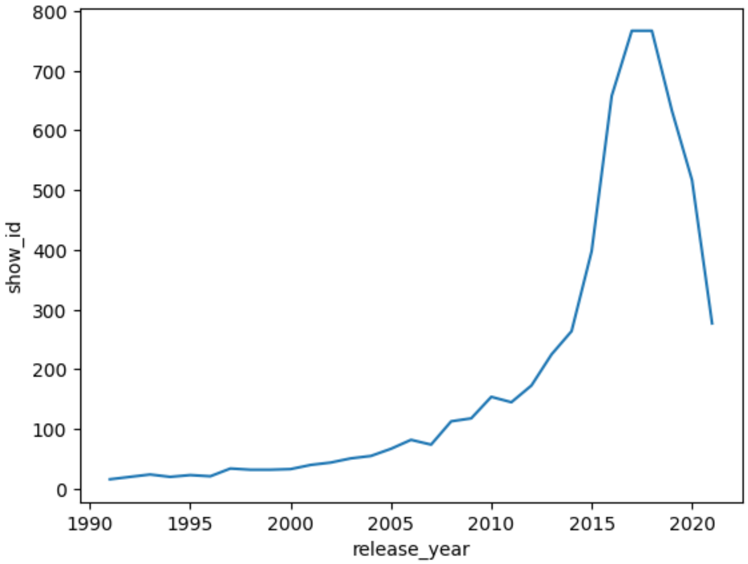
```
df_2=df_1[df_1['type']=='Movie'].groupby(['release_year'])['show_id'].nunique().sort_values()
df_2=df_2.reset_index()
df_2
```

```
      release_year  show_id
 0            1963        1
 1            1966        1
 2            1946        1
 3            1947        1
 4            1961        1
...            ...      ...
68            2020      517
69            2019      633
70            2016      658
71            2017      767
72            2018      767
73 rows × 2 columns
```

```
df_3=df_2[df_2['release_year']>=1991] #movies produced in last 30 years
```

```
sns.lineplot(data=df_3, x='release_year', y='show_id')
```

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



- There is an increase in the number of movies added on netflix over the past 20 t 30 years.
- There is a steep increase in the movies added from 2014 to 2018 and afterwards there is a decrease in the number of movies added.

Understanding what content is available in different countries

```
df_1['rating']=df_1['rating'].str.replace(' min','unk_rating')
df_1['rating'].nunique()
```

```
18
```

```
rating=df_1[df_1['rating']!='unk_rating'].groupby(['rating', 'country1'])['show_id'].nunique().sort_values(ascending=False).reset_index()
rating.columns=['rating','country1', 'no_of_shows']
rating
```

	rating	country1	no_of_shows
0	TV-MA	United States	1100
1	R	United States	660
2	TV-14	India	572
3	TV-14	United States	497
4	PG-13	United States	433
...	...	...	...
524	TV-G	Kenya	1
525	TV-G	Nigeria	1
526	TV-G	Peru	1
527	TV-G	South Africa	1
528	UR	United States	1

529 rows × 3 columns

- TV-MA contents are released more in the USA followed by rating of R in the USA
- UR contents are released very few in the US
- In India TV-14 contents are released more.

```
MVrating=df_1[(df_1['rating']!='unk_rating')& (df_1['type']!='Movie')].groupby(['rating', 'country1'])['show_id'].nunique().sort_values(asce
MVrating.columns=['rating','country1', 'no_of_shows']
MVrating#Analysis of ratings for movies
```

	rating	country1	no_of_shows
0	TV-MA	United States	382
1	TV-14	United States	221
2	TV-MA	unk_country	138
3	TV-PG	United States	124
4	TV-MA	United Kingdom	115
...	...	...	...
194	TV-MA	Malta	1
195	TV-MA	Malaysia	1
196	TV-MA	Luxembourg	1
197	TV-MA	Lebanon	1
198	TV-Y7-FV	Canada	1

199 rows × 3 columns

```
rating.loc[rating['country1']=='India']#analysis of ratings in India
```

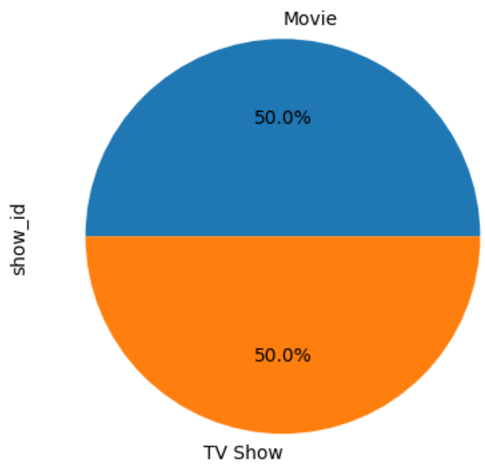
	rating	country1	no_of_shows
2	TV-14	India	572
7	TV-MA	India	266
15	TV-PG	India	144
162	NR	India	7

- In India TV-14 shows are produced more.
- 7 TV-Y7-FV, R are produced very few.

Does Netflix has more focus on tvshows than movies in recent years

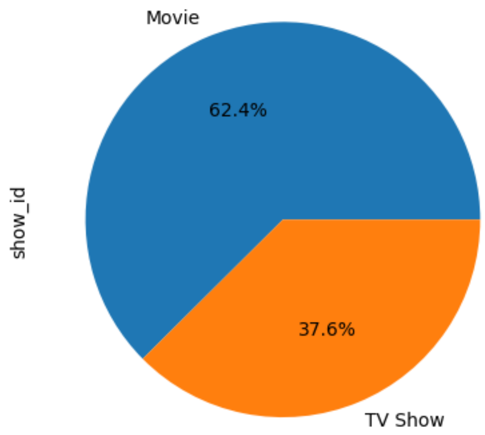
```
past_data=df_1[(df_1['year']==2008) & (df_1['year']>2000)].groupby('type')['show_id'].nunique().plot(kind='pie', autopct='%1.1f%%')
past_data
```

```
<Axes: ylabel='show_id'>
```



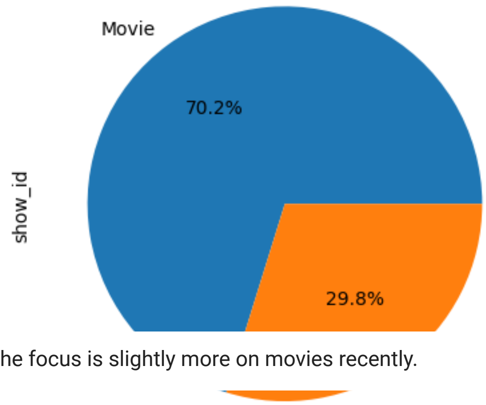
```
past_data=df_1[(df_1['year']<=2016) & (df_1['year']>2000)].groupby('type')['show_id'].nunique().plot(kind='pie', autopct='%1.1f%%')
past_data
```

```
<Axes: ylabel='show_id'>
```



```
recent_data=df_1[(df_1['year']>2016)].groupby('type')['show_id'].nunique().plot(kind='pie', autopct='%1.1f%%')
recent_data
```

```
<Axes: ylabel='show_id'>
```



No. The focus is slightly more on movies recently.