NETFLIX BUSINESS CASE STUDY

In [ ]:

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

In [ ]:

1. Downloading Netflix data set

In [ ]:

```
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv" -O netflix.csv
```

```
--2023-07-12 18:59:54--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 108.157.172.173, 108.157.172.176, 108.157.172.10, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|108.157.172.173|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3399671 (3.2M) [text/plain]
Saving to: ‘netflix.csv’

netflix.csv         100%[===================>]   3.24M  --.-KB/s    in 0.1s

2023-07-12 18:59:55 (24.6 MB/s) - ‘netflix.csv’ saved [3399671/3399671]
```

In [ ]:

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

In [ ]:

2. Analyzing Data

In [ ]:

```
# Gives first 5 rows of loaded data
nf.head()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...

In [ ]:

In [ ]:

```
# Information about data
nf.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
 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 [ ]:

```
# Show unique values of each column

print(nf['show_id'].unique())
print(nf['type'].unique())
print(nf['title'].unique())
print(nf['director'].unique())
print(nf['cast'].unique())
print(nf['country'].unique())
print(nf['date_added'].unique())
print(nf['release_year'].unique())
print(nf['rating'].unique())
print(nf['duration'].unique())
print(nf['listed_in'].unique())
print(nf['description'].unique())
```

```
['s1' 's2' 's3' ... 's8805' 's8806' 's8807']
['Movie' 'TV Show']
['Dick Johnson Is Dead' 'Blood & Water' 'Ganglands' ... 'Zombieland'
 'Zoom' 'Zubaan']
['Kirsten Johnson' nan 'Julien Leclercq' ... 'Majid Al Ansari'
 'Peter Hewitt' 'Mozez Singh']
[nan
 'Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmo
re Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake Ka-Ncube, Odwa Gwanya, Mekaila Mathys, Sandi Schultz, Duan
e Williams, Shamilla Miller, Patrick Mofokeng'
 'Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabiha Akkari, Sofia Lesaffre, Salim Kechiouche, Noureddine Farihi, Geert Van Rampelbe
rg, Bakary Diombera'
 ...
 'Jesse Eisenberg, Woody Harrelson, Emma Stone, Abigail Breslin, Amber Heard, Bill Murray, Derek Graf'
 'Tim Allen, Courteney Cox, Chevy Chase, Kate Mara, Ryan Newman, Michael Cassidy, Spencer Breslin, Rip Torn, Kevin Zegers'
 'Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna Malik, Malkeet Rauni, Anita Shabdish, Chittaranjan Tri
pathy']
['United States' 'South Africa' nan 'India'
 'United States, Ghana, Burkina Faso, United Kingdom, Germany, Ethiopia'
 'United Kingdom' 'Germany, Czech Republic' 'Mexico' 'Turkey' 'Australia'
 'United States, India, France' 'Finland' 'China, Canada, United States'
 'South Africa, United States, Japan' 'Nigeria' 'Japan'
 'Spain, United States' 'France' 'Belgium' 'United Kingdom, United States'
 'United States, United Kingdom' 'France, United States' 'South Korea'
 'Spain' 'United States, Singapore' 'United Kingdom, Australia, France'
 'United Kingdom, Australia, France, United States'
 'United States, Canada' 'Germany, United States'
 'South Africa, United States' 'United States, Mexico'
 'United States, Italy, France, Japan'
 'United States, Italy, Romania, United Kingdom'
 'Australia, United States' 'Argentina, Venezuela'
 'United States, United Kingdom, Canada' 'China, Hong Kong' 'Russia'
 'Canada' 'Hong Kong' 'United States, China, Hong Kong'
 'Italy, United States' 'United States, Germany'
 'United Kingdom, Canada, United States' ', South Korea' 'Ireland'
 'India, Nepal' 'New Zealand, Australia, France, United States' 'Italy'
 'Italy, Brazil, Greece' 'Argentina' 'Jordan' 'Colombia'
 'United States, Japan' 'Belgium, United Kingdom'
 'Switzerland, United Kingdom, Australia' 'Israel, United States'
 'Canada, United States' 'Brazil' 'Argentina, Spain' 'Taiwan'
 'United States, Nigeria' 'Bulgaria, United States'
 'Spain, United Kingdom, United States' 'United States, China'
 'United States, France' 'Spain, France, United Kingdom, United States'
 ', France, Algeria' 'Poland' 'Germany'
 'France, Israel, Germany, United States, United Kingdom' 'New Zealand'
 'Saudi Arabia' 'Thailand' 'Indonesia' 'Egypt, Denmark, Germany'
 'United States, Switzerland' 'Hong Kong, Canada, United States'
 'Kuwait, United States' 'France, Canada, United States, Spain'
 'France, Netherlands, Singapore' 'France, Belgium'
 'Ireland, United States, United Kingdom' 'Egypt' 'Malaysia' 'Israel'
 'Australia, New Zealand' 'United Kingdom, Germany' 'Belgium, Netherlands'
 'South Korea, Czech Republic' 'Australia, Germany' 'Vietnam'
 'United Kingdom, Belgium' 'United Kingdom, Australia, United States'
 'France, Japan, United States'
 'United Kingdom, Germany, Spain, United States'
 'United Kingdom, United States, France, Italy'
 'United States, Germany, Canada'
 'United States, France, Italy, United Kingdom'
 'United States, United Kingdom, Germany, Hungary'
 'United States, New Zealand' 'Sweden' 'China' 'Lebanon' 'Romania'
 'Finland, Germany' 'Lebanon, Syria' 'Philippines' 'Iceland' 'Denmark'
 'United States, India' 'Philippines, Singapore, Indonesia'
 'China, United States, Canada' 'Lebanon, United Arab Emirates'
 'Canada, United States, Denmark' 'United Arab Emirates'
 'Mexico, France, Colombia' 'Netherlands' 'Germany, United States, France'
 'United States, Bulgaria'
 'United Kingdom, France, Germany, United States' 'Norway, Denmark'
 'Syria, France, Lebanon, Qatar' 'United States, Czech Republic'
 'Mauritius' 'Canada, South Africa' 'Austria' 'Mexico, Brazil'
 'Germany, France' 'Mexico, United States'
 'United Kingdom, France, Spain, United States' 'United States, Australia'
 'United States, United Kingdom, France' 'United States, Russia'
 'United States, United Kingdom, New Zealand' 'Australia, United Kingdom'
 'Canada, Nigeria, United States'
 'France, United States, United Kingdom, Canada' 'France, United Kingdom'
 'India, United Kingdom' 'Canada, United States, Mexico'
 'United Kingdom, Germany, United States'
 'Czech Republic, United Kingdom, United States' 'China, United Kingdom'
 'Italy, United Kingdom' 'China, Taiwan'
 'United States, Brazil, Japan, Spain, India'
```

```
'United States, China, United Kingdom' 'Cameroon'
'Lebanon, Palestine, Denmark, Qatar' 'Japan, United States'
'Uruguay, Germany' 'Egypt, Saudi Arabia'
'United Kingdom, France, Poland, Germany, United States'
'Ireland, Switzerland, United Kingdom, France, United States'
'United Kingdom, South Africa, France'
'Ireland, United Kingdom, France, Germany' 'Russia, United States'
'United Kingdom, United States, France' 'United Kingdom,'
'United States, India, United Kingdom' 'Kenya' 'Spain, Argentina'
'India, United Kingdom, France, Qatar' 'Belgium, France'
'Argentina, Chile' 'United States, Thailand' 'Chile, Brazil'
'United States, Colombia' 'Canada, United States, United Kingdom'
'Uruguay' 'Luxembourg' 'United States, Cambodia, Romania' 'Bangladesh'
'Spain, Belgium, United States'
'United Kingdom, United States, Australia'
'Canada, United States, France' 'Portugal, United States'
'Portugal, Spain' 'India, United States' 'United Kingdom, Ireland'
'United Kingdom, Spain, United States' 'Hungary, United States'
'United States, South Korea' 'Canada, United States, Cayman Islands'
'India, France' 'France, Canada' 'Canada, Hungary, United States'
'Norway' 'Canada, United Kingdom, United States'
'United Kingdom, Germany, France, United States' 'Denmark, United States'
'Senegal' 'France, Algeria'
'United Kingdom, Finland, Germany, United States, Australia, Japan, France, Ireland'
'Philippines, Canada, United Kingdom, United States'
'Ireland, France, Iceland, United States, Mexico, Belgium, United Kingdom, Hong Kong'
'Singapore' 'Kuwait' 'United States, France, Serbia'
'United States, Italy' 'Spain, Italy'
'United States, Ireland, United Kingdom, India'
'United Kingdom, Singapore' 'Hong Kong, United States'
'United States, Malta, France, United Kingdom'
'United States, China, Canada' 'Canada, United States, Ireland'
'Lebanon, Canada, France' 'Japan, Canada, United States'
'Spain, France, Canada' 'Denmark, Singapore, Canada, United States'
'United States, France, Denmark' 'United States, China, Colombia'
'Spain, Thailand, United States' 'Mexico, Spain'
'Ireland, Luxembourg, Belgium' 'China, United States' 'Canada, Belgium'
'Canada, United Kingdom'
'Lebanon, United Arab Emirates, France, Switzerland, Germany'
'France, Belgium, Italy' 'Lebanon, United States, United Arab Emirates'
'Lebanon, France' 'France, Lebanon' 'France, Lebanon, United Kingdom'
'France, Norway, Lebanon, Belgium'
'Sweden, Czech Republic, United Kingdom, Denmark, Netherlands'
'United States, United Kingdom, India' 'Indonesia, Netherlands'
'Turkey, South Korea' 'Serbia, United States' 'Namibia'
'United Kingdom, Kenya' 'United Kingdom, France, Germany, Spain'
'United Kingdom, France, United States, Belgium, Luxembourg, China, Germany'
'Thailand, United States' 'United States, France, Canada, Belgium'
'United Kingdom, China' 'Germany, China, United Kingdom'
'Australia, New Zealand, United States'
'Hong Kong, Iceland, United States' 'France, Australia, Germany'
'United States, Belgium, Canada, France' 'South Africa, Angola'
'United States, Philippines'
'United States, United Kingdom, Canada, China'
'United States, Canada, United Kingdom' 'Turkey, United States'
'Peru, Germany, Norway' 'Mozambique' 'Brazil, France'
'China, Spain, South Korea, United States' 'Spain, Germany'
'Hong Kong, China' 'France, Belgium, Luxembourg, Cambodia,'
'United Kingdom, Australia' 'Belarus' 'Indonesia, United Kingdom'
'Switzerland, France, Belgium, United States' 'Ghana'
'Spain, France, Canada, United States' 'Chile, Italy'
'United Kingdom, Nigeria' 'Chile' 'France, Egypt' 'Egypt, France'
'France, Brazil, Spain, Belgium' 'Egypt, Algeria'
'Canada, South Korea, United States' 'Nigeria, United Kingdom'
'United States, France, Canada' 'Poland, United States'
'United Arab Emirates, Jordan, Lebanon, Saudi Arabia'
'United States, Mexico, Spain, Malta'
'Saudi Arabia, United Arab Emirates' 'Zimbabwe'
'United Kingdom, Germany, United Arab Emirates, New Zealand'
'Romania, United States' 'Canada, Nigeria'
'Saudi Arabia, Netherlands, Germany, Jordan, United Arab Emirates, United States'
'United Kingdom, Spain' 'Finland, France'
'United Kingdom, Germany, United States, France'
'India, United Kingdom, China, Canada, Japan, South Korea, United States'
'Italy, United Kingdom, France' 'United States, Mexico, Colombia'
'Turkey, India' 'Italy, Turkey' 'United Kingdom, United States, Japan'
'France, Belgium, United States' 'Puerto Rico, United States, Colombia'
'Uruguay, Argentina' 'United States, United Kingdom, Japan'
'United States, Argentina' 'United Kingdom, Italy'
'Ireland, United Kingdom'
'United Kingdom, France, Belgium, Canada, United States'
'Netherlands, Germany, Denmark, United Kingdom' 'Hungary'
'Austria, Germany' 'Taiwan, China'
'United Kingdom, United States, Ireland' 'South Korea, United States'
'Brazil, United Kingdom' 'Pakistan, United States'
'Romania, France, Switzerland, Germany' 'Romania, United Kingdom'
'France, Malta, United States' 'Cyprus'
'United Kingdom, France, Belgium, Ireland, United States'
'United States, Norway, Canada' 'Kenya, United States'
'France, South Korea, Japan, United States' 'Taiwan, Malaysia'
'Uruguay, Argentina, Germany, Spain'
'United States, United Kingdom, France, Germany, Japan'
'United States, France, Japan' 'United Kingdom, France, United States'
'Spain, France, United States' 'Indonesia, South Korea, Singapore'
'United States, Spain' 'Netherlands, Germany, Italy, Canada'
'Spain, Germany, Denmark, United States' 'Norway, Sweden'
'South Korea, Canada, United States, China' 'Argentina, Uruguay, Serbia'
'France, Japan' 'Mauritius, South Africa' 'United States, Poland'
'United Kingdom, United States, Germany, Denmark, Belgium, Japan'
'India, Germany' 'India, United Kingdom, Canada, United States'
'Philippines, United States' 'Romania, Bulgaria, Hungary'
'Uruguay, Guatemala' 'France, Senegal, Belgium' 'United Kingdom, Canada'
```

```
'Mexico, United States, Spain, Colombia' 'Canada, Norway'
'Singapore, United States' 'Finland, Germany, Belgium'
'United Kingdom, France' 'United States, Chile'
'United Kingdom, Japan, United States' 'Spain, United Kingdom'
'Argentina, United States, Mexico' 'United States, South Korea, Japan'
'Canada, Australia' 'United Kingdom, Hungary, Australia' 'Italy, Belgium'
'United States, United Kingdom, Germany' 'Switzerland'
'Singapore, Malaysia'
'France, Belgium, Luxembourg, Romania, Canada, United States'
'South Africa, Nigeria' 'Spain, France' 'United Kingdom, Hong Kong'
'Pakistan' 'Brazil, United States'
'Denmark, Brazil, France, Portugal, Sweden' 'India, Turkey'
'Malaysia, Singapore, Hong Kong' 'Philippines, Singapore'
'Australia, Canada' 'Taiwan, China, France, United States'
'Germany, Italy' 'Colombia, Peru, United Kingdom'
'Thailand, China, United States' 'Argentina, United States'
'Sweden, United States' 'Uruguay, Spain, Mexico'
'France, Luxembourg, Canada' 'Denmark, Spain' 'Chile, Argentina'
'United Kingdom, Belgium, Sweden' 'Canada, Brazil' 'Italy, France'
'Canada, Germany' 'Pakistan, United Arab Emirates' 'Ghana, United States'
'Mexico, Finland' 'United Arab Emirates, United Kingdom, India'
'Netherlands, Belgium' 'United States, Taiwan'
'Austria, Iraq, United States' 'United Kingdom, Malawi'
'Paraguay, Argentina' 'United Kingdom, Russia, United States'
'India, Pakistan' 'Indonesia, Singapore' 'Spain, Belgium'
'Iceland, Sweden, Belgium' 'Croatia' 'Uruguay, Argentina, Spain'
'United Kingdom, Ireland, United States'
'Canada, Germany, France, United States' 'United Kingdom, Japan'
'Norway, Denmark, Netherlands, Sweden' 'Hong Kong, China, United States'
'Ireland, Canada' 'Italy, Switzerland, France, Germany'
'Mexico, Netherlands' 'United States, Sweden' 'Germany, France, Russia'
'France, Iran, United States' 'United Kingdom, India'
'Russia, Poland, Serbia' 'Spain, Portugal' 'Peru' 'Mexico, Argentina'
'United Kingdom, Canada, United States, Cayman Islands'
'Indonesia, United States'
'United States, Israel, United Kingdom, Canada'
'Norway, Iceland, United States' 'Czech Republic, United States'
'United Kingdom, India, United States' 'United Kingdom, West Germany'
'India, Australia' 'United States,'
'Belgium, United Kingdom, United States' 'India, Germany, Austria'
'United States, Brazil, South Korea, Mexico, Japan, Germany'
'Spain, Mexico' 'China, Japan' 'Argentina, France'
'China, United States, United Kingdom'
'France, Luxembourg, United States' 'China, United States, Australia'
'Colombia, Mexico' 'United States, Canada, Ireland' 'Chile, Peru'
'Argentina, Italy' 'Canada, Japan, United States'
'United Kingdom, Canada, United States, Germany'
'Italy, Switzerland, Albania, Poland' 'United States, Japan, Canada'
'Cambodia' 'Italy, United States, Argentina'
'Saudi Arabia, Syria, Egypt, Lebanon, Kuwait'
'United States, Canada, Indonesia, United Kingdom, China, Singapore'
'Spain, Colombia'
'United Kingdom, South Africa, Australia, United States' 'Bulgaria'
'Argentina, Brazil, France, Poland, Germany, Denmark'
'United Kingdom, Spain, United States, Germany' 'Philippines, Qatar'
'Netherlands, Belgium, Germany, Jordan'
'United Arab Emirates, United States' 'Norway, Germany, Sweden'
'South Korea, China' 'Georgia' 'Soviet Union, India'
'Australia, United Arab Emirates' 'Canada, Germany, South Africa'
'South Korea, China, United States' 'India, Soviet Union' 'India, Mexico'
'Georgia, Germany, France' 'United Arab Emirates, Romania'
'India, Malaysia' 'Germany, Jordan, Netherlands'
'Turkey, France, Germany, Poland' 'Greece, United States'
'France, United Kingdom, United States' 'Norway, Germany'
'France, Morocco' 'Cambodia, United States' 'United States, Denmark'
'United States, Colombia, Mexico'
'United Kingdom, Italy, Israel, Peru, United States'
'Argentina, Uruguay, Spain, France'
'United Kingdom, France, United States, Belgium'
'France, Canada, China, Cambodia'
'United Kingdom, France, Belgium, United States' 'Chile, France'
'Netherlands, United States' 'France, United Kingdom, India'
'Czech Republic, Slovakia' 'Singapore, France' 'Spain, Switzerland'
'United States, Australia, China' 'South Africa, United States, Germany'
'United States, United Kingdom, Australia' 'Spain, Italy, Argentina'
'Chile, Spain, Argentina, Germany' 'West Germany'
'Austria, Czech Republic' 'Lebanon, Qatar'
'United Kingdom, Jordan, Qatar, Iran' 'France, South Korea, Japan'
'Israel, Germany, France' 'Canada, Japan, Netherlands'
'United States, Hungary' 'France, Germany' 'France, Qatar'
'United Kingdom, Germany, Canada' 'Ireland, South Africa'
'Chile, United States, France' 'Belgium, France, Netherlands'
'United Kingdom, Ukraine, United States'
'Germany, Australia, France, China' 'Norway, United States'
'United States, Bermuda, Ecuador'
'United States, Hungary, Ireland, Canada'
'United Kingdom, Egypt, United States'
'United States, France, United Kingdom' 'Spain, Mexico, France'
'United States, South Africa' 'Hong Kong, China, Singapore'
'South Africa, China, United States' 'Denmark, France, Poland'
'New Zealand, United Kingdom' 'Netherlands, Denmark, South Africa'
'Iran, France' 'United Kingdom, United States, France, Germany'
'Australia, France' 'Ireland, United Kingdom, United States'
'United Kingdom, France, Germany' 'Canada, Luxembourg'
'Brazil, Netherlands, United States, Colombia, Austria, Germany'
'France, Canada, Belgium' 'Canada, France'
'Bulgaria, United States, Spain, Canada' 'Sweden, Netherlands'
'France, United States, Mexico'
'Australia, United Kingdom, United Arab Emirates, Canada'
'Australia, Armenia, Japan, Jordan, Mexico, Mongolia, New Zealand, Philippines, South Africa, Sweden, United States, Uruguay'
'India, Iran' 'France, Belgium, Spain'
'Denmark, Sweden, Israel, United States' 'United States, Iceland'
'United Kingdom, Russia' 'United States, Israel, Italy, South Africa'
```

```
'United Kingdom, Russia' 'United States, Israel, Italy, South Africa'
'Netherlands, Denmark, France, Germany' 'South Korea, Japan'
'United Kingdom, Pakistan' 'France, New Zealand'
'United Kingdom, Czech Republic, United States, Germany, Bahamas'
'China, Germany, India, United States' 'Germany, Sri Lanka'
'United States, India, Bangladesh' 'United States, Canada, France'
'Brazil, France, Germany' 'Germany, United States, Hong Kong, Singapore'
'France, Germany, Switzerland'
'Germany, France, Luxembourg, United Kingdom, United States'
'United Kingdom, Canada, Italy' 'Czech Republic, France'
'Taiwan, Hong Kong, United States, China' 'Germany, Australia'
'United Kingdom, Poland, United States' 'Denmark, Zimbabwe'
'United Kingdom, South Africa' 'Finland, Sweden, Norway, Latvia, Germany'
'South Africa, United States, New Zealand, Canada'
'United States, Italy, United Kingdom, Liechtenstein'
'Denmark, France, Belgium, Italy, Netherlands, United States, United Kingdom'
'United States, Australia, Mexico'
'United Kingdom, Czech Republic, Germany, United States'
'France, China, Japan, United States' 'United States, South Korea, China'
'Germany, Belgium' 'Pakistan, Norway, United States'
'United States, Canada, Belgium, United Kingdom' 'Venezuela'
'Canada, France, Italy, Morocco, United States' 'Canada, Spain, France'
'United States, Indonesia' 'Spain, France, Italy'
'United Arab Emirates, United States, United Kingdom'
'United Kingdom, Israel, Russia' 'Spain, Cuba' 'United States, Brazil'
'United States, France, Mexico' 'United States, Nicaragua'
'United Kingdom, United States, Spain, Germany, Greece, Canada'
'Italy, Canada, France' 'United Kingdom, Denmark, Canada, Croatia'
'Italy, Germany' 'United States, France, United Kingdom, Japan'
'United States, United Kingdom, Denmark, Sweden'
'United States, United Kingdom, Italy'
'United States, France, Canada, Spain' 'Russia, United States, China'
'United States, Canada, Germany' 'Ireland, United States'
'United States, United Arab Emirates' 'United States, Ireland'
'Ireland, United Kingdom, Italy, United States' 'Poland,'
'Slovenia, Croatia, Germany, Czech Republic, Qatar'
'Canada, United Kingdom, Netherlands' 'United States, Spain, Germany'
'India, Japan' 'China, South Korea, United States'
'United Kingdom, France, Belgium' 'Canada, Ireland, United States'
'United Kingdom, United States, Dominican Republic'
'United States, Senegal' 'Germany, United Kingdom, United States'
'South Africa, Germany, Netherlands, France'
'Canada, United States, United Kingdom, France, Luxembourg'
'Ireland, United States, France' 'Germany, United States, Canada'
'United Kingdom, Germany, Canada, United States'
'United States, France, Canada, Lebanon, Qatar'
'Netherlands, Belgium, United Kingdom, United States'
'France, Belgium, China, United States' 'United States, Chile, Israel'
'United Kingdom, Norway, Denmark, Germany, Sweden'
'Norway, Denmark, Sweden' 'China, India, Nepal'
'Colombia, Mexico, United States' 'United Kingdom, South Korea'
'Denmark, China' 'United States, Greece, Brazil' 'South Korea, France'
'United States, Australia, Samoa, United Kingdom'
'Germany, United Kingdom' 'Argentina, Chile, Peru' 'Turkey, Azerbaijan'
'Poland, West Germany' 'Germany, United States, Sweden' 'Canada, Spain'
'United States, Cambodia' 'United States, Greece'
'Norway, United Kingdom, France, Ireland' 'United Kingdom, Poland'
'Israel, Sweden, Germany, Netherlands' 'Switzerland, France'
'Italy, India' 'United States, Botswana'
'Chile, Argentina, France, Spain, United States'
'United States, India, South Korea, China'
'Denmark, Germany, Belgium, United Kingdom, France'
'Denmark, Germany, Belgium, United Kingdom, France, Sweden'
'France, Switzerland, Spain, United States, United Arab Emirates'
'Brazil, India, China, United States'
'Denmark, France, United States, Sweden' 'Australia, Iraq'
'China, Morocco, Hong Kong' 'Canada, United States, Germany'
'United Kingdom, Thailand' 'Venezuela, Colombia'
'Colombia, United States' 'France, Germany, Czech Republic, Belgium'
'Switzerland, Vatican City, Italy, Germany, France'
'Portugal, France, Poland, United States'
'United States, New Zealand, Japan'
'United States, Netherlands, Japan, France' 'India, Switzerland'
'Canada, India' 'United States, Morocco' 'Singapore, Japan, France'
'Canada, Mexico, Germany, South Africa'
'United Kingdom, United States, Canada'
'Germany, France, United States, Canada, United Kingdom'
'United States, Uruguay' 'India, Canada'
'Ireland, Canada, United Kingdom, United States'
'United States, Germany, Australia' 'Australia, France, Ireland'
'Australia, India' 'United States, United Kingdom, Canada, Japan'
'Sweden, United Kingdom, Finland' 'Hong Kong, Taiwan'
'United States, United Kingdom, Spain, South Korea' 'Guatemala' 'Ukraine'
'Italy, South Africa, West Germany, Australia, United States'
'United States, Germany, United Kingdom, Australia'
'Italy, France, Switzerland' 'Canada, France, United States'
'Switzerland, United States' 'Thailand, Canada, United States'
'China, Hong Kong, United States' 'United Kingdom, New Zealand'
'Czech Republic, United Kingdom, France'
'Australia, United Kingdom, Canada' 'Jamaica, United States'
'Australia, United Kingdom, United States, New Zealand, Italy, France'
'France, United States, Canada'
'United Kingdom, France, Canada, Belgium, United States'
'Denmark, United Kingdom, Sweden' 'United States, Hong Kong'
'United States, Kazakhstan'
'Argentina, France, United States, Germany, Qatar'
'United States, Germany, United Kingdom'
'United States, Germany, United Kingdom, Italy'
'United States, New Zealand, United Kingdom' 'Finland, United States'
'Spain, France, Uruguay' 'France, Canada, United States'
'United States, Canada, China'
'Ireland, Canada, Luxembourg, United States, United Kingdom, Philippines, India'
'United States, Czech Republic, United Kingdom' 'Israel, Germany'
```

```
 'Mexico, France'
 'Israel, Germany, Poland, Luxembourg, Belgium, France, United States'
 'Austria, United States' 'United Kingdom, Lithuania'
 'United States, Greece, United Kingdom'
 'United Kingdom, China, United States, India'
 'United States, Sweden, Norway' 'United Kingdom, United States, Morocco'
 'United States, United Kingdom, Morocco' 'Spain, Canada, United States'
 'United States, India, United Arab Emirates'
 'United Kingdom, Canada, France, United States' 'India, Germany, France'
 'Belgium, Ireland, Netherlands, Germany, Afghanistan'
 'France, Canada, Italy, United States, China'
 'Ireland, United Kingdom, Greece, France, Netherlands'
 'Denmark, Indonesia, Finland, Norway, United Kingdom, Israel, France, United States, Germany, Netherlands'
 'New Zealand, United States'
 'United States, Australia, South Africa, United Kingdom'
 'United States, Germany, Mexico'
 'Somalia, Kenya, Sudan, South Africa, United States'
 'United States, Canada, Japan, Panama' 'United Kingdom, Spain, Belgium'
 'Serbia, South Korea, Slovenia'
 'Denmark, United Kingdom, South Africa, Sweden, Belgium'
 'Germany, Canada, United States'
 'Ireland, Canada, United States, United Kingdom'
 'New Zealand, United Kingdom, Australia'
 'United Kingdom, Australia, Canada, United States'
 'Germany, United States, Italy' 'United States, Venezuela'
 'United Kingdom, Canada, Japan'
 'United Kingdom, United States, Czech Republic'
 'United Kingdom, China, United States' 'United Kingdom, Brazil, Germany'
 'United Kingdom, Namibia, South Africa, Zimbabwe, United States'
 'Canada, United States, India, United Kingdom'
 'Switzerland, United Kingdom, United States'
 'United Kingdom, India, Sweden'
 'United States, Brazil, India, Uganda, China'
 'Peru, United States, United Kingdom'
 'Germany, United States, United Kingdom, Canada'
 'Canada, India, Thailand, United States, United Arab Emirates'
 'United States, East Germany, West Germany'
 'France, Netherlands, South Africa, Finland'
 'Egypt, Austria, United States' 'Russia, Spain'
 'Croatia, Slovenia, Serbia, Montenegro' 'Japan, Canada'
 'United States, France, South Korea, Indonesia'
 'United Arab Emirates, Jordan']
['September 25, 2021' 'September 24, 2021' 'September 23, 2021' ...
 'December 6, 2018' 'March 9, 2016' 'January 11, 2020']
[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]
['PG-13' 'TV-MA' 'PG' 'TV-14' 'TV-PG' 'TV-Y' 'TV-Y7' 'R' 'TV-G' 'G'
 'NC-17' '74 min' '84 min' '66 min' 'NR' nan 'TV-Y7-FV' 'UR']
['90 min' '2 Seasons' '1 Season' '91 min' '125 min' '9 Seasons' '104 min'
 '127 min' '4 Seasons' '67 min' '94 min' '5 Seasons' '161 min' '61 min'
 '166 min' '147 min' '103 min' '97 min' '106 min' '111 min' '3 Seasons'
 '110 min' '105 min' '96 min' '124 min' '116 min' '98 min' '23 min'
 '115 min' '122 min' '99 min' '88 min' '100 min' '6 Seasons' '102 min'
 '93 min' '95 min' '85 min' '83 min' '113 min' '13 min' '182 min' '48 min'
 '145 min' '87 min' '92 min' '80 min' '117 min' '128 min' '119 min'
 '143 min' '114 min' '118 min' '108 min' '63 min' '121 min' '142 min'
 '154 min' '120 min' '82 min' '109 min' '101 min' '86 min' '229 min'
 '76 min' '89 min' '156 min' '112 min' '107 min' '129 min' '135 min'
 '136 min' '165 min' '150 min' '133 min' '70 min' '84 min' '140 min'
 '78 min' '7 Seasons' '64 min' '59 min' '139 min' '69 min' '148 min'
 '189 min' '141 min' '130 min' '138 min' '81 min' '132 min' '10 Seasons'
 '123 min' '65 min' '68 min' '66 min' '62 min' '74 min' '131 min' '39 min'
 '46 min' '38 min' '8 Seasons' '17 Seasons' '126 min' '155 min' '159 min'
 '137 min' '12 min' '273 min' '36 min' '34 min' '77 min' '60 min' '49 min'
 '58 min' '72 min' '204 min' '212 min' '25 min' '73 min' '29 min' '47 min'
 '32 min' '35 min' '71 min' '149 min' '33 min' '15 min' '54 min' '224 min'
 '162 min' '37 min' '75 min' '79 min' '55 min' '158 min' '164 min'
 '173 min' '181 min' '185 min' '21 min' '24 min' '51 min' '151 min'
 '42 min' '22 min' '134 min' '177 min' '13 Seasons' '52 min' '14 min'
 '53 min' '8 min' '57 min' '28 min' '50 min' '9 min' '26 min' '45 min'
 '171 min' '27 min' '44 min' '146 min' '20 min' '157 min' '17 min'
 '203 min' '41 min' '30 min' '194 min' '15 Seasons' '233 min' '237 min'
 '230 min' '195 min' '253 min' '152 min' '190 min' '160 min' '208 min'
 '180 min' '144 min' '5 min' '174 min' '170 min' '192 min' '209 min'
 '187 min' '172 min' '16 min' '186 min' '11 min' '193 min' '176 min'
 '56 min' '169 min' '40 min' '10 min' '3 min' '168 min' '312 min'
 '153 min' '214 min' '31 min' '163 min' '19 min' '12 Seasons' nan
 '179 min' '11 Seasons' '43 min' '200 min' '196 min' '167 min' '178 min'
 '228 min' '18 min' '205 min' '201 min' '191 min']
['Documentaries' 'International TV Shows, TV Dramas, TV Mysteries'
 'Crime TV Shows, International TV Shows, TV Action & Adventure'
 'Docuseries, Reality TV'
 'International TV Shows, Romantic TV Shows, TV Comedies'
 'TV Dramas, TV Horror, TV Mysteries' 'Children & Family Movies'
 'Dramas, Independent Movies, International Movies'
 'British TV Shows, Reality TV' 'Comedies, Dramas'
 'Crime TV Shows, Docuseries, International TV Shows'
 'Dramas, International Movies' 'Children & Family Movies, Comedies'
 'British TV Shows, Crime TV Shows, Docuseries' 'TV Comedies, TV Dramas'
 'Documentaries, International Movies'
 'Crime TV Shows, Spanish-Language TV Shows, TV Dramas' 'Thrillers'
 'International TV Shows, Spanish-Language TV Shows, TV Action & Adventure'
 'International TV Shows, TV Action & Adventure, TV Dramas'
 'Comedies, International Movies'
 'Comedies, International Movies, Romantic Movies'
 'Docuseries, International TV Shows, Reality TV'
 'Comedies, International Movies, Music & Musicals' 'Comedies'
 'Horror Movies, Sci-Fi & Fantasy' 'TV Comedies'
 'British TV Shows, International TV Shows, TV Comedies'
 'British TV Shows, International TV Shows, TV Comedies'
 'International TV Shows, TV Dramas, TV Thrillers' "Kids' TV"
 'Dramas, International Movies, Thrillers'
 'Action & Adventure, Dramas, International Movies'
 "Kids' TV, TV Comedies" 'Action & Adventure, Dramas'
 "Kids' TV, TV Sci-Fi & Fantasy"
 'Action & Adventure, Classic Movies, Dramas'
 'Dramas, Horror Movies, Thrillers'
 'Action & Adventure, Horror Movies, Thrillers' 'Action & Adventure'
 'Dramas, Thrillers' 'International TV Shows, TV Dramas'
 'International TV Shows, TV Dramas, TV Sci-Fi & Fantasy'
 'Action & Adventure, Anime Features, International Movies' 'Reality TV'
 'Docuseries, International TV Shows'
 'Documentaries, International Movies, Sports Movies'
 'International TV Shows, Reality TV, Romantic TV Shows'
 'British TV Shows, Docuseries, International TV Shows'
 'Anime Series, International TV Shows'
 'Comedies, Dramas, International Movies'
 'Crime TV Shows, TV Comedies, TV Dramas'
 'Action & Adventure, Comedies, Dramas' "Anime Series, Kids' TV"
 'International Movies, Thrillers' "Kids' TV, Korean TV Shows"
 'Documentaries, Sports Movies' 'Sci-Fi & Fantasy, Thrillers'
 'Dramas, International Movies, Romantic Movies'
 'Documentaries, Music & Musicals'
 "Kids' TV, TV Comedies, TV Sci-Fi & Fantasy" "British TV Shows, Kids' TV"
 'Docuseries, Science & Nature TV' 'Children & Family Movies, Dramas'
 "Kids' TV, TV Dramas, Teen TV Shows"
 'Crime TV Shows, International TV Shows, Spanish-Language TV Shows'
 'Docuseries, International TV Shows, Spanish-Language TV Shows' 'Dramas'
 'Comedies, Romantic Movies' 'Dramas, Romantic Movies'
 'Comedies, Dramas, Independent Movies'
 'Crime TV Shows, TV Action & Adventure, TV Comedies'
 'Children & Family Movies, Music & Musicals'
 'Action & Adventure, Classic Movies, Cult Movies'
 'International TV Shows, TV Action & Adventure, TV Comedies'
 'Action & Adventure, Sci-Fi & Fantasy' 'Action & Adventure, Comedies'
 'Classic Movies, Comedies, Dramas' 'Comedies, Cult Movies'
 'Comedies, Cult Movies, Music & Musicals' 'Comedies, Music & Musicals'
 'TV Shows' 'Action & Adventure, International Movies'
 'Anime Series, International TV Shows, Teen TV Shows'
 'Action & Adventure, Children & Family Movies, Cult Movies'
 'Comedies, Dramas, Romantic Movies'
 'Comedies, Cult Movies, Sci-Fi & Fantasy' 'Classic Movies, Dramas'
 'Action & Adventure, Children & Family Movies, Comedies'
 'Dramas, Faith & Spirituality' 'Documentaries, LGBTQ Movies'
 'Action & Adventure, Classic Movies' 'Docuseries'
 'International TV Shows, TV Comedies' 'Dramas, Independent Movies'
 'Action & Adventure, Comedies, International Movies'
 'International TV Shows, Spanish-Language TV Shows, TV Dramas'
 'Crime TV Shows, International TV Shows, TV Dramas'
 'Action & Adventure, Horror Movies, International Movies'
 'Comedies, International Movies, Sci-Fi & Fantasy'
 'Action & Adventure, International Movies, Music & Musicals'
 'Dramas, International Movies, Music & Musicals'
 'Horror Movies, International Movies' 'Reality TV, Teen TV Shows'
 'Crime TV Shows, TV Dramas, TV Mysteries'
 'International TV Shows, Reality TV'
 'International TV Shows, TV Comedies, TV Dramas'
 'Dramas, Independent Movies, Romantic Movies' 'Horror Movies'
 'Documentaries, LGBTQ Movies, Sports Movies'
 'Horror Movies, International Movies, Thrillers'
 'Action & Adventure, Anime Features'
 'TV Dramas, TV Mysteries, TV Sci-Fi & Fantasy'
 'International TV Shows, Spanish-Language TV Shows, TV Comedies'
 'Children & Family Movies, Comedies, Music & Musicals'
 'Comedies, Independent Movies'
 'Anime Series, International TV Shows, Romantic TV Shows'
 'Classic Movies, Dramas, Independent Movies'
 'International TV Shows, Romantic TV Shows, Spanish-Language TV Shows'
 'International TV Shows, TV Dramas, Teen TV Shows' 'Stand-Up Comedy'
 'Action & Adventure, Anime Features, Children & Family Movies'
 'International TV Shows, Romantic TV Shows, TV Dramas'
 'International Movies, Music & Musicals'
 'TV Action & Adventure, TV Dramas, TV Mysteries'
 'Horror Movies, Independent Movies, International Movies'
 'Comedies, Cult Movies, International Movies'
 'Classic Movies, Dramas, International Movies' 'Movies'
 'Crime TV Shows, Docuseries'
 'Children & Family Movies, Comedies, Sci-Fi & Fantasy'
 'Anime Series, International TV Shows, TV Thrillers'
 'Action & Adventure, Horror Movies, Sci-Fi & Fantasy'
 'Classic Movies, Comedies, Cult Movies' 'TV Dramas, Teen TV Shows'
 'Action & Adventure, Sci-Fi & Fantasy, Thrillers'
 'Children & Family Movies, Comedies, Dramas' 'Dramas, Sports Movies'
 'Action & Adventure, Dramas, Sci-Fi & Fantasy'
 'Action & Adventure, Comedies, Cult Movies'
 'Dramas, Independent Movies, Thrillers' 'TV Dramas, TV Sci-Fi & Fantasy'
 'Action & Adventure, International Movies, Thrillers'
 'British TV Shows, International TV Shows, Reality TV'
 'TV Action & Adventure, TV Dramas, Teen TV Shows' 'Anime Series'
 'Crime TV Shows, TV Action & Adventure, TV Sci-Fi & Fantasy'
 'Crime TV Shows, International TV Shows, TV Comedies'
 'Stand-Up Comedy & Talk Shows, TV Comedies'
 'Classic & Cult TV, TV Action & Adventure, TV Dramas'
 'Children & Family Movies, Sports Movies'
 'TV Action & Adventure, TV Sci-Fi & Fantasy'
 'Anime Series, Stand-Up Comedy & Talk Shows' 'TV Dramas'
 'Anime Features, Children & Family Movies, International Movies'
 'Classic & Cult TV, Crime TV Shows, International TV Shows'
 'Crime TV Shows, International TV Shows, Romantic TV Shows'
 'Horror Movies, LGBTQ Movies'
 'Action & Adventure, Dramas, Romantic Movies'
 'Documentaries, International Movies, Music & Musicals'
```

```
'Documentaries, International Movies, Music & Musicals'
'TV Comedies, TV Dramas, Teen TV Shows'
'Children & Family Movies, Comedies, Sports Movies'
'Children & Family Movies, Dramas, International Movies'
'Comedies, Documentaries, International Movies'
'Romantic TV Shows, TV Dramas' 'Anime Series, TV Horror, TV Thrillers'
'International Movies, Romantic Movies'
'TV Action & Adventure, TV Dramas, TV Sci-Fi & Fantasy'
"Kids' TV, Korean TV Shows, TV Comedies"
'British TV Shows, Crime TV Shows, International TV Shows'
'Crime TV Shows, TV Horror, TV Mysteries'
'Docuseries, International TV Shows, Science & Nature TV'
'British TV Shows, International TV Shows, TV Dramas'
"Kids' TV, TV Action & Adventure, TV Sci-Fi & Fantasy"
'International Movies, Romantic Movies, Thrillers'
'Action & Adventure, Cult Movies, International Movies'
'Action & Adventure, Comedies, Sci-Fi & Fantasy'
"International TV Shows, Kids' TV, TV Mysteries"
'Action & Adventure, Thrillers'
'Dramas, Faith & Spirituality, International Movies'
'Action & Adventure, Classic Movies, Comedies'
'Action & Adventure, Comedies, Sports Movies'
'Action & Adventure, Children & Family Movies, Classic Movies'
'Action & Adventure, Children & Family Movies, Dramas'
'Horror Movies, Thrillers' 'Action & Adventure, Romantic Movies'
'Dramas, Romantic Movies, Sci-Fi & Fantasy'
'Dramas, Music & Musicals, Romantic Movies'
'Anime Series, Crime TV Shows, International TV Shows'
'Reality TV, Romantic TV Shows'
'International Movies, Music & Musicals, Romantic Movies'
'Reality TV, TV Action & Adventure, TV Mysteries'
'Crime TV Shows, TV Dramas'
'International TV Shows, Reality TV, Spanish-Language TV Shows'
'Crime TV Shows, TV Dramas, TV Thrillers' 'British TV Shows, Docuseries'
'International TV Shows, Korean TV Shows, TV Comedies'
'Action & Adventure, Anime Features, Classic Movies'
'TV Action & Adventure, TV Dramas, TV Horror'
'Crime TV Shows, International TV Shows, TV Thrillers'
'Anime Series, Crime TV Shows, TV Horror' 'Anime Features, Documentaries'
'Comedies, Horror Movies'
'International TV Shows, Spanish-Language TV Shows, Stand-Up Comedy & Talk Shows'
'Children & Family Movies, Documentaries, International Movies'
'Romantic TV Shows, TV Comedies, TV Dramas'
'Dramas, Faith & Spirituality, Romantic Movies'
'Dramas, Independent Movies, LGBTQ Movies'
'Comedies, Independent Movies, LGBTQ Movies'
'Action & Adventure, Cult Movies, Sci-Fi & Fantasy'
'Cult Movies, Horror Movies' 'Action & Adventure, Dramas, Sports Movies'
'Anime Series, Romantic TV Shows, Teen TV Shows'
'Dramas, International Movies, LGBTQ Movies'
'Dramas, Romantic Movies, Thrillers'
'Children & Family Movies, Dramas, Faith & Spirituality'
'Dramas, International Movies, Sports Movies'
'Action & Adventure, Horror Movies'
'Documentaries, International Movies, LGBTQ Movies'
'Dramas, Independent Movies, Sci-Fi & Fantasy'
'Comedies, Independent Movies, International Movies'
'Reality TV, TV Horror, TV Thrillers'
'TV Action & Adventure, TV Horror, TV Sci-Fi & Fantasy'
'International TV Shows, TV Horror, TV Sci-Fi & Fantasy'
'Independent Movies, International Movies, Thrillers'
'Independent Movies, Thrillers' 'Documentaries, Dramas'
'Action & Adventure, Sports Movies'
'Dramas, International Movies, Sci-Fi & Fantasy'
'Comedies, Independent Movies, Romantic Movies'
'Horror Movies, Romantic Movies, Sci-Fi & Fantasy'
'International TV Shows, Stand-Up Comedy & Talk Shows'
'Action & Adventure, Anime Features, Horror Movies'
'Cult Movies, Dramas, Music & Musicals' 'TV Dramas, TV Thrillers'
'Crime TV Shows, International TV Shows, Korean TV Shows'
'TV Horror, TV Mysteries, TV Thrillers'
'Comedies, Horror Movies, International Movies'
'Crime TV Shows, Docuseries, TV Mysteries'
'Comedies, International Movies, Sports Movies'
'Classic Movies, Music & Musicals' 'Reality TV, TV Comedies, TV Horror'
'Children & Family Movies, Faith & Spirituality, Music & Musicals'
'International TV Shows, Korean TV Shows, Stand-Up Comedy & Talk Shows'
'Dramas, Music & Musicals'
'Docuseries, Science & Nature TV, TV Action & Adventure'
"British TV Shows, Kids' TV, TV Dramas"
'International TV Shows, Korean TV Shows, Romantic TV Shows'
'Horror Movies, Independent Movies'
"Anime Series, Kids' TV, TV Action & Adventure"
'Comedies, Dramas, Music & Musicals' 'TV Horror, Teen TV Shows'
'Comedies, LGBTQ Movies, Thrillers'
'Docuseries, Reality TV, Science & Nature TV'
'Crime TV Shows, Spanish-Language TV Shows, TV Action & Adventure'
'Romantic TV Shows, Teen TV Shows' 'TV Comedies, Teen TV Shows'
'Romantic TV Shows, TV Dramas, Teen TV Shows'
'Children & Family Movies, Sci-Fi & Fantasy'
'Romantic TV Shows, TV Action & Adventure, TV Dramas'
'Comedies, International Movies, LGBTQ Movies' 'Dramas, Sci-Fi & Fantasy'
"Kids' TV, TV Thrillers"
'TV Action & Adventure, TV Comedies, TV Sci-Fi & Fantasy'
'British TV Shows, Romantic TV Shows, TV Dramas'
'Anime Series, International TV Shows, Spanish-Language TV Shows'
'Docuseries, TV Comedies' 'Comedies, Romantic Movies, Sports Movies'
'TV Action & Adventure, TV Comedies, TV Dramas'
'Children & Family Movies, Dramas, Sports Movies'
'Action & Adventure, Dramas, Independent Movies'
'Spanish-Language TV Shows, TV Dramas' 'Dramas, LGBTQ Movies'
'TV Horror, TV Mysteries, TV Sci-Fi & Fantasy'
'Action & Adventure, Dramas, Faith & Spirituality'
```

```
 'International TV Shows, TV Mysteries, TV Thrillers'
 'British TV Shows, Classic & Cult TV, International TV Shows'
 'Action & Adventure, Comedies, Independent Movies' 'Music & Musicals'
 "British TV Shows, Kids' TV, TV Comedies"
 'Docuseries, Spanish-Language TV Shows'
 'Dramas, Independent Movies, Sports Movies'
 'TV Dramas, TV Mysteries, TV Thrillers'
 'Comedies, LGBTQ Movies, Music & Musicals'
 'International TV Shows, TV Action & Adventure, TV Mysteries'
 "Kids' TV, TV Comedies, Teen TV Shows"
 'International TV Shows, TV Dramas, TV Horror'
 'Comedies, International Movies, Thrillers'
 'Classic & Cult TV, TV Action & Adventure, TV Sci-Fi & Fantasy'
 'International TV Shows, TV Horror, TV Mysteries'
 'Children & Family Movies, Documentaries'
 'Music & Musicals, Romantic Movies' 'Romantic Movies'
 'Children & Family Movies, Classic Movies, Comedies'
 'TV Action & Adventure, TV Dramas'
 'Dramas, LGBTQ Movies, Romantic Movies'
 'Children & Family Movies, Comedies, Romantic Movies'
 'Comedies, Sports Movies' 'International Movies'
 'International TV Shows, Romantic TV Shows, TV Mysteries'
 'Stand-Up Comedy & Talk Shows'
 'Action & Adventure, International Movies, Romantic Movies'
 'Reality TV, TV Comedies' 'Cult Movies, Dramas, International Movies'
 "Kids' TV, TV Dramas"
 'Crime TV Shows, International TV Shows, TV Mysteries'
 'Action & Adventure, Sci-Fi & Fantasy, Sports Movies'
 'TV Dramas, TV Sci-Fi & Fantasy, TV Thrillers'
 'Romantic TV Shows, TV Dramas, TV Sci-Fi & Fantasy'
 'Docuseries, TV Sci-Fi & Fantasy' 'Anime Features, International Movies'
 "British TV Shows, Classic & Cult TV, Kids' TV"
 'British TV Shows, Reality TV, Romantic TV Shows'
 'Documentaries, Faith & Spirituality, International Movies'
 "Kids' TV, Reality TV, TV Dramas" 'LGBTQ Movies, Thrillers'
 'TV Action & Adventure, TV Mysteries, TV Sci-Fi & Fantasy'
 'Reality TV, Science & Nature TV'
 "Kids' TV, TV Action & Adventure, TV Comedies"
 'International TV Shows, Romantic TV Shows, TV Action & Adventure'
 'Children & Family Movies, Dramas, Independent Movies'
 'Comedies, Music & Musicals, Romantic Movies'
 'International TV Shows, Korean TV Shows, Reality TV'
 'Classic & Cult TV, TV Dramas, TV Sci-Fi & Fantasy'
 'Anime Features, Children & Family Movies'
 'Action & Adventure, International Movies, Sci-Fi & Fantasy'
 'Crime TV Shows, TV Action & Adventure, TV Dramas'
 'Classic & Cult TV, TV Action & Adventure, TV Horror'
 'International TV Shows, Korean TV Shows, TV Dramas'
 'International TV Shows, TV Action & Adventure, TV Horror'
 'Action & Adventure, Comedies, Romantic Movies'
 'International TV Shows, Korean TV Shows, TV Action & Adventure'
 "Classic & Cult TV, Kids' TV, TV Action & Adventure"
 'Anime Series, International TV Shows, TV Horror'
 'International TV Shows, Korean TV Shows, TV Horror'
 'Children & Family Movies, Comedies, International Movies'
 'International Movies, Sci-Fi & Fantasy'
 'International Movies, Sci-Fi & Fantasy, Thrillers'
 'Children & Family Movies, Dramas, Romantic Movies'
 'Anime Series, Romantic TV Shows' 'Comedies, Dramas, LGBTQ Movies'
 'British TV Shows, International TV Shows, TV Action & Adventure'
 'Docuseries, Science & Nature TV, TV Comedies'
 'International TV Shows, Stand-Up Comedy & Talk Shows, TV Comedies'
 'Children & Family Movies, Dramas, Music & Musicals'
 'Action & Adventure, Independent Movies, International Movies'
 'Action & Adventure, Children & Family Movies, Sci-Fi & Fantasy'
 'Horror Movies, Independent Movies, Sci-Fi & Fantasy'
 'TV Dramas, TV Sci-Fi & Fantasy, Teen TV Shows'
 'Anime Features, International Movies, Sci-Fi & Fantasy'
 'Dramas, Independent Movies, Music & Musicals'
 "Kids' TV, TV Comedies, TV Dramas"
 'Children & Family Movies, Documentaries, Sports Movies'
 'Independent Movies, Sci-Fi & Fantasy, Thrillers'
 'Anime Features, Music & Musicals, Sci-Fi & Fantasy'
 'TV Comedies, TV Dramas, TV Sci-Fi & Fantasy'
 'Crime TV Shows, TV Action & Adventure'
 'Comedies, Faith & Spirituality, Romantic Movies'
 "Kids' TV, TV Action & Adventure"
 'Action & Adventure, Independent Movies'
 'International TV Shows, Reality TV, TV Comedies'
 'Docuseries, Reality TV, Teen TV Shows'
 'Crime TV Shows, International TV Shows, Reality TV'
 'Anime Series, Teen TV Shows'
 'Crime TV Shows, Romantic TV Shows, TV Dramas'
 'Anime Features, Romantic Movies'
 'Horror Movies, Sci-Fi & Fantasy, Thrillers'
 'International TV Shows, TV Comedies, TV Sci-Fi & Fantasy'
 'International TV Shows, Romantic TV Shows'
 'Anime Features, Music & Musicals'
 'Anime Features, International Movies, Romantic Movies'
 'International TV Shows, Romantic TV Shows, Teen TV Shows'
 'Docuseries, Stand-Up Comedy & Talk Shows'
 'Horror Movies, Independent Movies, Thrillers'
 'TV Action & Adventure, TV Comedies, TV Horror'
 'Documentaries, Stand-Up Comedy' "Kids' TV, Spanish-Language TV Shows"
 "British TV Shows, Kids' TV, TV Thrillers"
 "Kids' TV, TV Action & Adventure, TV Dramas"
 'Anime Series, Crime TV Shows' 'Dramas, Sci-Fi & Fantasy, Thrillers'
 'TV Comedies, TV Dramas, TV Horror'
 'Children & Family Movies, Comedies, LGBTQ Movies'
 'International TV Shows, TV Action & Adventure, TV Sci-Fi & Fantasy'
 'Docuseries, TV Dramas'
 'Horror Movies, International Movies, Romantic Movies'
```

```
'Crime TV Shows, Docuseries, Science & Nature TV'
'International Movies, Music & Musicals, Thrillers'
"Kids' TV, Spanish-Language TV Shows, Teen TV Shows"
'Comedies, Horror Movies, Independent Movies'
'Action & Adventure, International Movies, Sports Movies'
'Action & Adventure, Independent Movies, Sci-Fi & Fantasy'
'Horror Movies, LGBTQ Movies, Music & Musicals'
'Comedies, Music & Musicals, Sports Movies'
'TV Horror, TV Mysteries, Teen TV Shows' 'Romantic TV Shows, TV Comedies'
"Kids' TV, Reality TV, Science & Nature TV"
'International Movies, Romantic Movies, Sci-Fi & Fantasy'
'TV Comedies, TV Horror, TV Thrillers' 'TV Action & Adventure'
'International TV Shows, Spanish-Language TV Shows, TV Horror'
'Crime TV Shows, TV Action & Adventure, TV Thrillers'
'Music & Musicals, Stand-Up Comedy' 'British TV Shows, TV Comedies'
'TV Comedies, TV Sci-Fi & Fantasy, Teen TV Shows'
'TV Comedies, TV Sci-Fi & Fantasy'
'Romantic TV Shows, Spanish-Language TV Shows, TV Comedies'
'Crime TV Shows, International TV Shows, TV Sci-Fi & Fantasy'
'British TV Shows, International TV Shows, Romantic TV Shows'
"Crime TV Shows, Kids' TV"
'Horror Movies, International Movies, Sci-Fi & Fantasy'
'TV Comedies, TV Mysteries'
'Cult Movies, Horror Movies, Independent Movies'
'British TV Shows, Docuseries, TV Comedies' 'Comedies, Documentaries'
'Reality TV, Science & Nature TV, TV Action & Adventure'
'TV Comedies, TV Dramas, TV Mysteries'
'Crime TV Shows, TV Comedies, Teen TV Shows'
"Docuseries, Kids' TV, Science & Nature TV"
'Reality TV, Spanish-Language TV Shows'
'Action & Adventure, Anime Features, Sci-Fi & Fantasy'
"Crime TV Shows, Kids' TV, TV Comedies"
'Dramas, Faith & Spirituality, Independent Movies'
'Documentaries, Faith & Spirituality'
'British TV Shows, International TV Shows, Stand-Up Comedy & Talk Shows'
'Comedies, Dramas, Faith & Spirituality' 'Classic & Cult TV, TV Comedies'
'Dramas, Romantic Movies, Sports Movies'
'Stand-Up Comedy & Talk Shows, TV Mysteries, TV Sci-Fi & Fantasy'
'TV Sci-Fi & Fantasy, TV Thrillers'
'Comedies, Independent Movies, Music & Musicals'
'Comedies, Cult Movies, Independent Movies'
'Documentaries, Dramas, International Movies'
'British TV Shows, TV Horror, TV Thrillers'
'British TV Shows, Docuseries, Science & Nature TV'
'Children & Family Movies, Comedies, Cult Movies' 'Sports Movies'
'Sci-Fi & Fantasy' 'Comedies, LGBTQ Movies'
'Comedies, Independent Movies, Thrillers'
'Classic Movies, Cult Movies, Dramas'
'British TV Shows, TV Comedies, TV Dramas'
'Action & Adventure, Children & Family Movies, Independent Movies'
'Action & Adventure, Documentaries, International Movies'
'Children & Family Movies, Independent Movies'
'Comedies, Cult Movies, Dramas'
'International TV Shows, TV Horror, TV Thrillers'
'Classic Movies, Thrillers' 'Crime TV Shows, TV Dramas, TV Horror'
'British TV Shows, Docuseries, Reality TV'
'Documentaries, LGBTQ Movies, Music & Musicals'
'Classic Movies, Dramas, Romantic Movies'
'Crime TV Shows, Romantic TV Shows, Spanish-Language TV Shows'
'Classic Movies, Cult Movies, Horror Movies'
'Anime Series, Crime TV Shows, TV Thrillers'
'Children & Family Movies, Classic Movies'
'Classic Movies, Comedies, International Movies'
'Comedies, Sci-Fi & Fantasy' 'Action & Adventure, Cult Movies, Dramas'
'Documentaries, Faith & Spirituality, Music & Musicals'
'British TV Shows, Classic & Cult TV, TV Comedies'
'International Movies, Sports Movies' 'International TV Shows'
"Classic & Cult TV, Kids' TV, Spanish-Language TV Shows"
'Romantic TV Shows, Spanish-Language TV Shows, TV Dramas'
'Children & Family Movies, Comedies, Faith & Spirituality'
'British TV Shows, Crime TV Shows, TV Dramas'
'Classic Movies, Dramas, Music & Musicals'
'Cult Movies, Horror Movies, Thrillers'
'Action & Adventure, Classic Movies, Sci-Fi & Fantasy'
'TV Action & Adventure, TV Comedies'
'Classic Movies, Comedies, Music & Musicals' 'Independent Movies'
'Documentaries, Horror Movies'
'Classic & Cult TV, TV Horror, TV Mysteries'
'Comedies, Faith & Spirituality, International Movies'
'Dramas, Horror Movies, Sci-Fi & Fantasy'
'British TV Shows, TV Dramas, TV Sci-Fi & Fantasy'
'Comedies, Cult Movies, Horror Movies'
'Comedies, Cult Movies, Sports Movies' 'Classic Movies, Documentaries'
'Action & Adventure, Faith & Spirituality, Sci-Fi & Fantasy'
'Action & Adventure, Children & Family Movies'
'International TV Shows, Reality TV, TV Action & Adventure'
'Docuseries, Science & Nature TV, TV Dramas' 'Anime Features'
'Action & Adventure, Horror Movies, Independent Movies'
'Action & Adventure, Classic Movies, International Movies'
'Cult Movies, Independent Movies, Thrillers'
'Crime TV Shows, TV Comedies'
'Classic Movies, Cult Movies, Documentaries'
"Classic & Cult TV, Kids' TV, TV Comedies"
'Classic Movies, Dramas, LGBTQ Movies'
'Classic Movies, Dramas, Sports Movies' 'Action & Adventure, Cult Movies'
'Action & Adventure, Comedies, Music & Musicals'
'Classic Movies, Horror Movies, Thrillers'
'Classic Movies, Comedies, Independent Movies'
'Children & Family Movies, Classic Movies, Dramas'
'Dramas, Faith & Spirituality, Sports Movies'
'Classic Movies, Comedies, Romantic Movies'
'Dramas, Horror Movies, Music & Musicals'
'Classic Movies, Independent Movies, Thrillers'
```

```
 'Classic Movies, Independent Movies, Thrillers'
 'Children & Family Movies, Faith & Spirituality'
 'Classic Movies, Comedies, Sports Movies'
 'Comedies, Dramas, Sports Movies'
 'Action & Adventure, Romantic Movies, Sci-Fi & Fantasy'
 'Classic & Cult TV, TV Sci-Fi & Fantasy'
 'Comedies, Cult Movies, LGBTQ Movies'
 'Comedies, Horror Movies, Sci-Fi & Fantasy'
 'Action & Adventure, Comedies, Horror Movies'
 'Classic & Cult TV, Crime TV Shows, TV Dramas'
 'Action & Adventure, Documentaries, Sports Movies'
 'International Movies, LGBTQ Movies, Romantic Movies'
 'Cult Movies, Dramas, Thrillers']
['As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inventive and comical ways to help them bo
th face the inevitable.'
 'After crossing paths at a party, a Cape Town teen sets out to prove whether a private-school swimming star is her sister who was
abducted at birth.'
 'To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team of robbers are pulled into a violent an
d deadly turf war.'
 ...
 'Looking to survive in a world taken over by zombies, a dorky college student teams with an urban roughneck and a pair of grifter
sisters.'
 'Dragged from civilian life, a former superhero must train a new crop of youthful saviors when the military preps for an attack b
y a familiar villain.'
 "A scrappy but poor boy worms his way into a tycoon's dysfunctional family, while facing his fear of music and the truth about his
past."]
```

Business problem : which type of shows/movies to produce

Documentaries International TV Shows Crime TV Shows Docuseries TV Dramas Children & Family Movies Dramas British TV Shows Comedies TV Comedies Thrillers Horror Movies Kids' TV Action & Adventure Reality TV Anime Series International Movies Sci-Fi & Fantasy Classic Movies TV Shows Stand-Up Comedy TV Action & Adventure Movies Stand-Up Comedy & Talk Shows Classic & Cult TV Anime Features Romantic TV Shows Cult Movies Independent Movies TV Horror Spanish-Language TV Shows Music & Musicals Romantic Movies LGBTQ Movies TV Sci-Fi & Fantasy Sports Movies TV Dramas International TV Shows Reality TV Romantic TV Shows TV Horror Independent Movies Dramas Docuseries International Movies Comedies Crime TV Shows Spanish-Language TV Shows TV Action & Adventure Sci-Fi & Fantasy TV Comedies TV Sci-Fi & Fantasy Classic Movies Horror Movies Thrillers Anime Features Kids' TV Korean TV Shows Sports Movies Music & Musicals Science & Nature TV Romantic Movies Cult Movies Children & Family Movies Faith & Spirituality LGBTQ Movies Teen TV Shows TV Mysteries Stand-Up Comedy & Talk Shows Documentaries TV Thrillers Classic & Cult TV Stand-Up Comedy

Netflix is already covering 73 different types of movies & TV Shows, so adding one more does not make any sense. They are already doing great there.

In [ ]:

```

```

Business problem: how they can grow the business in different countries

While data is very useful to answer the above question, Honestly i think more than 50% of the answer is outside of data provided.

Suggestions:

They have over 10000 movies or tv shows available on their platform, as of mid-2021, they have over 222M Subscribers globally.

there are roughly 8 billion people on earth. without exception everyone likes some kind of celluloid entertainment.

So there is no reason, not to think that netflix can (and should) capture all 8 billion of them.

they have 222M subscribers there are 8000 M people thats 2.775% only, they can do a lot better, if they take strategic measures.

1. **Take the game more regionally (roots)**
2. **more regional is more universal.**

Lets focus on the rest of the 50% of what the data is saying, to get 100% of the marks available for grabs

In [ ]:

```

```

Understanding each column

In [ ]:

```
print(nf['show_id'].unique())
print(nf['type'].unique())
print(nf['title'].unique())
print(nf['director'].unique())
print(nf['cast'].unique())
print(nf['country'].unique())
print(nf['date_added'].unique())
print(nf['release_year'].unique())
print(nf['rating'].unique())
print(nf['duration'].unique())
print(nf['listed_in'].unique())
print(nf['description'].unique())
```

```
['s1' 's2' 's3' ... 's8805' 's8806' 's8807']
['Movie' 'TV Show']
['Dick Johnson Is Dead' 'Blood & Water' 'Ganglands' ... 'Zombieland'
 'Zoom' 'Zubaan']
['Kirsten Johnson' nan 'Julien Leclercq' ... 'Majid Al Ansari'
```

```
 'Peter Hewitt' 'Mozez Singh']
[nan
 'Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmo
re Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake Ka-Ncube, Odwa Gwanya, Mekaila Mathys, Sandi Schultz, Duan
e Williams, Shamilla Miller, Patrick Mofokeng'
 'Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabiha Akkari, Sofia Lesaffre, Salim Kechiouche, Noureddine Farihi, Geert Van Rampelbe
rg, Bakary Diombera'
 ...
 'Jesse Eisenberg, Woody Harrelson, Emma Stone, Abigail Breslin, Amber Heard, Bill Murray, Derek Graf'
 'Tim Allen, Courteney Cox, Chevy Chase, Kate Mara, Ryan Newman, Michael Cassidy, Spencer Breslin, Rip Torn, Kevin Zegers'
 'Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna Malik, Malkeet Rauni, Anita Shabdish, Chittaranjan Tri
pathy']
['United States' 'South Africa' nan 'India'
 'United States, Ghana, Burkina Faso, United Kingdom, Germany, Ethiopia'
 'United Kingdom' 'Germany, Czech Republic' 'Mexico' 'Turkey' 'Australia'
 'United States, India, France' 'Finland' 'China, Canada, United States'
 'South Africa, United States, Japan' 'Nigeria' 'Japan'
 'Spain, United States' 'France' 'Belgium' 'United Kingdom, United States'
 'United States, United Kingdom' 'France, United States' 'South Korea'
 'Spain' 'United States, Singapore' 'United Kingdom, Australia, France'
 'United Kingdom, Australia, France, United States'
 'United States, Canada' 'Germany, United States'
 'South Africa, United States' 'United States, Mexico'
 'United States, Italy, France, Japan'
 'United States, Italy, Romania, United Kingdom'
 'Australia, United States' 'Argentina, Venezuela'
 'United States, United Kingdom, Canada' 'China, Hong Kong' 'Russia'
 'Canada' 'Hong Kong' 'United States, China, Hong Kong'
 'Italy, United States' 'United States, Germany'
 'United Kingdom, Canada, United States' ', South Korea' 'Ireland'
 'India, Nepal' 'New Zealand, Australia, France, United States' 'Italy'
 'Italy, Brazil, Greece' 'Argentina' 'Jordan' 'Colombia'
 'United States, Japan' 'Belgium, United Kingdom'
 'Switzerland, United Kingdom, Australia' 'Israel, United States'
 'Canada, United States' 'Brazil' 'Argentina, Spain' 'Taiwan'
 'United States, Nigeria' 'Bulgaria, United States'
 'Spain, United Kingdom, United States' 'United States, China'
 'United States, France' 'Spain, France, United Kingdom, United States'
 ', France, Algeria' 'Poland' 'Germany'
 'France, Israel, Germany, United States, United Kingdom' 'New Zealand'
 'Saudi Arabia' 'Thailand' 'Indonesia' 'Egypt, Denmark, Germany'
 'United States, Switzerland' 'Hong Kong, Canada, United States'
 'Kuwait, United States' 'France, Canada, United States, Spain'
 'France, Netherlands, Singapore' 'France, Belgium'
 'Ireland, United States, United Kingdom' 'Egypt' 'Malaysia' 'Israel'
 'Australia, New Zealand' 'United Kingdom, Germany' 'Belgium, Netherlands'
 'South Korea, Czech Republic' 'Australia, Germany' 'Vietnam'
 'United Kingdom, Belgium' 'United Kingdom, Australia, United States'
 'France, Japan, United States'
 'United Kingdom, Germany, Spain, United States'
 'United Kingdom, United States, France, Italy'
 'United States, Germany, Canada'
 'United States, France, Italy, United Kingdom'
 'United States, United Kingdom, Germany, Hungary'
 'United States, New Zealand' 'Sweden' 'China' 'Lebanon' 'Romania'
 'Finland, Germany' 'Lebanon, Syria' 'Philippines' 'Iceland' 'Denmark'
 'United States, India' 'Philippines, Singapore, Indonesia'
 'China, United States, Canada' 'Lebanon, United Arab Emirates'
 'Canada, United States, Denmark' 'United Arab Emirates'
 'Mexico, France, Colombia' 'Netherlands' 'Germany, United States, France'
 'United States, Bulgaria'
 'United Kingdom, France, Germany, United States' 'Norway, Denmark'
 'Syria, France, Lebanon, Qatar' 'United States, Czech Republic'
 'Mauritius' 'Canada, South Africa' 'Austria' 'Mexico, Brazil'
 'Germany, France' 'Mexico, United States'
 'United Kingdom, France, Spain, United States' 'United States, Australia'
 'United States, United Kingdom, France' 'United States, Russia'
 'United States, United Kingdom, New Zealand' 'Australia, United Kingdom'
 'Canada, Nigeria, United States'
 'France, United States, United Kingdom, Canada' 'France, United Kingdom'
 'India, United Kingdom' 'Canada, United States, Mexico'
 'United Kingdom, Germany, United States'
 'Czech Republic, United Kingdom, United States' 'China, United Kingdom'
 'Italy, United Kingdom' 'China, Taiwan'
 'United States, Brazil, Japan, Spain, India'
 'United States, China, United Kingdom' 'Cameroon'
 'Lebanon, Palestine, Denmark, Qatar' 'Japan, United States'
 'Uruguay, Germany' 'Egypt, Saudi Arabia'
 'United Kingdom, France, Poland, Germany, United States'
 'Ireland, Switzerland, United Kingdom, France, United States'
 'United Kingdom, South Africa, France'
 'Ireland, United Kingdom, France, Germany' 'Russia, United States'
 'United Kingdom, United States, France' 'United Kingdom,'
 'United States, India, United Kingdom' 'Kenya' 'Spain, Argentina'
 'India, United Kingdom, France, Qatar' 'Belgium, France'
 'Argentina, Chile' 'United States, Thailand' 'Chile, Brazil'
 'United States, Colombia' 'Canada, United States, United Kingdom'
 'Uruguay' 'Luxembourg' 'United States, Cambodia, Romania' 'Bangladesh'
 'Spain, Belgium, United States'
 'United Kingdom, United States, Australia'
 'Canada, United States, France' 'Portugal, United States'
 'Portugal, Spain' 'India, United States' 'United Kingdom, Ireland'
 'United Kingdom, Spain, United States' 'Hungary, United States'
 'United States, South Korea' 'Canada, United States, Cayman Islands'
 'India, France' 'France, Canada' 'Canada, Hungary, United States'
 'Norway' 'Canada, United Kingdom, United States'
 'United Kingdom, Germany, France, United States' 'Denmark, United States'
 'Senegal' 'France, Algeria'
 'United Kingdom, Finland, Germany, United States, Australia, Japan, France, Ireland'
 'Philippines, Canada, United Kingdom, United States'
 'Ireland, France, Iceland, United States, Mexico, Belgium, United Kingdom, Hong Kong'
 'Singapore' 'Kuwait' 'United States, France, Serbia'
 'United States, Italy' 'Spain, Italy'
```

```
'United States, Italy' 'Spain, Italy'
'United States, Ireland, United Kingdom, India'
'United Kingdom, Singapore' 'Hong Kong, United States'
'United States, Malta, France, United Kingdom'
'United States, China, Canada' 'Canada, United States, Ireland'
'Lebanon, Canada, France' 'Japan, Canada, United States'
'Spain, France, Canada' 'Denmark, Singapore, Canada, United States'
'United States, France, Denmark' 'United States, China, Colombia'
'Spain, Thailand, United States' 'Mexico, Spain'
'Ireland, Luxembourg, Belgium' 'China, United States' 'Canada, Belgium'
'Canada, United Kingdom'
'Lebanon, United Arab Emirates, France, Switzerland, Germany'
'France, Belgium, Italy' 'Lebanon, United States, United Arab Emirates'
'Lebanon, France' 'France, Lebanon' 'France, Lebanon, United Kingdom'
'France, Norway, Lebanon, Belgium'
'Sweden, Czech Republic, United Kingdom, Denmark, Netherlands'
'United States, United Kingdom, India' 'Indonesia, Netherlands'
'Turkey, South Korea' 'Serbia, United States' 'Namibia'
'United Kingdom, Kenya' 'United Kingdom, France, Germany, Spain'
'United Kingdom, France, United States, Belgium, Luxembourg, China, Germany'
'Thailand, United States' 'United States, France, Canada, Belgium'
'United Kingdom, China' 'Germany, China, United Kingdom'
'Australia, New Zealand, United States'
'Hong Kong, Iceland, United States' 'France, Australia, Germany'
'United States, Belgium, Canada, France' 'South Africa, Angola'
'United States, Philippines'
'United States, United Kingdom, Canada, China'
'United States, Canada, United Kingdom' 'Turkey, United States'
'Peru, Germany, Norway' 'Mozambique' 'Brazil, France'
'China, Spain, South Korea, United States' 'Spain, Germany'
'Hong Kong, China' 'France, Belgium, Luxembourg, Cambodia,'
'United Kingdom, Australia' 'Belarus' 'Indonesia, United Kingdom'
'Switzerland, France, Belgium, United States' 'Ghana'
'Spain, France, Canada, United States' 'Chile, Italy'
'United Kingdom, Nigeria' 'Chile' 'France, Egypt' 'Egypt, France'
'France, Brazil, Spain, Belgium' 'Egypt, Algeria'
'Canada, South Korea, United States' 'Nigeria, United Kingdom'
'United States, France, Canada' 'Poland, United States'
'United Arab Emirates, Jordan, Lebanon, Saudi Arabia'
'United States, Mexico, Spain, Malta'
'Saudi Arabia, United Arab Emirates' 'Zimbabwe'
'United Kingdom, Germany, United Arab Emirates, New Zealand'
'Romania, United States' 'Canada, Nigeria'
'Saudi Arabia, Netherlands, Germany, Jordan, United Arab Emirates, United States'
'United Kingdom, Spain' 'Finland, France'
'United Kingdom, Germany, United States, France'
'India, United Kingdom, China, Canada, Japan, South Korea, United States'
'Italy, United Kingdom, France' 'United States, Mexico, Colombia'
'Turkey, India' 'Italy, Turkey' 'United Kingdom, United States, Japan'
'France, Belgium, United States' 'Puerto Rico, United States, Colombia'
'Uruguay, Argentina' 'United States, United Kingdom, Japan'
'United States, Argentina' 'United Kingdom, Italy'
'Ireland, United Kingdom'
'United Kingdom, France, Belgium, Canada, United States'
'Netherlands, Germany, Denmark, United Kingdom' 'Hungary'
'Austria, Germany' 'Taiwan, China'
'United Kingdom, United States, Ireland' 'South Korea, United States'
'Brazil, United Kingdom' 'Pakistan, United States'
'Romania, France, Switzerland, Germany' 'Romania, United Kingdom'
'France, Malta, United States' 'Cyprus'
'United Kingdom, France, Belgium, Ireland, United States'
'United States, Norway, Canada' 'Kenya, United States'
'France, South Korea, Japan, United States' 'Taiwan, Malaysia'
'Uruguay, Argentina, Germany, Spain'
'United States, United Kingdom, France, Germany, Japan'
'United States, France, Japan' 'United Kingdom, France, United States'
'Spain, France, United States' 'Indonesia, South Korea, Singapore'
'United States, Spain' 'Netherlands, Germany, Italy, Canada'
'Spain, Germany, Denmark, United States' 'Norway, Sweden'
'South Korea, Canada, United States, China' 'Argentina, Uruguay, Serbia'
'France, Japan' 'Mauritius, South Africa' 'United States, Poland'
'United Kingdom, United States, Germany, Denmark, Belgium, Japan'
'India, Germany' 'India, United Kingdom, Canada, United States'
'Philippines, United States' 'Romania, Bulgaria, Hungary'
'Uruguay, Guatemala' 'France, Senegal, Belgium' 'United Kingdom, Canada'
'Mexico, United States, Spain, Colombia' 'Canada, Norway'
'Singapore, United States' 'Finland, Germany, Belgium'
'United Kingdom, France' 'United States, Chile'
'United Kingdom, Japan, United States' 'Spain, United Kingdom'
'Argentina, United States, Mexico' 'United States, South Korea, Japan'
'Canada, Australia' 'United Kingdom, Hungary, Australia' 'Italy, Belgium'
'United States, United Kingdom, Germany' 'Switzerland'
'Singapore, Malaysia'
'France, Belgium, Luxembourg, Romania, Canada, United States'
'South Africa, Nigeria' 'Spain, France' 'United Kingdom, Hong Kong'
'Pakistan' 'Brazil, United States'
'Denmark, Brazil, France, Portugal, Sweden' 'India, Turkey'
'Malaysia, Singapore, Hong Kong' 'Philippines, Singapore'
'Australia, Canada' 'Taiwan, China, France, United States'
'Germany, Italy' 'Colombia, Peru, United Kingdom'
'Thailand, China, United States' 'Argentina, United States'
'Sweden, United States' 'Uruguay, Spain, Mexico'
'France, Luxembourg, Canada' 'Denmark, Spain' 'Chile, Argentina'
'United Kingdom, Belgium, Sweden' 'Canada, Brazil' 'Italy, France'
'Canada, Germany' 'Pakistan, United Arab Emirates' 'Ghana, United States'
'Mexico, Finland' 'United Arab Emirates, United Kingdom, India'
'Netherlands, Belgium' 'United States, Taiwan'
'Austria, Iraq, United States' 'United Kingdom, Malawi'
'Paraguay, Argentina' 'United Kingdom, Russia, United States'
'India, Pakistan' 'Indonesia, Singapore' 'Spain, Belgium'
'Iceland, Sweden, Belgium' 'Croatia' 'Uruguay, Argentina, Spain'
'United Kingdom, Ireland, United States'
'Canada, Germany, France, United States' 'United Kingdom, Japan'
```

```
 'Norway, Denmark, Netherlands, Sweden' 'Hong Kong, China, United States'
 'Ireland, Canada' 'Italy, Switzerland, France, Germany'
 'Mexico, Netherlands' 'United States, Sweden' 'Germany, France, Russia'
 'France, Iran, United States' 'United Kingdom, India'
 'Russia, Poland, Serbia' 'Spain, Portugal' 'Peru' 'Mexico, Argentina'
 'United Kingdom, Canada, United States, Cayman Islands'
 'Indonesia, United States'
 'United States, Israel, United Kingdom, Canada'
 'Norway, Iceland, United States' 'Czech Republic, United States'
 'United Kingdom, India, United States' 'United Kingdom, West Germany'
 'India, Australia' 'United States,'
 'Belgium, United Kingdom, United States' 'India, Germany, Austria'
 'United States, Brazil, South Korea, Mexico, Japan, Germany'
 'Spain, Mexico' 'China, Japan' 'Argentina, France'
 'China, United States, United Kingdom'
 'France, Luxembourg, United States' 'China, United States, Australia'
 'Colombia, Mexico' 'United States, Canada, Ireland' 'Chile, Peru'
 'Argentina, Italy' 'Canada, Japan, United States'
 'United Kingdom, Canada, United States, Germany'
 'Italy, Switzerland, Albania, Poland' 'United States, Japan, Canada'
 'Cambodia' 'Italy, United States, Argentina'
 'Saudi Arabia, Syria, Egypt, Lebanon, Kuwait'
 'United States, Canada, Indonesia, United Kingdom, China, Singapore'
 'Spain, Colombia'
 'United Kingdom, South Africa, Australia, United States' 'Bulgaria'
 'Argentina, Brazil, France, Poland, Germany, Denmark'
 'United Kingdom, Spain, United States, Germany' 'Philippines, Qatar'
 'Netherlands, Belgium, Germany, Jordan'
 'United Arab Emirates, United States' 'Norway, Germany, Sweden'
 'South Korea, China' 'Georgia' 'Soviet Union, India'
 'Australia, United Arab Emirates' 'Canada, Germany, South Africa'
 'South Korea, China, United States' 'India, Soviet Union' 'India, Mexico'
 'Georgia, Germany, France' 'United Arab Emirates, Romania'
 'India, Malaysia' 'Germany, Jordan, Netherlands'
 'Turkey, France, Germany, Poland' 'Greece, United States'
 'France, United Kingdom, United States' 'Norway, Germany'
 'France, Morocco' 'Cambodia, United States' 'United States, Denmark'
 'United States, Colombia, Mexico'
 'United Kingdom, Italy, Israel, Peru, United States'
 'Argentina, Uruguay, Spain, France'
 'United Kingdom, France, United States, Belgium'
 'France, Canada, China, Cambodia'
 'United Kingdom, France, Belgium, United States' 'Chile, France'
 'Netherlands, United States' 'France, United Kingdom, India'
 'Czech Republic, Slovakia' 'Singapore, France' 'Spain, Switzerland'
 'United States, Australia, China' 'South Africa, United States, Germany'
 'United States, United Kingdom, Australia' 'Spain, Italy, Argentina'
 'Chile, Spain, Argentina, Germany' 'West Germany'
 'Austria, Czech Republic' 'Lebanon, Qatar'
 'United Kingdom, Jordan, Qatar, Iran' 'France, South Korea, Japan'
 'Israel, Germany, France' 'Canada, Japan, Netherlands'
 'United States, Hungary' 'France, Germany' 'France, Qatar'
 'United Kingdom, Germany, Canada' 'Ireland, South Africa'
 'Chile, United States, France' 'Belgium, France, Netherlands'
 'United Kingdom, Ukraine, United States'
 'Germany, Australia, France, China' 'Norway, United States'
 'United States, Bermuda, Ecuador'
 'United States, Hungary, Ireland, Canada'
 'United Kingdom, Egypt, United States'
 'United States, France, United Kingdom' 'Spain, Mexico, France'
 'United States, South Africa' 'Hong Kong, China, Singapore'
 'South Africa, China, United States' 'Denmark, France, Poland'
 'New Zealand, United Kingdom' 'Netherlands, Denmark, South Africa'
 'Iran, France' 'United Kingdom, United States, France, Germany'
 'Australia, France' 'Ireland, United Kingdom, United States'
 'United Kingdom, France, Germany' 'Canada, Luxembourg'
 'Brazil, Netherlands, United States, Colombia, Austria, Germany'
 'France, Canada, Belgium' 'Canada, France'
 'Bulgaria, United States, Spain, Canada' 'Sweden, Netherlands'
 'France, United States, Mexico'
 'Australia, United Kingdom, United Arab Emirates, Canada'
 'Australia, Armenia, Japan, Jordan, Mexico, Mongolia, New Zealand, Philippines, South Africa, Sweden, United States, Uruguay'
 'India, Iran' 'France, Belgium, Spain'
 'Denmark, Sweden, Israel, United States' 'United States, Iceland'
 'United Kingdom, Russia' 'United States, Israel, Italy, South Africa'
 'Netherlands, Denmark, France, Germany' 'South Korea, Japan'
 'United Kingdom, Pakistan' 'France, New Zealand'
 'United Kingdom, Czech Republic, United States, Germany, Bahamas'
 'China, Germany, India, United States' 'Germany, Sri Lanka'
 'United States, India, Bangladesh' 'United States, Canada, France'
 'Brazil, France, Germany' 'Germany, United States, Hong Kong, Singapore'
 'France, Germany, Switzerland'
 'Germany, France, Luxembourg, United Kingdom, United States'
 'United Kingdom, Canada, Italy' 'Czech Republic, France'
 'Taiwan, Hong Kong, United States, China' 'Germany, Australia'
 'United Kingdom, Poland, United States' 'Denmark, Zimbabwe'
 'United Kingdom, South Africa' 'Finland, Sweden, Norway, Latvia, Germany'
 'South Africa, United States, New Zealand, Canada'
 'United States, Italy, United Kingdom, Liechtenstein'
 'Denmark, France, Belgium, Italy, Netherlands, United States, United Kingdom'
 'United States, Australia, Mexico'
 'United Kingdom, Czech Republic, Germany, United States'
 'France, China, Japan, United States' 'United States, South Korea, China'
 'Germany, Belgium' 'Pakistan, Norway, United States'
 'United States, Canada, Belgium, United Kingdom' 'Venezuela'
 'Canada, France, Italy, Morocco, United States' 'Canada, Spain, France'
 'United States, Indonesia' 'Spain, France, Italy'
 'United Arab Emirates, United States, United Kingdom'
 'United Kingdom, Israel, Russia' 'Spain, Cuba' 'United States, Brazil'
 'United States, France, Mexico' 'United States, Nicaragua'
 'United Kingdom, United States, Spain, Germany, Greece, Canada'
 'Italy, Canada, France' 'United Kingdom, Denmark, Canada, Croatia'
 'Italy, Germany' 'United States, France, United Kingdom, Japan'
 'United States, United Kingdom, Denmark, Sweden'
 'United States, United Kingdom, Italy'
 'United States, France, Canada, Spain' 'Russia, United States, China'
 'United States, Canada, Germany' 'Ireland, United States'
 'United States, United Arab Emirates' 'United States, Ireland'
 'Ireland, United Kingdom, Italy, United States' 'Poland,'
 'Slovenia, Croatia, Germany, Czech Republic, Qatar'
 'Canada, United Kingdom, Netherlands' 'United States, Spain, Germany'
 'India, Japan' 'China, South Korea, United States'
 'United Kingdom, France, Belgium' 'Canada, Ireland, United States'
 'United Kingdom, United States, Dominican Republic'
 'United States, Senegal' 'Germany, United Kingdom, United States'
 'South Africa, Germany, Netherlands, France'
 'Canada, United States, United Kingdom, France, Luxembourg'
 'Ireland, United States, France' 'Germany, United States, Canada'
 'United Kingdom, Germany, Canada, United States'
 'United States, France, Canada, Lebanon, Qatar'
 'Netherlands, Belgium, United Kingdom, United States'
 'France, Belgium, China, United States' 'United States, Chile, Israel'
 'United Kingdom, Norway, Denmark, Germany, Sweden'
 'Norway, Denmark, Sweden' 'China, India, Nepal'
 'Colombia, Mexico, United States' 'United Kingdom, South Korea'
 'Denmark, China' 'United States, Greece, Brazil' 'South Korea, France'
 'United States, Australia, Samoa, United Kingdom'
 'Germany, United Kingdom' 'Argentina, Chile, Peru' 'Turkey, Azerbaijan'
 'Poland, West Germany' 'Germany, United States, Sweden' 'Canada, Spain'
 'United States, Cambodia' 'United States, Greece'
 'Norway, United Kingdom, France, Ireland' 'United Kingdom, Poland'
 'Israel, Sweden, Germany, Netherlands' 'Switzerland, France'
 'Italy, India' 'United States, Botswana'
 'Chile, Argentina, France, Spain, United States'
 'United States, India, South Korea, China'
 'Denmark, Germany, Belgium, United Kingdom, France'
 'Denmark, Germany, Belgium, United Kingdom, France, Sweden'
 'France, Switzerland, Spain, United States, United Arab Emirates'
 'Brazil, India, China, United States'
 'Denmark, France, United States, Sweden' 'Australia, Iraq'
 'China, Morocco, Hong Kong' 'Canada, United States, Germany'
 'United Kingdom, Thailand' 'Venezuela, Colombia'
 'Colombia, United States' 'France, Germany, Czech Republic, Belgium'
 'Switzerland, Vatican City, Italy, Germany, France'
 'Portugal, France, Poland, United States'
 'United States, New Zealand, Japan'
 'United States, Netherlands, Japan, France' 'India, Switzerland'
 'Canada, India' 'United States, Morocco' 'Singapore, Japan, France'
 'Canada, Mexico, Germany, South Africa'
 'United Kingdom, United States, Canada'
 'Germany, France, United States, Canada, United Kingdom'
 'United States, Uruguay' 'India, Canada'
 'Ireland, Canada, United Kingdom, United States'
 'United States, Germany, Australia' 'Australia, France, Ireland'
 'Australia, India' 'United States, United Kingdom, Canada, Japan'
 'Sweden, United Kingdom, Finland' 'Hong Kong, Taiwan'
 'United States, United Kingdom, Spain, South Korea' 'Guatemala' 'Ukraine'
 'Italy, South Africa, West Germany, Australia, United States'
 'United States, Germany, United Kingdom, Australia'
 'Italy, France, Switzerland' 'Canada, France, United States'
 'Switzerland, United States' 'Thailand, Canada, United States'
 'China, Hong Kong, United States' 'United Kingdom, New Zealand'
 'Czech Republic, United Kingdom, France'
 'Australia, United Kingdom, Canada' 'Jamaica, United States'
 'Australia, United Kingdom, United States, New Zealand, Italy, France'
 'France, United States, Canada'
 'United Kingdom, France, Canada, Belgium, United States'
 'Denmark, United Kingdom, Sweden' 'United States, Hong Kong'
 'United States, Kazakhstan'
 'Argentina, France, United States, Germany, Qatar'
 'United States, Germany, United Kingdom'
 'United States, Germany, United Kingdom, Italy'
 'United States, New Zealand, United Kingdom' 'Finland, United States'
 'Spain, France, Uruguay' 'France, Canada, United States'
 'United States, Canada, China'
 'Ireland, Canada, Luxembourg, United States, United Kingdom, Philippines, India'
 'United States, Czech Republic, United Kingdom' 'Israel, Germany'
 'Mexico, France'
 'Israel, Germany, Poland, Luxembourg, Belgium, France, United States'
 'Austria, United States' 'United Kingdom, Lithuania'
 'United States, Greece, United Kingdom'
 'United Kingdom, China, United States, India'
 'United States, Sweden, Norway' 'United Kingdom, United States, Morocco'
 'United States, United Kingdom, Morocco' 'Spain, Canada, United States'
 'United States, India, United Arab Emirates'
 'United Kingdom, Canada, France, United States' 'India, Germany, France'
 'Belgium, Ireland, Netherlands, Germany, Afghanistan'
 'France, Canada, Italy, United States, China'
 'Ireland, United Kingdom, Greece, France, Netherlands'
 'Denmark, Indonesia, Finland, Norway, United Kingdom, Israel, France, United States, Germany, Netherlands'
 'New Zealand, United States'
 'United States, Australia, South Africa, United Kingdom'
 'United States, Germany, Mexico'
 'Somalia, Kenya, Sudan, South Africa, United States'
 'United States, Canada, Japan, Panama' 'United Kingdom, Spain, Belgium'
 'Serbia, South Korea, Slovenia'
 'Denmark, United Kingdom, South Africa, Sweden, Belgium'
 'Germany, Canada, United States'
 'Ireland, Canada, United States, United Kingdom'
 'New Zealand, United Kingdom, Australia'
 'United Kingdom, Australia, Canada, United States'
 'Germany, United States, Italy' 'United States, Venezuela'
 'United Kingdom, Canada, Japan'
 'United Kingdom, United States, Czech Republic'
 'United Kingdom, China, United States' 'United Kingdom, Brazil, Germany'
 United Kingdom, China, United States  United Kingdom, Brazil, Germany
 'United Kingdom, Namibia, South Africa, Zimbabwe, United States'
 'Canada, United States, India, United Kingdom'
 'Switzerland, United Kingdom, United States'
 'United Kingdom, India, Sweden'
 'United States, Brazil, India, Uganda, China'
 'Peru, United States, United Kingdom'
 'Germany, United States, United Kingdom, Canada'
 'Canada, India, Thailand, United States, United Arab Emirates'
 'United States, East Germany, West Germany'
 'France, Netherlands, South Africa, Finland'
 'Egypt, Austria, United States' 'Russia, Spain'
 'Croatia, Slovenia, Serbia, Montenegro' 'Japan, Canada'
 'United States, France, South Korea, Indonesia'
 'United Arab Emirates, Jordan']
['September 25, 2021' 'September 24, 2021' 'September 23, 2021' ...
 'December 6, 2018' 'March 9, 2016' 'January 11, 2020']
[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]
['PG-13' 'TV-MA' 'PG' 'TV-14' 'TV-PG' 'TV-Y' 'TV-Y7' 'R' 'TV-G' 'G'
 'NC-17' '74 min' '84 min' '66 min' 'NR' nan 'TV-Y7-FV' 'UR']
['90 min' '2 Seasons' '1 Season' '91 min' '125 min' '9 Seasons' '104 min'
 '127 min' '4 Seasons' '67 min' '94 min' '5 Seasons' '161 min' '61 min'
 '166 min' '147 min' '103 min' '97 min' '106 min' '111 min' '3 Seasons'
 '110 min' '105 min' '96 min' '124 min' '116 min' '98 min' '23 min'
 '115 min' '122 min' '99 min' '88 min' '100 min' '6 Seasons' '102 min'
 '93 min' '95 min' '85 min' '83 min' '113 min' '13 min' '182 min' '48 min'
 '145 min' '87 min' '92 min' '80 min' '117 min' '128 min' '119 min'
 '143 min' '114 min' '118 min' '108 min' '63 min' '121 min' '142 min'
 '154 min' '120 min' '82 min' '109 min' '101 min' '86 min' '229 min'
 '76 min' '89 min' '156 min' '112 min' '107 min' '129 min' '135 min'
 '136 min' '165 min' '150 min' '133 min' '70 min' '84 min' '140 min'
 '78 min' '7 Seasons' '64 min' '59 min' '139 min' '69 min' '148 min'
 '189 min' '141 min' '130 min' '138 min' '81 min' '132 min' '10 Seasons'
 '123 min' '65 min' '68 min' '66 min' '62 min' '74 min' '131 min' '39 min'
 '46 min' '38 min' '8 Seasons' '17 Seasons' '126 min' '155 min' '159 min'
 '137 min' '12 min' '273 min' '36 min' '34 min' '77 min' '60 min' '49 min'
 '58 min' '72 min' '204 min' '212 min' '25 min' '73 min' '29 min' '47 min'
 '32 min' '35 min' '71 min' '149 min' '33 min' '15 min' '54 min' '224 min'
 '162 min' '37 min' '75 min' '79 min' '55 min' '158 min' '164 min'
 '173 min' '181 min' '185 min' '21 min' '24 min' '51 min' '151 min'
 '42 min' '22 min' '134 min' '177 min' '13 Seasons' '52 min' '14 min'
 '53 min' '8 min' '57 min' '28 min' '50 min' '9 min' '26 min' '45 min'
 '171 min' '27 min' '44 min' '146 min' '20 min' '157 min' '17 min'
 '203 min' '41 min' '30 min' '194 min' '15 Seasons' '233 min' '237 min'
 '230 min' '195 min' '253 min' '152 min' '190 min' '160 min' '208 min'
 '180 min' '144 min' '5 min' '174 min' '170 min' '192 min' '209 min'
 '187 min' '172 min' '16 min' '186 min' '11 min' '193 min' '176 min'
 '56 min' '169 min' '40 min' '10 min' '3 min' '168 min' '312 min'
 '153 min' '214 min' '31 min' '163 min' '19 min' '12 Seasons' nan
 '179 min' '11 Seasons' '43 min' '200 min' '196 min' '167 min' '178 min'
 '228 min' '18 min' '205 min' '201 min' '191 min']
['Documentaries' 'International TV Shows, TV Dramas, TV Mysteries'
 'Crime TV Shows, International TV Shows, TV Action & Adventure'
 'Docuseries, Reality TV'
 'International TV Shows, Romantic TV Shows, TV Comedies'
 'TV Dramas, TV Horror, TV Mysteries' 'Children & Family Movies'
 'Dramas, Independent Movies, International Movies'
 'British TV Shows, Reality TV' 'Comedies, Dramas'
 'Crime TV Shows, Docuseries, International TV Shows'
 'Dramas, International Movies' 'Children & Family Movies, Comedies'
 'British TV Shows, Crime TV Shows, Docuseries' 'TV Comedies, TV Dramas'
 'Documentaries, International Movies'
 'Crime TV Shows, Spanish-Language TV Shows, TV Dramas' 'Thrillers'
 'International TV Shows, Spanish-Language TV Shows, TV Action & Adventure'
 'International TV Shows, TV Action & Adventure, TV Dramas'
 'Comedies, International Movies'
 'Comedies, International Movies, Romantic Movies'
 'Docuseries, International TV Shows, Reality TV'
 'Comedies, International Movies, Music & Musicals' 'Comedies'
 'Horror Movies, Sci-Fi & Fantasy' 'TV Comedies'
 'British TV Shows, International TV Shows, TV Comedies'
 'International TV Shows, TV Dramas, TV Thrillers' "Kids' TV"
 'Dramas, International Movies, Thrillers'
 'Action & Adventure, Dramas, International Movies'
 "Kids' TV, TV Comedies" 'Action & Adventure, Dramas'
 "Kids' TV, TV Sci-Fi & Fantasy"
 'Action & Adventure, Classic Movies, Dramas'
 'Dramas, Horror Movies, Thrillers'
 'Action & Adventure, Horror Movies, Thrillers' 'Action & Adventure'
 'Dramas, Thrillers' 'International TV Shows, TV Dramas'
 'International TV Shows, TV Dramas, TV Sci-Fi & Fantasy'
 'Action & Adventure, Anime Features, International Movies' 'Reality TV'
 'Docuseries, International TV Shows'
 'Documentaries, International Movies, Sports Movies'
 'International TV Shows, Reality TV, Romantic TV Shows'
 'British TV Shows, Docuseries, International TV Shows'
 'Anime Series, International TV Shows'
 'Comedies, Dramas, International Movies'
 'Crime TV Shows, TV Comedies, TV Dramas'
 'Action & Adventure, Comedies, Dramas' "Anime Series, Kids' TV"
 'International Movies, Thrillers' "Kids' TV, Korean TV Shows"
 'Documentaries, Sports Movies' 'Sci-Fi & Fantasy, Thrillers'
 'Dramas, International Movies, Romantic Movies'
 'Documentaries, Music & Musicals'
 "Kids' TV, TV Comedies, TV Sci-Fi & Fantasy" "British TV Shows, Kids' TV"
 'Docuseries, Science & Nature TV' 'Children & Family Movies, Dramas'
 "Kids' TV, TV Dramas, Teen TV Shows"
 'Crime TV Shows, International TV Shows, Spanish-Language TV Shows'
```

```
'Docuseries, International TV Shows, Spanish-Language TV Shows' 'Dramas'
'Comedies, Romantic Movies' 'Dramas, Romantic Movies'
'Comedies, Dramas, Independent Movies'
'Crime TV Shows, TV Action & Adventure, TV Comedies'
'Children & Family Movies, Music & Musicals'
'Action & Adventure, Classic Movies, Cult Movies'
'International TV Shows, TV Action & Adventure, TV Comedies'
'Action & Adventure, Sci-Fi & Fantasy' 'Action & Adventure, Comedies'
'Classic Movies, Comedies, Dramas' 'Comedies, Cult Movies'
'Comedies, Cult Movies, Music & Musicals' 'Comedies, Music & Musicals'
'TV Shows' 'Action & Adventure, International Movies'
'Anime Series, International TV Shows, Teen TV Shows'
'Action & Adventure, Children & Family Movies, Cult Movies'
'Comedies, Dramas, Romantic Movies'
'Comedies, Cult Movies, Sci-Fi & Fantasy' 'Classic Movies, Dramas'
'Action & Adventure, Children & Family Movies, Comedies'
'Dramas, Faith & Spirituality' 'Documentaries, LGBTQ Movies'
'Action & Adventure, Classic Movies' 'Docuseries'
'International TV Shows, TV Comedies' 'Dramas, Independent Movies'
'Action & Adventure, Comedies, International Movies'
'International TV Shows, Spanish-Language TV Shows, TV Dramas'
'Crime TV Shows, International TV Shows, TV Dramas'
'Action & Adventure, Horror Movies, International Movies'
'Comedies, International Movies, Sci-Fi & Fantasy'
'Action & Adventure, International Movies, Music & Musicals'
'Dramas, International Movies, Music & Musicals'
'Horror Movies, International Movies' 'Reality TV, Teen TV Shows'
'Crime TV Shows, TV Dramas, TV Mysteries'
'International TV Shows, Reality TV'
'International TV Shows, TV Comedies, TV Dramas'
'Dramas, Independent Movies, Romantic Movies' 'Horror Movies'
'Documentaries, LGBTQ Movies, Sports Movies'
'Horror Movies, International Movies, Thrillers'
'Action & Adventure, Anime Features'
'TV Dramas, TV Mysteries, TV Sci-Fi & Fantasy'
'International TV Shows, Spanish-Language TV Shows, TV Comedies'
'Children & Family Movies, Comedies, Music & Musicals'
'Comedies, Independent Movies'
'Anime Series, International TV Shows, Romantic TV Shows'
'Classic Movies, Dramas, Independent Movies'
'International TV Shows, Romantic TV Shows, Spanish-Language TV Shows'
'International TV Shows, TV Dramas, Teen TV Shows' 'Stand-Up Comedy'
'Action & Adventure, Anime Features, Children & Family Movies'
'International TV Shows, Romantic TV Shows, TV Dramas'
'International Movies, Music & Musicals'
'TV Action & Adventure, TV Dramas, TV Mysteries'
'Horror Movies, Independent Movies, International Movies'
'Comedies, Cult Movies, International Movies'
'Classic Movies, Dramas, International Movies' 'Movies'
'Crime TV Shows, Docuseries'
'Children & Family Movies, Comedies, Sci-Fi & Fantasy'
'Anime Series, International TV Shows, TV Thrillers'
'Action & Adventure, Horror Movies, Sci-Fi & Fantasy'
'Classic Movies, Comedies, Cult Movies' 'TV Dramas, Teen TV Shows'
'Action & Adventure, Sci-Fi & Fantasy, Thrillers'
'Children & Family Movies, Comedies, Dramas' 'Dramas, Sports Movies'
'Action & Adventure, Dramas, Sci-Fi & Fantasy'
'Action & Adventure, Comedies, Cult Movies'
'Dramas, Independent Movies, Thrillers' 'TV Dramas, TV Sci-Fi & Fantasy'
'Action & Adventure, International Movies, Thrillers'
'British TV Shows, International TV Shows, Reality TV'
'TV Action & Adventure, TV Dramas, Teen TV Shows' 'Anime Series'
'Crime TV Shows, TV Action & Adventure, TV Sci-Fi & Fantasy'
'Crime TV Shows, International TV Shows, TV Comedies'
'Stand-Up Comedy & Talk Shows, TV Comedies'
'Classic & Cult TV, TV Action & Adventure, TV Dramas'
'Children & Family Movies, Sports Movies'
'TV Action & Adventure, TV Sci-Fi & Fantasy'
'Anime Series, Stand-Up Comedy & Talk Shows' 'TV Dramas'
'Anime Features, Children & Family Movies, International Movies'
'Classic & Cult TV, Crime TV Shows, International TV Shows'
'Crime TV Shows, International TV Shows, Romantic TV Shows'
'Horror Movies, LGBTQ Movies'
'Action & Adventure, Dramas, Romantic Movies'
'Documentaries, International Movies, Music & Musicals'
'TV Comedies, TV Dramas, Teen TV Shows'
'Children & Family Movies, Comedies, Sports Movies'
'Children & Family Movies, Dramas, International Movies'
'Comedies, Documentaries, International Movies'
'Romantic TV Shows, TV Dramas' 'Anime Series, TV Horror, TV Thrillers'
'International Movies, Romantic Movies'
'TV Action & Adventure, TV Dramas, TV Sci-Fi & Fantasy'
"Kids' TV, Korean TV Shows, TV Comedies"
'British TV Shows, Crime TV Shows, International TV Shows'
'Crime TV Shows, TV Horror, TV Mysteries'
'Docuseries, International TV Shows, Science & Nature TV'
'British TV Shows, International TV Shows, TV Dramas'
"Kids' TV, TV Action & Adventure, TV Sci-Fi & Fantasy"
'International Movies, Romantic Movies, Thrillers'
'Action & Adventure, Cult Movies, International Movies'
'Action & Adventure, Comedies, Sci-Fi & Fantasy'
"International TV Shows, Kids' TV, TV Mysteries"
'Action & Adventure, Thrillers'
'Dramas, Faith & Spirituality, International Movies'
'Action & Adventure, Classic Movies, Comedies'
'Action & Adventure, Comedies, Sports Movies'
'Action & Adventure, Children & Family Movies, Classic Movies'
'Action & Adventure, Children & Family Movies, Dramas'
'Horror Movies, Thrillers' 'Action & Adventure, Romantic Movies'
'Dramas, Romantic Movies, Sci-Fi & Fantasy'
'Dramas, Music & Musicals, Romantic Movies'
'Anime Series, Crime TV Shows, International TV Shows'
```

```
'Reality TV, Romantic TV Shows'
'International Movies, Music & Musicals, Romantic Movies'
'Reality TV, TV Action & Adventure, TV Mysteries'
'Crime TV Shows, TV Dramas'
'International TV Shows, Reality TV, Spanish-Language TV Shows'
'Crime TV Shows, TV Dramas, TV Thrillers' 'British TV Shows, Docuseries'
'International TV Shows, Korean TV Shows, TV Comedies'
'Action & Adventure, Anime Features, Classic Movies'
'TV Action & Adventure, TV Dramas, TV Horror'
'Crime TV Shows, International TV Shows, TV Thrillers'
'Anime Series, Crime TV Shows, TV Horror' 'Anime Features, Documentaries'
'Comedies, Horror Movies'
'International TV Shows, Spanish-Language TV Shows, Stand-Up Comedy & Talk Shows'
'Children & Family Movies, Documentaries, International Movies'
'Romantic TV Shows, TV Comedies, TV Dramas'
'Dramas, Faith & Spirituality, Romantic Movies'
'Dramas, Independent Movies, LGBTQ Movies'
'Comedies, Independent Movies, LGBTQ Movies'
'Action & Adventure, Cult Movies, Sci-Fi & Fantasy'
'Cult Movies, Horror Movies' 'Action & Adventure, Dramas, Sports Movies'
'Anime Series, Romantic TV Shows, Teen TV Shows'
'Dramas, International Movies, LGBTQ Movies'
'Dramas, Romantic Movies, Thrillers'
'Children & Family Movies, Dramas, Faith & Spirituality'
'Dramas, International Movies, Sports Movies'
'Action & Adventure, Horror Movies'
'Documentaries, International Movies, LGBTQ Movies'
'Dramas, Independent Movies, Sci-Fi & Fantasy'
'Comedies, Independent Movies, International Movies'
'Reality TV, TV Horror, TV Thrillers'
'TV Action & Adventure, TV Horror, TV Sci-Fi & Fantasy'
'International TV Shows, TV Horror, TV Sci-Fi & Fantasy'
'Independent Movies, International Movies, Thrillers'
'Independent Movies, Thrillers' 'Documentaries, Dramas'
'Action & Adventure, Sports Movies'
'Dramas, International Movies, Sci-Fi & Fantasy'
'Comedies, Independent Movies, Romantic Movies'
'Horror Movies, Romantic Movies, Sci-Fi & Fantasy'
'International TV Shows, Stand-Up Comedy & Talk Shows'
'Action & Adventure, Anime Features, Horror Movies'
'Cult Movies, Dramas, Music & Musicals' 'TV Dramas, TV Thrillers'
'Crime TV Shows, International TV Shows, Korean TV Shows'
'TV Horror, TV Mysteries, TV Thrillers'
'Comedies, Horror Movies, International Movies'
'Crime TV Shows, Docuseries, TV Mysteries'
'Comedies, International Movies, Sports Movies'
'Classic Movies, Music & Musicals' 'Reality TV, TV Comedies, TV Horror'
'Children & Family Movies, Faith & Spirituality, Music & Musicals'
'International TV Shows, Korean TV Shows, Stand-Up Comedy & Talk Shows'
'Dramas, Music & Musicals'
'Docuseries, Science & Nature TV, TV Action & Adventure'
"British TV Shows, Kids' TV, TV Dramas"
'International TV Shows, Korean TV Shows, Romantic TV Shows'
'Horror Movies, Independent Movies'
"Anime Series, Kids' TV, TV Action & Adventure"
'Comedies, Dramas, Music & Musicals' 'TV Horror, Teen TV Shows'
'Comedies, LGBTQ Movies, Thrillers'
'Docuseries, Reality TV, Science & Nature TV'
'Crime TV Shows, Spanish-Language TV Shows, TV Action & Adventure'
'Romantic TV Shows, Teen TV Shows' 'TV Comedies, Teen TV Shows'
'Romantic TV Shows, TV Dramas, Teen TV Shows'
'Children & Family Movies, Sci-Fi & Fantasy'
'Romantic TV Shows, TV Action & Adventure, TV Dramas'
'Comedies, International Movies, LGBTQ Movies' 'Dramas, Sci-Fi & Fantasy'
"Kids' TV, TV Thrillers"
'TV Action & Adventure, TV Comedies, TV Sci-Fi & Fantasy'
'British TV Shows, Romantic TV Shows, TV Dramas'
'Anime Series, International TV Shows, Spanish-Language TV Shows'
'Docuseries, TV Comedies' 'Comedies, Romantic Movies, Sports Movies'
'TV Action & Adventure, TV Comedies, TV Dramas'
'Children & Family Movies, Dramas, Sports Movies'
'Action & Adventure, Dramas, Independent Movies'
'Spanish-Language TV Shows, TV Dramas' 'Dramas, LGBTQ Movies'
'TV Horror, TV Mysteries, TV Sci-Fi & Fantasy'
'Action & Adventure, Dramas, Faith & Spirituality'
'International TV Shows, TV Mysteries, TV Thrillers'
'British TV Shows, Classic & Cult TV, International TV Shows'
'Action & Adventure, Comedies, Independent Movies' 'Music & Musicals'
"British TV Shows, Kids' TV, TV Comedies"
'Docuseries, Spanish-Language TV Shows'
'Dramas, Independent Movies, Sports Movies'
'TV Dramas, TV Mysteries, TV Thrillers'
'Comedies, LGBTQ Movies, Music & Musicals'
'International TV Shows, TV Action & Adventure, TV Mysteries'
"Kids' TV, TV Comedies, Teen TV Shows"
'International TV Shows, TV Dramas, TV Horror'
'Comedies, International Movies, Thrillers'
'Classic & Cult TV, TV Action & Adventure, TV Sci-Fi & Fantasy'
'International TV Shows, TV Horror, TV Mysteries'
'Children & Family Movies, Documentaries'
'Music & Musicals, Romantic Movies' 'Romantic Movies'
'Children & Family Movies, Classic Movies, Comedies'
'TV Action & Adventure, TV Dramas'
'Dramas, LGBTQ Movies, Romantic Movies'
'Children & Family Movies, Comedies, Romantic Movies'
'Comedies, Sports Movies' 'International Movies'
'International TV Shows, Romantic TV Shows, TV Mysteries'
'Stand-Up Comedy & Talk Shows'
'Action & Adventure, International Movies, Romantic Movies'
'Reality TV, TV Comedies' 'Cult Movies, Dramas, International Movies'
"Kids' TV, TV Dramas"
'Crime TV Shows, International TV Shows, TV Mysteries'
'Action & Adventure, Sci-Fi & Fantasy, Sports Movies'
```

```
 'Action & Adventure, Sci-Fi & Fantasy, Sports Movies'
 'TV Dramas, TV Sci-Fi & Fantasy, TV Thrillers'
 'Romantic TV Shows, TV Dramas, TV Sci-Fi & Fantasy'
 'Docuseries, TV Sci-Fi & Fantasy' 'Anime Features, International Movies'
 "British TV Shows, Classic & Cult TV, Kids' TV"
 'British TV Shows, Reality TV, Romantic TV Shows'
 'Documentaries, Faith & Spirituality, International Movies'
 "Kids' TV, Reality TV, TV Dramas" 'LGBTQ Movies, Thrillers'
 'TV Action & Adventure, TV Mysteries, TV Sci-Fi & Fantasy'
 'Reality TV, Science & Nature TV'
 "Kids' TV, TV Action & Adventure, TV Comedies"
 'International TV Shows, Romantic TV Shows, TV Action & Adventure'
 'Children & Family Movies, Dramas, Independent Movies'
 'Comedies, Music & Musicals, Romantic Movies'
 'International TV Shows, Korean TV Shows, Reality TV'
 'Classic & Cult TV, TV Dramas, TV Sci-Fi & Fantasy'
 'Anime Features, Children & Family Movies'
 'Action & Adventure, International Movies, Sci-Fi & Fantasy'
 'Crime TV Shows, TV Action & Adventure, TV Dramas'
 'Classic & Cult TV, TV Action & Adventure, TV Horror'
 'International TV Shows, Korean TV Shows, TV Dramas'
 'International TV Shows, TV Action & Adventure, TV Horror'
 'Action & Adventure, Comedies, Romantic Movies'
 'International TV Shows, Korean TV Shows, TV Action & Adventure'
 "Classic & Cult TV, Kids' TV, TV Action & Adventure"
 'Anime Series, International TV Shows, TV Horror'
 'International TV Shows, Korean TV Shows, TV Horror'
 'Children & Family Movies, Comedies, International Movies'
 'International Movies, Sci-Fi & Fantasy'
 'International Movies, Sci-Fi & Fantasy, Thrillers'
 'Children & Family Movies, Dramas, Romantic Movies'
 'Anime Series, Romantic TV Shows' 'Comedies, Dramas, LGBTQ Movies'
 'British TV Shows, International TV Shows, TV Action & Adventure'
 'Docuseries, Science & Nature TV, TV Comedies'
 'International TV Shows, Stand-Up Comedy & Talk Shows, TV Comedies'
 'Children & Family Movies, Dramas, Music & Musicals'
 'Action & Adventure, Independent Movies, International Movies'
 'Action & Adventure, Children & Family Movies, Sci-Fi & Fantasy'
 'Horror Movies, Independent Movies, Sci-Fi & Fantasy'
 'TV Dramas, TV Sci-Fi & Fantasy, Teen TV Shows'
 'Anime Features, International Movies, Sci-Fi & Fantasy'
 'Dramas, Independent Movies, Music & Musicals'
 "Kids' TV, TV Comedies, TV Dramas"
 'Children & Family Movies, Documentaries, Sports Movies'
 'Independent Movies, Sci-Fi & Fantasy, Thrillers'
 'Anime Features, Music & Musicals, Sci-Fi & Fantasy'
 'TV Comedies, TV Dramas, TV Sci-Fi & Fantasy'
 'Crime TV Shows, TV Action & Adventure'
 'Comedies, Faith & Spirituality, Romantic Movies'
 "Kids' TV, TV Action & Adventure"
 'Action & Adventure, Independent Movies'
 'International TV Shows, Reality TV, TV Comedies'
 'Docuseries, Reality TV, Teen TV Shows'
 'Crime TV Shows, International TV Shows, Reality TV'
 'Anime Series, Teen TV Shows'
 'Crime TV Shows, Romantic TV Shows, TV Dramas'
 'Anime Features, Romantic Movies'
 'Horror Movies, Sci-Fi & Fantasy, Thrillers'
 'International TV Shows, TV Comedies, TV Sci-Fi & Fantasy'
 'International TV Shows, Romantic TV Shows'
 'Anime Features, Music & Musicals'
 'Anime Features, International Movies, Romantic Movies'
 'International TV Shows, Romantic TV Shows, Teen TV Shows'
 'Docuseries, Stand-Up Comedy & Talk Shows'
 'Horror Movies, Independent Movies, Thrillers'
 'TV Action & Adventure, TV Comedies, TV Horror'
 'Documentaries, Stand-Up Comedy' "Kids' TV, Spanish-Language TV Shows"
 "British TV Shows, Kids' TV, TV Thrillers"
 "Kids' TV, TV Action & Adventure, TV Dramas"
 'Anime Series, Crime TV Shows' 'Dramas, Sci-Fi & Fantasy, Thrillers'
 'TV Comedies, TV Dramas, TV Horror'
 'Children & Family Movies, Comedies, LGBTQ Movies'
 'International TV Shows, TV Action & Adventure, TV Sci-Fi & Fantasy'
 'Docuseries, TV Dramas'
 'Horror Movies, International Movies, Romantic Movies'
 'Crime TV Shows, Docuseries, Science & Nature TV'
 'International Movies, Music & Musicals, Thrillers'
 "Kids' TV, Spanish-Language TV Shows, Teen TV Shows"
 'Comedies, Horror Movies, Independent Movies'
 'Action & Adventure, International Movies, Sports Movies'
 'Action & Adventure, Independent Movies, Sci-Fi & Fantasy'
 'Horror Movies, LGBTQ Movies, Music & Musicals'
 'Comedies, Music & Musicals, Sports Movies'
 'TV Horror, TV Mysteries, Teen TV Shows' 'Romantic TV Shows, TV Comedies'
 "Kids' TV, Reality TV, Science & Nature TV"
 'International Movies, Romantic Movies, Sci-Fi & Fantasy'
 'TV Comedies, TV Horror, TV Thrillers' 'TV Action & Adventure'
 'International TV Shows, Spanish-Language TV Shows, TV Horror'
 'Crime TV Shows, TV Action & Adventure, TV Thrillers'
 'Music & Musicals, Stand-Up Comedy' 'British TV Shows, TV Comedies'
 'TV Comedies, TV Sci-Fi & Fantasy, Teen TV Shows'
 'TV Comedies, TV Sci-Fi & Fantasy'
 'Romantic TV Shows, Spanish-Language TV Shows, TV Comedies'
 'Crime TV Shows, International TV Shows, TV Sci-Fi & Fantasy'
 'British TV Shows, International TV Shows, Romantic TV Shows'
 "Crime TV Shows, Kids' TV"
 'Horror Movies, International Movies, Sci-Fi & Fantasy'
 'TV Comedies, TV Mysteries'
 'Cult Movies, Horror Movies, Independent Movies'
 'British TV Shows, Docuseries, TV Comedies' 'Comedies, Documentaries'
 'Reality TV, Science & Nature TV, TV Action & Adventure'
 'TV Comedies, TV Dramas, TV Mysteries'
 'Crime TV Shows, TV Comedies, Teen TV Shows'
 "Docuseries, Kids' TV, Science & Nature TV"
 'Reality TV, Spanish-Language TV Shows'
 'Action & Adventure, Anime Features, Sci-Fi & Fantasy'
 "Crime TV Shows, Kids' TV, TV Comedies"
 'Dramas, Faith & Spirituality, Independent Movies'
 'Documentaries, Faith & Spirituality'
 'British TV Shows, International TV Shows, Stand-Up Comedy & Talk Shows'
 'Comedies, Dramas, Faith & Spirituality' 'Classic & Cult TV, TV Comedies'
 'Dramas, Romantic Movies, Sports Movies'
 'Stand-Up Comedy & Talk Shows, TV Mysteries, TV Sci-Fi & Fantasy'
 'TV Sci-Fi & Fantasy, TV Thrillers'
 'Comedies, Independent Movies, Music & Musicals'
 'Comedies, Cult Movies, Independent Movies'
 'Documentaries, Dramas, International Movies'
 'British TV Shows, TV Horror, TV Thrillers'
 'British TV Shows, Docuseries, Science & Nature TV'
 'Children & Family Movies, Comedies, Cult Movies' 'Sports Movies'
 'Sci-Fi & Fantasy' 'Comedies, LGBTQ Movies'
 'Comedies, Independent Movies, Thrillers'
 'Classic Movies, Cult Movies, Dramas'
 'British TV Shows, TV Comedies, TV Dramas'
 'Action & Adventure, Children & Family Movies, Independent Movies'
 'Action & Adventure, Documentaries, International Movies'
 'Children & Family Movies, Independent Movies'
 'Comedies, Cult Movies, Dramas'
 'International TV Shows, TV Horror, TV Thrillers'
 'Classic Movies, Thrillers' 'Crime TV Shows, TV Dramas, TV Horror'
 'British TV Shows, Docuseries, Reality TV'
 'Documentaries, LGBTQ Movies, Music & Musicals'
 'Classic Movies, Dramas, Romantic Movies'
 'Crime TV Shows, Romantic TV Shows, Spanish-Language TV Shows'
 'Classic Movies, Cult Movies, Horror Movies'
 'Anime Series, Crime TV Shows, TV Thrillers'
 'Children & Family Movies, Classic Movies'
 'Classic Movies, Comedies, International Movies'
 'Comedies, Sci-Fi & Fantasy' 'Action & Adventure, Cult Movies, Dramas'
 'Documentaries, Faith & Spirituality, Music & Musicals'
 'British TV Shows, Classic & Cult TV, TV Comedies'
 'International Movies, Sports Movies' 'International TV Shows'
 "Classic & Cult TV, Kids' TV, Spanish-Language TV Shows"
 'Romantic TV Shows, Spanish-Language TV Shows, TV Dramas'
 'Children & Family Movies, Comedies, Faith & Spirituality'
 'British TV Shows, Crime TV Shows, TV Dramas'
 'Classic Movies, Dramas, Music & Musicals'
 'Cult Movies, Horror Movies, Thrillers'
 'Action & Adventure, Classic Movies, Sci-Fi & Fantasy'
 'TV Action & Adventure, TV Comedies'
 'Classic Movies, Comedies, Music & Musicals' 'Independent Movies'
 'Documentaries, Horror Movies'
 'Classic & Cult TV, TV Horror, TV Mysteries'
 'Comedies, Faith & Spirituality, International Movies'
 'Dramas, Horror Movies, Sci-Fi & Fantasy'
 'British TV Shows, TV Dramas, TV Sci-Fi & Fantasy'
 'Comedies, Cult Movies, Horror Movies'
 'Comedies, Cult Movies, Sports Movies' 'Classic Movies, Documentaries'
 'Action & Adventure, Faith & Spirituality, Sci-Fi & Fantasy'
 'Action & Adventure, Children & Family Movies'
 'International TV Shows, Reality TV, TV Action & Adventure'
 'Docuseries, Science & Nature TV, TV Dramas' 'Anime Features'
 'Action & Adventure, Horror Movies, Independent Movies'
 'Action & Adventure, Classic Movies, International Movies'
 'Cult Movies, Independent Movies, Thrillers'
 'Crime TV Shows, TV Comedies'
 'Classic Movies, Cult Movies, Documentaries'
 "Classic & Cult TV, Kids' TV, TV Comedies"
 'Classic Movies, Dramas, LGBTQ Movies'
 'Classic Movies, Dramas, Sports Movies' 'Action & Adventure, Cult Movies'
 'Action & Adventure, Comedies, Music & Musicals'
 'Classic Movies, Horror Movies, Thrillers'
 'Classic Movies, Comedies, Independent Movies'
 'Children & Family Movies, Classic Movies, Dramas'
 'Dramas, Faith & Spirituality, Sports Movies'
 'Classic Movies, Comedies, Romantic Movies'
 'Dramas, Horror Movies, Music & Musicals'
 'Classic Movies, Independent Movies, Thrillers'
 'Children & Family Movies, Faith & Spirituality'
 'Classic Movies, Comedies, Sports Movies'
 'Comedies, Dramas, Sports Movies'
 'Action & Adventure, Romantic Movies, Sci-Fi & Fantasy'
 'Classic & Cult TV, TV Sci-Fi & Fantasy'
 'Comedies, Cult Movies, LGBTQ Movies'
 'Comedies, Horror Movies, Sci-Fi & Fantasy'
 'Action & Adventure, Comedies, Horror Movies'
 'Classic & Cult TV, Crime TV Shows, TV Dramas'
 'Action & Adventure, Documentaries, Sports Movies'
 'International Movies, LGBTQ Movies, Romantic Movies'
 'Cult Movies, Dramas, Thrillers']
['As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inventive and comical ways to help them bo
th face the inevitable.'
 'After crossing paths at a party, a Cape Town teen sets out to prove whether a private-school swimming star is her sister who was
abducted at birth.'
 'To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team of robbers are pulled into a violent an
d deadly turf war.'
 ...
 'Looking to survive in a world taken over by zombies, a dorky college student teams with an urban roughneck and a pair of grifter
sisters.'
 'Dragged from civilian life, a former superhero must train a new crop of youthful saviors when the military preps for an attack b
y a familiar villain.'
 "A scrappy but poor boy worms his way into a tycoon's dysfunctional family, while facing his fear of music and the truth about his
past."]
```

Columns

Show_id: Unique ID for every Movie / Tv Show Type: Identifier - A Movie or TV Show Title: Title of the Movie / Tv Show Director: Director of the Movie Cast: Actors involved in the movie/show Country: Country where the movie/show was produced Date_added: Date it was added on Netflix Release_year: Actual Release year of the movie/show Rating: TV Rating of the movie/show Duration: Total Duration - in minutes or number of seasons Listed_in: Genre Description: The summary description

In [ ]:

```

```

In [ ]:

```
# Data frame
nf.shape
```

Out[ ]:

```
(8807, 12)
```

In [ ]:

```
# Rows
print(nf.shape[0])
# Columns
print(nf.shape[1])
```

```
8807
12
```

In [ ]:

```
# Column names
print(nf.columns)
```

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

This data has 8807 rows, 12 columns which are

['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added','release_year', 'rating', 'duration', 'listed_in', 'description']

3. Preparing Data

In [ ]:

```
# Explode Data to make a new line item for each and item in the below columns
```

In [ ]:

```
nf = nf.assign(director = nf["director"].str.split(", ")).explode("director")
nf = nf.assign(cast = nf["cast"].str.split(", ")).explode("cast")
nf = nf.assign(country = nf["country"].str.split(", ")).explode("country")
nf = nf.assign(listed_in = nf["listed_in"].str.split(", ")).explode("listed_in")
```

In [ ]:

```
# Reset the index column
nf.reset_index(drop = True, inplace = True)
```

In [ ]:

```
# check the df
nf.head()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...
2	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...
3	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Mysteries	After crossing paths at a party, a Cape Town t...
4	s2	TV Show	Blood & Water	NaN	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...

In [ ]:

```
# df matrix shape
nf.shape
```

Out[ ]:

```
(201991, 12)
```

In [ ]:

```
# df data types
nf.dtypes
```

Out[ ]:

Out[ ]:

```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

In [ ]:

```

```

In [ ]:

```
# df info
nf.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      151348 non-null  object
 4   cast          199845 non-null  object
 5   country       190094 non-null  object
 6   date_added    201833 non-null  object
 7   release_year  201991 non-null  int64 
 8   rating        201924 non-null  object
 9   duration      201988 non-null  object
 10  listed_in     201991 non-null  object
 11  description   201991 non-null  object
dtypes: int64(1), object(11)
memory usage: 18.5+ MB
```

In [ ]:

```
# Change the datatype - date_added
nf["date_added"] = pd.to_datetime(nf["date_added"])
```

In [ ]:

```
# df info
nf.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      151348 non-null  object        
 4   cast          199845 non-null  object        
 5   country       190094 non-null  object        
 6   date_added    201833 non-null  datetime64[ns]
 7   release_year  201991 non-null  int64         
 8   rating        201924 non-null  object        
 9   duration      201988 non-null  object        
 10  listed_in     201991 non-null  object        
 11  description   201991 non-null  object        
dtypes: datetime64[ns](1), int64(1), object(10)
memory usage: 18.5+ MB
```

In [ ]:

```

```

In [ ]:

```
# Split date column - date_added
```

In [ ]:

```
nf.insert(7, "month_added_on_netflix", nf["date_added"].dt.month_name())
nf.head(3)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	September	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...
2	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...

In [ ]:

```
nf.insert(8, "year_added_on_netflix", nf["date_added"].dt.year)
nf.head(3)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	September	2021.0	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...
2	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...

In [ ]:

```
nf
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	duration	listed_in	descripti
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	September	2021.0	2020	PG-13	90 min	Documentaries	As h fath nears t end of li filmn
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2 Seasons	International TV Shows	Aft crossi paths a party Cape To
2	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2 Seasons	TV Dramas	Aft crossi paths a party Cape To
3	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2 Seasons	TV Mysteries	Aft crossi paths a party Cape To
4	s2	TV Show	Blood & Water	NaN	Khosi Ngema	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2 Seasons	International TV Shows	Aft crossi paths a party Cape To
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
201986	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	2019-03-02	March	2019.0	2015	TV-14	111 min	International Movies	A scrap but po boy wor his w into a t
201987	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	2019-03-02	March	2019.0	2015	TV-14	111 min	Music & Musicals	A scrap but po boy wor his w into a t
201988	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2019-03-02	March	2019.0	2015	TV-14	111 min	Dramas	A scrap but po boy wor his w into a t
201989	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2019-03-02	March	2019.0	2015	TV-14	111 min	International Movies	A scrap but po boy wor his w into a t
201990	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2019-03-02	March	2019.0	2015	TV-14	111 min	Music & Musicals	A scrap but po boy wor his w into a t

201991 rows × 14 columns

In [ ]:

In [ ]:

```
# Split date column - duration
```

In [ ]:

```
# Change the data type
nf["duration"] = nf["duration"].astype(str)
```

In [ ]:

```
# df info
nf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201991 entries, 0 to 201990
Data columns (total 14 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                151348 non-null  object
 4   cast                    199845 non-null  object
 5   country                 190094 non-null  object
 6   date_added              201833 non-null  datetime64[ns]
 7   month_added_on_netflix  201833 non-null  object
 8   year_added_on_netflix   201833 non-null  float64
 9   release_year            201991 non-null  int64
 10  rating                  201924 non-null  object
 11  duration                201991 non-null  object
 12  listed_in               201991 non-null  object
 13  description             201991 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(1), object(11)
memory usage: 21.6+ MB
```

In [ ]:

```
def splitdata(x):
  if x == np.NaN:
    return 0
  else:
    y = x.split()[0]
  return y
```

In [ ]:

```
nf.insert(11, "duration_in_minutes", nf[nf["type"]=="Movie"]["duration"].apply(splitdata))
nf.insert(11, "no_of_seasons", nf[nf["type"]=="TV Show"]["duration"].apply(splitdata))
```

In [ ]:

```
nf.head(3)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	no_of_seasons	duration_in_minutes	duratior
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	September	2021.0	2020	PG-13	NaN	90	90 mi
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2	NaN	2 Seasons
2	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2	NaN	2 Seasons

In [ ]:

```

```

In [ ]:

```
# Delete unwanted columns
```

In [ ]:

```
nf.drop(["description","duration"],axis = 1, inplace=True)
nf.head(3)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	no_of_seasons	duration_in_minutes	li
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	September	2021.0	2020	PG-13	NaN	90	Docume
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2	NaN	Interr TV

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	no_of_seasons	duration_in_minutes	li
2	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2	NaN	TV I

In [ ]:

```
# df info
nf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201991 entries, 0 to 201990
Data columns (total 14 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                151348 non-null  object        
 4   cast                    199845 non-null  object        
 5   country                 190094 non-null  object        
 6   date_added              201833 non-null  datetime64[ns]
 7   month_added_on_netflix  201833 non-null  object        
 8   year_added_on_netflix   201833 non-null  float64       
 9   release_year            201991 non-null  int64         
 10  rating                  201924 non-null  object        
 11  no_of_seasons           56148 non-null   object        
 12  duration_in_minutes     145843 non-null  object        
 13  listed_in               201991 non-null  object        
dtypes: datetime64[ns](1), float64(1), int64(1), object(11)
memory usage: 21.6+ MB
```

In [ ]:

```

```

In [ ]:

```
# missing values (imputation)
```

In [ ]:

```
# Column wise null values
nf.isna().sum()
```

Out[ ]:

```
show_id                        0
type                           0
title                          0
director                   50643
cast                        2146
country                    11897
date_added                   158
month_added_on_netflix       158
year_added_on_netflix        158
release_year                   0
rating                        67
no_of_seasons             145843
duration_in_minutes        56148
listed_in                      0
dtype: int64
```

Columns with Nulls

1. **director**
2. **cast**
3. **country**
4. **date_added**
5. **month_added_on_netflix**
6. **year_added_on_netflix**
7. **rating**
8. **no_of_seasons**
9. **duration_in_minutes**

In [ ]:

```
# Delete rows with null values
nf.dropna(subset=["date_added","month_added_on_netflix","year_added_on_netflix","rating"], inplace = True)
```

In [ ]:

```
# Column wise null values
nf.isna().sum()
```

Out[ ]:

```
show_id                        0
type                           0
title                          0
director                   50425
cast                        2146
country                    11894
date_added                     0
month_added_on_netflix         0
year_added_on_netflix          0
release_year                   0
rating                         0
no_of_seasons             145834
no_of_seasons               145854
duration_in_minutes          55932
listed_in                        0
dtype: int64
```

In [ ]:

```

```

In [ ]:

```
# Fill data - no_of_seasons, duration_in_minutes
nf[["no_of_seasons","duration_in_minutes"]] = nf[["no_of_seasons","duration_in_minutes"]].fillna(0)
```

In [ ]:

```
# Column wise null values
nf.isna().sum()
```

Out[ ]:

```
show_id                      0
type                         0
title                        0
director                 50425
cast                      2146
country                  11894
date_added                   0
month_added_on_netflix       0
year_added_on_netflix        0
release_year                 0
rating                       0
no_of_seasons                0
duration_in_minutes          0
listed_in                    0
dtype: int64
```

In [ ]:

```

```

Fill data - director

for i in nf[(nf["director"].isnull())]["release_year"].unique(): for j in nf[(nf["director"].isnull())]["country"].unique(): if j != np.NaN: director_mode = nf[(nf["release_year"]==i) & (nf["country"]==j)]["director"].mode() if len(director_mode) != 0: nf.loc[(nf["release_year"]==i) & (nf["country"]==j), "director"] = nf.loc[(nf["release_year"]==i) & (nf["country"]==j), "director"].fillna(director_mode[0])

In [ ]:

```
nf.head(3)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	no_of_seasons	duration_in_minutes	li
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	September	2021.0	2020	PG-13	0	90	Docume
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2	0	Interr TV
2	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021-09-24	September	2021.0	2021	TV-MA	2	0	TV I

In [ ]:

```
# Column wise null values
nf.isna().sum()
```

Out[ ]:

```
show_id                      0
type                         0
title                        0
director                 50425
cast                      2146
country                  11894
date_added                   0
month_added_on_netflix       0
year_added_on_netflix        0
release_year                 0
rating                       0
no_of_seasons                0
duration_in_minutes          0
listed_in                    0
dtype: int64
```

In [ ]:

```

```

In [ ]:

```
# Fill null - director
nf["director"].fillna("Unknown Director", inplace= True)
```

In [ ]:

```
# Column wise null values
# Column wise null values
nf.isna().sum()
```

Out[ ]:

```
show_id                      0
type                         0
title                        0
director                     0
cast                      2146
country                  11894
date_added                   0
month_added_on_netflix       0
year_added_on_netflix        0
release_year                 0
rating                       0
no_of_seasons                0
duration_in_minutes          0
listed_in                    0
dtype: int64
```

In [ ]:

In [ ]:

Fill null - country

for i in nf[(nf["country"].isnull())]["director"].unique(): for j in nf[(nf["country"].isnull())]["release_year"].unique(): country_mode = nf[(nf["director"]==i) & (nf["release_year"]==j)]["country"].mode() if len(country_mode) != 0: nf.loc[(nf["director"]==i) & (nf["release_year"]==j), "country"] = nf.loc[(nf["director"]==i) & (nf["release_year"]==j) , "country"].fillna(country_mode[0])

In [ ]:

```
# Column wise null values
nf.isna().sum()
```

Out[ ]:

```
show_id                      0
type                         0
title                        0
director                     0
cast                      2146
country                  11894
date_added                   0
month_added_on_netflix       0
year_added_on_netflix        0
release_year                 0
rating                       0
no_of_seasons                0
duration_in_minutes          0
listed_in                    0
dtype: int64
```

In [ ]:

```
# Fill na - country
nf["country"].fillna("Unknown Country", inplace= True)
```

In [ ]:

```
# Column wise null values
nf.isna().sum()
```

Out[ ]:

```
show_id                     0
type                        0
title                       0
director                    0
cast                     2146
country                     0
date_added                  0
month_added_on_netflix      0
year_added_on_netflix       0
release_year                0
rating                      0
no_of_seasons               0
duration_in_minutes         0
listed_in                   0
dtype: int64
```

In [ ]:

In [ ]:

Fill null - cast

for i in nf[(nf["cast"].isnull())]["country"].unique(): if i != "Unknown Country": for j in nf[(nf["cast"].isnull())]["release_year"].unique(): cast_mode = nf[(nf["country"]==i) & (nf["release_year"]==j)]["cast"].mode() if len(cast_mode) != 0: nf.loc[(nf["country"]==i) & (nf["release_year"]==j), "cast"] = nf.loc[(nf["country"]==i) & (nf["release_year"]==j), "cast"].fillna(cast_mode[0])

In [ ]:

```
# Fill na - cast
nf["cast"].fillna("Unknown Cast", inplace= True)
```

In [ ]:

```

```

In [ ]:

```
# Column wise null values
nf.isna().sum()
```

Out[ ]:

```
show_id                    0
type                       0
title                      0
director                   0
cast                       0
country                    0
date_added                 0
month_added_on_netflix     0
year_added_on_netflix      0
release_year               0
rating                     0
no_of_seasons              0
duration_in_minutes        0
listed_in                  0
dtype: int64
```

In [ ]:

```
# Working with duration_in_minutes column
nf.loc[nf["duration_in_minutes"] == "nan","duration_in_minutes"] = 0
```

In [ ]:

```
# Cleaning data complete
```

In [ ]:

```

```

In [ ]:

```
# Converting datatypes
nf.dtypes
```

Out[ ]:

```
show_id                           object
type                              object
title                             object
director                          object
cast                              object
country                           object
date_added                datetime64[ns]
month_added_on_netflix            object
year_added_on_netflix            float64
release_year                       int64
rating                            object
no_of_seasons                     object
duration_in_minutes               object
listed_in                         object
dtype: object
```

In [ ]:

```
nf = nf.astype({
    "show_id": str,
    "type": str,
    "title": str,
    "director": str,
    "cast": str,
    "country": str,
    "month_added_on_netflix": str,
    "year_added_on_netflix": int,
    "rating": str,
    "no_of_seasons": int,
    "duration_in_minutes": int,
    "listed_in": str})
```

In [ ]:

```

```

In [ ]:

```
nf.tail()
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	no_of_seasons	duration_in_minutes
201986	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	2019-03-02	March	2019	2015	TV-14	0	111
201987	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	2019-03-02	March	2019	2015	TV-14	0	111

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	no_of_seasons	duration_in_minutes
201988	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2019-03-02	March	2019	2015	TV-14	0	111
201989	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2019-03-02	March	2019	2015	TV-14	0	111
201990	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2019-03-02	March	2019	2015	TV-14	0	111

In [ ]:

In [ ]:

```
nf.columns
```

Out[ ]:

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'month_added_on_netflix', 'year_added_on_netflix', 'release_year',
       'rating', 'no_of_seasons', 'duration_in_minutes', 'listed_in'],
      dtype='object')
```

4. Data Analysis

Dataframe

In [ ]:

```
# Dataframe
nf.head(3)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	no_of_seasons	duration_in_minutes	
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown Cast	United States	2021-09-25	September	2021	2020	PG-13	0	90	Docu
1	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata	South Africa	2021-09-24	September	2021	2021	TV-MA	2	0	Int
2	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata	South Africa	2021-09-24	September	2021	2021	TV-MA	2	0	T

In [ ]:

```
# df info
nf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201766 entries, 0 to 201990
Data columns (total 14 columns):
 #   Column                  Non-Null Count   Dtype         
---  ------                  --------------   -----         
 0   show_id                 201766 non-null  object        
 1   type                    201766 non-null  object        
 2   title                   201766 non-null  object        
 3   director                201766 non-null  object        
 4   cast                    201766 non-null  object        
 5   country                 201766 non-null  object        
 6   date_added              201766 non-null  datetime64[ns]
 7   month_added_on_netflix  201766 non-null  object        
 8   year_added_on_netflix   201766 non-null  int64         
 9   release_year            201766 non-null  int64         
 10  rating                  201766 non-null  object        
 11  no_of_seasons           201766 non-null  int64         
 12  duration_in_minutes     201766 non-null  int64         
 13  listed_in               201766 non-null  object        
dtypes: datetime64[ns](1), int64(4), object(9)
memory usage: 23.1+ MB
```

In [ ]:

```
# Entire nf dataframe
print(type(nf))
print(nf.ndim) # dimensions
print()
print(nf.shape) # rows  - 201766 # columns - 14
print(nf.shape[0]) # rows  - 201766
print(nf.shape[1]) # columns - 14
print()
print(nf.size) # tot df elements
print()
print(len(nf)) # rows count
# print(nf.dtype) # AttributeError: 'DataFrame' object has no attribute 'dtype'
```

```
<class 'pandas.core.frame.DataFrame'>
2

(201766, 14)
201766
14

2824724
```

```
201766
```

In [ ]:

Column of Dataframe

In [ ]:

```
# Each Column - type
nf["type"]
```

Out[ ]:

```
0           Movie
1         TV Show
2         TV Show
3         TV Show
4         TV Show
           ...
201986      Movie
201987      Movie
201988      Movie
201989      Movie
201990      Movie
Name: type, Length: 201766, dtype: object
```

In [ ]:

```
# Each Column in
print(type(nf['type'])) # Column - <class 'pandas.core.series.Series'>
print()
print(nf['type'].unique())
print(  type(nf['type'].unique()) ) # Unique text (categorical) - <class 'numpy.ndarray'>
print()
print(nf.groupby('type')['title'].nunique())
print(type(nf.groupby('type')['title'].nunique())) # Column - <class 'pandas.core.series.Series'>
```

```
<class 'pandas.core.series.Series'>

['Movie' 'TV Show']
<class 'numpy.ndarray'>

type
Movie      6129
TV Show    2664
Name: title, dtype: int64
<class 'pandas.core.series.Series'>
```

In [ ]:

Reports

In [ ]:

```
# Report 1A
# Types of content that were released on nf
types_of_content = nf['type'].unique()

print(type(types_of_content))
print(types_of_content)
```

```
<class 'numpy.ndarray'>
['Movie' 'TV Show']
```

In [ ]:

```
print('Types of Content:',types_of_content[0],"&", types_of_content[1])
```

```
Types of Content: Movie & TV Show
```

In [ ]:

In [ ]:

```
# Report 1B
# Count of content that were released on nf
count_type_of_content = nf.groupby('type')['title'].nunique()

print(type(count_type_of_content))
print()
print("Total content count :",count_type_of_content.sum())
print("Total Movies content count :",count_type_of_content.values[0])
print("Total TV Shows content count :",count_type_of_content.values[1])
```

```
<class 'pandas.core.series.Series'>

Total content count : 8793
Total Movies content count : 6129
Total TV Shows content count : 2664
```

In [ ]:

In [ ]:

```python
# Report 2
# When did netflix start uploading content on its platform ?
start_year = nf['year_added_on_netflix'].min()
print("First year content added :",start_year)
```

```
First year content added : 2008
```

In [ ]:

In [ ]:

```python
# Report 3
# Latest content uploaded on its platform ?
latest_year = nf['year_added_on_netflix'].max()
print("Latest year content added :",latest_year)
```

```
Latest year content added : 2021
```

In [ ]:

In [ ]:

```python
# Save df
nf.to_csv('latest_nf.csv', sep=',')
```

In [ ]:

```
!ls
```

```
latest_nf.csv  netflix.csv  sample_data
```

In [ ]:

In [ ]:

```python
# Report 4
# Content released before 2008
content_rlsd_b4_2008 = nf[nf["release_year"] < min(nf["year_added_on_netflix"])]["title"].nunique()
movies_rlsd_b4_2008 = nf[(nf["release_year"] < min(nf["year_added_on_netflix"])) & (nf["type"] == "Movie")]["title"].nunique()
tvshow_rsld_b4_2008 = nf[(nf["release_year"] < min(nf["year_added_on_netflix"])) & (nf["type"] == "TV Show")]["title"].nunique()
print("content_rlsd_befor_2008:", content_rlsd_b4_2008)
print("movies _rlsd_b4_2008:", movies_rlsd_b4_2008)
print("tv-shows_rlsd_befor_2008:", tvshow_rsld_b4_2008)
```

```
content_rlsd_befor_2008: 1045
movies _rlsd_b4_2008: 922
tv-shows_rlsd_befor_2008: 123
```

In [ ]:

In [ ]:

```python
# Report 5
# How many directors content on nf till date ?
# How many Movie and how many TV directors ?
```

In [ ]:

```python
directors = nf.groupby('type')['director'].nunique().sort_values(ascending=False)
print('Total directors :',directors.sum())
print('Movie directors :', directors[0])
print('TV directors :', directors[1])
```

```
Total directors : 5077
Movie directors : 4777
TV directors : 300
```

In [ ]:

In [ ]:

```python
# Report 6
# How many cast content on nf till date ?
# How many Movie and how many tv cast ?
```

In [ ]:

```python
cast = nf.groupby('type')['cast'].nunique().sort_values(ascending=False)
print('Total cast :',cast.sum())
print('Movie cast :', cast[0])
print('TV cast :', cast[1])
```

```
Total cast : 40747
Movie cast : 25945
TV cast : 14802
```

In [ ]:

```

```

In [ ]:

```
# Report 7
# Top movie director, who rlsd most movies on nf ?
```

In [ ]:

```
tdirm = nf[nf['type'] == 'Movie'].groupby('director')['title'].nunique().sort_values(ascending=False)[1:2]
print('Top director :', tdirm.index[0])
print('# movies :', tdirm.values[0])
```

```
Top director : Rajiv Chilaka
# movies : 22
```

In [ ]:

```

```

In [ ]:

```
# Report 8
# Top movie director, who rlsd most movies on nf ?
```

In [ ]:

```
tdirt = nf[nf['type'] == 'TV Show'].groupby('director')['title'].nunique().sort_values(ascending=False)[1:2]
print('Top director :', tdirt.index[0])
print('# tv shows :', tdirt.values[0])
```

```
Top director : Ken Burns
# tv shows : 3
```

In [ ]:

```

```

In [ ]:

```
# Report 9
# Top movie cast, who rlsd most movies on nf ?
```

In [ ]:

```
tcasm = nf[nf['type'] == 'Movie'].groupby('cast')['title'].nunique().sort_values(ascending=False)
print('Top cast :', tcasm.index[0])
print('# movies acted :', tcasm.values[0])
```

```
Top cast : Unknown Cast
# movies acted : 475
```

In [ ]:

```

```

In [ ]:

```
# Report 10
# Top tv show cast, who rlsd most movies on nf ?
```

In [ ]:

```
tcast = nf[nf['type'] == 'TV Show'].groupby('cast')['title'].nunique().sort_values(ascending=False)
print('Top cast :', tcast.index[0])
print('# tv shows acted :', tcast.values[0])
```

```
Top cast : Unknown Cast
# tv shows acted : 350
```

In [ ]:

```

```

In [ ]:

```
# Report 11
# Which year max movies uploaded on nf ?
# How many ?
```

In [ ]:

```
nf[nf['type'] == 'Movie'].groupby('year_added_on_netflix')['title'].nunique().sort_values(ascending = False)
```

Out[ ]:

```
year_added_on_netflix
2019    1424
2020    1284
2018    1237
2021     993
2017     837
2016     253
2015      56
2014      19
2011      13
2013       6
2012       3
2009       2
```

```
2008       1
2010       1
Name: title, dtype: int64
```

In [ ]:

```
year = nf[nf['type'] == 'Movie'].groupby('year_added_on_netflix')['title'].nunique().sort_values(ascending = False)
```

In [ ]:

```
print('Year with most movie releases :',year.index[0])
print('No. of movies:',year.values[0])
```

```
Year with most movie releases : 2019
No. of movies: 1424
```

In [ ]:

```
type(year)
```

Out[ ]:

```
pandas.core.series.Series
```

In [ ]:

```

```

In [ ]:

```
# Report 12
# Which month max movies uploaded on nf ?
# How many ?
```

In [ ]:

```
nf[nf['type'] == 'Movie'].groupby('month_added_on_netflix')['title'].nunique().sort_values(ascending = False)
```

Out[ ]:

```
month_added_on_netflix
July         565
April        550
December     547
January      545
October      545
March        528
August       519
September    519
November     498
June         492
May          439
February     382
Name: title, dtype: int64
```

In [ ]:

```
month = nf[nf['type'] == 'Movie'].groupby('month_added_on_netflix')['title'].nunique().sort_values(ascending = False)
span = latest_year - start_year
```

In [ ]:

```
print('Month with most movie releases :',month.index[0])
print('No. of movies:',month.values[0])
print('in the span of : ', span, ' years.' )
```

```
Month with most movie releases : July
No. of movies: 565
in the span of :  13  years.
```

In [ ]:

```

```

In [ ]:

```
# Report 13
# Oldest movie on nf ? Name , date & year.
```

Conditions

(nf['release_year'] == min(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie')

Movie name

nf[(nf['release_year'] == min(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie')] ['title']

Date added

nf[(nf['release_year'] == min(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie')] ['date_added']

Release year

nf[(nf['release_year'] == min(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie')] ['release_year']

In [ ]:

```
# Calculations
oldest_movie = nf[ (nf['release_year'] == min(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie') ] ['title'].values[0]
added_on = nf[ (nf['release_year'] == min(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie') ] ['date_added'].dt.date.values[0]
original_rel_yr = nf[ (nf['release_year'] == min(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie') ] ['release_year'].values[0]
```

In [ ]:

```
print('Oldest movie on nf: ', oldest_movie)
print('Added in nf on :', added_on)
print('Film originally released in year :', original_rel_yr)
```

```
Oldest movie on nf:  Prelude to War
Added in nf on : 2017-03-31
Film originally released in year : 1942
```

In [ ]:

In [ ]:

```
# Report 14
# Latest movie on nf ? Name , date & year.
```

In [ ]:

```
# Calculations
latest_movie = nf[ (nf['release_year'] == max(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie') ] ['title'].values[0]
added_on = nf[ (nf['release_year'] == max(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie') ] ['date_added'].dt.date.values[0]
original_rel_yr = nf[ (nf['release_year'] == max(nf[nf['type'] == 'Movie']['release_year'])) & (nf['type'] == 'Movie') ] ['release_year'].values[0]
```

In [ ]:

```
print('Latest movie on nf: ', latest_movie)
print('Added in nf on :', added_on)
print('Film originally released in year :', original_rel_yr)
```

```
Latest movie on nf:  My Little Pony: A New Generation
Added in nf on : 2021-09-24
Film originally released in year : 2021
```

In [ ]:

In [ ]:

```
# Report 15
# Which year max tv shows uploaded on nf ?
# How many ?
```

In [ ]:

```
nf[nf['type'] == 'TV Show'].groupby('year_added_on_netflix')['title'].nunique().sort_values(ascending = False)
```

Out[ ]:

```
year_added_on_netflix
2020    595
2019    592
2021    505
2018    411
2017    349
2016    175
2015     26
2013      5
2014      5
2008      1
Name: title, dtype: int64
```

In [ ]:

```
year = nf[nf['type'] == 'TV Show'].groupby('year_added_on_netflix')['title'].nunique().sort_values(ascending = False)
```

In [ ]:

```
print('Year with most tv shows released :',year.index[0])
print('No. of tv shows :',year.values[0])
```

```
Year with most tv shows released : 2020
No. of tv shows : 595
```

In [ ]:

In [ ]:

```
# Report 16
# Which month max tv shows uploaded on nf ?
# How many ?
```

In [ ]:

```
month = nf[nf['type'] == 'TV Show'].groupby('month_added_on_netflix')['title'].nunique().sort_values(ascending = False)
span = latest_year - start_year
```

In [ ]:

```
print('Month with most tv shows released :',month.index[0])
print('No. of tv shows :',month.values[0])
print('in the span of : ', span, ' years.' )
```

```
Month with most tv shows released : December
No. of tv shows : 265
in the span of :  13  years.
```

In [ ]:

```

```

In [ ]:

```
# Report 17
# Oldest tv show on nf ? Name , date & year.
```

In [ ]:

```
# Calculations
oldest_tvshow = nf[ (nf['release_year'] == min(nf[nf['type'] == 'TV Show']['release_year'])) & (nf['type'] == 'TV Show') ] ['title'].values[0]
added_on = nf[ (nf['release_year'] == min(nf[nf['type'] == 'TV Show']['release_year'])) & (nf['type'] == 'TV Show') ] ['date_added'].dt.date.values[0]
original_rel_yr = nf[ (nf['release_year'] == min(nf[nf['type'] == 'TV Show']['release_year'])) & (nf['type'] == 'TV Show') ] ['release_year'].values[0]
```

In [ ]:

```
print('Oldest tv show on nf: ', oldest_tvshow)
print('Added in nf on :', added_on)
print('Show originally released in year :', original_rel_yr)
```

```
Oldest tv show on nf:  Pioneers: First Women Filmmakers*
Added in nf on : 2018-12-30
Show originally released in year : 1925
```

In [ ]:

```

```

In [ ]:

```
# Report 18
# Latest tv show on nf ? Name , date & year.
```

In [ ]:

```
# Calculations
latest_tvshow = nf[ (nf['release_year'] == max(nf[nf['type'] == 'TV Show']['release_year'])) & (nf['type'] == 'TV Show') ] ['title'].values[0]
added_on = nf[ (nf['release_year'] == max(nf[nf['type'] == 'TV Show']['release_year'])) & (nf['type'] == 'TV Show') ] ['date_added'].dt.date.values[0]
original_rel_yr = nf[ (nf['release_year'] == max(nf[nf['type'] == 'TV Show']['release_year'])) & (nf['type'] == 'TV Show') ] ['release_year'].values[0]
```

In [ ]:

```
print('Latest tv show on nf: ', latest_tvshow)
print('Added in nf on :', added_on)
print('Show originally released in year :', original_rel_yr)
```

```
Latest tv show on nf:  Blood & Water
Added in nf on : 2021-09-24
Show originally released in year : 2021
```

In [ ]:

```

```

In [ ]:

```
# Report 19
# First content uploaded in nf ? date ?
```

In [ ]:

```
# Date, Name of, Type of first content release on nf
first_content = nf[nf["date_added"] == min(nf["date_added"])]["date_added"].dt.date
name = nf[nf["date_added"] == min(nf["date_added"])]["title"]
content_type = nf[nf["date_added"] == min(nf["date_added"])]["type"]
```

In [ ]:

```
print("Date first content added on nf:",first_content.values[0])
print("Name first content added on nf:",name.values[0])
print("Type of first content :",content_type.values[0])
```

```
Date first content added on nf: 2008-01-01
Name first content added on nf: To and From New York
Type of first content : Movie
```

In [ ]:

```
 
```

In [ ]:

```
# Report 19
# Latest content uploaded in nf ? date ?
```

In [ ]:

```
# Date, Name of, Type of latest content release on nf
latest_content = nf[nf["date_added"] == max(nf["date_added"])]["date_added"].dt.date
name = nf[nf["date_added"] == max(nf["date_added"])]["title"]
content_type = nf[nf["date_added"] == max(nf["date_added"])]["type"]
```

In [ ]:

```
print("Date latest content added on nf:",latest_content.values[0])
print("Name latest content added on nf:",name.values[0])
print("Type of latest content :",content_type.values[0])
```

```
Date latest content added on nf: 2021-09-25
Name latest content added on nf: Dick Johnson Is Dead
Type of latest content : Movie
```

In [ ]:

```
 
```

In [ ]:

```
# Report 20
# Top 10
```

In [ ]:

```
# Directors
top_10_directors = nf.groupby('director')['title'].nunique().sort_values(ascending = False)[1:11]
```

In [ ]:

```
print("Top 10 Directors by Unique Title:\n",top_10_directors)
```

```
Top 10 Directors by Unique Title:
 director
Rajiv Chilaka          22
Jan Suter              21
Raúl Campos            19
Marcus Raboy           16
Suhas Kadav            16
Jay Karas              15
Cathy Garcia-Molina    13
Jay Chapman            12
Martin Scorsese        12
Youssef Chahine        12
Name: title, dtype: int64
```

In [ ]:

```
# Cast
top_10_cast = nf.groupby('cast')['title'].nunique().sort_values(ascending = False)[1:11]
```

In [ ]:

```
print("Top 10 cast by Unique Title:\n",top_10_cast)
```

```
Top 10 cast by Unique Title:
 cast
Anupam Kher          43
Shah Rukh Khan       35
Julie Tejwani        33
Naseeruddin Shah     32
Takahiro Sakurai     32
Rupa Bhimani         31
Om Puri              30
Akshay Kumar         30
Yuki Kaji            29
Amitabh Bachchan     28
Name: title, dtype: int64
```

In [ ]:

```
# Country
top_10_country = nf.groupby('country')['title'].nunique().sort_values(ascending = False)[0:10]
```

In [ ]:

```
print("Top 10 Countries by Unique Title:\n",top_10_country)
```

```
Top 10 Countries by Unique Title:
 country
United States      3683
India              1046
Unknown Country     829
United Kingdom      803
Canada              445
France              393
Japan               316
Spain               232
South Korea         231
Germany             226
Name: title, dtype: int64
```

```
Name: title, dtype: int64
```

In [ ]:

```
# Genre
top_10_genre = nf.groupby('listed_in')['title'].nunique().sort_values(ascending = False)[0:10]
```

In [ ]:

```
print("Top 10 Genres by Unique Title:\n",top_10_genre)
```

```
Top 10 Genres by Unique Title:
 listed_in
International Movies        2752
Dramas                      2426
Comedies                    1674
International TV Shows      1349
Documentaries                869
Action & Adventure           859
TV Dramas                    762
Independent Movies           756
Children & Family Movies     641
Romantic Movies              616
Name: title, dtype: int64
```

In [ ]:

```
nf
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	month_added_on_netflix	year_added_on_netflix	release_year	rating	no_of_seasons	duration_in_minute
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown Cast	United States	2021-09-25	September	2021	2020	PG-13	0	9
1	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata	South Africa	2021-09-24	September	2021	2021	TV-MA	2	
2	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata	South Africa	2021-09-24	September	2021	2021	TV-MA	2	
3	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata	South Africa	2021-09-24	September	2021	2021	TV-MA	2	
4	s2	TV Show	Blood & Water	Unknown Director	Khosi Ngema	South Africa	2021-09-24	September	2021	2021	TV-MA	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	.
201986	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	2019-03-02	March	2019	2015	TV-14	0	11
201987	s8807	Movie	Zubaan	Mozez Singh	Anita Shabdish	India	2019-03-02	March	2019	2015	TV-14	0	11
201988	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2019-03-02	March	2019	2015	TV-14	0	11
201989	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2019-03-02	March	2019	2015	TV-14	0	11
201990	s8807	Movie	Zubaan	Mozez Singh	Chittaranjan Tripathy	India	2019-03-02	March	2019	2015	TV-14	0	11

201766 rows × 14 columns

In [ ]:

5. Data Visualization

1. Pie Chart

Type of content released most on nf

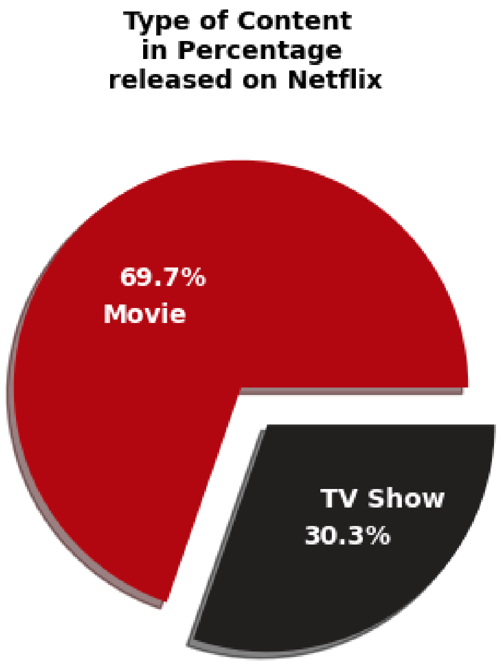
In [ ]:

```
# How does the data look
data = nf.groupby('type')['title'].nunique()

print(data)
print()
print(data.nunique()) # how many unique types
print()
print(data.index) # Unique indexes
print()
print(data.values) # Unique values
```

```
type
Movie      6129
TV Show    2664
Name: title, dtype: int64

2

Index(['Movie', 'TV Show'], dtype='object', name='type')

[6129 2664]
```

In [ ]:

```
# Plot
plt.figure(figsize = (8,4))
nf_r = "#b20710"
nf_g = "#221f1f"

data = nf.groupby('type')['title'].nunique()
index = data.index # ['Movie', 'TV Show']

plt.pie(data.values, labels = index,
        labeldistance = 0.4,
        shadow = True,
        explode = (0,0.2),
        autopct="%.1f%%",
        colors = [nf_r,nf_g],
        textprops ={"color":"white", "fontweight":"bold","fontsize":"medium"})

plt.title("Type of Content \n in Percentage \n released on Netflix", fontsize = 10, fontweight = "bold")
plt.show()
```



In [ ]:

```

```

2. Line Chart

Yearly content released on nf

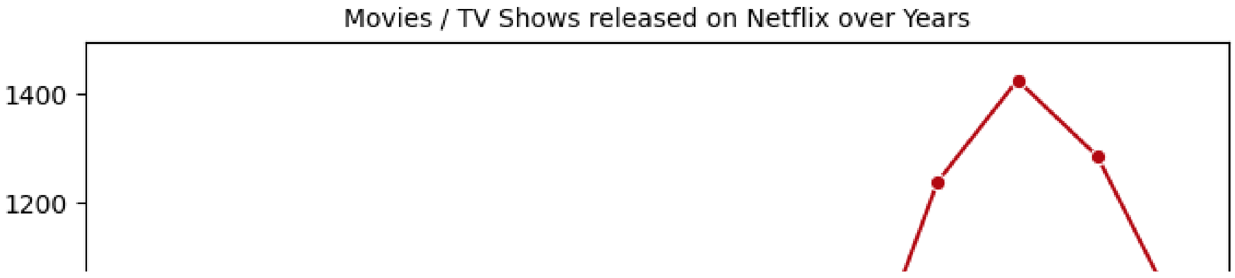
In [ ]:

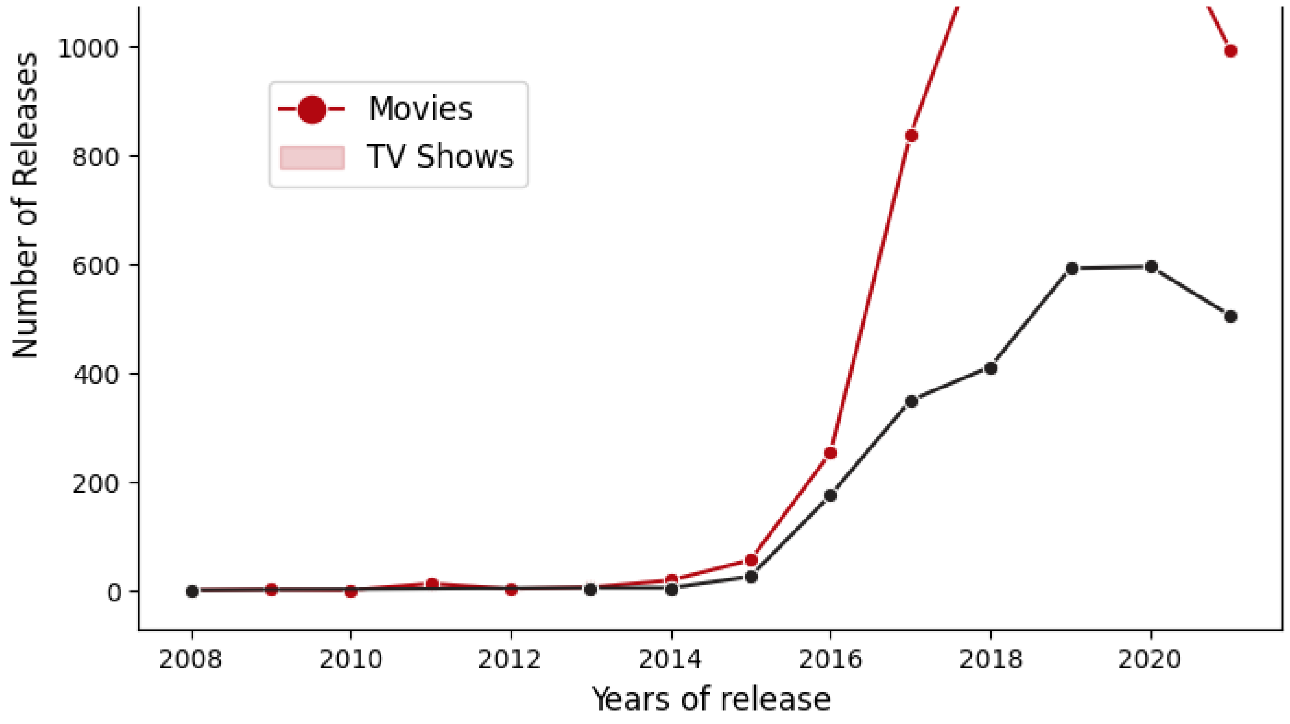
```
# How does the data look

# yearwise_movies_released
m = nf[nf["type"] == "Movie"].groupby("year_added_on_netflix")["title"].nunique()
print(m.index)
print(m.values)
print()
# yearwise_tvshows_released
t = nf[nf["type"] == "TV Show"].groupby("year_added_on_netflix")["title"].nunique()
print(t.index)
print(t.values)
```

```
Int64Index([2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018,
            2019, 2020, 2021],
           dtype='int64', name='year_added_on_netflix')
[   1    2    1   13    3    6   19   56  253  837 1237 1424 1284  993]

Int64Index([2008, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021], dtype='int64', name='year_added_on_netflix')
[  1   5   5  26 175 349 411 592 595 505]
```

In [ ]:

```
plt.figure(figsize = (8,6))
nf_r = "#b20710"
nf_g = "#221f1f"

sns.lineplot(x = m.index, y = m.values, marker="o", color = nf_r);
sns.lineplot(x = t.index, y = t.values, marker="o", color = nf_g);

plt.xlabel("Years of release", fontsize = 12)
plt.ylabel("Number of Releases", fontsize = 12)
plt.legend(["Movies","TV Shows"], loc = "lower left", bbox_to_anchor = (0.1, 0.5), markerscale = 2, fontsize=12)

plt.title("Movies / TV Shows released on Netflix over Years", fontsize = 10)
plt.show()
```





In [ ]:

3. Bar plot

Yearly content released on nf

In [ ]:

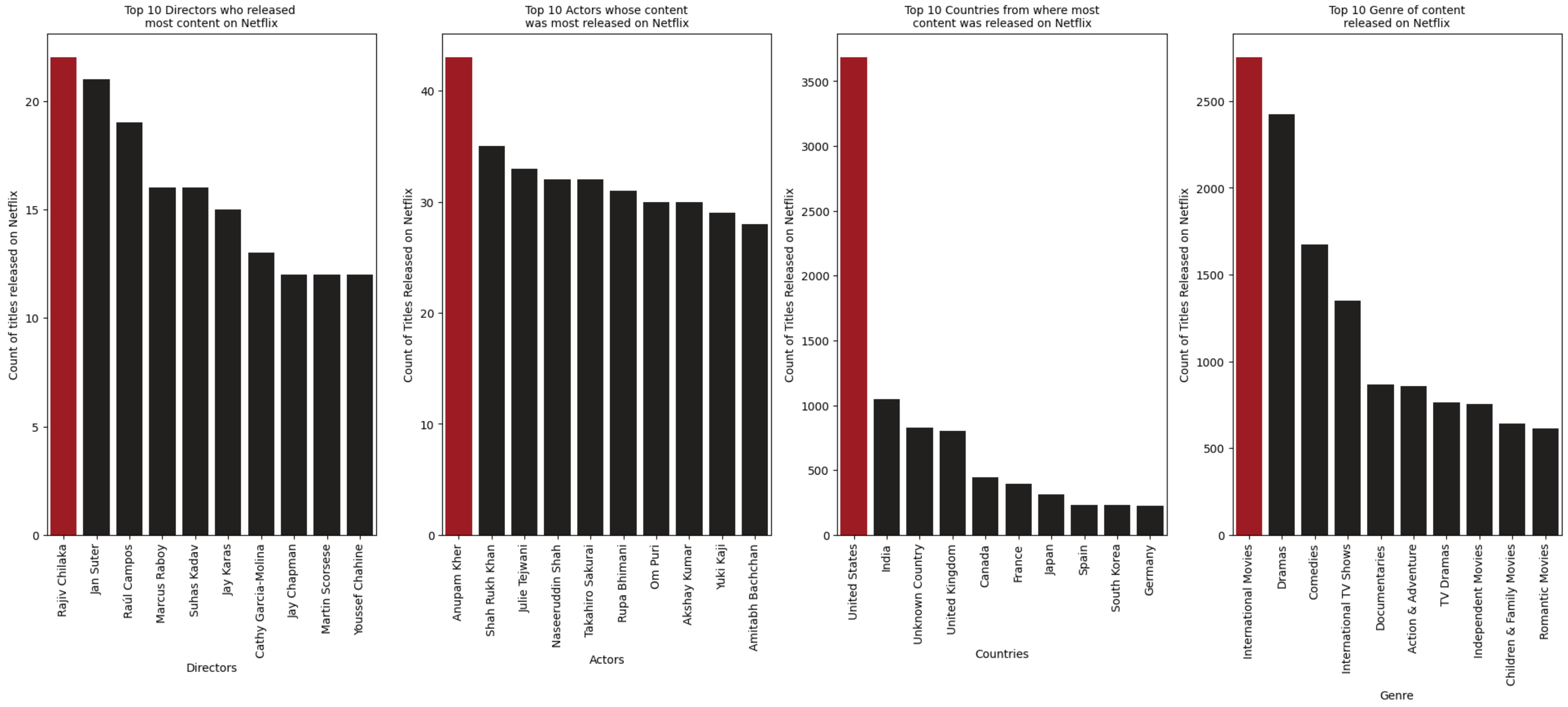
```
top_10_directors
```

Out[ ]:

```
director
Rajiv Chilaka          22
Jan Suter              21
Raúl Campos            19
Marcus Raboy           16
Suhas Kadav            16
Jay Karas              15
Cathy Garcia-Molina    13
Jay Chapman            12
Martin Scorsese        12
Youssef Chahine        12
Name: title, dtype: int64
```

In [ ]:

```
# Plot
plt.figure(figsize = (24,8))
nf_r = "#b20710"
nf_g = "#221f1f"

#------------------------------------------------------------

# Bar Plot for Top 10 Directors
plt.subplot(1,4,1)

top_10_dirs = pd.DataFrame(top_10_directors).reset_index()
ttd_cp = [nf_g if(x < max(top_10_dirs["title"])) else nf_r for x in top_10_dirs["title"]]

sns.barplot(data = top_10_dirs, x = 'director', y = 'title', palette = ttd_cp )

plt.xticks(rotation = 90)
plt.xlabel("Directors")
plt.title("Top 10 Directors who released \nmost content on Netflix", size = 10)
plt.ylabel("Count of titles released on Netflix")

#------------------------------------------------------------

# Bar Plot for Top 10 Actors
plt.subplot(1,4,2)

top_10_actors = pd.DataFrame(top_10_cast).reset_index()
tta_cp= [nf_g if(x < max(top_10_actors["title"])) else nf_r for x in top_10_actors["title"]]

sns.barplot(data = top_10_actors, x = "cast", y = "title", palette = tta_cp )

plt.xticks(rotation = 90)
plt.xlabel("Actors")
plt.title("Top 10 Actors whose content\nwas most released on Netflix", size = 10)
plt.ylabel("Count of Titles Released on Netflix")

#------------------------------------------------------------

# Bar Plot for Top 10 Countries
plt.subplot(1,4,3)

top_10_cntry = pd.DataFrame(top_10_country).reset_index()
ttc_cp = [nf_g if(x < max(top_10_cntry["title"])) else nf_r for x in top_10_cntry["title"]]

sns.barplot(data = top_10_cntry, x = "country", y = "title", palette = ttc_cp )

plt.xticks(rotation = 90)
plt.xlabel("Countries")
plt.title("Top 10 Countries from where most\ncontent was released on Netflix", size = 10)
plt.ylabel("Count of Titles Released on Netflix")

#----------------------------------------------------------

# Bar Plot for Top 10 Genre
plt.subplot(1,4,4)

top_10_gen = pd.DataFrame(top_10_genre).reset_index()
ttg_cp = [nf_g if(x < max(top_10_gen["title"])) else nf_r for x in top_10_gen["title"]]

sns.barplot(data = top_10_gen, x = "listed_in", y = "title", palette = ttg_cp )

plt.title("Top 10 Genre of content\nreleased on Netflix", size = 10
          )
plt.xticks(rotation = 90)
plt.xlabel("Genre")
plt.ylabel("Count of Titles Released on Netflix")

#----------------------------------------------------------

plt.show()
```



In [ ]:

4. Distplot

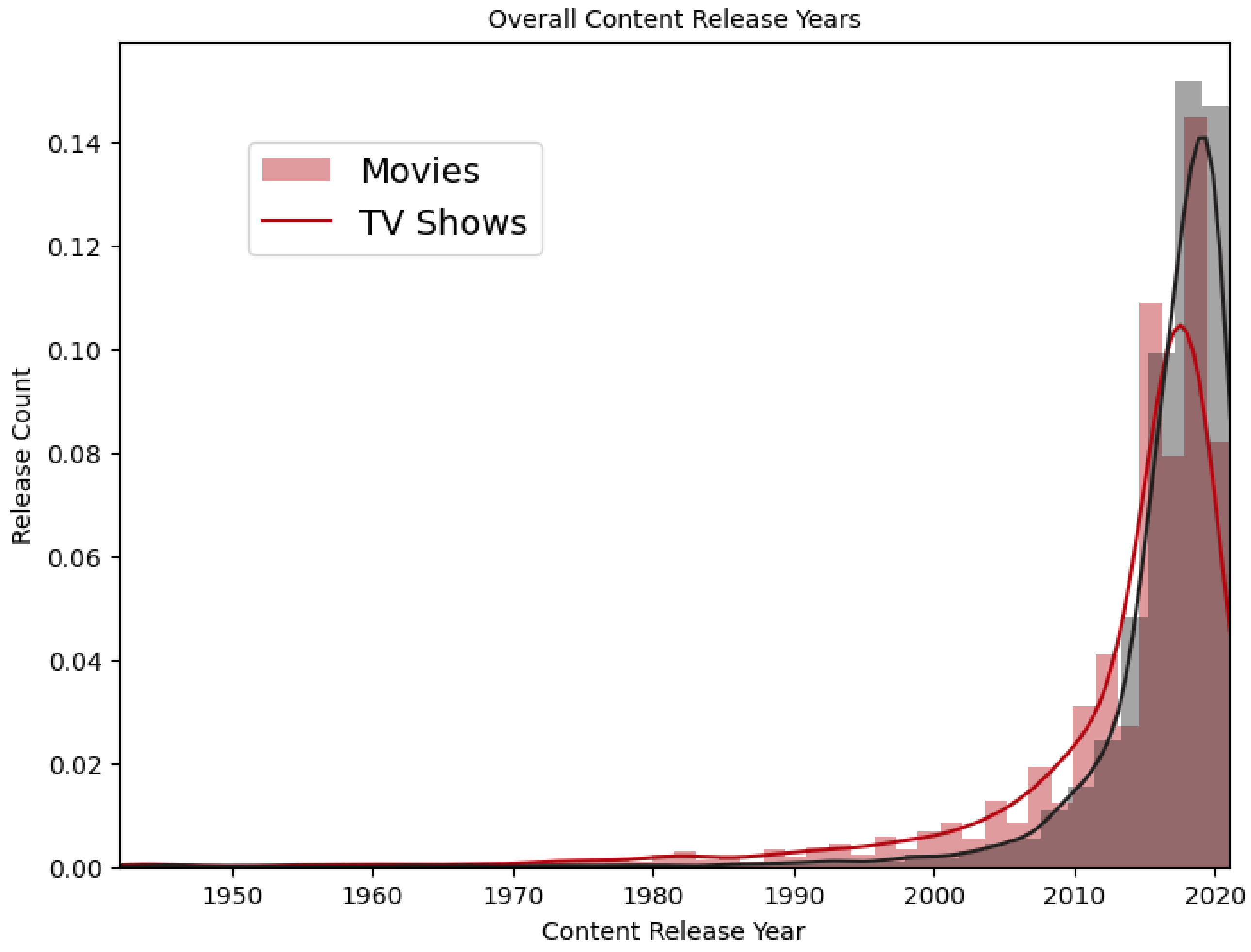
Yearly content released to public not nf

In [ ]:

```
# Plot
m = nf[nf["type"]=="Movie"].drop_duplicates(subset="title") # movie_title_duplicates_removed
t = nf[nf["type"]=="TV Show"].drop_duplicates(subset="title") # tvshow_title_duplicates_removed

max_year = [max(t["release_year"]) if max(t["release_year"]) > max(m["release_year"]) else max(m["release_year"])]
min_year = [min(t["release_year"]) if min(t["release_year"]) > min(m["release_year"]) else min(m["release_year"])]

plt.figure(figsize = (8,6))
nf_r = "#b20710"
nf_g = "#221f1f"

sns.distplot(m["release_year"], color = nf_r, label = "Movies")
sns.distplot(t["release_year"], color = nf_g, label = "TV Shows")

plt.xlim(min_year[0],max_year[0])
plt.legend(["Movies","TV Shows"], loc = "upper left", bbox_to_anchor = (0.1, 0.9), fontsize=14)
plt.xlabel("Content Release Year")
plt.ylabel("Release Count")
plt.title("Overall Content Release Years",fontsize=10)

plt.show()
```

```
<ipython-input-274-4f4a403dcb06>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(m["release_year"], color = nf_r, label = "Movies")
<ipython-input-274-4f4a403dcb06>:13: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(t["release_year"], color = nf_g, label = "TV Shows")
```



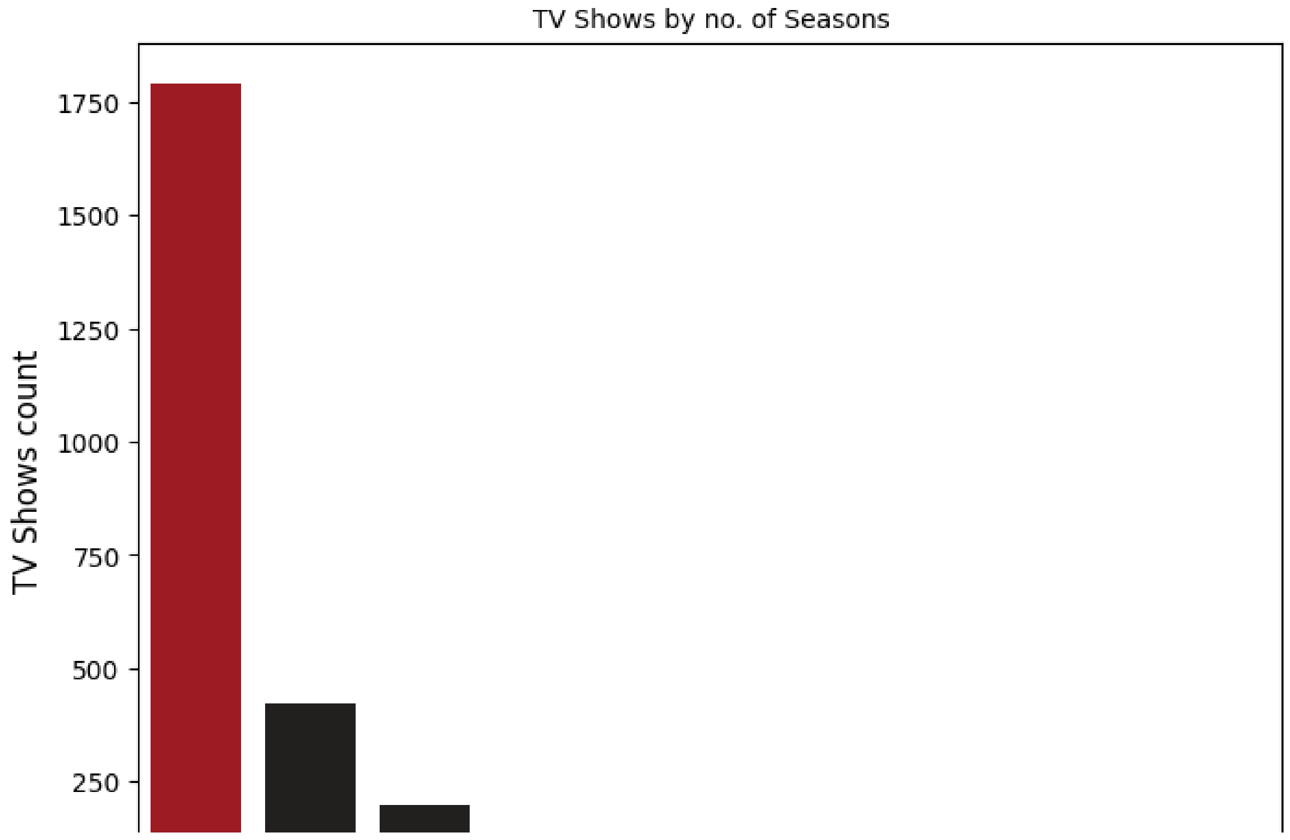
In [ ]:

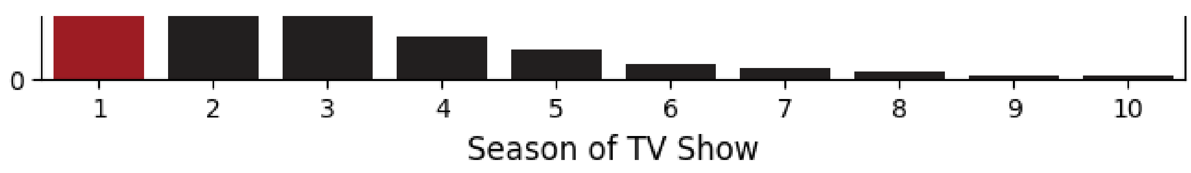
5. Countplot

Count no. of TV Shows by no. of Seasons

In [ ]:

```
# Plot
plt.figure(figsize = (8,6))
nf_r = "#b20710"
nf_g = "#221f1f"

# Count Plot - count number of TV Shows by seasons
t = nf[(nf["no_of_seasons"]!=0) & (nf["type"]=="TV Show") ].drop_duplicates(subset="title") # tvshows_without_duplicate_titles
ts_cp = [nf_g if(x < t["no_of_seasons"].value_counts().values[0]) else nf_r for x in t["no_of_seasons"].value_counts().values] #
topseasons_cp

sns.countplot(data = t, x = "no_of_seasons", palette = ts_cp)

plt.xlim(-0.5,9.5)
plt.xlabel("Season of TV Show", fontsize = 12)
plt.ylabel("TV Shows count", fontsize = 12)
plt.title("TV Shows by no. of Seasons",fontsize=10)

plt.show()
```





In [ ]:

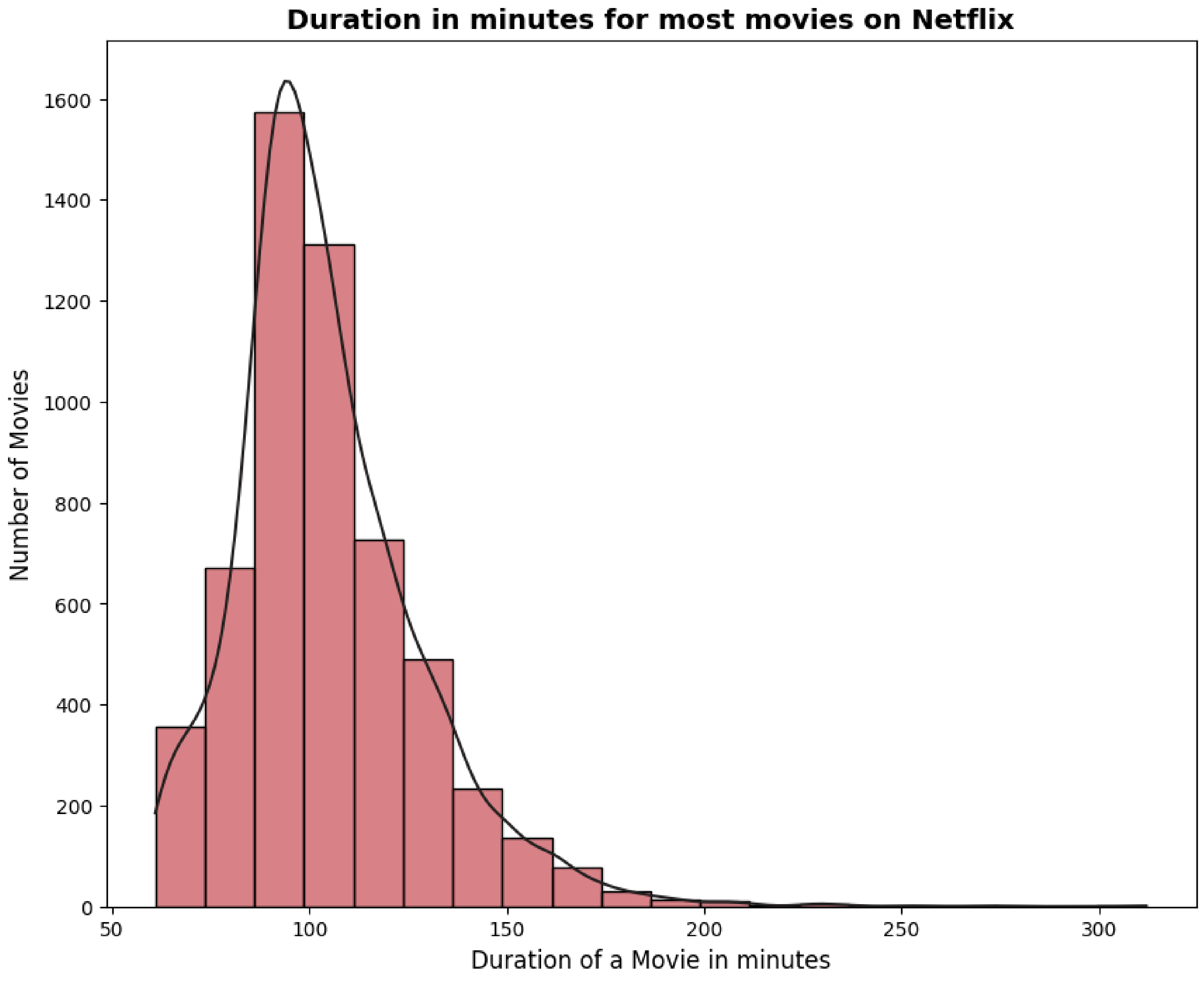
```

```

6. Histogram

What duration movies are most released on nf

In [ ]:

```
# Plot
plt.figure(figsize = (10,8))
nf_r = "#b20710"
nf_g = "#221f1f"

m = nf[(nf["duration_in_minutes"]>60) & (nf["type"] == "Movie")].drop_duplicates(subset = "title") # duration_of_movie
td_cp = [nf_g if(x < m["duration_in_minutes"].value_counts().values[0]) else nf_r for x in m["duration_in_minutes"].value_counts(
).values] # topduration_cp

line = sns.histplot(data = m, x = "duration_in_minutes", bins = 20, kde = True, color = nf_r)
line.lines[0].set_color(nf_g)

plt.xlabel("Duration of a Movie in minutes", fontsize=12)
plt.ylabel("Number of Movies", fontsize=12)
plt.title("Duration in minutes for most movies on Netflix", fontsize = 14, fontweight = "bold")

plt.show()
```



In [ ]:

```

```

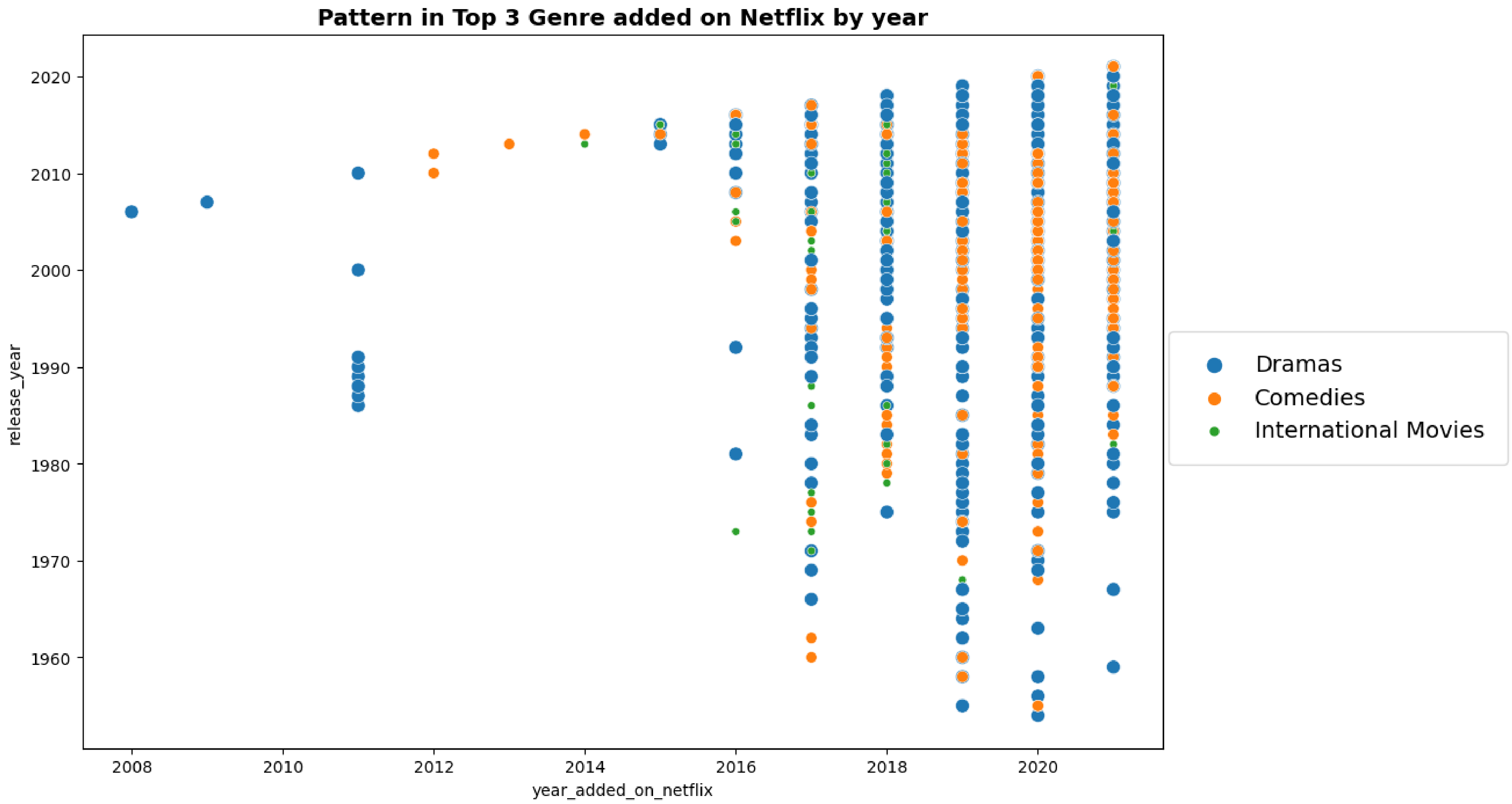
7. Scatter plot

Top 3 genre release pattern by year

In [ ]:

```
# Plot
top_3_genre = nf[(nf["listed_in"].isin(top_10_genre[:3].index))].drop_duplicates(subset="title")

plt.figure(figsize = (12,8))
nf_r = "#b20710"
nf_g = "#221f1f"

sns.scatterplot(data = top_3_genre, x = "year_added_on_netflix", y = "release_year", size = "listed_in", sizes = (25,75), hue="li
sted_in")
```

```
plt.legend(loc = "upper right", bbox_to_anchor = (1.33, 0.6), fontsize=14, borderpad= 1)
plt.title("Pattern in Top 3 Genre added on Netflix by year", fontsize = 14, fontweight = "bold")

plt.show()
```



In [ ]:

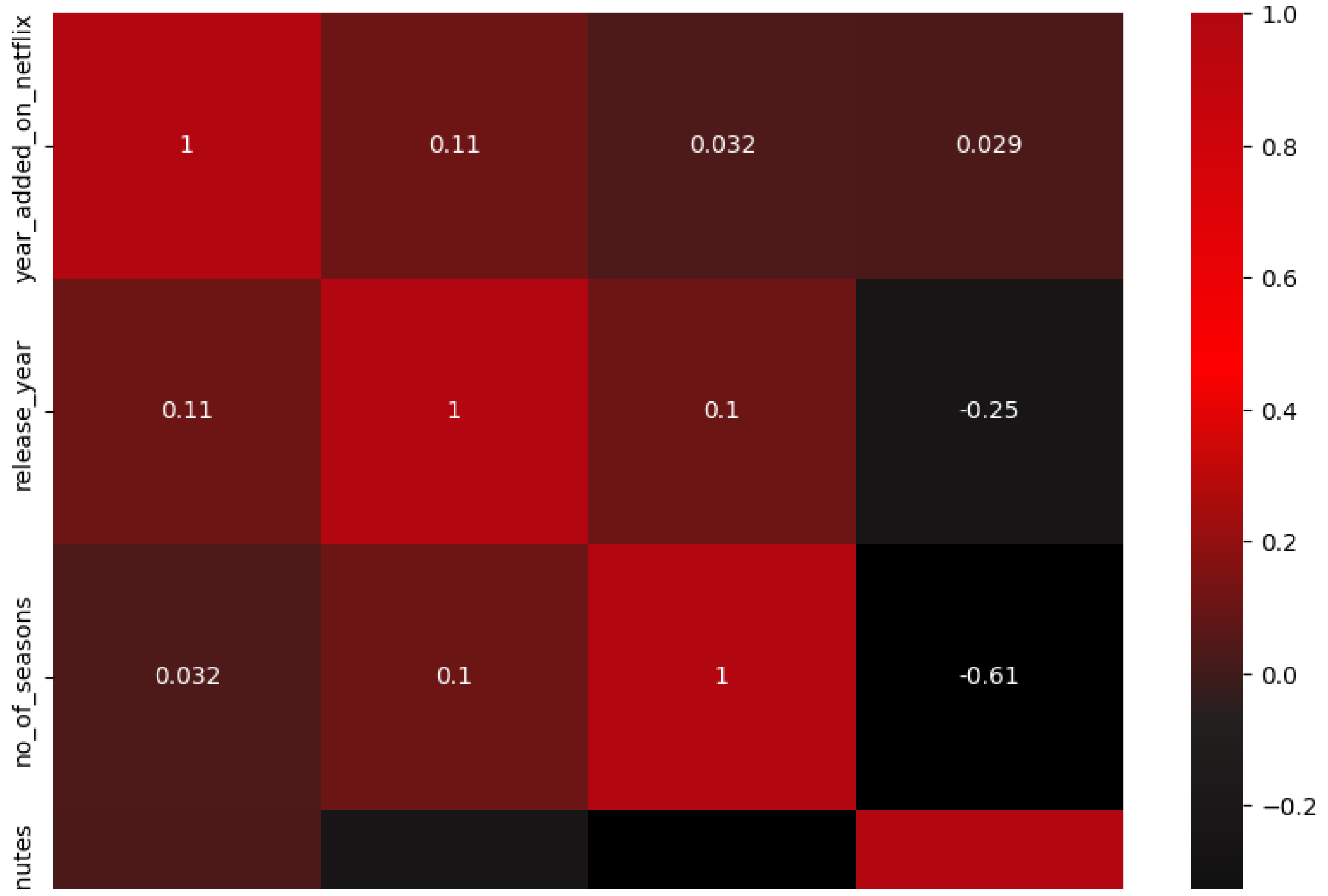
8. Correlation Heat Map

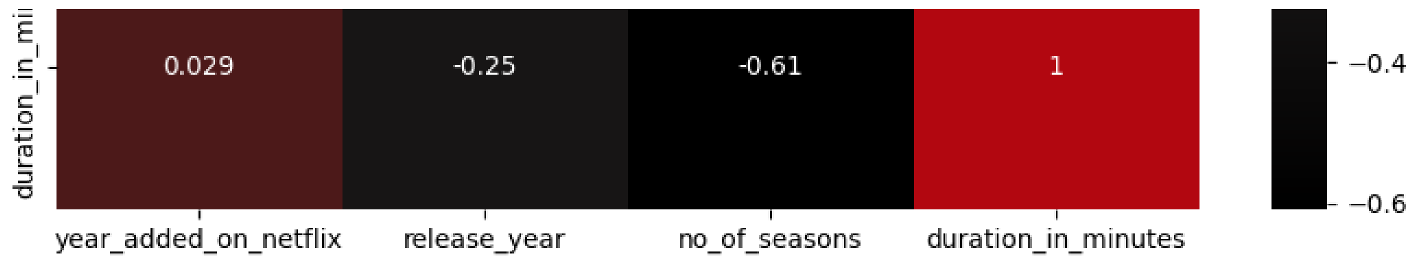
Immediate percantage of relation between the immediate related data

In [ ]:

```
# Plot
nf_no_duplicate = nf.drop_duplicates(subset="title")
netflix_correlation = nf_no_duplicate.corr()

plt.figure(figsize = (10,8))
nf_r = "#b20710"
nf_g = "#221f1f"

cmap = clr.LinearSegmentedColormap.from_list('netflix_colors', ["black",nf_g,"red",nf_r])
sns.heatmap(netflix_correlation, annot=True, cmap = cmap)

plt.show()
```

```
<ipython-input-291-3bc66674110f>:3: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  netflix_correlation = netflix_data_no_duplicate.corr()
```





In [ ]:

In [ ]:

9. Pair plot

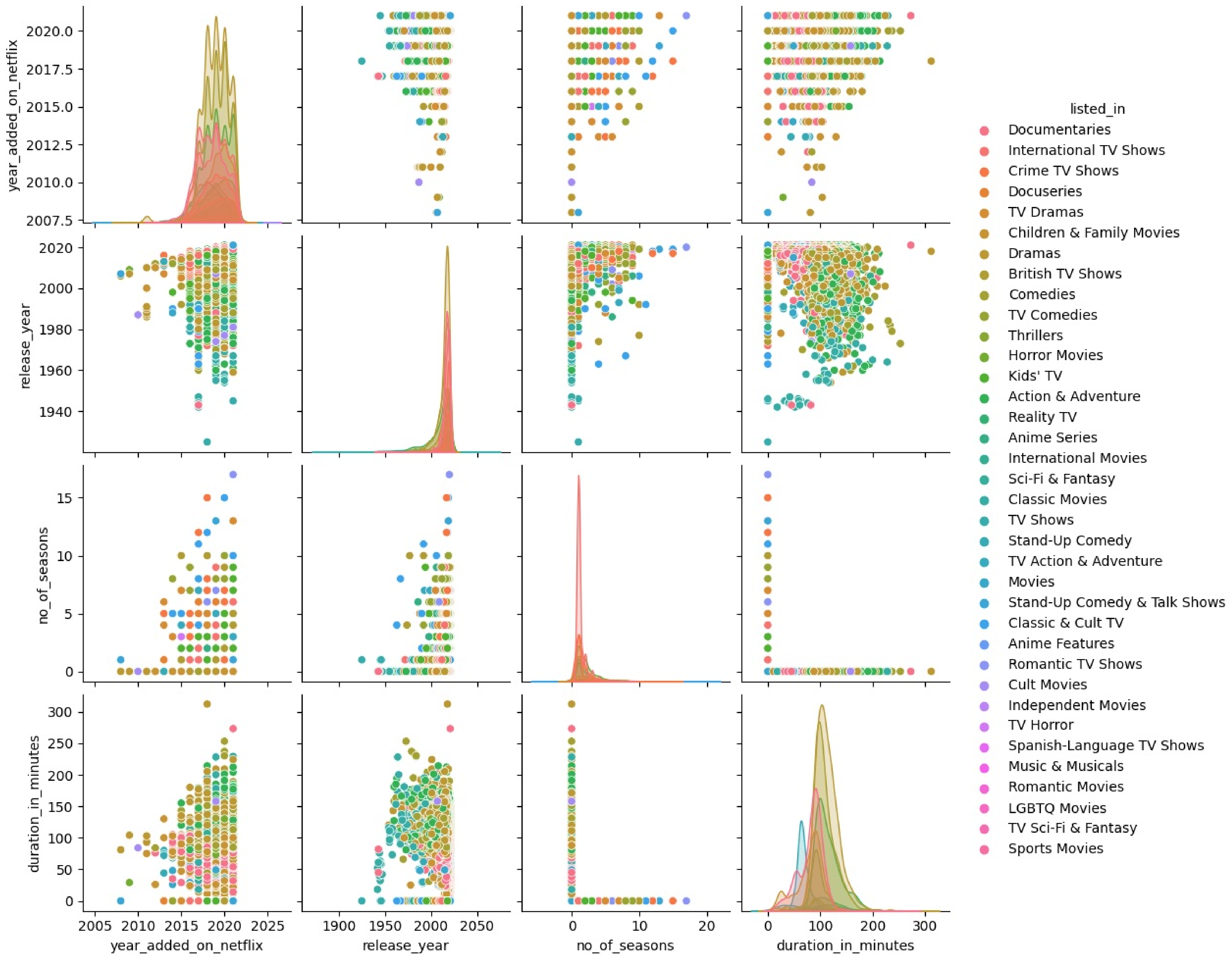
How Genre of both Movies and TV Shows are ploted compared to other columns like :

1. **Year they were added on netflix**
2. **The year in which they originally got released**
3. **no of seasons in case of a TV Show genre**
4. **duration in minutes in case of a movie genre**

In [ ]:

```
plt.figure(figsize=(20,15))
netflix_duplicate_free_data = nf.drop_duplicates(subset = "title")
sns.pairplot(data = netflix_duplicate_free_data, hue = "listed_in")

plt.show()
```

```
<Figure size 2000x1500 with 0 Axes>
```



In [ ]:

How year in which a Movie and TV Shows were added on Netflix are ploted compared to other columns like

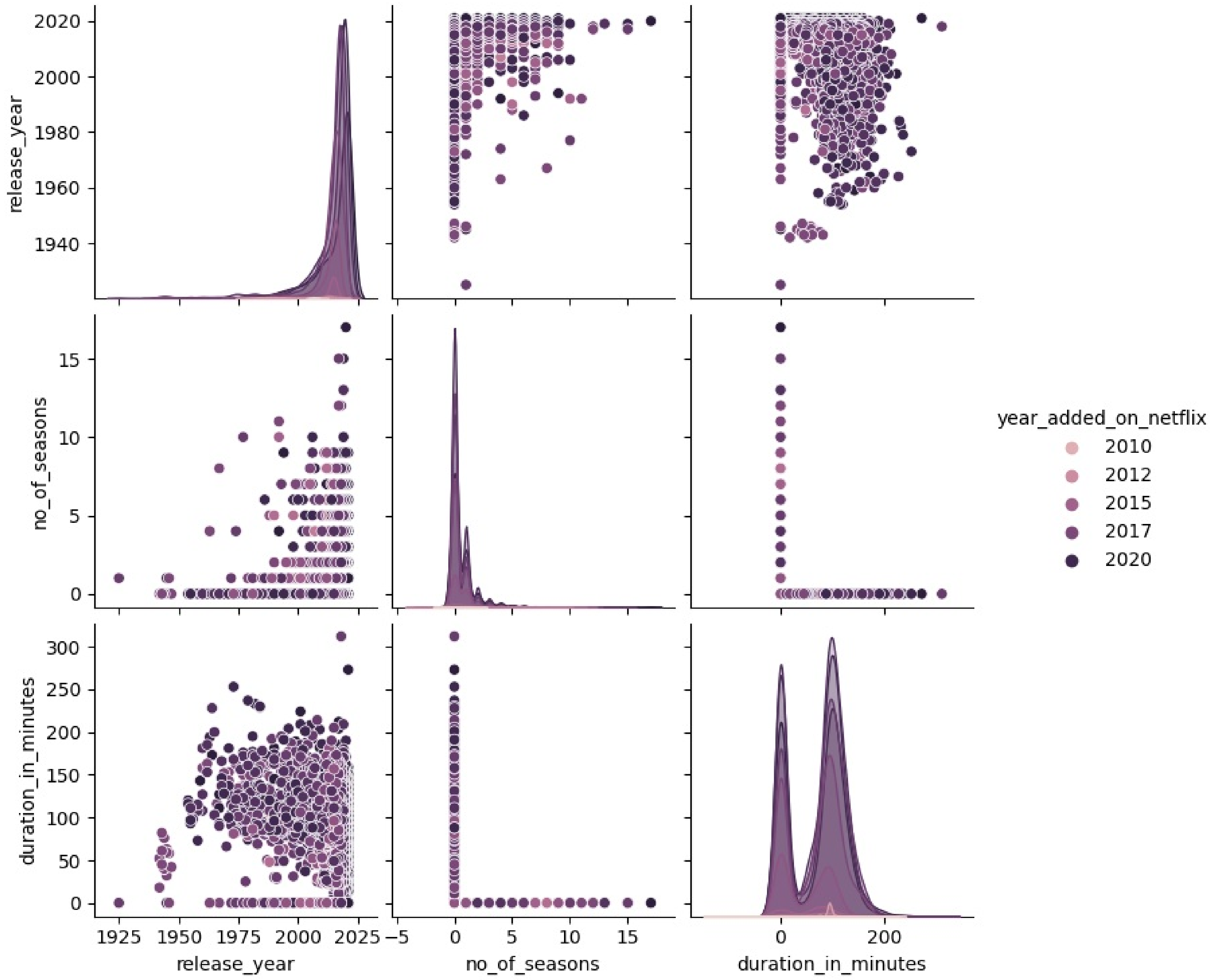
- **The year in which they originally got released**
- **no of seasons in case of a TV Show release year on Netflix**

- no of seasons in case of a TV Show release year on Netflix
- **duration in minutes in case of a movie release year on Netflix**

In [ ]:

```
plt.figure(figsize=(30,20))
netflix_duplicate_free_data = nf.drop_duplicates(subset = "title")
sns.pairplot(data = netflix_duplicate_free_data, hue = "year_added_on_netflix")
plt.show()
```

```
<Figure size 3000x2000 with 0 Axes>
```



In [ ]:

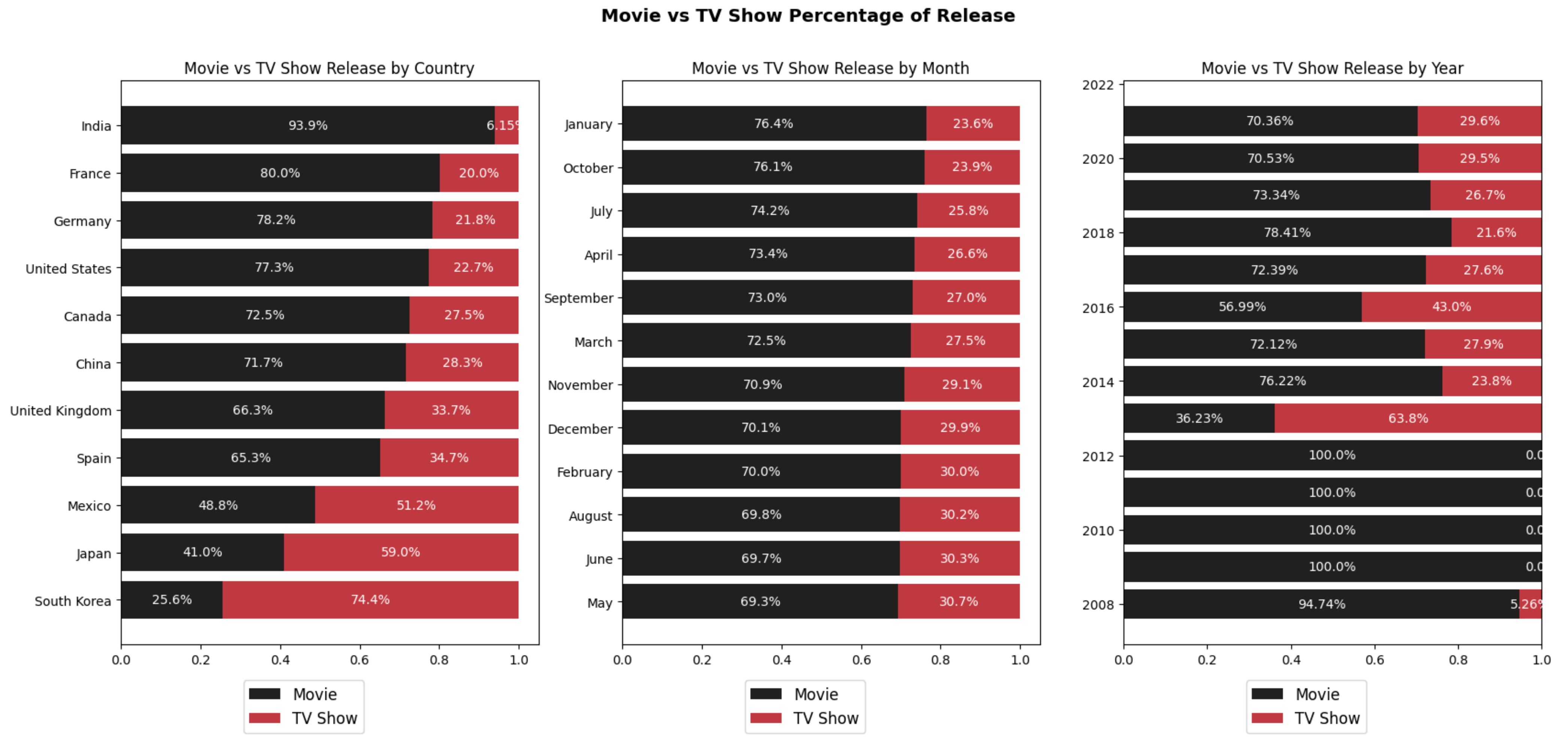
```

```

10. Bar plot Horizontal

In [ ]:

```
# Plot
plt.figure(figsize=(20,8))
nf_r = "#b20710"
nf_g = "#221f1f"

# Top Countries by Movies % release
plt.subplot(1,3,1)
top_countries = nf[nf["country"] != "Unknown Country"]['country'].value_counts()[:11].index

countrywise_release_data = nf[(nf["country"] != "Unknown Country")][['type', 'country']].groupby('country')['type'].value_counts(
).unstack().loc[top_countries]
countrywise_release_data['sum'] = countrywise_release_data.sum(axis=1)
countrywise_release_data_ratio = (countrywise_release_data.T / countrywise_release_data['sum']).T[['Movie', 'TV Show']].sort_value
s(by='Movie',ascending=False)[::-1]

plt.barh(countrywise_release_data_ratio.index, countrywise_release_data_ratio['Movie'], color=nf_g, label='Movie')
plt.barh(countrywise_release_data_ratio.index, countrywise_release_data_ratio['TV Show'], left=countrywise_release_data_ratio['Mov
ie'], color=nf_r, alpha=0.8, label='TV Show')

for i in countrywise_release_data_ratio.index:
  plt.annotate(f"{countrywise_release_data_ratio['Movie'][i]*100:.3}%", xy=(countrywise_release_data_ratio['Movie'][i]/2, i),
               va = 'center', ha='center',fontsize = 10, color='white')
for i in countrywise_release_data_ratio.index:
  plt.annotate(f"{countrywise_release_data_ratio['TV Show'][i]*100:.3}%", xy=(countrywise_release_data_ratio['Movie'][i]+countrywi
se_release_data_ratio['TV Show'][i]/2, i),
                 va = 'center', ha='center',fontsize=10, color='white',)
plt.legend(loc = "upper right", bbox_to_anchor = (0.6, -0.05), fontsize=12)
plt.title("Movie vs TV Show Release by Country", fontsize = 12)

# Top Months by Movies % release
plt.subplot(1,3,2)
top_month_release = nf["month_added_on_netflix"].value_counts().index

monthwise_release_data = nf[["type", "month_added_on_netflix"]].groupby("month_added_on_netflix")["type"].value_counts().unstack(
).loc[top_month_release]
```

```
monthwise_release_data["sum"] = monthwise_release_data.sum(axis=1)
monthwise_release_data_ratio = (monthwise_release_data.T / monthwise_release_data["sum"]).T[["Movie", "TV Show"]].sort_values(by=
"Movie",ascending=False)[::-1]

plt.barh(monthwise_release_data_ratio.index, monthwise_release_data_ratio["Movie"], color=nf_g, label="Movie")
plt.barh(monthwise_release_data_ratio.index, monthwise_release_data_ratio["TV Show"], left=monthwise_release_data_ratio["Movie"],
color=nf_r, alpha=0.8, label="TV Show")

for i in monthwise_release_data_ratio.index:
  plt.annotate(f"{monthwise_release_data_ratio['Movie'][i]*100:.3}%", xy=(monthwise_release_data_ratio["Movie"][i]/2, i),
               va = "center", ha="center",fontsize = 10, color="white")
for i in monthwise_release_data_ratio.index:
  plt.annotate(f"{monthwise_release_data_ratio['TV Show'][i]*100:.3}%", xy=(monthwise_release_data_ratio["Movie"][i]+monthwise_rel
ease_data_ratio["TV Show"][i]/2, i),
                 va = "center", ha="center",fontsize=10, color="white",)
plt.legend(loc = "upper right", bbox_to_anchor = (0.6, -0.05), fontsize=12)
plt.title("Movie vs TV Show Release by Month", fontsize = 12)

# Top Year by Movies % release
plt.subplot(1,3,3)
top_year_release = nf["year_added_on_netflix"].value_counts().index

yearwise_release_data = nf[["type", "year_added_on_netflix"]].groupby("year_added_on_netflix")["type"].value_counts().unstack().l
oc[top_year_release]
yearwise_release_data["sum"] = yearwise_release_data.sum(axis=1)
yearwise_release_data_ratio = (yearwise_release_data.T / yearwise_release_data["sum"]).T[["Movie", "TV Show"]].sort_values(by="Mo
vie",ascending=False)[::-1]
yearwise_release_data_ratio["TV Show"] = yearwise_release_data_ratio["TV Show"].fillna(0)

plt.barh(yearwise_release_data_ratio.index, yearwise_release_data_ratio["Movie"], color=nf_g, label="Movie")
plt.barh(yearwise_release_data_ratio.index, yearwise_release_data_ratio["TV Show"], left=yearwise_release_data_ratio["Movie"], col
or=nf_r, alpha=0.8, label="TV Show")

for i in yearwise_release_data_ratio.index:
  plt.annotate(f"{yearwise_release_data_ratio['Movie'][i]*100:.4}%", xy=(yearwise_release_data_ratio["Movie"][i]/2, i),
               va = "center", ha="center",fontsize = 10, color="white")
for i in yearwise_release_data_ratio.index:
  plt.annotate(f"{yearwise_release_data_ratio['TV Show'][i]*100:.3}%", xy=(yearwise_release_data_ratio["Movie"][i]+yearwise_releas
e_data_ratio["TV Show"][i]/2, i),
                 va = "center", ha="center",fontsize=10, color="white",)
plt.legend(loc = "upper right", bbox_to_anchor = (0.6, -0.05), fontsize=12)
plt.title("Movie vs TV Show Release by Year", fontsize = 12)
plt.suptitle("Movie vs TV Show Percentage of Release", fontsize=14, fontweight ="bold")

plt.show()
```



Looking at the visualisation above we can understand the following:

1. **Movie vs TV Show Release by Top 10 countries: India has released the highest number of movies till date on Netflix. Mexico has released almost equal number of movies and TV shows on Netflix. Taiwan has released most TV shows.**
2. **Movie vs TV Show releases by Month of release on Netflix We can observe that almost every month other than May, June and August more than 70% releases have been movies.**
3. **Movie vs TV show release on Netflix by Year We can see that initial 5 years the content released by Netflix was movies. Netflix started uploading TV shows more from 2013. Since 2018 we can observe that the percent of TV shows have been constantly increase year by year and at the same time the content of upload of movies is reducing at the same level. But having said, it is not a significant change.**

In [ ]:

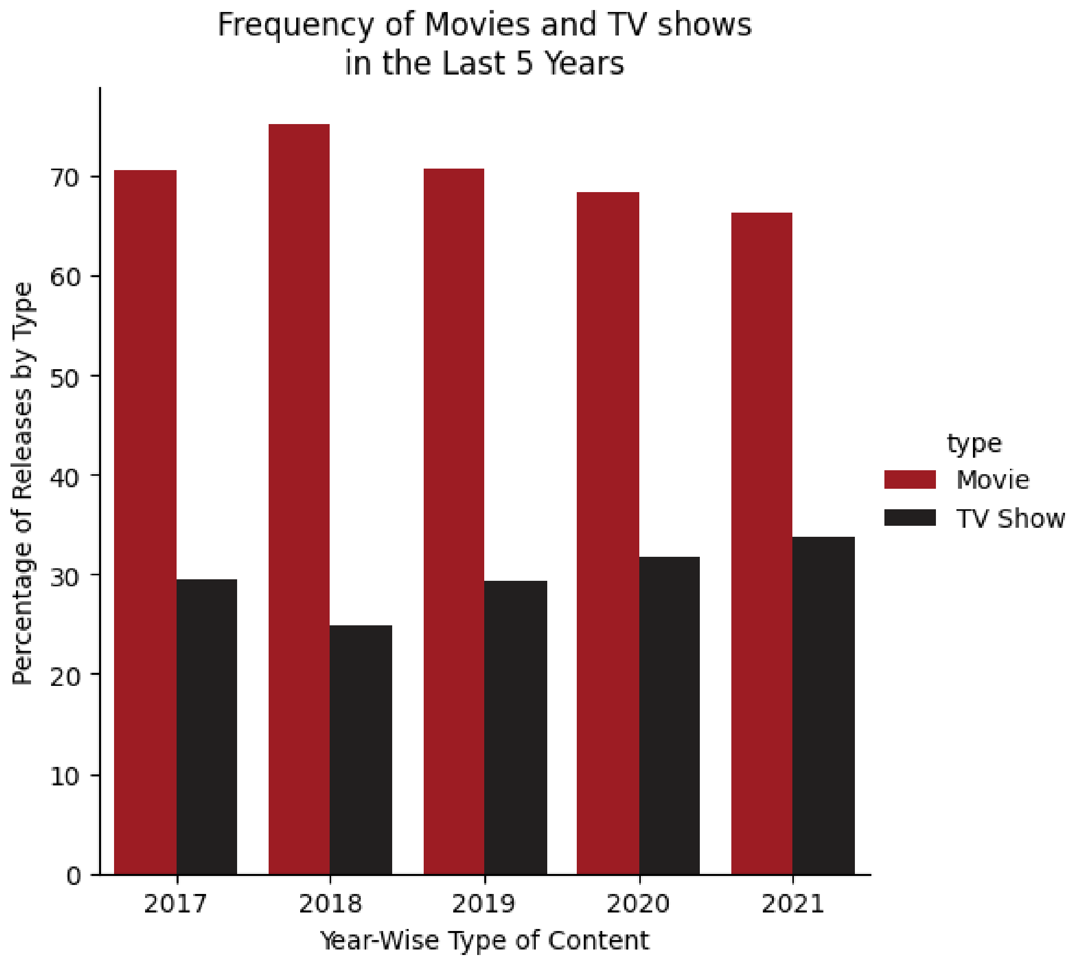
10. Dodge Bar chart

last 5 years content uploade differentiation between Movies and TV Shows

In [ ]:

```
plt.figure(figsize=(10,8))
nf_r = "#b20710"
nf_g = "#221f1f"

# Last 5 Years TV Shows vs Movies released on Netflix
netflix_drop_duplicate = nf.drop_duplicates(subset = "title")
last_5_years_data_by_type = netflix_drop_duplicate[netflix_drop_duplicate["year_added_on_netflix"]>2016].groupby("year_added_on_ne
tflix")["type"].value_counts(normalize = True).mul(100).rename('percent').reset_index().round(2)
dbc = sns.catplot(data = last_5_years_data_by_type, x = "year_added_on_netflix", y = "percent", hue ="type", kind = "bar", legend
= True, palette=[nf_r,nf_g])
dbc.set_axis_labels("Year-Wise Type of Content","Percentage of Releases by Type")
plt.title("Frequency of Movies and TV shows\nin the Last 5 Years")
plt.show()
```

```
<Figure size 1000x800 with 0 Axes>
```

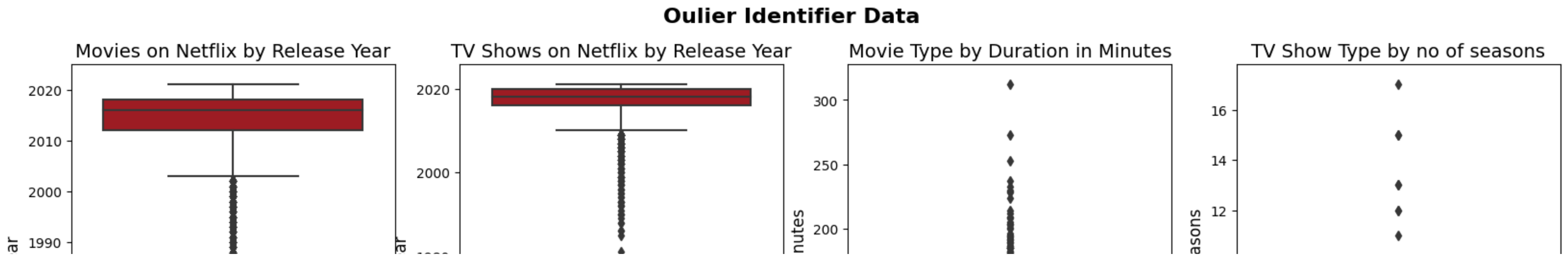


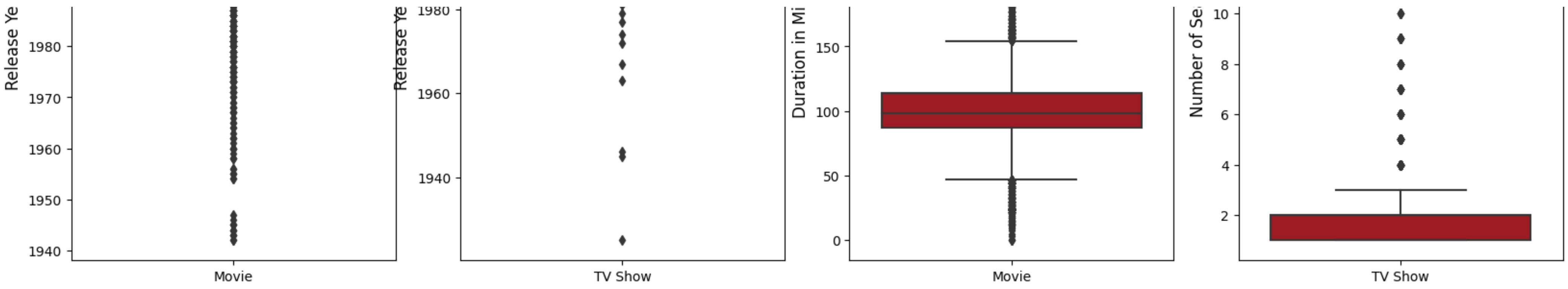
In [ ]:

Box Plot - Outliers Identification

In [ ]:

```
netflix_no_duplicate = nf.drop_duplicates(subset = "title")
netflix_movie_no_duplicate = netflix_no_duplicate[netflix_no_duplicate["type"] == "Movie"]
netflix_tvshow_no_duplicate = netflix_no_duplicate[netflix_no_duplicate["type"] == "TV Show"]

plt.figure(figsize=(20,6))
netflix_red = "#b20710"
netflix_grey = "#221f1f"

plt.subplot(1,4,1)
sns.boxplot(data = netflix_movie_no_duplicate, x = "type", y ="release_year", color= netflix_red)
plt.title("Movies on Netflix by Release Year", fontsize = 14)
plt.xlabel("")
plt.ylabel("Release Year", fontsize = 12)
plt.subplot(1,4,2)
sns.boxplot(data = netflix_tvshow_no_duplicate, x = "type", y ="release_year", color = netflix_red)
plt.title("TV Shows on Netflix by Release Year", fontsize = 14)
plt.xlabel("")
plt.ylabel("Release Year", fontsize = 12)
plt.subplot(1,4,3)
sns.boxplot(data = netflix_movie_no_duplicate, x = "type", y ="duration_in_minutes", color = netflix_red)
plt.title("Movie Type by Duration in Minutes", fontsize = 14)
plt.xlabel("")
plt.ylabel("Duration in Minutes", fontsize = 12)
plt.subplot(1,4,4)
sns.boxplot(data = netflix_tvshow_no_duplicate[netflix_tvshow_no_duplicate["no_of_seasons"]!=0], x = "type", y ="no_of_seasons",
color = netflix_red)
plt.title("TV Show Type by no of seasons", fontsize = 14)
plt.xlabel("")
plt.ylabel("Number of Seasons", fontsize = 12)
plt.suptitle("Oulier Identifier Data", fontweight = "bold", fontsize = 16)
plt.show()
```





In [ ]:

Understand the outlier data is very important to understand the pattern and content being uploaded by netflix. We can undertand the following from the above plotting. The most Movies released on Netflix are from the original release year sometime after 2012 till around 2018. The rest of the movies are distributed outside can be considered as outliers. The most TV Shows released on Netflix are from the original release year sometime after 2018 till around 2020. The rest of the movies are distributed outside can be considered as outliers. The movies that most released on Netflix have a runtime or are of duration between 90 minutes to 120 minutes. The TV Shows that most released on Netflix have a seasons between 1 to 2 seasons. 6) Recommendations:

1. Netflix should consider to release movies between the duration of 90 minutes to 120 minutes as that seems to be working presently for them.
2. As most of the movies that seems to working for netflix are from after 2012, they can try to consider to remove any of the old movies that are not working for them any more and consider to onboard more latest movies.
3. Netflix can cosider onboarding more movies from India as it seems that movies are being most viewed in India over all.
4. Netflix can cosider onboarding more TV Shows from Taiwan as it seems that movies are being most viewed in Taiwan over all.
5. It is recommended that Netflix can consider uploading more "International Movie" Genre as looks like the audience from various countries might be watch more due to which it is working for them.
6. Though TV Shows are slowing pickup it is recommended that you keep onboarding Movies on a regular basis and not reduce the number as they seemed to have always worked more than TV Shows.
7. It is recommended to onboard more comedy genre content as we can see that there has been a sudden increase especially in the last 3 years.

In [ ]:

In [ ]:

In [ ]:

In [ ]: