

# 1. Problem Statement:

Netflix aims to enhance its content production to increase its subscriber base and revenue in different countries. The company wants to analyze viewer preferences and trends across regions to determine the most suitable types of shows and movies to produce. Additionally, Netflix seeks insights into how it can effectively grow its business in diverse markets.

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

data = pd.read_csv("/content/netflix_case_study.csv")
data.head()

{"repr_error":"'str' object has no attribute
'empty'", "type": "dataframe", "variable_name": "data"}

netflix_shows=data[data['type']=='TV Show']
netflix_movies=data[data['type']=='Movie']
```

## 2. Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary

```
print("shape", data.shape)
print(data.dtypes)
print(data.isnull().sum())
```

```
shape (8807, 12)
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
show_id      0
type         0
title        0
director     2634
cast         825
country      831
date_added   10
release_year  0
rating       4
duration     3
listed_in    0
description  0
dtype: int64

```

```
print(data.describe(include="all"))
```

	show_id	type	title	director \
count	8807	8807	8807	6173
unique	8807	2	8807	4528
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka
freq	1	6131	1	19
mean	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN

	cast	country	date_added
release_year \			
count	7982	7976	8797
8807.000000			
unique	7692	748	1767
NaN			
top	David Attenborough	United States	January 1, 2020
NaN			
freq	19	2818	109
NaN			
mean	NaN	NaN	NaN
2014.180198			
std	NaN	NaN	NaN
8.819312			
min	NaN	NaN	NaN
1925.000000			
25%	NaN	NaN	NaN
2013.000000			
50%	NaN	NaN	NaN
2017.000000			

75%	NaN	NaN	NaN
2019.000000			
max	NaN	NaN	NaN
2021.000000			

	rating	duration	listed_in \
count	8803	8804	8807
unique	17	220	514
top	TV-MA	1 Season	Dramas, International Movies
freq	3207	1793	362
mean	NaN	NaN	NaN
std	NaN	NaN	NaN
min	NaN	NaN	NaN
25%	NaN	NaN	NaN
50%	NaN	NaN	NaN
75%	NaN	NaN	NaN
max	NaN	NaN	NaN

	description
count	8807
unique	8775
top	Paranormal activity at a lush, abandoned prope...
freq	4
mean	NaN
std	NaN
min	NaN
25%	NaN
50%	NaN
75%	NaN
max	NaN

```
# Conversion of categorical attributes to 'category' (if required)
# For example, if 'Country' and 'Rating' are categorical attributes
# data["country"] = data["country"].astype("category")
# data["rating"] = data["rating"].astype("category")
# print(data["country"])
# print(data["rating"])
```

### #3. Non-Graphical Analysis: Value counts and unique attributes

```
# Non-Graphical Analysis: Value counts and unique attributes
categorical_columns = ["type", "country", "rating", "duration"]
for col in categorical_columns:
    print(data[col].value_counts())

Movie      6131
TV Show    2676
Name: type, dtype: int64
United States      2818
India              972
```

United Kingdom	419
Japan	245
South Korea	199

...

Romania, Bulgaria, Hungary	1
Uruguay, Guatemala	1
France, Senegal, Belgium	1
Mexico, United States, Spain, Colombia	1
United Arab Emirates, Jordan	1

Name: country, Length: 748, dtype: int64

TV-MA	3207
-------	------

TV-14	2160
-------	------

TV-PG	863
-------	-----

R	799
---	-----

PG-13	490
-------	-----

TV-Y7	334
-------	-----

TV-Y	307
------	-----

PG	287
----	-----

TV-G	220
------	-----

NR	80
----	----

G	41
---	----

TV-Y7-FV	6
----------	---

NC-17	3
-------	---

UR	3
----	---

74 min	1
--------	---

84 min	1
--------	---

66 min	1
--------	---

Name: rating, dtype: int64

1 Season	1793
----------	------

2 Seasons	425
-----------	-----

3 Seasons	199
-----------	-----

90 min	152
--------	-----

94 min	146
--------	-----

...

16 min	1
--------	---

186 min	1
---------	---

193 min	1
---------	---

189 min	1
---------	---

191 min	1
---------	---

Name: duration, Length: 220, dtype: int64

```
numerical_columns = ["release_year"]
```

```
for col in numerical_columns:
```

```
    print(data[col].value_counts())
```

2018	1147
------	------

2017	1032
------	------

2019	1030
------	------

2020	953
------	-----

2016	902
------	-----

```

...
1959      1
1925      1
1961      1
1947      1
1966      1
Name: release_year, Length: 74, dtype: int64

unique_attributes = {}
for col in data.columns:
    unique_attributes[col] = data[col].unique()

for column, values in unique_attributes.items():
    print(column)
    print(values)

show_id
['s1' 's2' 's3' ... 's8805' 's8806' 's8807']
type
['Movie' 'TV Show']
title
['Dick Johnson Is Dead' 'Blood & Water' 'Ganglands' ... 'Zombieland'
 'Zoom' 'Zubaan']
director
['Kirsten Johnson' nan 'Julien Leclercq' ... 'Majid Al Ansari'
 'Peter Hewitt' 'Mozez Singh']
cast
[nan
 'Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon
 Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmore
 Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake
 Ka-Ncube, Odwa Gwanya, Mekaila Mathys, Sandi Schultz, Duane Williams,
 Shamilla Miller, Patrick Mofokeng'
 'Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabiha Akkari, Sofia
 Lesaffre, Salim Kechiouche, Nouredine Farihi, Geert Van Rampelberg,
 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 Tripathy']
country
['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'  
'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']
date_added
['September 25, 2021' 'September 24, 2021' 'September 23, 2021' ...
'December 6, 2018' 'March 9, 2016' 'January 11, 2020']
release_year
[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]
rating
['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']
duration
['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']
listed_in
['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'  
'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']
description
['As her father nears the end of his life, filmmaker Kirsten Johnson
stages his death in inventive and comical ways to help them both 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 and 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 by 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."]

```

#### #4. Visual Analysis - Univariate, Bivariate after pre-processing of the data

```

# 2. Handling null values
# a. For categorical variables with null values, update those rows as
unknown_column_name.
# Example : Replace missing value with Unknown Actor for missing value
in Actors column.
# b. Replace with 0 for continuous variables having null values.
categorical_cols = ['type', 'title', 'director', 'cast', 'country',
'rating', 'duration']
for col in categorical_cols:
    data[col] = data[col].fillna('unknown_'+ col)
# Handling null values in continuous variables
continuous_cols = ['release_year']
data[continuous_cols] = data[continuous_cols].fillna(0)

data.isnull().sum()

```

show_id	0
type	0
title	0
director	0
cast	0
country	0
date_added	10
release_year	0
rating	0

```

duration          0
listed_in         0
description       0
dtype: int64

data.head()

{"repr_error":"'str' object has no attribute
'empty'", "type": "dataframe", "variable_name": "data"}

# 1. Un-nesting the columns
# a. Un-nest the columns those have cells with multiple comma
separated values by creating multiple rows
# Identify columns with comma-separated values that need to be
unnested
columns_to_unnest = ['cast', 'director', 'country'] # Add more
columns as needed

# Create a list to store DataFrames with unnested columns
unnested_dfs = []

# Iterate over each column to unnest
for column in columns_to_unnest:
    # Split values by comma and create separate rows
    unnested_df = data.assign(**{column:
data[column].str.split(',').explode(column)
    unnested_dfs.append(unnested_df)

# Concatenate all unnested DataFrames
netflix_data = pd.concat(unnested_dfs, ignore_index=True)

# Drop original rows with comma-separated values
netflix_data.dropna(subset=columns_to_unnest, inplace=True)

# Reset index
netflix_data.reset_index(drop=True, inplace=True)

netflix_data.shape

(85413, 12)

netflix_data.head()

{"repr_error":"'str' object has no attribute
'empty'", "type": "dataframe", "variable_name": "netflix_data"}

netflix_data.shape

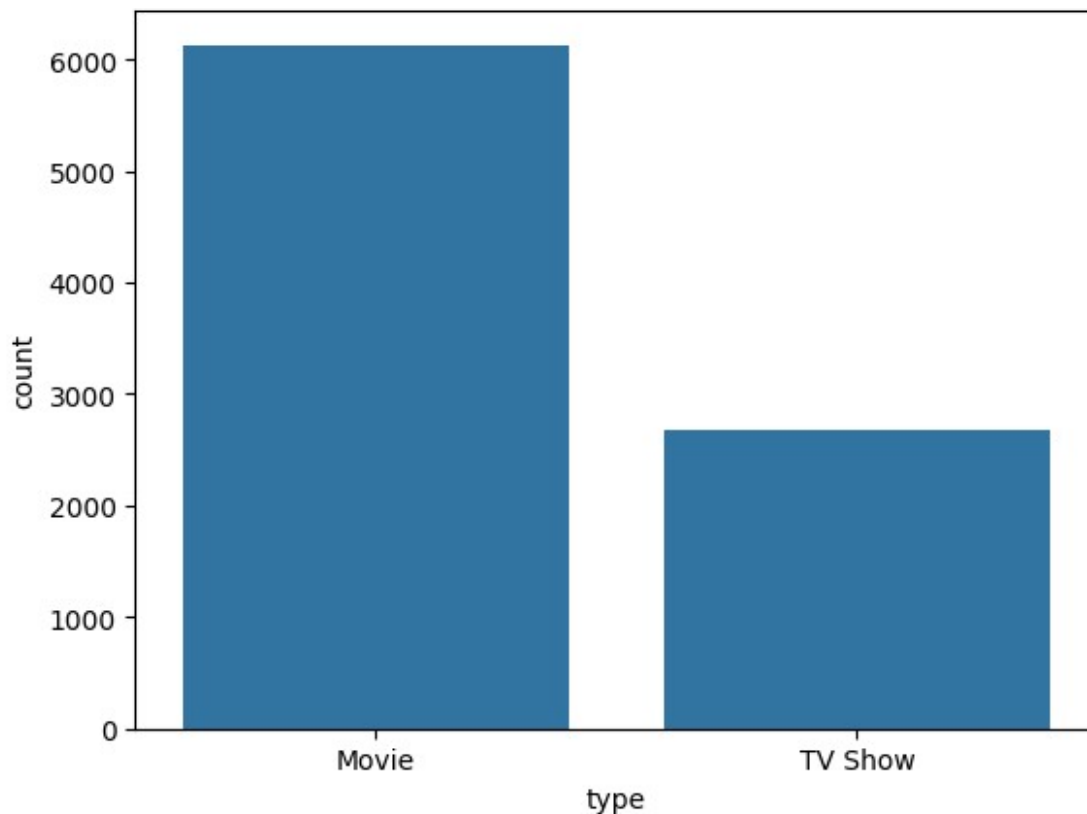
(85413, 12)

```

## 4.1 For continuous variable(s): Distplot, countplot, histogram for univariate analysis

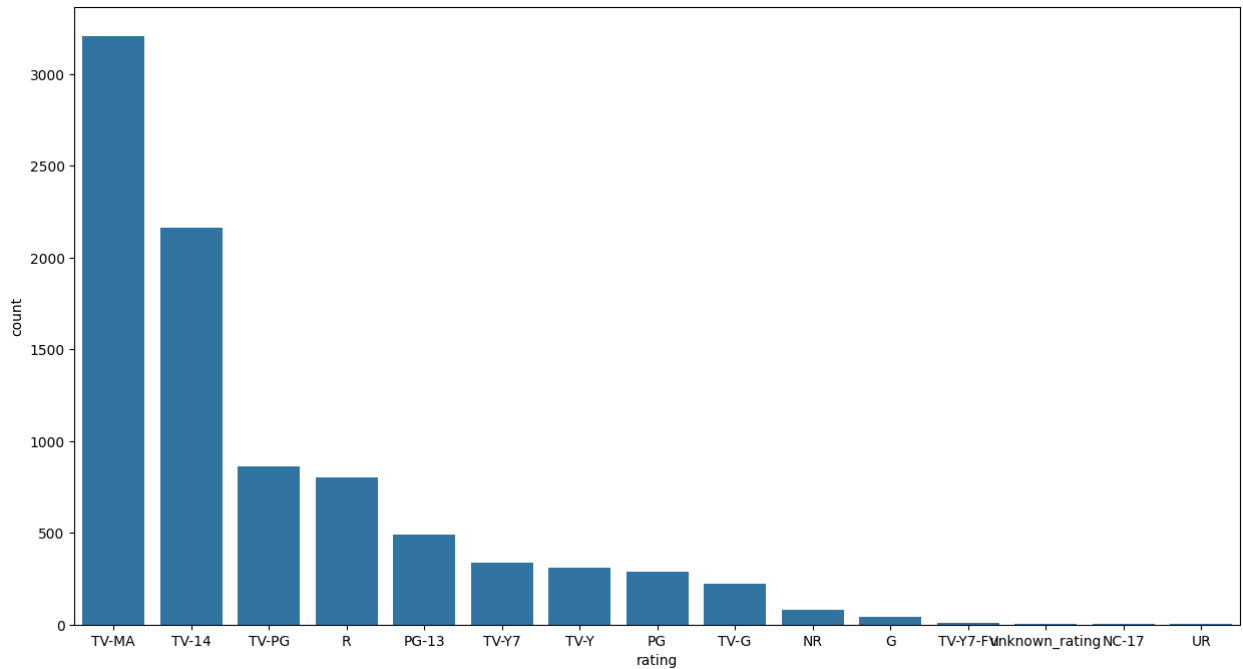
```
sns.countplot(data=data, x="type")
```

```
<Axes: xlabel='type', ylabel='count'>
```



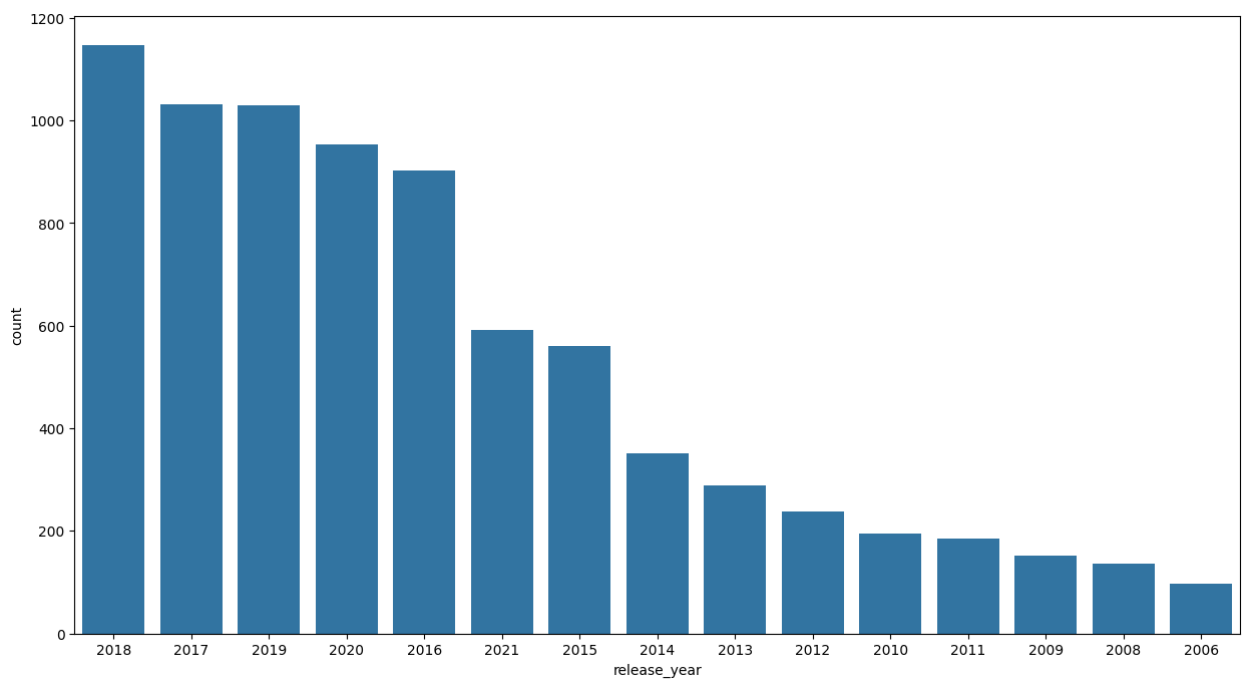
```
plt.figure(figsize=(15, 8))  
sns.countplot(x="rating", data=data,  
order=data["rating"].value_counts().index[0:15])
```

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



```
plt.figure(figsize=(15, 8))
sns.countplot(x="release_year", data=data,
order=data["release_year"].value_counts().index[0:15])
```

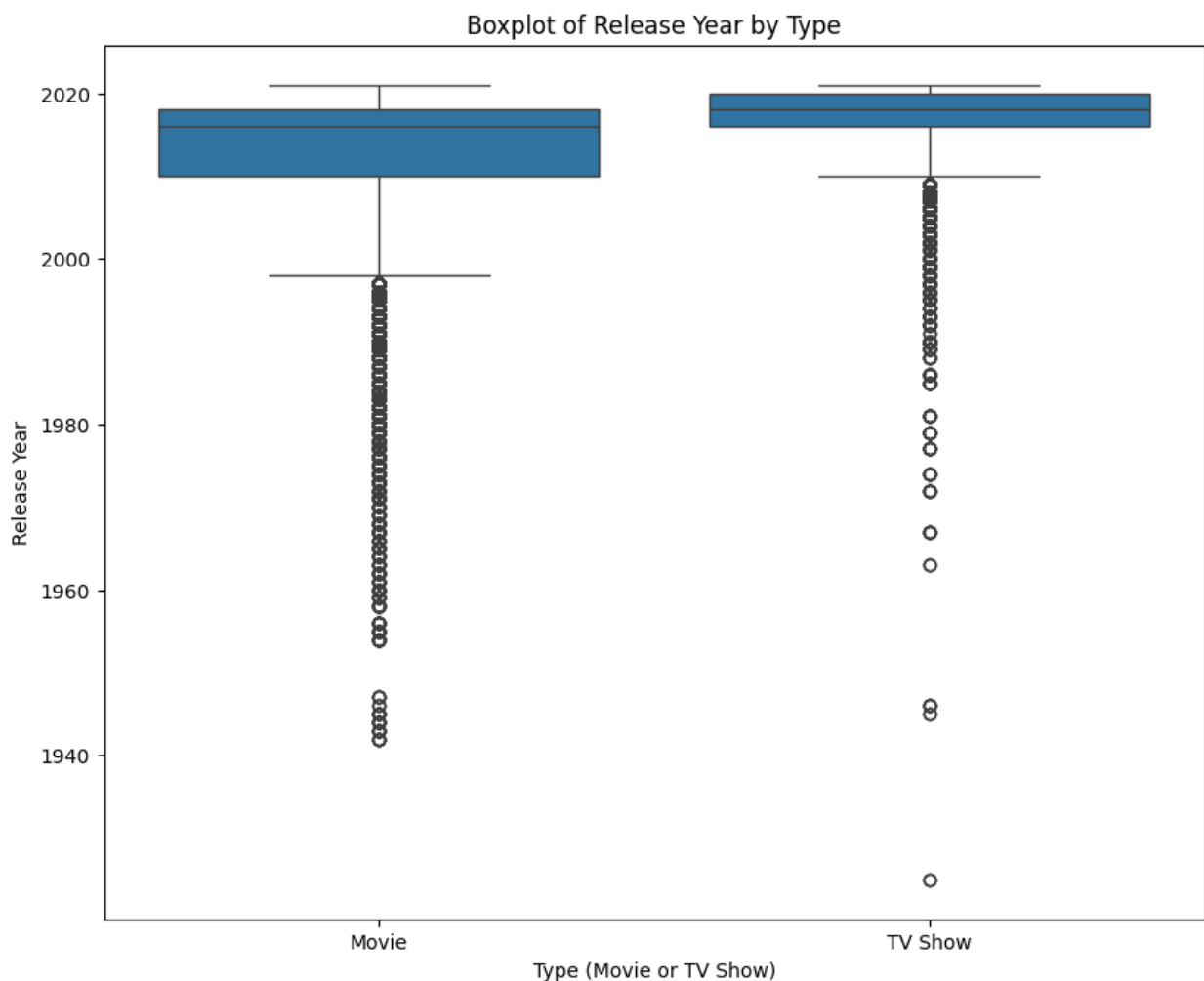
<Axes: xlabel='release\_year', ylabel='count'>



## 4.2 For categorical variable(s): Boxplot

```
categorical_column = 'type'
numerical_column = 'release_year'

# Create boxplot
plt.figure(figsize=(10, 8))
sns.boxplot(x=categorical_column, y=numerical_column,
            data=netflix_data)
plt.title('Boxplot of Release Year by Type')
plt.xlabel('Type (Movie or TV Show)')
plt.ylabel('Release Year')
plt.show()
```



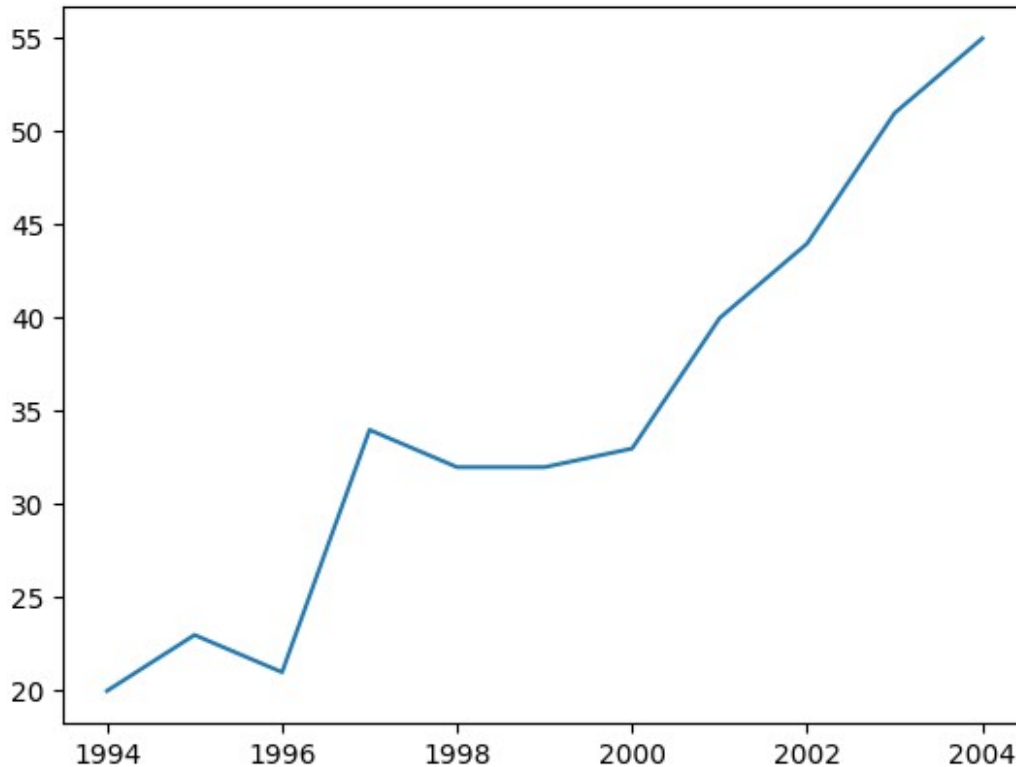
```
# How has the number of movies released per year changed over the last 20-30 years?
# Release_year, Movies
movies = data.loc[data["type"]=="Movie"]
```

```

movies_per_year = movies.groupby('release_year').size()
current_year = pd.Timestamp.now().year
start_year = current_year-30
end_year = current_year-20
movies_per_year = movies_per_year.loc[start_year:end_year]
plt.plot(movies_per_year.index, movies_per_year.values,
label="Movies")

```

[<matplotlib.lines.Line2D at 0x79de8649cfa0>]

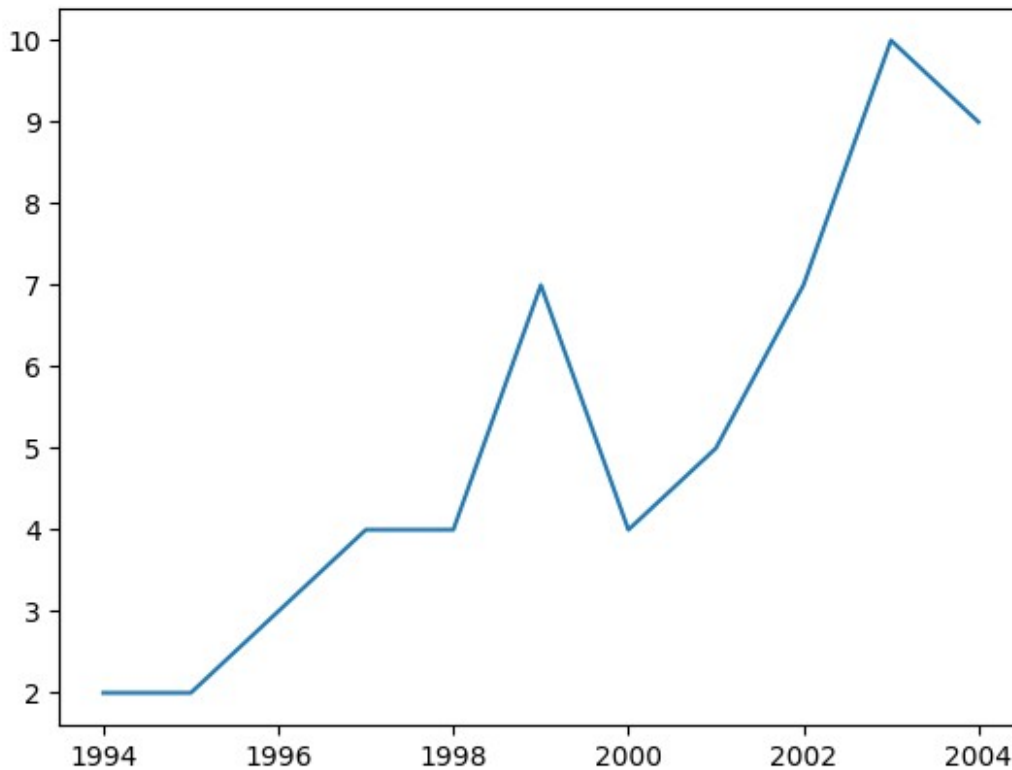


```

tv_shows = data.loc[data["type"]=="TV Show"]
tv_shows_per_year = tv_shows.groupby('release_year').size()
current_year = pd.Timestamp.now().year
start_year = current_year-30
end_year = current_year-20
tv_shows_per_year = tv_shows_per_year.loc[start_year:end_year]
plt.plot(tv_shows_per_year.index, tv_shows_per_year.values, label='TV
Shows')

```

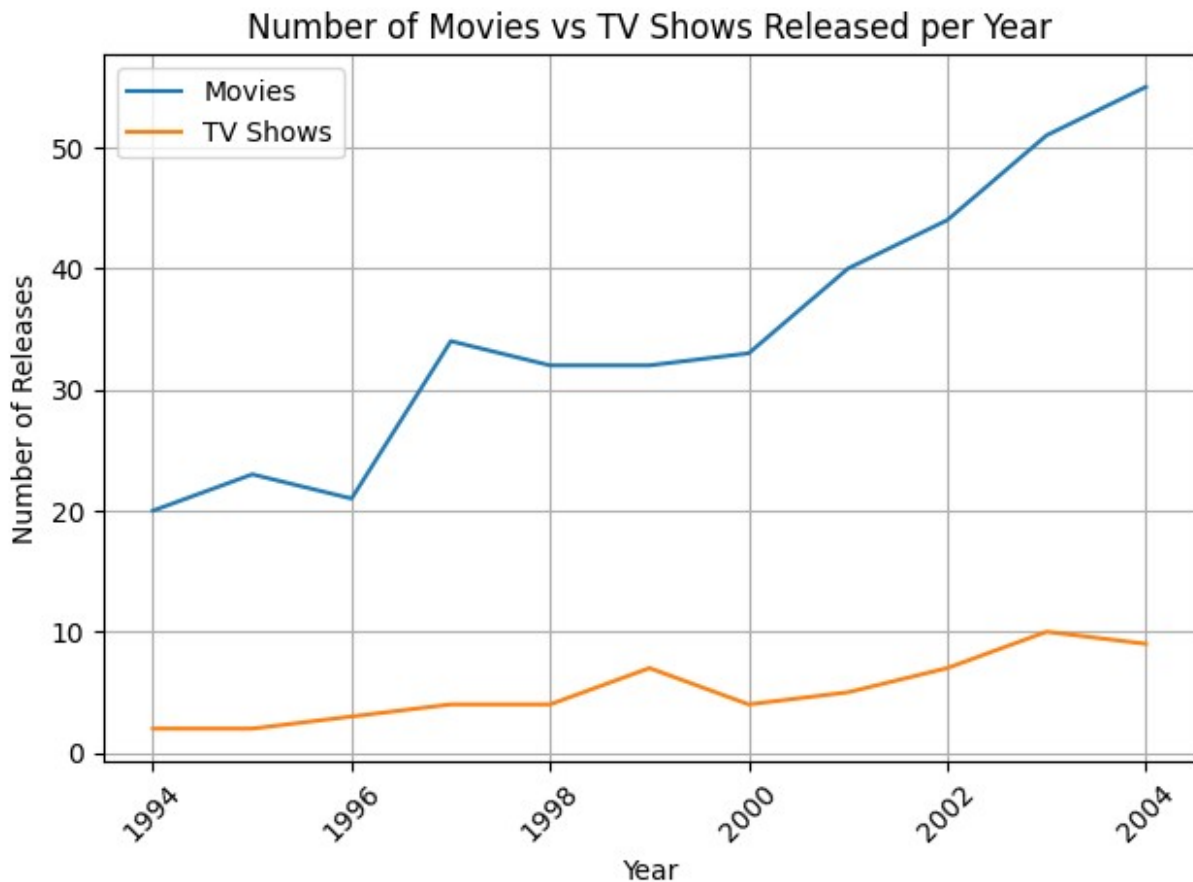
[<matplotlib.lines.Line2D at 0x79de7de34d90>]



```
# Comparison of tv shows vs. movies.
# Type
movies = data.loc[data["type"]=="Movie"]
movies_per_year = movies.groupby('release_year').size()
current_year = pd.Timestamp.now().year
start_year = current_year-30
end_year = current_year-20
movies_per_year = movies_per_year.loc[start_year:end_year]
tv_shows = data.loc[data["type"]=="TV Show"]
tv_shows_per_year = tv_shows.groupby('release_year').size()
current_year = pd.Timestamp.now().year
start_year = current_year-30
end_year = current_year-20
tv_shows_per_year = tv_shows_per_year.loc[start_year:end_year]
plt.plot(movies_per_year.index, movies_per_year.values,
label="Movies")
plt.plot(tv_shows_per_year.index, tv_shows_per_year.values, label='TV
Shows')
plt.title('Number of Movies vs TV Shows Released per Year')
plt.xlabel('Year')
plt.ylabel('Number of Releases')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
```



```
plt.tight_layout()
plt.show()
```



```
# What is the best time to launch a TV show?
# Date_added, Type
tv_shows = data.loc[data["type"] == "TV Show"]
tv_shows["date_added"] = pd.to_datetime(tv_shows["date_added"])
tv_shows["month_added"] = tv_shows["date_added"].dt.month
tv_shows_by_month = tv_shows.groupby("month_added").size()
plt.title('Number of TV Shows Added to Netflix by Month')
plt.xlabel('Month')
plt.ylabel('Number of TV Shows')
plt.xticks(rotation=0)
plt.tight_layout()
plt.bar(tv_shows_by_month.index, tv_shows_by_month, color='orange')
plt.show()
```

<ipython-input-92-1d4b276cde0a>:4: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation:

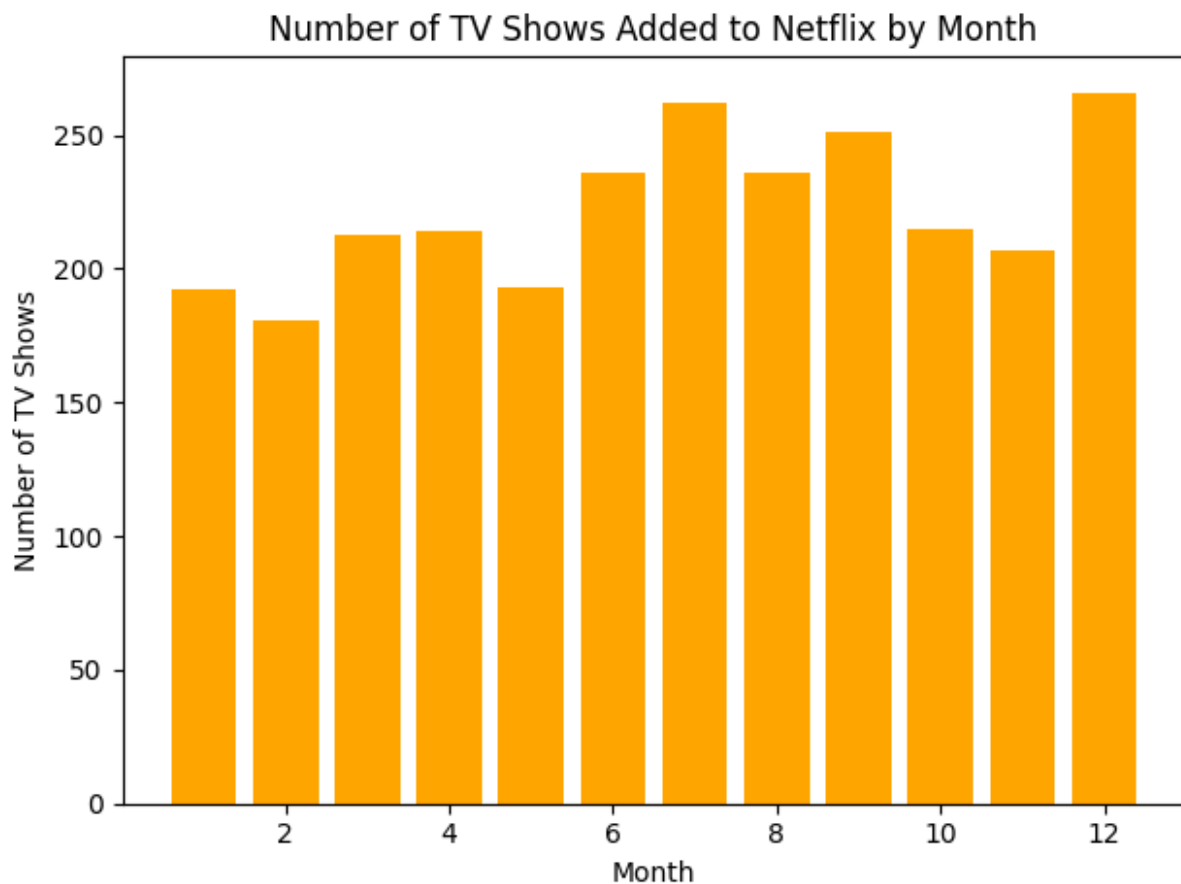
[https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
tv_shows["date_added"] = pd.to_datetime(tv_shows["date_added"])
<ipython-input-92-1d4b276cde0a>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation:

[https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
tv_shows["month_added"] = tv_shows["date_added"].dt.month
```



```
# Analysis of actors/directors of different types of shows/movies.
# Director, Type, cast, Genre
movies = data.loc[data["type"] == "Movie"]
tv_shows = data.loc[data["type"] == "TV Show"]

def split_cast_director(data):
    if pd.isna(data):
        return data.split(', ')
    else:
```

```

    return

movies["Actors"] = movies["cast"].apply(split_cast_director)
movies["Directors"] = movies["director"].apply(split_cast_director)
tv_shows["Actors"] = tv_shows["cast"].apply(split_cast_director)
tv_shows["Directors"] =
tv_shows["director"].apply(split_cast_director)

movie_actors = movies["Actors"].explode().value_counts()
tv_shows_actors = tv_shows["Actors"].explode().value_counts()
movie_directors = movies["Directors"].explode().value_counts()
tv_shows_directors = tv_shows["Directors"].explode().value_counts()

plt.figure(figsize=(10, 6))
sns.barplot(x=movie_actors.head(10).values,
y=movie_actors.head(10).index, palette="Blues_d")
plt.title('Top Actors in Movies')
plt.xlabel('Number of Appearances')
plt.ylabel('Actor')
plt.show()

# Plot for top directors in movies
plt.figure(figsize=(10, 6))
sns.barplot(x=movie_directors.head(10).values,
y=movie_directors.head(10).index, palette="BuGn_r")
plt.title('Top Directors in Movies')
plt.xlabel('Number of Movies Directed')
plt.ylabel('Director')
plt.show()

# Plot for top actors in TV shows
plt.figure(figsize=(10, 6))
sns.barplot(x=tv_shows_actors.head(10).values,
y=tv_shows_actors.head(10).index, palette="Oranges_r")
plt.title('Top Actors in TV Shows')
plt.xlabel('Number of Appearances')
plt.ylabel('Actor')
plt.show()

# Plot for top directors in TV shows
plt.figure(figsize=(10, 6))
sns.barplot(x=tv_shows_directors.head(10).values,
y=tv_shows_directors.head(10).index, palette="Reds_r")
plt.title('Top Directors in TV Shows')
plt.xlabel('Number of TV Shows Directed')
plt.ylabel('Director')
plt.show()

```

```
<ipython-input-93-446c47a40877>:12: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation:

[https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
movies["Actors"] = movies["cast"].apply(split_cast_director)
```

```
<ipython-input-93-446c47a40877>:13: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation:

[https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
movies["Directors"] = movies["director"].apply(split_cast_director)
```

```
<ipython-input-93-446c47a40877>:14: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation:

[https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
tv_shows["Actors"] = tv_shows["cast"].apply(split_cast_director)
```

```
<ipython-input-93-446c47a40877>:15: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation:

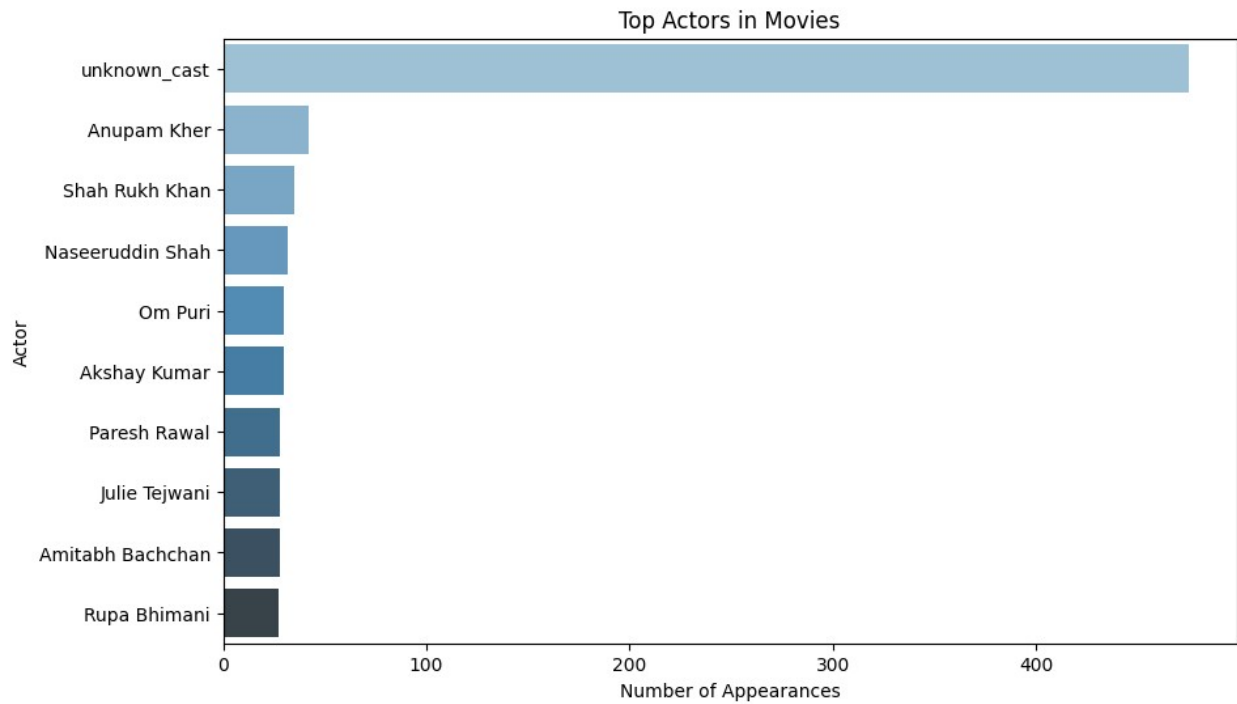
[https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
tv_shows["Directors"] =  
tv_shows["director"].apply(split_cast_director)
```

```
<ipython-input-93-446c47a40877>:24: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

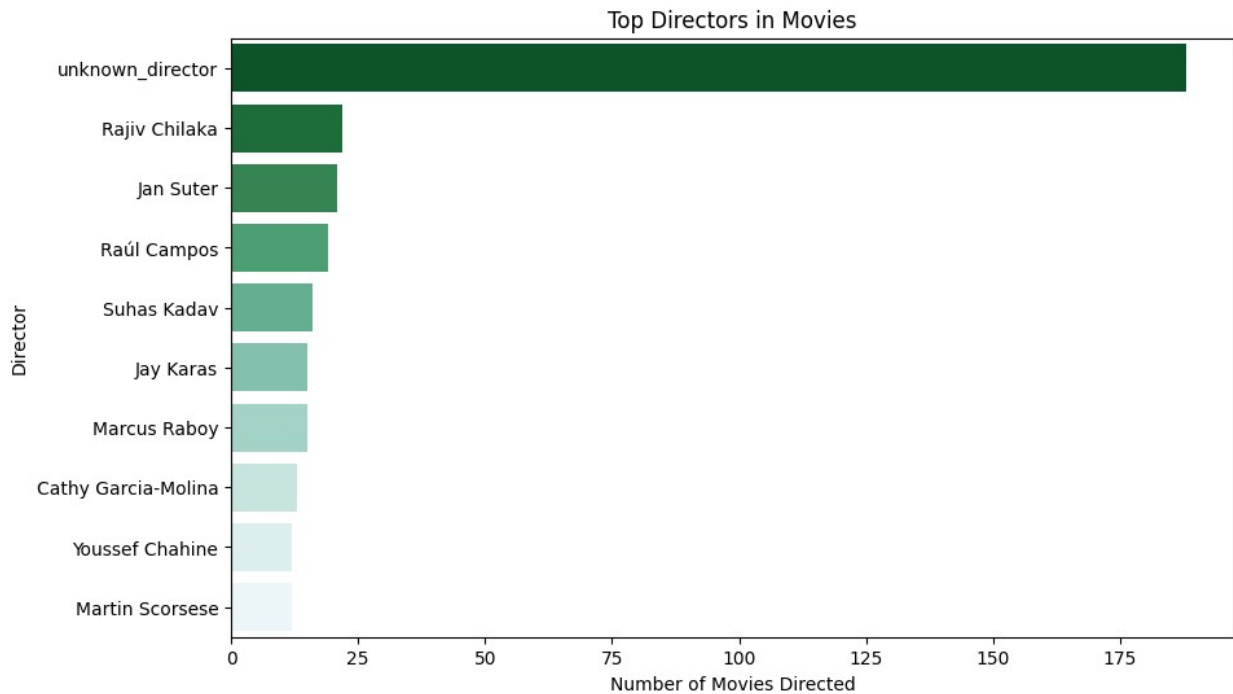
```
sns.barplot(x=movie_actors.head(10).values,  
y=movie_actors.head(10).index, palette="Blues_d")
```



```
<ipython-input-93-446c47a40877>:32: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.
```

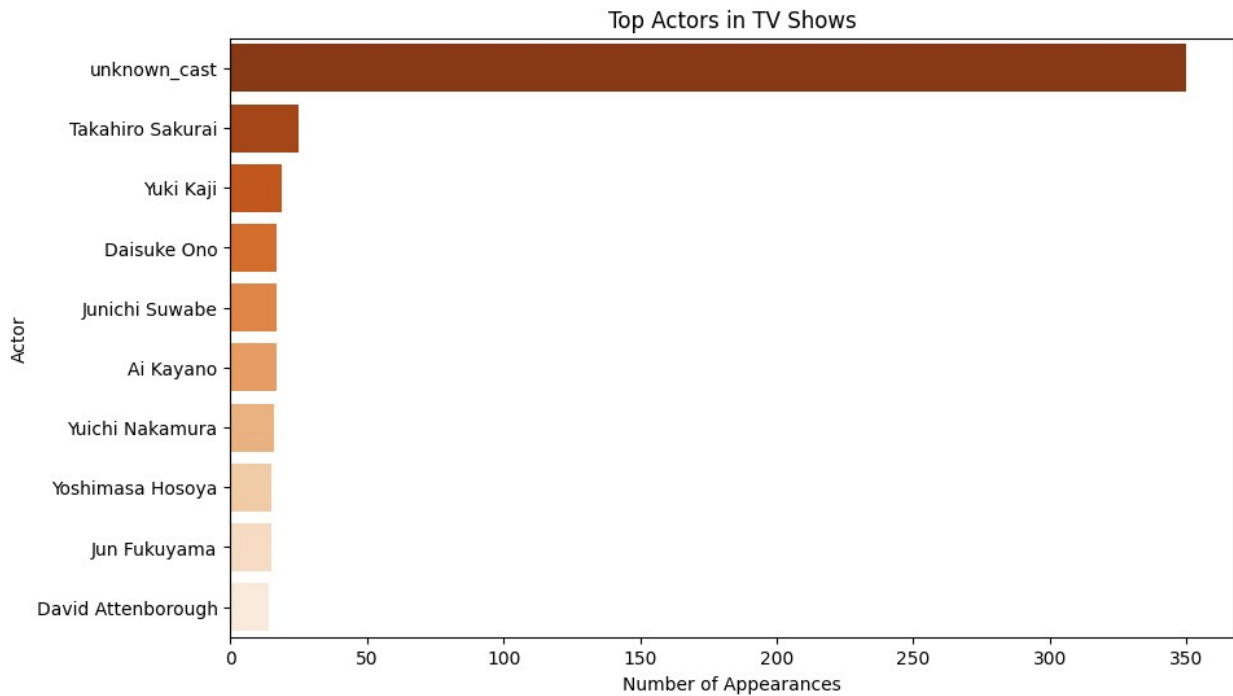
```
sns.barplot(x=movie_directors.head(10).values,  
y=movie_directors.head(10).index, palette="BuGn_r")
```



```
<ipython-input-93-446c47a40877>:40: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

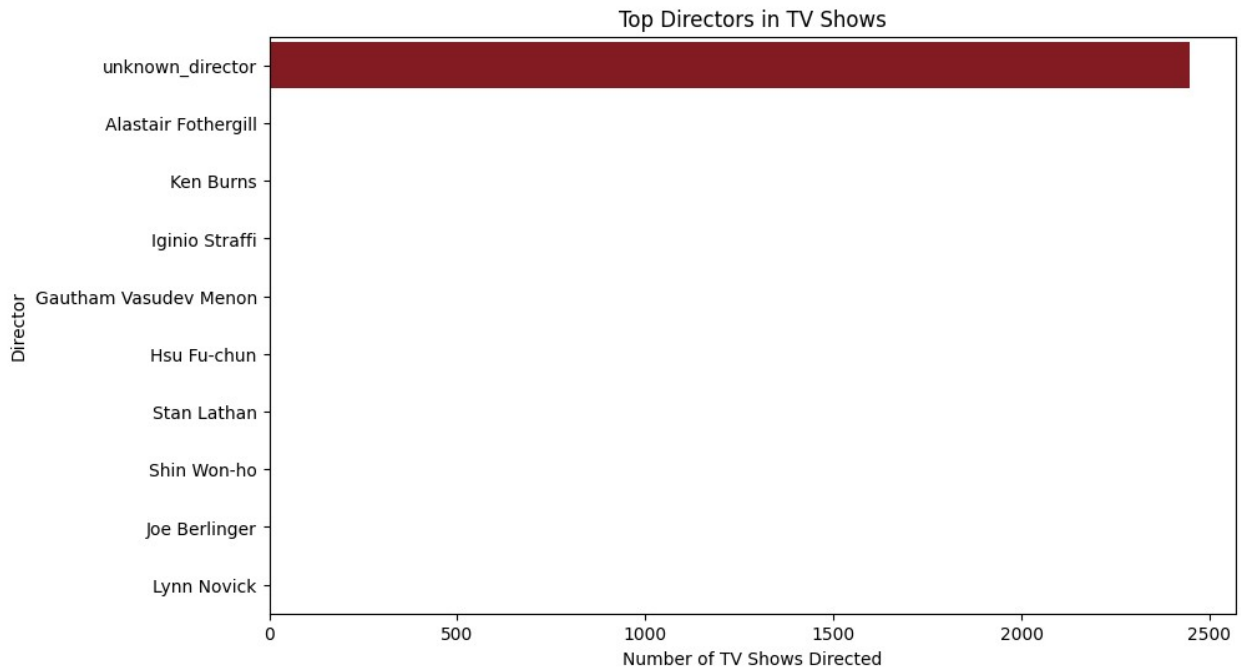
```
sns.barplot(x=tv_shows_actors.head(10).values,  
y=tv_shows_actors.head(10).index, palette="Oranges_r")
```



```
<ipython-input-93-446c47a40877>:48: FutureWarning:
```

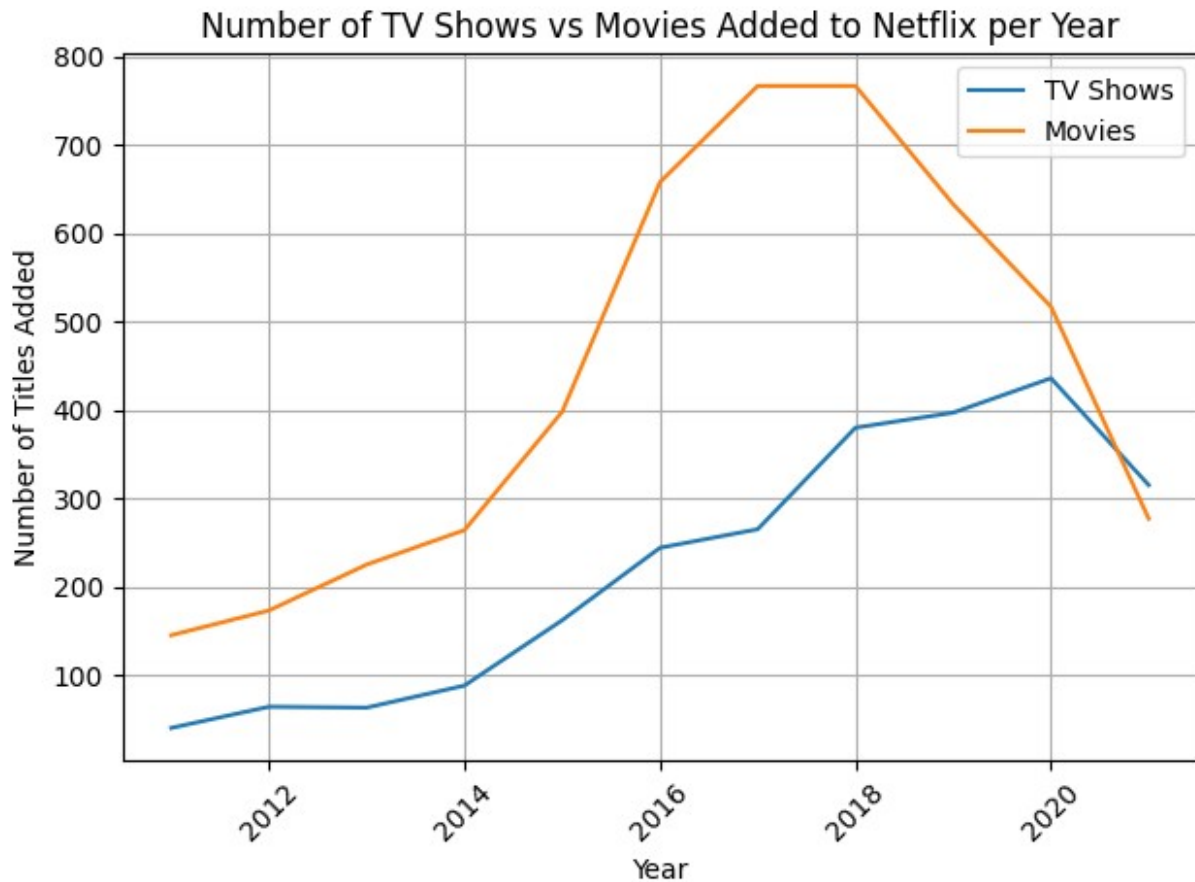
```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.
```

```
sns.barplot(x=tv_shows_directors.head(10).values,  
y=tv_shows_directors.head(10).index, palette="Reds_r")
```

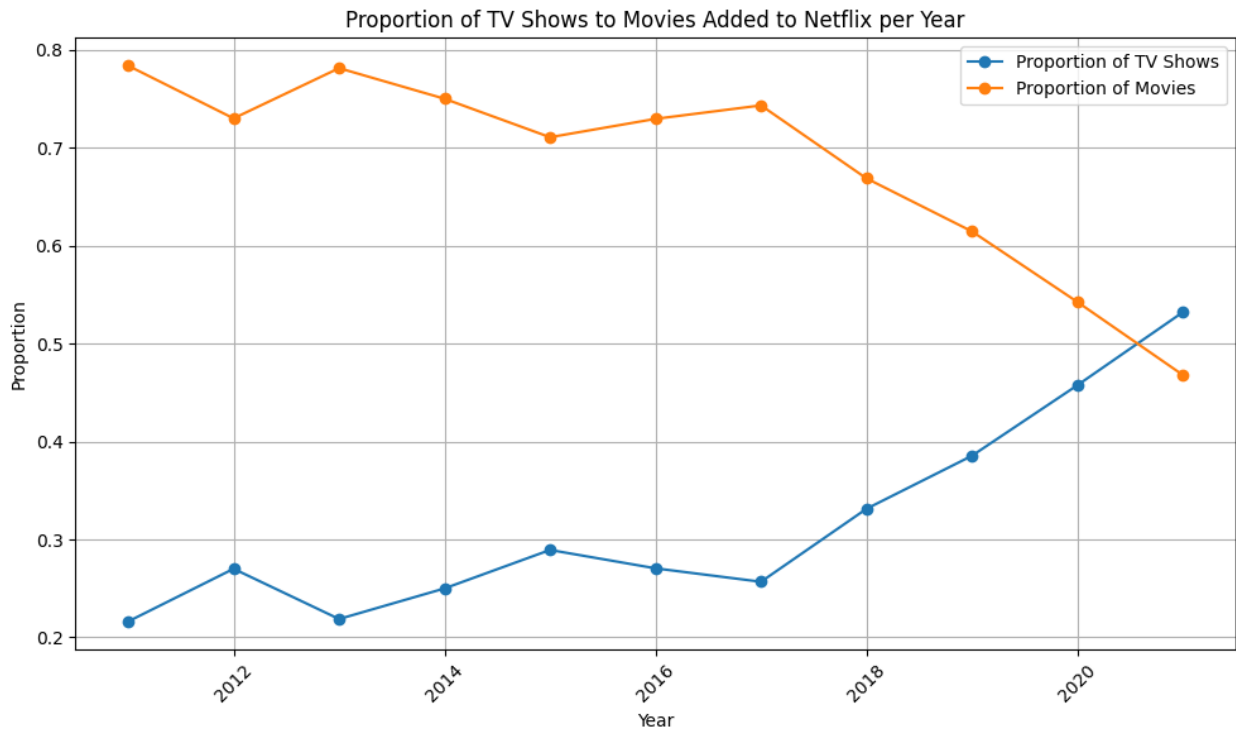


```
# Does Netflix has more focus on TV Shows than movies in recent years
# type, release_year
movies = data.loc[(data["type"] == "Movie") & (data['release_year'] >=
2011)]
tv_shows = data.loc[(data["type"] == "TV Show") &
(data['release_year'] >= 2011)]
movie_per_years = movies.groupby('release_year').size()
tv_shows_per_years = tv_shows.groupby('release_year').size()
# print(movie_per_years)
# print(tv_shows_per_years)
plt.plot(tv_shows_per_years.index, tv_shows_per_years.values,
label='TV Shows')
plt.plot(movie_per_years.index, movie_per_years.values,
label="Movies")
plt.title('Number of TV Shows vs Movies Added to Netflix per Year')
plt.xlabel('Year')
plt.ylabel('Number of Titles Added')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

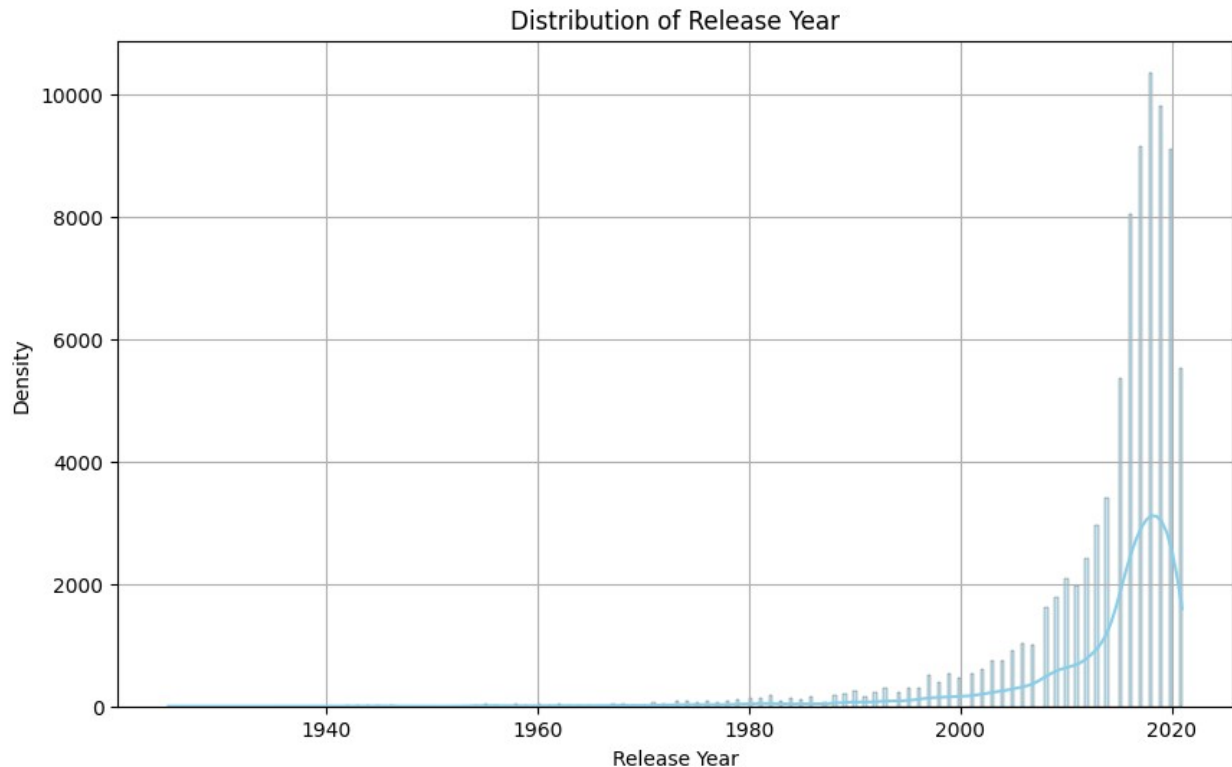




```
# print(tv_shows_per_years)
# print(movie_per_years)
proportion_tv_shows = tv_shows_per_years / (tv_shows_per_years +
movie_per_years)
proportion_movies = movie_per_years / (tv_shows_per_years +
movie_per_years)
# print(proportion_tv_shows)
# print(proportion_movies)
plt.figure(figsize=(10, 6))
plt.plot(proportion_tv_shows.index, proportion_tv_shows.values,
label='Proportion of TV Shows', marker='o')
plt.plot(proportion_movies.index, proportion_movies.values,
label='Proportion of Movies', marker='o')
plt.title('Proportion of TV Shows to Movies Added to Netflix per
Year')
plt.xlabel('Year')
plt.ylabel('Proportion')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
plt.figure(figsize=(10, 6))
sns.histplot(netflix_data['release_year'], kde=True, color='skyblue',
edgecolor='black')
plt.title('Distribution of Release Year')
plt.xlabel('Release Year')
plt.ylabel('Density')
plt.grid(True)
plt.show()
```



```
# 2. Comparison of tv shows vs. movies.
# a. Find the number of movies produced in each country and pick the
top 10
# countries.
# Hint : We want you to apply group by each country and find the count
of unique titles of movies
# b. Find the number of Tv-Shows produced in each country and pick the
top 10 countries.
# Hint : We want you to apply group by each country and find the count
of unique
# titles of Tv-shows

movies_data = data[data['type'] == 'Movie']

# Group by country and count the number of unique movie titles
movie_counts_by_country = movies_data.groupby('country')
['title'].nunique()

# Pick the top 10 countries with the highest number of movies
top_10_movie_countries =
movie_counts_by_country.sort_values(ascending=False).head(10)

# print("Top 10 countries with the most movies produced:")
# print(top_10_movie_countries)

# Filter the DataFrame to contain only TV shows
```

```

tv_shows_data = data[data['type'] == 'TV Show']

# Group by country and count the number of unique TV show titles
tv_show_counts_by_country = tv_shows_data.groupby('country')
['title'].nunique()

# Pick the top 10 countries with the highest number of TV shows
top_10_tv_show_countries =
tv_show_counts_by_country.sort_values(ascending=False).head(10)

# print("\nTop 10 countries with the most TV shows produced:")
# print(top_10_tv_show_countries)

fig, axs = plt.subplots(2, 1, figsize=(10, 12))

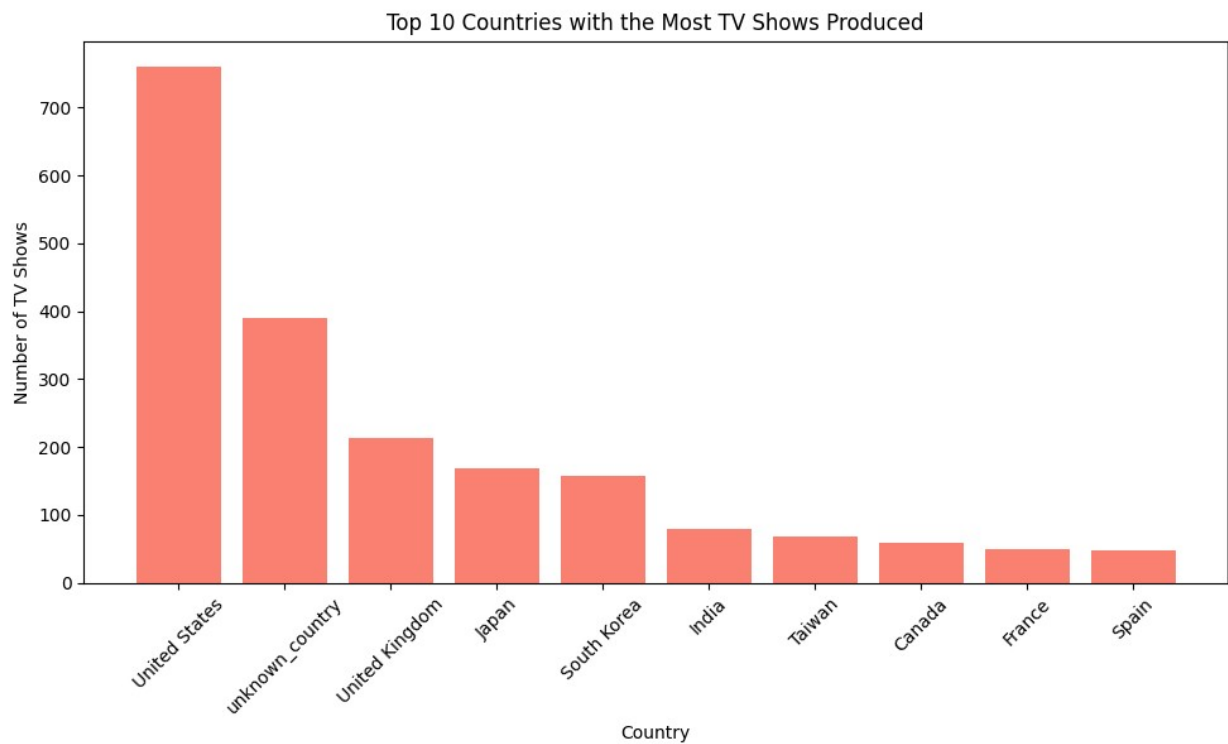
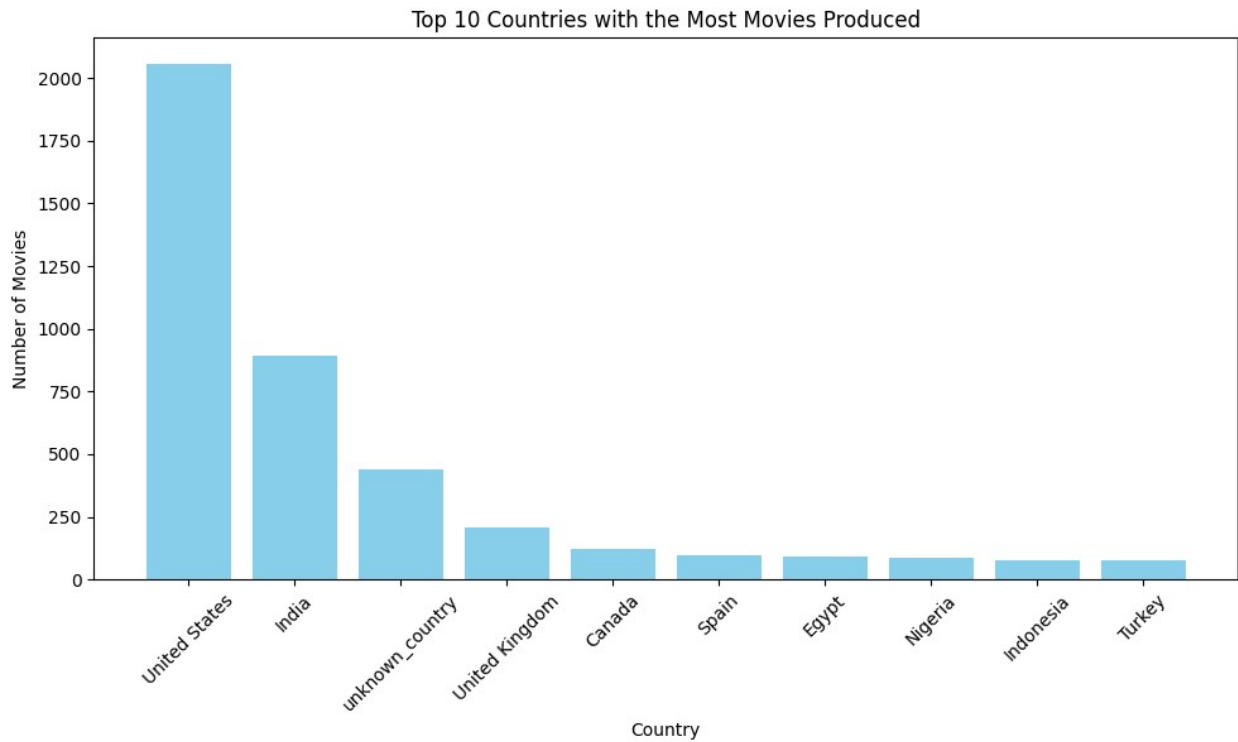
# Plot top 10 countries with the most movies produced
axs[0].bar(top_10_movie_countries.index,
top_10_movie_countries.values, color='skyblue')
axs[0].set_title('Top 10 Countries with the Most Movies Produced')
axs[0].set_xlabel('Country')
axs[0].set_ylabel('Number of Movies')
axs[0].tick_params(axis='x', rotation=45)

# Plot top 10 countries with the most TV shows produced
axs[1].bar(top_10_tv_show_countries.index,
top_10_tv_show_countries.values, color='salmon')
axs[1].set_title('Top 10 Countries with the Most TV Shows Produced')
axs[1].set_xlabel('Country')
axs[1].set_ylabel('Number of TV Shows')
axs[1].tick_params(axis='x', rotation=45)

# Adjust layout
plt.tight_layout()

# Show plot
plt.show()

```



```
# 3. What is the best time to launch a TV show?
# a. Find which is the best week to release the Tv-show or the movie.
Do the analysis
# separately for Tv-shows and Movies
# Hint : We expect you to create a new column and group by each week
```

and count the total number of movies/ tv shows.  
# b. Find which is the best month to release the Tv-show or the movie.  
Do the analysis separately for Tv-shows and Movies  
# Hint : We expect you to create a new column and group by each month  
and count the total number of movies/ tv shows.

```
data['date_added'] = pd.to_datetime(data['date_added'])

# Extract week and month from the 'date_added' column
data['week'] = data['date_added'].dt.week
data['month'] = data['date_added'].dt.month

# Group by week and count the number of TV shows and movies
weekly_counts_tv_shows = data[data['type'] == 'TV Show'].groupby('week').size()
weekly_counts_movies = data[data['type'] == 'Movie'].groupby('week').size()

# Group by month and count the number of TV shows and movies
monthly_counts_tv_shows = data[data['type'] == 'TV Show'].groupby('month').size()
monthly_counts_movies = data[data['type'] == 'Movie'].groupby('month').size()

# Find the best week to release TV shows and movies
best_week_tv_shows = weekly_counts_tv_shows.idxmax()
best_week_movies = weekly_counts_movies.idxmax()

# Find the best month to release TV shows and movies
best_month_tv_shows = monthly_counts_tv_shows.idxmax()
best_month_movies = monthly_counts_movies.idxmax()

print("Best week to release TV shows:", best_week_tv_shows)
print("Best week to release Movies:", best_week_movies)
print("Best month to release TV shows:", best_month_tv_shows)
print("Best month to release Movies:", best_month_movies)
# Plotting the results
plt.figure(figsize=(10, 6))

# Plot weekly counts
plt.subplot(2, 1, 1)
weekly_counts_tv_shows.plot(label='TV Shows')
weekly_counts_movies.plot(label='Movies')
plt.title('Weekly Counts of TV Shows and Movies Added to Netflix')
plt.xlabel('Week')
plt.ylabel('Number of Titles')
plt.axvline(x=best_week_tv_shows, color='r', linestyle='--',
label='Best Week TV Shows')
plt.axvline(x=best_week_movies, color='g', linestyle='--', label='Best Week Movies')
```

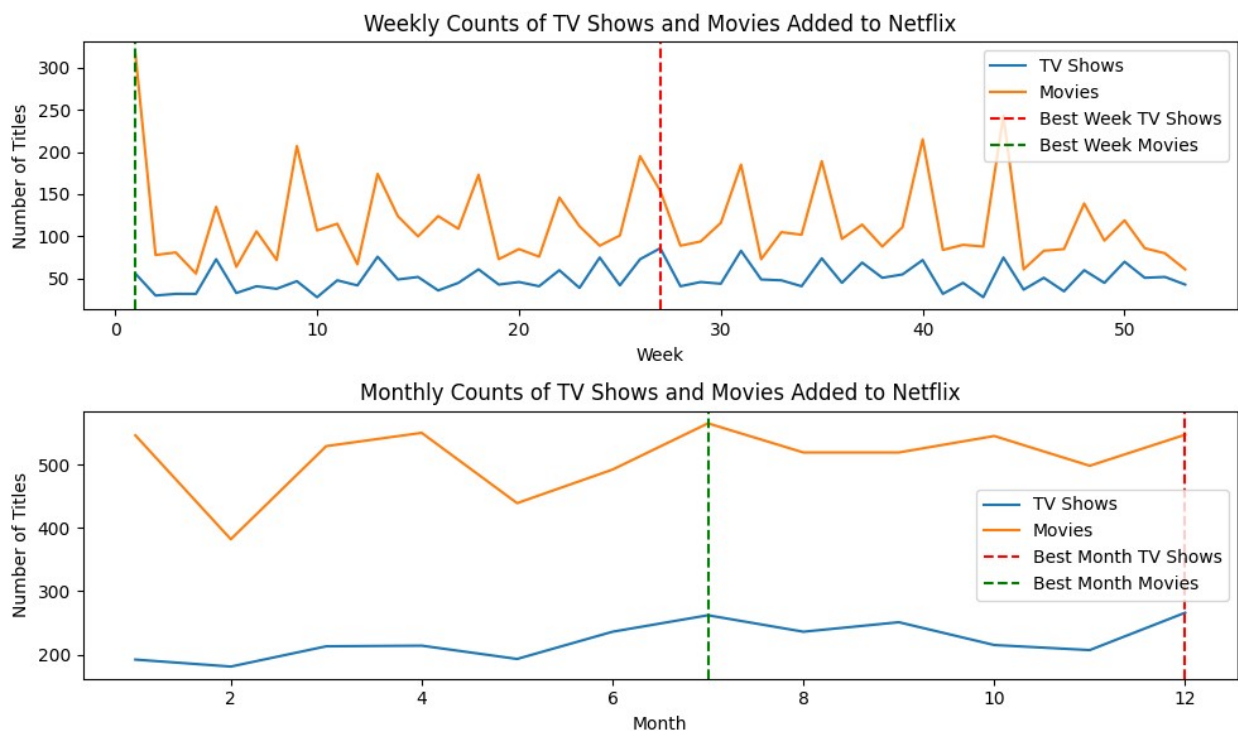
```
plt.legend()

# Plot monthly counts
plt.subplot(2, 1, 2)
monthly_counts_tv_shows.plot(label='TV Shows')
monthly_counts_movies.plot(label='Movies')
plt.title('Monthly Counts of TV Shows and Movies Added to Netflix')
plt.xlabel('Month')
plt.ylabel('Number of Titles')
plt.axvline(x=best_month_tv_shows, color='r', linestyle='--',
label='Best Month TV Shows')
plt.axvline(x=best_month_movies, color='g', linestyle='--',
label='Best Month Movies')
plt.legend()

plt.tight_layout()
plt.show()
```

<ipython-input-100-200a436d9732>:4: FutureWarning:  
Series.dt.weekofyear and Series.dt.week have been deprecated. Please  
use Series.dt.isocalendar().week instead.  
data['week'] = data['date\_added'].dt.week

Best week to release TV shows: 27.0  
Best week to release Movies: 1.0  
Best month to release TV shows: 12.0  
Best month to release Movies: 7.0



```

# 4. Analysis of actors/directors of different types of shows/movies.
# a. Identify the top 10 directors who have appeared in most movies or
TV shows.
# Hint : We want you to group by each actor and find the count of
unique titles of Tv-shows/movies
# b. Identify the top 10 directors who have appeared in most movies or
TV shows.
# Hint : We want you to group by each director and find the count of
unique titles
# of Tv-shows/movies

# Filter data for TV shows and movies separately
tv_shows = netflix_data[netflix_data['type'] == 'TV Show']
movies = netflix_data[netflix_data['type'] == 'Movie']

# Create DataFrames for actors in TV shows and movies and count the
number of unique titles they've appeared in
tv_show_actor_counts = tv_shows.groupby('cast')['title'].nunique()
movie_actor_counts = movies.groupby('cast')['title'].nunique()

# Select the top 10 actors based on the counts for TV shows and movies
separately
top_10_tv_show_actors =
tv_show_actor_counts.sort_values(ascending=False).head(10)
top_10_movie_actors =
movie_actor_counts.sort_values(ascending=False).head(10)

# print("Top 10 actors who have appeared in the most TV shows:")
# print(top_10_tv_show_actors)

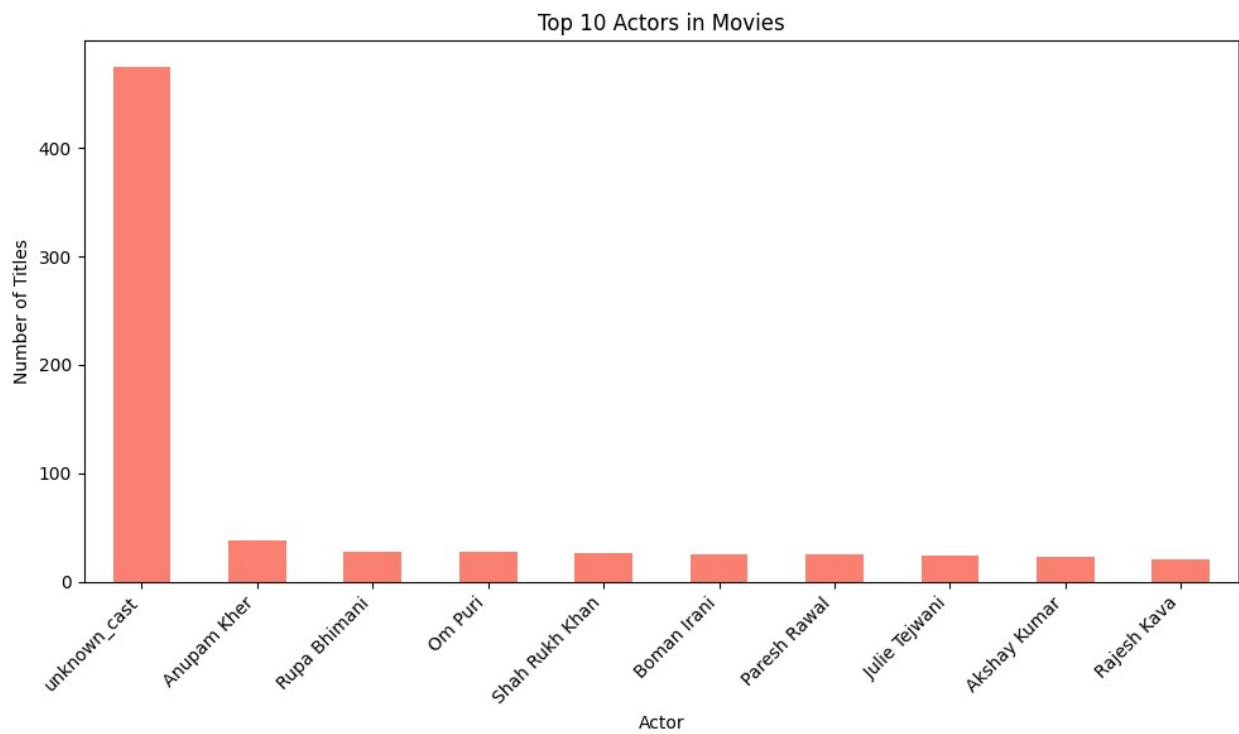
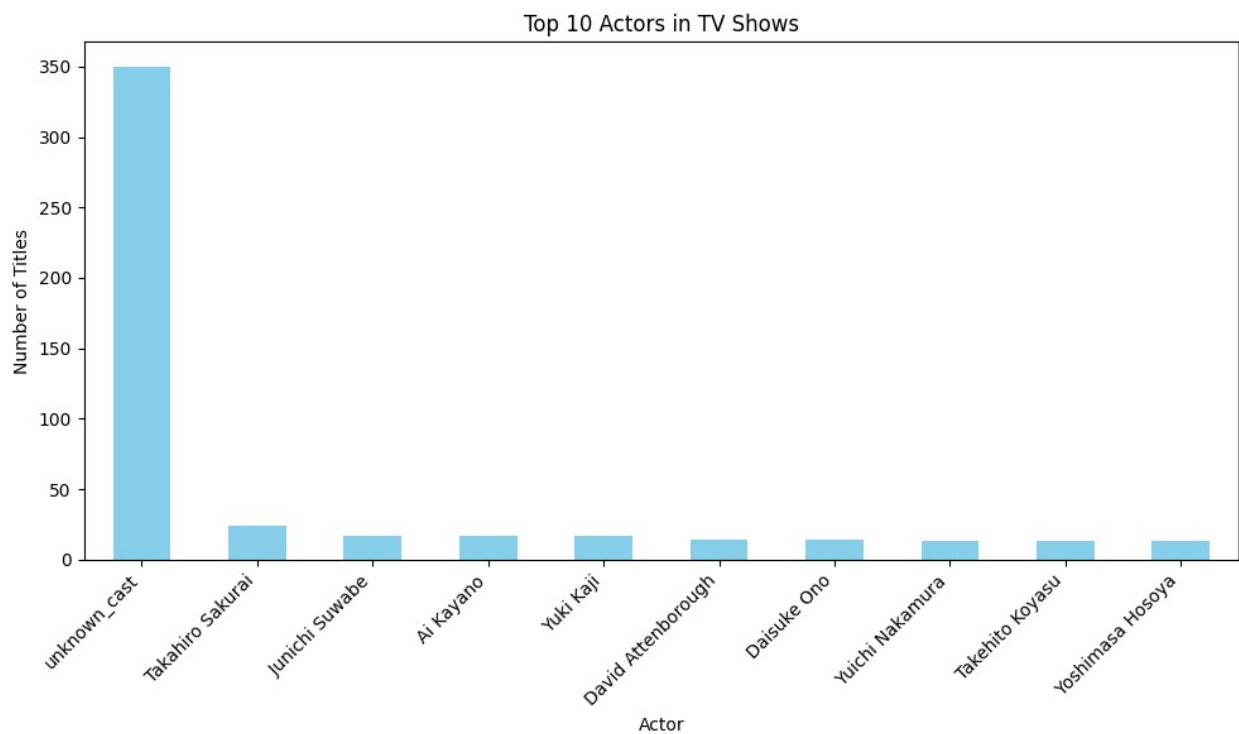
# print("\nTop 10 actors who have appeared in the most movies:")
# print(top_10_movie_actors)
# Plotting the top 10 actors who have appeared in the most TV shows
plt.figure(figsize=(10, 6))
top_10_tv_show_actors.plot(kind='bar', color='skyblue')
plt.title('Top 10 Actors in TV Shows')
plt.xlabel('Actor')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Plotting the top 10 actors who have appeared in the most movies
plt.figure(figsize=(10, 6))
top_10_movie_actors.plot(kind='bar', color='salmon')
plt.title('Top 10 Actors in Movies')
plt.xlabel('Actor')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45, ha='right')

```



```
plt.tight_layout()
plt.show()
```



```

# Create DataFrames for directors in TV shows and movies and count the
# number of unique titles they've directed
tv_show_director_counts = tv_shows.groupby('director')
['title'].nunique()
movie_director_counts = movies.groupby('director')['title'].nunique()

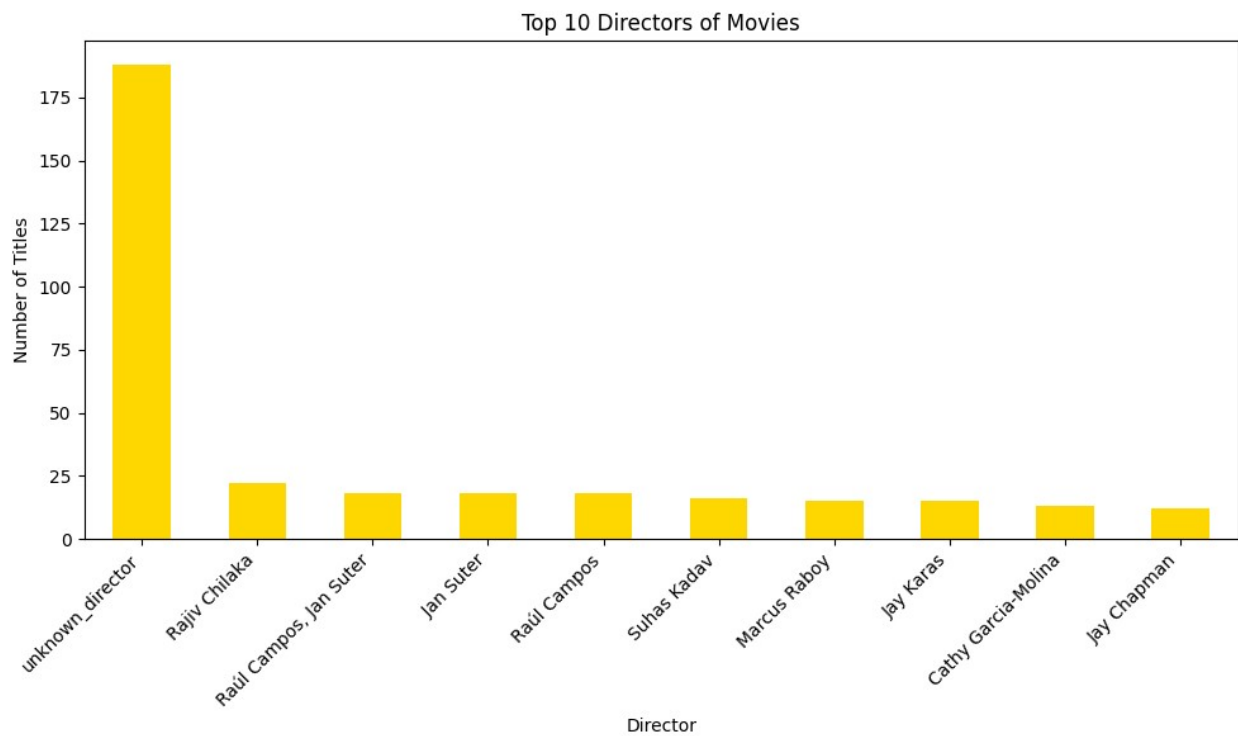
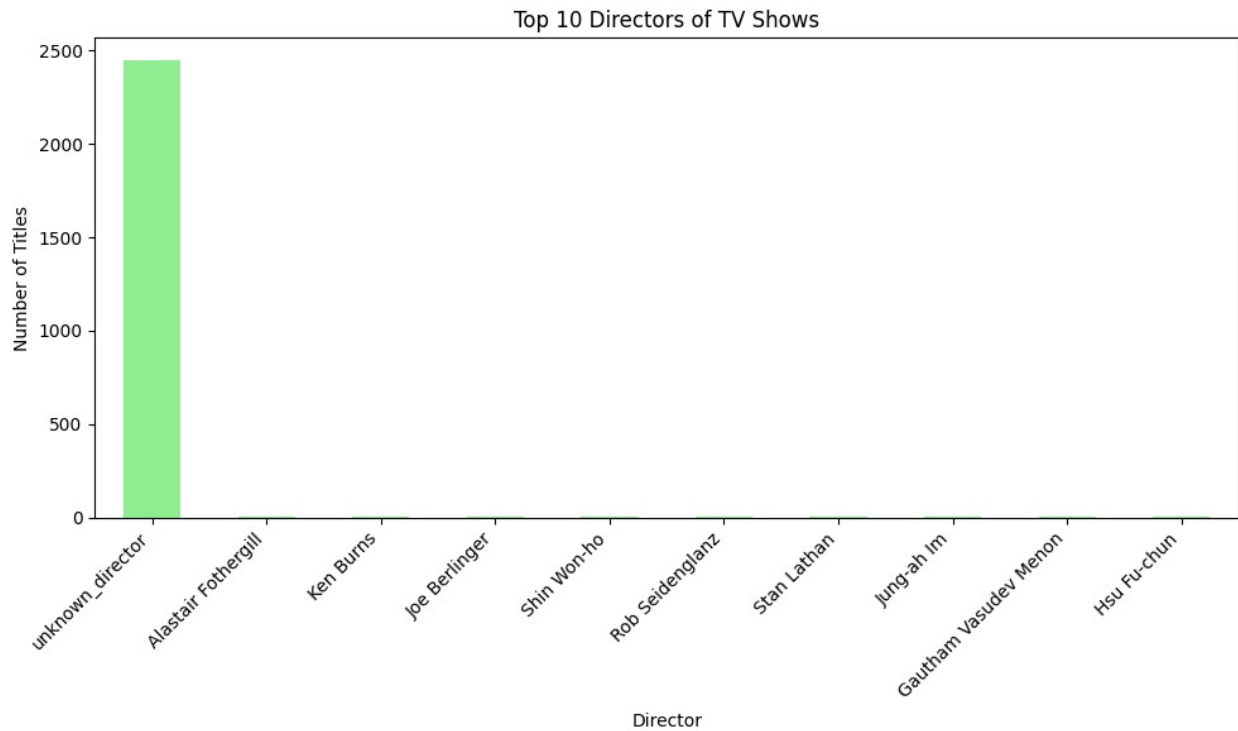
# Select the top 10 directors based on the counts for TV shows and
# movies separately
top_10_tv_show_directors =
tv_show_director_counts.sort_values(ascending=False).head(10)
top_10_movie_directors =
movie_director_counts.sort_values(ascending=False).head(10)

# print("Top 10 directors who have directed the most TV shows:")
# print(top_10_tv_show_directors)

# print("\nTop 10 directors who have directed the most movies:")
# print(top_10_movie_directors)
# Plotting the top 10 directors who have directed the most TV shows
plt.figure(figsize=(10, 6))
top_10_tv_show_directors.plot(kind='bar', color='lightgreen')
plt.title('Top 10 Directors of TV Shows')
plt.xlabel('Director')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Plotting the top 10 directors who have directed the most movies
plt.figure(figsize=(10, 6))
top_10_movie_directors.plot(kind='bar', color='gold')
plt.title('Top 10 Directors of Movies')
plt.xlabel('Director')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

```



# 5. Which genre movies are more popular or produced more  
 # Hint : We want you to apply the word cloud on the genre columns to know which kind  
 # of genre is produced

```
from wordcloud import WordCloud
genres = data["listed_in"]
genre_text = " ".join(genres.dropna())
word_cloud = WordCloud(width=800, height=800,
background_color="white",
                        stopwords=None, min_font_size =
10).generate(genre_text)
# Display the generated word cloud
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(word_cloud)
plt.axis("off")
plt.tight_layout(pad = 0)

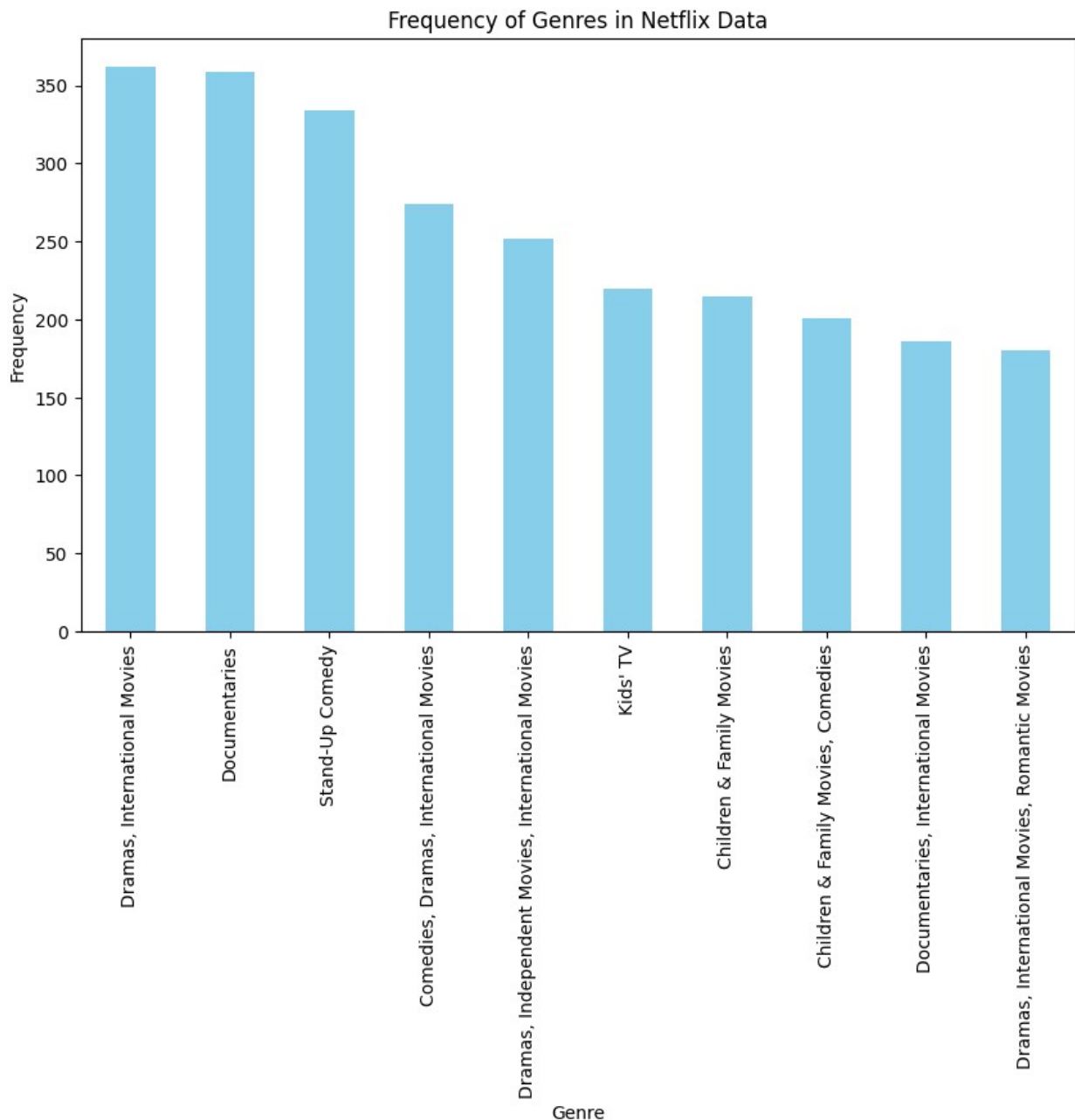
plt.show()
```



```
Kids' TV, Spanish-Language TV Shows, Teen TV Shows      1
Cult Movies, Dramas, Thrillers                          1
Name: listed_in, Length: 514, dtype: int64
```

```
import matplotlib.pyplot as plt
```

```
genre_counts[0:10].plot(kind='bar', figsize=(10, 6), color='skyblue')
plt.title('Frequency of Genres in Netflix Data')
plt.xlabel('Genre')
plt.ylabel('Frequency')
plt.show()
```



```

# 6. Find After how many days the movie will be added to Netflix after
the release of the movie (you can consider the recent past data)
# Hint : We want you to get the difference between the columns having
date added information and release year information and
# get the mode of difference. This will give an insight into what will
be the better time to add in Netflix
# Convert release_date and date_added columns to datetime objects
data['release_year'] = pd.to_datetime(data['release_year'],
format='%Y')
data['date_added'] = pd.to_datetime(data['date_added'])

# Calculate the difference in days between date_added and release_date
data['days_to_netflix'] = (data['date_added'] -
data['release_year']).dt.days
mode_days_to_netflix = data['days_to_netflix'].mode()[0]
mode_days_to_netflix

334.0

```

#6. Insights based on Non-Graphical and Visual Analysis (10 Points)

6.1 Comments on the distribution of the variables and relationship between them

6.2 Comments for each univariate and bivariate plot

```

# 1. Release year has more no of releases from 2000 to 2020
# 2. Movies produced in united states have more popularity and have
more views. Also observed that most watched genre is of
# Dramas, International movies and Documentaries
# 3. The more no of TV shows are produced with Mature Audience Rated
# 4. More no of movies and shows are being release in the month of
December and july respectively

```

#7. Business Insights - Should include patterns observed in the data along with what you can infer from it

```

# 1. Movies produced in united states have more popularity and have
more views. Also observed that most watched genre is of
# Dramas, International movies and Documentaries
# 2. More no of movies and shows are being release in the month of
December and july respectively
# 3. The more no of TV shows are produced with Mature Audience Rated
# 4. A movie takes 334 days to be added into netflix after it's
release date
# 5. More no of TV shows and movies are produced in United states

```

#8. Recommendations - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand

# 1. Focus on Producing Popular Genres: Invest more in producing dramas, international movies, and documentaries as they are the most-watched genres. This can attract a larger audience and increase viewership.

# 2. Strategic Release Timing: Plan to release more movies in July and more TV shows in December to capitalize on the observed trend of increased viewership during these months. This can help maximize exposure and viewership for new content.

# 3. Target Mature Audience: Allocate resources towards producing TV shows targeted at mature audiences since they have been observed to have higher production numbers. This can cater to a specific demographic and potentially increase viewership among mature audiences.

# 4. Optimize Content Addition Timeline: Streamline the process of adding movies to the Netflix library after their release date to reduce the average time lag of 334 days. This can help ensure that new content is available to viewers sooner, enhancing user satisfaction and engagement.

# 5. Diversify Content Sources: While the majority of content is produced in the United States, consider expanding partnerships and acquisitions globally to diversify the content library. This can offer viewers a wider range of choices and attract audiences from different regions.

# By implementing these recommendations, Netflix can capitalize on existing trends, optimize its content strategy, and potentially increase viewership and subscriber engagement.