```
import numpy as np
import pandas as pd
```

```
!pip install --upgrade gdown
```

```
Requirement already satisfied: gdown in /usr/local/lib/python3.10/dist-packages (5.2.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from gdown) (4.12.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from gdown) (3.16.1)
Requirement already satisfied: requests[socks] in /usr/local/lib/python3.10/dist-packages (from gdown) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from gdown) (4.66.6)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->gdown) (2.6)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gd
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdow
```

```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
To: /content/netflix.csv
100% 3.40M/3.40M [00:00<00:00, 20.6MB/s]
```

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

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descripti
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her fatl nears the e of his l filmr
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 cross paths a party, a Ca Town
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 family fror powerful di lc
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feu flirtations a toilet talk down am
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV	In a city coach centers kno

Next steps: Generate code with df View recommended plots New interactive sheet

```
df.head()
```

	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 Cape Town t...

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64 
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
df.columns
```

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

```
df.describe()
```

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

```
df.shape
```

```
(8807, 12)
```

```
df.size
```

```
105684
```

```
df.describe(include=['object'])
```

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

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

	0
show_id	0
type	0
title	0
director	2634
cast	825
country	831
date_added	10
release_year	0
rating	4
duration	3
listed_in	0
description	0

dtype: int64

```
df.isnull().mean()*100
```

	0
show_id	0.000000
type	0.000000
title	0.000000
director	29.908028
cast	9.367549
country	9.435676
date_added	0.113546
release_year	0.000000
rating	0.045418
duration	0.034064
listed_in	0.000000
description	0.000000

dtype: float64

```
df['rating'] = df['rating'].astype('category')
```

```
df.dtypes
```

	0
show_id	object
type	object
title	object
director	object
cast	object
country	object
date_added	object
release_year	int64
rating	category
duration	object
listed_in	object
description	object

dtype: object

```
df['date_added'] = df['date_added'].str.strip()
df['date_added'] = pd.to_datetime(df['date_added'])
```

```
unique_show=df['show_id'].unique()
```

```
print(f'{len(unique_show)}')
```

```
8807
```

```
unique_movie=df['title'].unique()
```

```
print(f'{len(unique_movie)}')
```

```
8807
```

```
movies_counts = df['title'].value_counts()
movies_counts
```

	count
title	
Dick Johnson Is Dead	1
Ip Man 2	1
Hannibal Buress: Comedy Camisado	1
Turbo FAST	1
Masha's Tales	1
...	...
Love for Sale 2	1
ROAD TO ROMA	1
Good Time	1
Captain Underpants Epic Choice-o-Rama	1
Zubaan	1

8807 rows × 1 columns

dtype: int64

```
df[df.duplicated()]
```

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description

```
num_duplicated_rows = df.duplicated().sum()
num_duplicated_rows
```

```
0
```

```
type_counts = df['type'].value_counts()
print("Movies vs. TV Shows:\n", type_counts)
```

```
Movies vs. TV Shows:
 type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

```
df.describe().transpose()
```

	count	mean	min	25%	50%	75%	max	std
date_added	8797	2019-05-17 05:59:08.436967168	2008-01-01 00:00:00	2018-04-06 00:00:00	2019-07-02 00:00:00	2020-08-19 00:00:00	2021-09-25 00:00:00	NaN
release_year	8807.0	2014.180198	1925.0	2013.0	2017.0	2019.0	2021.0	8.819312

```
mean_year = df['release_year'].mean()
median_year = df['release_year'].median()
print("Mean Release Year:", mean_year)
print("Median Release Year:", median_year)
```

```
Mean Release Year: 2014.1801975701146
Median Release Year: 2017.0
```

```
df['director'] = df['director'].fillna("Unknown")
df['cast'] = df['cast'].fillna("Unknown")
df['rating'] = df['rating'].fillna(df['rating'].mode(0))
df['duration'] = df['duration'].fillna(0)
df['country'] = df['country'].fillna("Unknown")
df['date_added'] = df['date_added'].fillna(0)
```

```
df.head()
```

	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	Unknown	United States	2021-09-25 00:00:00	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	Unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24 00:00:00	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

Next steps: Generate code with df View recommended plots New interactive sheet

```
df.cast.value_counts()
```

cast	count
Unknown	825
David Attenborough	19
Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mousam, Swapnil	14
Samuel West	10
Jeff Dunham	7
...	...
Nick Lachey, Vanessa Lachey	1
Takeru Sato, Kasumi Arimura, Haru, Kentaro Sakaguchi, Takayuki Yamada, Kendo Kobayashi, Ken Yasuda, Arata Furuta, Suzuki Matsuo, Koichi Yamadera, Arata Iura, Chikako Kaku, Kotaro Yoshida	1
Toyin Abraham, Sambasa Nzeribe, Chioma Chukwuka Akpotha, Chioma Omeruah, Chiwetalu Agu, Dele Odule, Femi Adebayo, Bayray McNwizu, Biodun Stephen	1
Neeraj Kabi, Geetanjali Kulkarni, Danish Husain, Sheeba Chaddha, Paras Priyadarshan, Anshul Chauhan, Anud Singh Dhaka, Shirin Sewani, Mihir Ahuja, Vasundhara Rajput	1
Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna Malik, Malkeet Rauni, Anita Shabdish, Chittaranjan Tripathy	1

7693 rows × 1 columns

```
column = "cast"
df[df[column].apply(lambda x: ", " in str(x))]
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
1	s2	TV Show	Blood & Water	Unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24 00:00:00	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...	Unknown	2021-09-24 00:00:00	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
4	s5	TV Show	Kota Factory	Unknown	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24 00:00:00	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H	Unknown	2021-09-24 00:00:00	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...

```
column = "country"
df[df[column].apply(lambda x: ", " in str(x))]
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descripti
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	2021-09-24 00:00:00	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	On a pho shoot Ghana, Americ model s
12	s13	Movie	Je Suis Karl	Christian Schwochow	Luna Wedler, Jannis Niewöhner, Milan Peschel, ...	Germany, Czech Republic	2021-09-23 00:00:00	2021	TV-MA	127 min	Dramas, International Movies	After most her family murdered ir ter
29	s30	Movie	Paranoia	Robert Luketic	Liam Hemsworth, Gary Oldman, Amber Heard, Harr...	United States, India, France	2021-09-19 00:00:00	2013	PG-13	106 min	Thrillers	Blackmail by l company CEO, a lo level
38	s39	Movie	Birth of the Dragon	George Nolfi	Billy Magnussen, Ron Yuan, Qu Jingjing, Terry ...	China, Canada, United States	2021-09-16 00:00:00	2017	PG-13	96 min	Action & Adventure, Dramas	A young Bru Lee ange kung traditionalis
					Denzel Washington,	South						Young C

```
column = "listed_in"
df[df[column].apply(lambda x: ", " in str(x))]
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
1	s2	TV Show	Blood & Water	Unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24 00:00:00	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...	Unknown	2021-09-24 00:00:00	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	Unknown	Unknown	Unknown	2021-09-24 00:00:00	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	Unknown	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24 00:00:00	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...
					Kate Siegel,						[illegible]	[illegible]

```
column = "director"
df[df[column].apply(lambda x: ", " in str(x))]
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	desc
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	Unknown	2021-09-24 00:00:00	2021	PG	91 min	Children & Family Movies	E divic br
16	s17	Movie	Europe's Most Dangerous Man: Otto Skorzeny in ...	Pedro de Echave García, Pablo Azorín Williams	Unknown	Unknown	2021-09-22 00:00:00	2020	TV-MA	67 min	Documentaries, International Movies	De di reveal
23	s24	Movie	Go! Go! Cory Carson: Chrissy Takes the Wheel	Alex Woo, Stanley Moore	Maisie Benson, Paul Killam, Kerry Gudjohnsen, ...	Unknown	2021-09-21 00:00:00	2021	TV-Y	61 min	Children & Family Movies	Fro gam hicc
30	s31	Movie	Ankahi Kahaniya	Ashwiny Iyer Tiwari, Abhishek Chaubey, Saket C...	Abhishek Banerjee, Rinku Rajguru, Delzad Hiwal...	Unknown	2021-09-17 00:00:00	2021	TV-14	111 min	Dramas, Independent Movies, International Movies	As b buzze the
68	s69	Movie	Schumacher	Hanns-Bruno Kammertöns, Vanessa Nöcker, Michae...	Michael Schumacher	Unknown	2021-09-15 00:00:00	2021	TV-14	113 min	Documentaries, International Movies, Sports Mo...	interv archi
...	...	...	...	...	...	...	...	...	...	...	...	

```
cast_df = df[["title", "cast"]]
cast_df["unnested_cast"] = cast_df ["cast"].apply(lambda x: str(x).split(", "))
cast_df = cast_df.explode("unnested_cast")
cast_df['cast'] = cast_df['cast'].str.strip()
cast_df.head(10)
```

```
<ipython-input-81-0a1074e8351c>:2: 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
  cast_df["unnested_cast"] = cast_df ["cast"].apply(lambda x: str(x).split(", "))
```

	title	cast	unnested_cast
0	Dick Johnson Is Dead	Unknown	Unknown
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Ama Qamata
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Khosi Ngema
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Gail Mabalane
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Thabang Molaba
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Dillon Windvogel
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Natasha Thahane
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Arno Greeff
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Xolile Tshabalala
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Getmore Sithole

Next steps: Generate code with cast_df View recommended plots New interactive sheet

```
country_df = df[["title", "country"]]
country_df["unnested_country"] = country_df ["country"].apply(lambda x: str(x).split(", "))
country_df = country_df.explode("unnested_country")
country_df.head(10)
```

```
<ipython-input-82-2552acf7a06d>:2: 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
  country_df["unnested_country"] = country_df ["country"].apply(lambda x: str(x).split(", "))
```

	title	country	unnested_country
0	Dick Johnson Is Dead	United States	United States
1	Blood & Water	South Africa	South Africa
2	Ganglands	Unknown	Unknown
3	Jailbirds New Orleans	Unknown	Unknown
4	Kota Factory	India	India
5	Midnight Mass	Unknown	Unknown
6	My Little Pony: A New Generation	Unknown	Unknown
7	Sankofa	United States, Ghana, Burkina Faso, United Kin...	United States
7	Sankofa	United States, Ghana, Burkina Faso, United Kin...	Ghana

Next steps: Generate code with country_df | View recommended plots | New interactive sheet

```
merge_df = pd.merge(left=cast_df, right=country_df, on="title")
merge_df.head()
```

	title	cast	unnested_cast	country	unnested_country
0	Dick Johnson Is Dead	Unknown	Unknown	United States	United States
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Ama Qamata	South Africa	South Africa
2	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Khosi Ngema	South Africa	South Africa
3	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Gail Mabalane	South Africa	South Africa
4	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	Thabang Molaba	South Africa	South Africa

Next steps: Generate code with merge_df | View recommended plots | New interactive sheet

```
listed_in_df = df[["title", "listed_in"]]
listed_in_df["unnested_listed_in"] = listed_in_df ["listed_in"].apply(lambda x: str(x).split(", "))
listed_in_df = listed_in_df.explode("unnested_listed_in")
listed_in_df.head(10)
```

```
<ipython-input-84-7235ff7161d1>:2: 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
  listed_in_df["unnested_listed_in"] = listed_in_df ["listed_in"].apply(lambda x: str(x).split(", "))
```

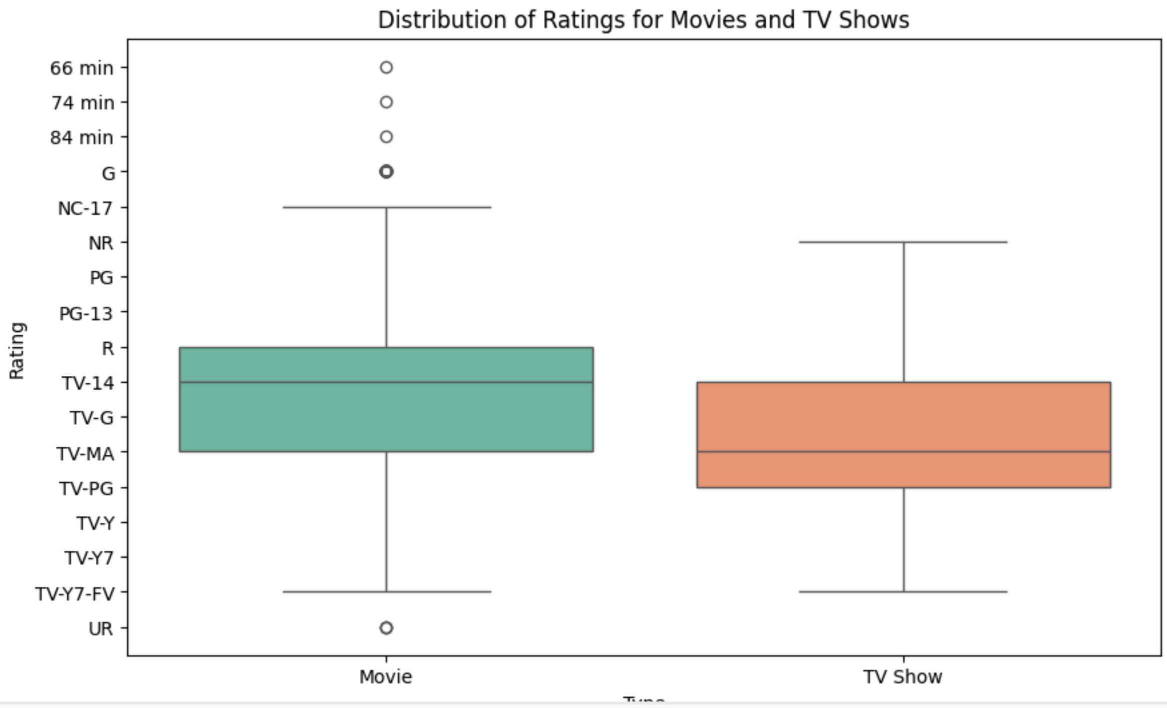
	title	listed_in	unnested_listed_in
0	Dick Johnson Is Dead	Documentaries	Documentaries
1	Blood & Water	International TV Shows, TV Dramas, TV Mysteries	International TV Shows
1	Blood & Water	International TV Shows, TV Dramas, TV Mysteries	TV Dramas
1	Blood & Water	International TV Shows, TV Dramas, TV Mysteries	TV Mysteries
2	Ganglands	Crime TV Shows, International TV Shows, TV Act...	Crime TV Shows
2	Ganglands	Crime TV Shows, International TV Shows, TV Act...	International TV Shows
2	Ganglands	Crime TV Shows, International TV Shows, TV Act...	TV Action & Adventure
3	Jailbirds New Orleans	Docuseries, Reality TV	Docuseries
3	Jailbirds New Orleans	Docuseries, Reality TV	Reality TV

Next steps: Generate code with listed_in_df | View recommended plots | New interactive sheet

Box plot

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

plt.figure(figsize=(10, 6))
sns.boxplot(x='type', y='rating', data=df, palette='Set2')
plt.title('Distribution of Ratings for Movies and TV Shows')
plt.xlabel('Type')
plt.ylabel('Rating')
plt.show()
```

```
<ipython-input-85-4d054de3f047>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and

  sns.boxplot(x='type', y='rating', data=df, palette='Set2')
```



Comment: The box plot reveals the spread and distribution of data, showing the range from minimum to maximum values, with quartiles indicating the interquartile range. It also highlights potential skewness in the data and points out anomalies that may need further investigation, such as unusually short or long durations of TV shows or movies. This visual is valuable for making quick decisions about trends and outliers in the dataset.

```
print(df['duration'].head(20))
print(df['duration'].isna().sum())
```

```
0        90 min
1     2 Seasons
2      1 Season
3      1 Season
4     2 Seasons
5      1 Season
6        91 min
7       125 min
8     9 Seasons
9       104 min
10     1 Season
11     1 Season
12      127 min
13       91 min
14     1 Season
15    4 Seasons
16       67 min
17    2 Seasons
18       94 min
19     1 Season
```

```
Name: duration, dtype: object
0
```

```
df['duration'] = df['duration'].fillna('0 Unknown')

df['duration'] = df['duration'].astype(str)


df[['Minutes', 'Units']] = df['duration'].str.split(' ', n=1, expand=True)

df[df['Units'].isna()]
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	Mi
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04 00:00:00	2017	74 min	0	Movies	Louis C.K. muses on religion, eternal love, gi...	

```
df['Units'].fillna('Unknown', inplace=True)


df['Minutes'].fillna(0, inplace=True)

df['Minutes'] = pd.to_numeric(df['Minutes'], errors='coerce')
```

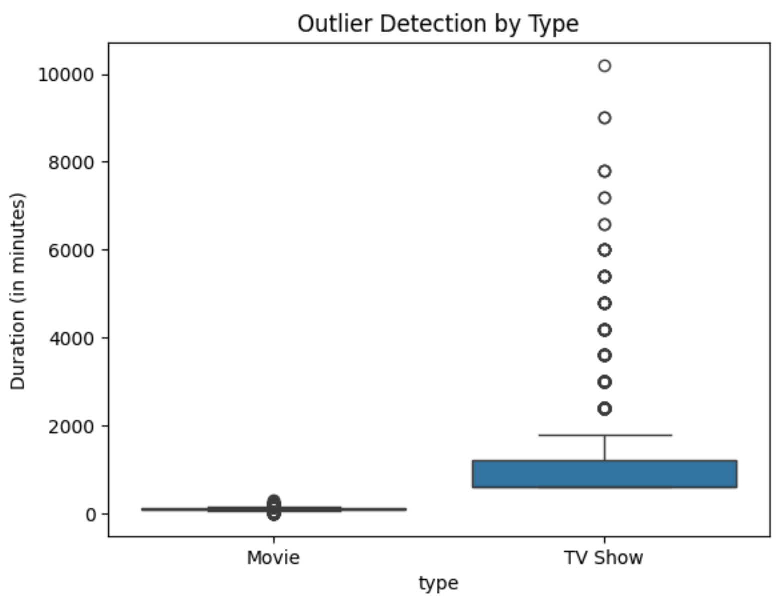
```
<ipython-input-89-458d51d7e774>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through cha
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  df['Units'].fillna('Unknown', inplace=True)
<ipython-input-89-458d51d7e774>:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through cha
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  df['Minutes'].fillna(0, inplace=True)
```

```
def calculate_duration(row):
    if 'Season' in row['Units']:
        return row['Minutes'] * 600
    elif 'min' in row['Units']:
        return row['Minutes']
    else:
        return None


df['Duration (in minutes)'] = df.apply(calculate_duration, axis=1)


print(df[['duration', 'Minutes', 'Units', 'Duration (in minutes)']].head())
```

```
    duration  Minutes    Units  Duration (in minutes)
0     90 min       90      min                   90.0
1  2 Seasons        2  Seasons                 1200.0
2   1 Season        1   Season                  600.0
3   1 Season        1   Season                  600.0
4  2 Seasons        2  Seasons                 1200.0
```

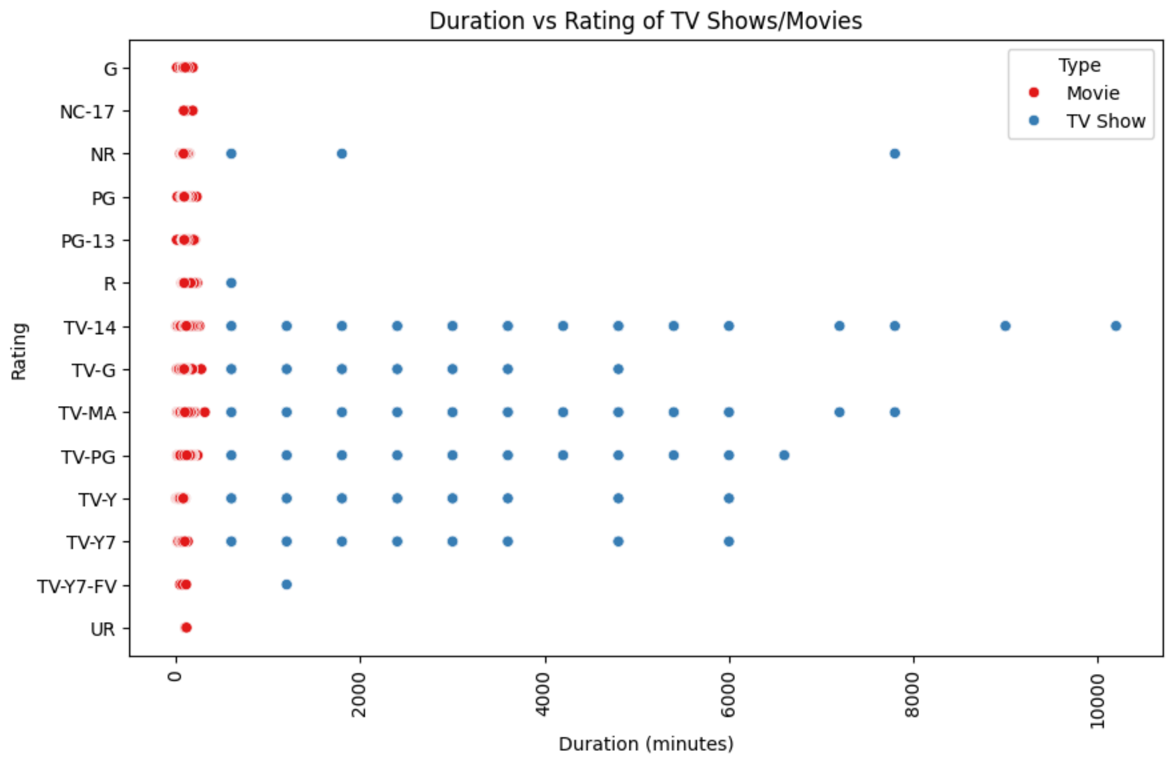
Outlier Detection

```
sns.boxplot(x='type', y='Duration (in minutes)', data=df)
plt.title('Outlier Detection by Type')
plt.show()
```



Scatter Plot

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

plt.figure(figsize=(10,6))
sns.scatterplot(data=df, x='Duration (in minutes)', y='rating', hue='type', palette='Set1')
plt.title('Duration vs Rating of TV Shows/Movies')
plt.xlabel('Duration (minutes)')
plt.ylabel('Rating')
plt.xticks(rotation=90)
plt.legend(title='Type', loc='upper right')
plt.show()
```



Comment: The scatter plot shows the relationship between Duration and Rating for Netflix content, revealing how content length might impact its rating. TV Shows and Movies are compared by color, showing potentially distinct patterns between the two. Outliers, like long movies with high ratings, may provide insights into audience preferences for content length.

```
type_counts = df['type'].value_counts()
print("Counts of each category in 'type':\n", type_counts)
```

```
Counts of each category in 'type':
 type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

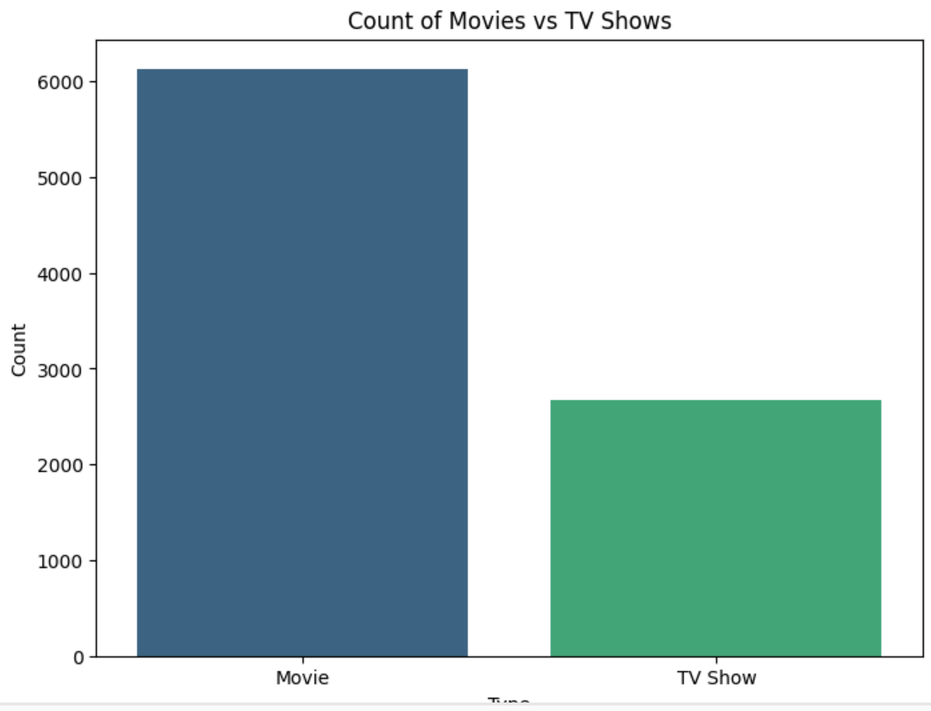
```
rating_counts = df['rating'].value_counts()
print("\nCounts of each category in 'rating':\n", rating_counts)
```

```

Counts of each category in 'rating':
 rating
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
UR             3
NC-17          3
74 min         1
84 min         1
66 min         1
Name: count, dtype: int64
```

count plot

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

# Count plot for 'type'
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='type', palette='viridis')
plt.title("Count of Movies vs TV Shows")
plt.xlabel("Type")
plt.ylabel("Count")
plt.show()
```

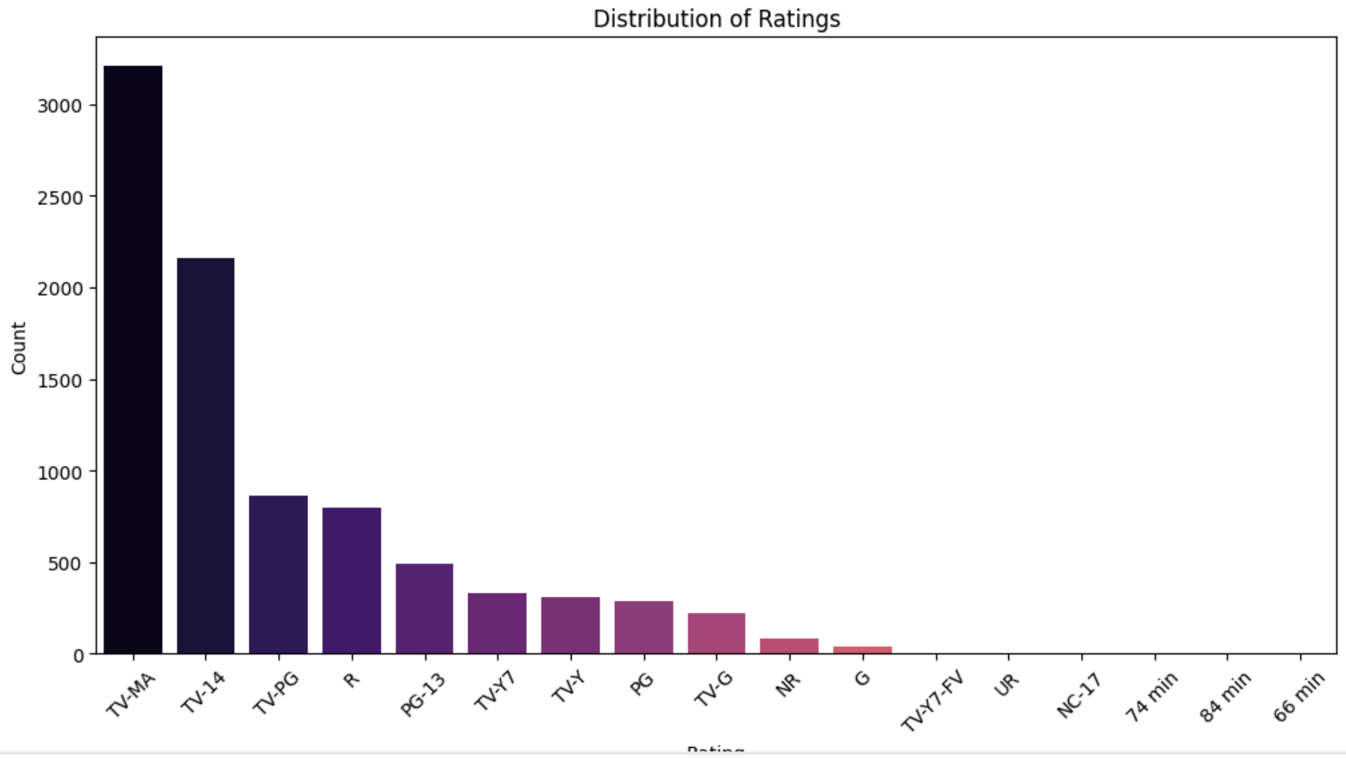
```
<ipython-input-96-e01b5a960d04>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and

  sns.countplot(data=df, x='type', palette='viridis')
```



Comment: Movies make up a larger proportion of Netflix content compared to TV Shows.This could indicate a greater emphasis on single-unit entertainment over episodic formats or reflect the ease of acquiring movie licenses.

```
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='rating', order=rating_counts.index, palette='magma')
plt.title("Distribution of Ratings")
plt.xlabel("Rating")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.show()
```

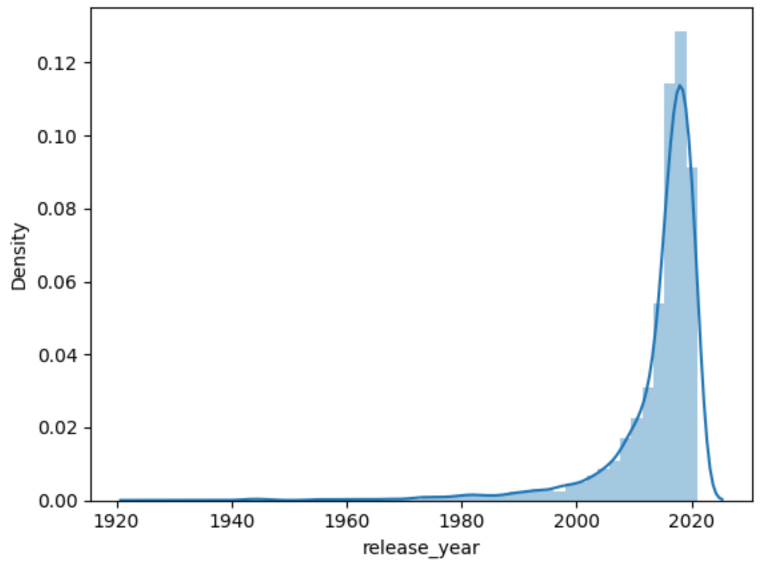
```
<ipython-input-97-484cb3f188d3>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and

  sns.countplot(data=df, x='rating', order=rating_counts.index, palette='magma')
```



Dist Plot

```
sns.distplot(df['release_year'])
```

```
<ipython-input-98-5635d90732bd>:1: 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(df['release_year'])
<Axes: xlabel='release_year', ylabel='Density'>
```



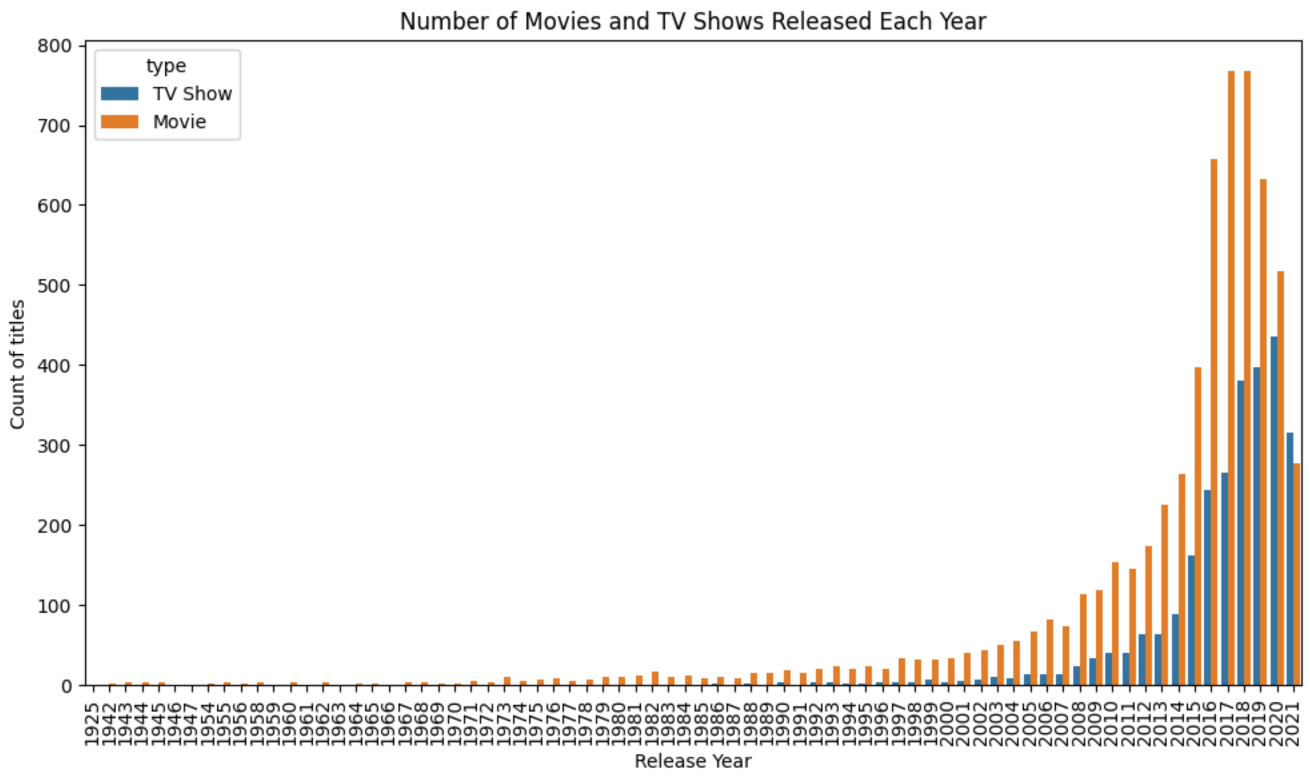
Comment: The distribution plot for release_year reveals the concentration of Netflix content based on its release year. Recent years tend to dominate, indicating Netflix's focus on newer, contemporary content, while earlier years show lower frequencies, reflecting limited availability of older titles or strategic content choices.

```
tv_shows = df[df['type'] == 'TV Show'].groupby('release_year').size()
movies = df[df['type'] == 'Movie'].groupby('release_year').size()
print(tv_shows)
print(movies)
```

```
1946      1
1963      1
1967      1
1972      1
1974      1
1977      1
1979      1
1981      1
1985      1
1986      2
1988      2
1989      1
1990      3
1991      1
1992      3
1993      4
1994      2
1995      2
1996      3
1997      4
1998      4
1999      7
2000      4
2001      5
2002      7
2003     10
2004      9
2005     13
2006     14
2007     14
2008     23
2009     34
2010     40
2011     40
2012     64
2013     63
2014     88
2015    162
2016    244
2017    265
2018    380
2019    397
2020    436
2021    315
dtype: int64
release_year
1942      2
1943      3
1944      3
1945      3
1946      1
       ...
2017    767
2018    767
2019    633
2020    517
2021    277
Length: 73, dtype: int64
```

Bar Plot

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

yearly_counts = df.groupby(['release_year', 'type']).size().reset_index(name='Count')

plt.figure(figsize=(10, 6))
sns.barplot(data=yearly_counts, x='release_year', y='Count', hue='type')
plt.title('Number of Movies and TV Shows Released Each Year')
plt.xlabel('Release Year')
plt.ylabel('Count of titles')
plt.xticks(rotation=90)
plt.legend(title='type')
plt.tight_layout()
plt.show()
```



Comment: Content production shows an upward trend, especially post-2015, reflecting Netflix's aggressive strategy for original productions and global market expansion.

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

df['loading_year']=df['date_added'].dt.year
df['loading_year']
```

	loading_year
0	2021
1	2021
2	2021
3	2021
4	2021
...	...
8802	2019
8803	2019
8804	2019
8805	2020
8806	2019

8807 rows × 1 columns

dtype: int32

```
df_temp=df[['loading_year','release_year']]
df_temp
```

	loading_year	release_year
0	2021	2020
1	2021	2021
2	2021	2021
3	2021	2021
4	2021	2021
...	...	...
8802	2019	2007
8803	2019	2018
8804	2019	2009
8805	2020	2006
8806	2019	2015

8807 rows × 2 columns

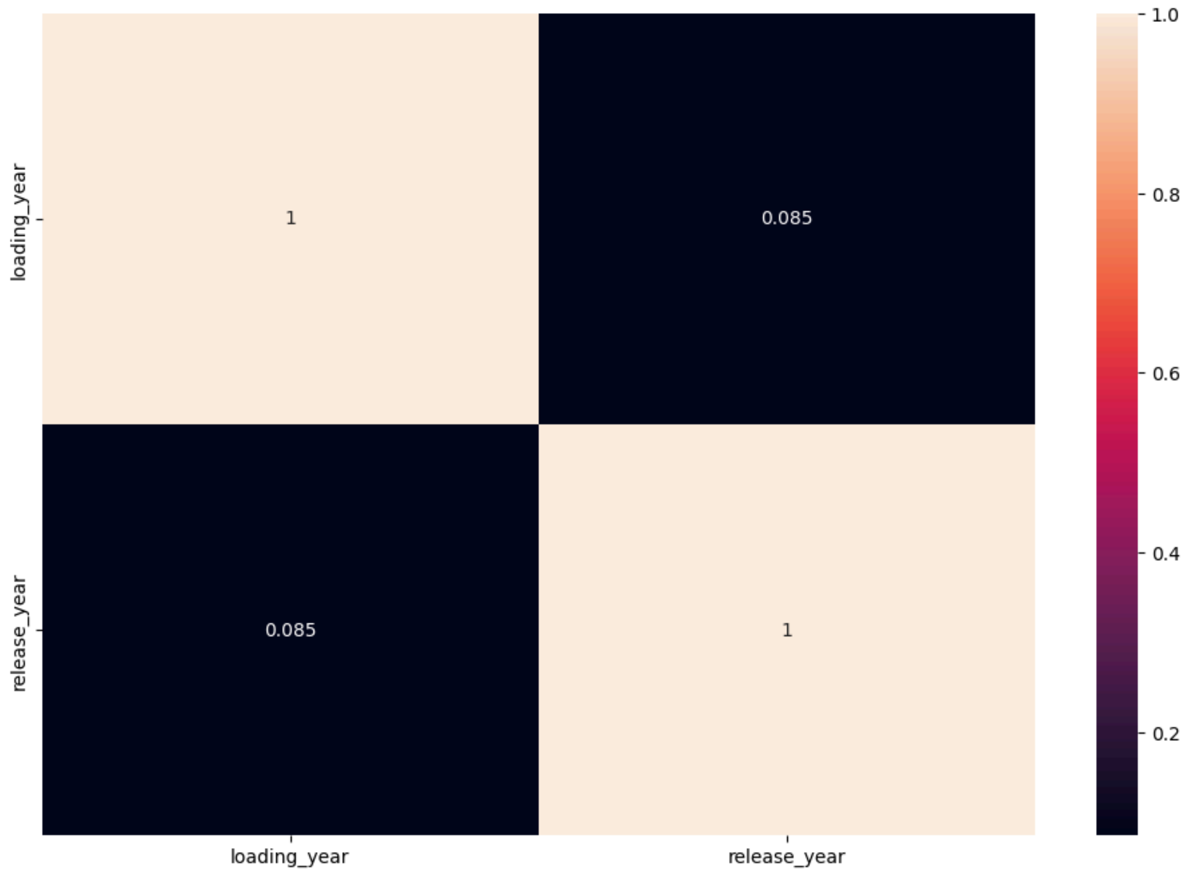
Next steps: Generate code with df_temp View recommended plots New interactive sheet

```
df_temp.corr()
```

	loading_year	release_year
loading_year	1.000000	0.085007
release_year	0.085007	1.000000

Heat Map

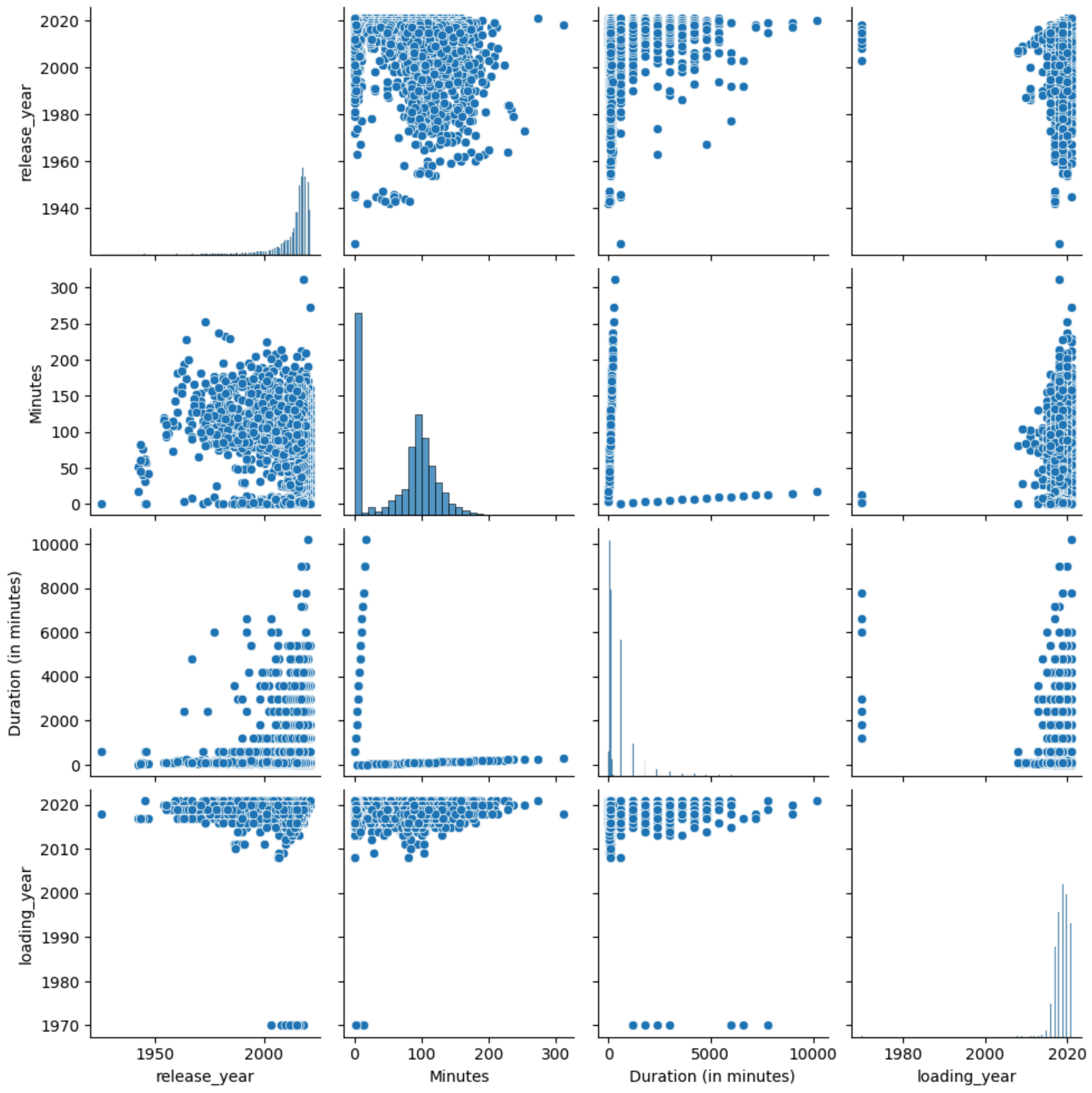
```
plt.figure(figsize=(12,8))
sns.heatmap(df_temp.corr(), annot=True)
plt.show()
```



Pair Plot

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7f45b360c7f0>
```



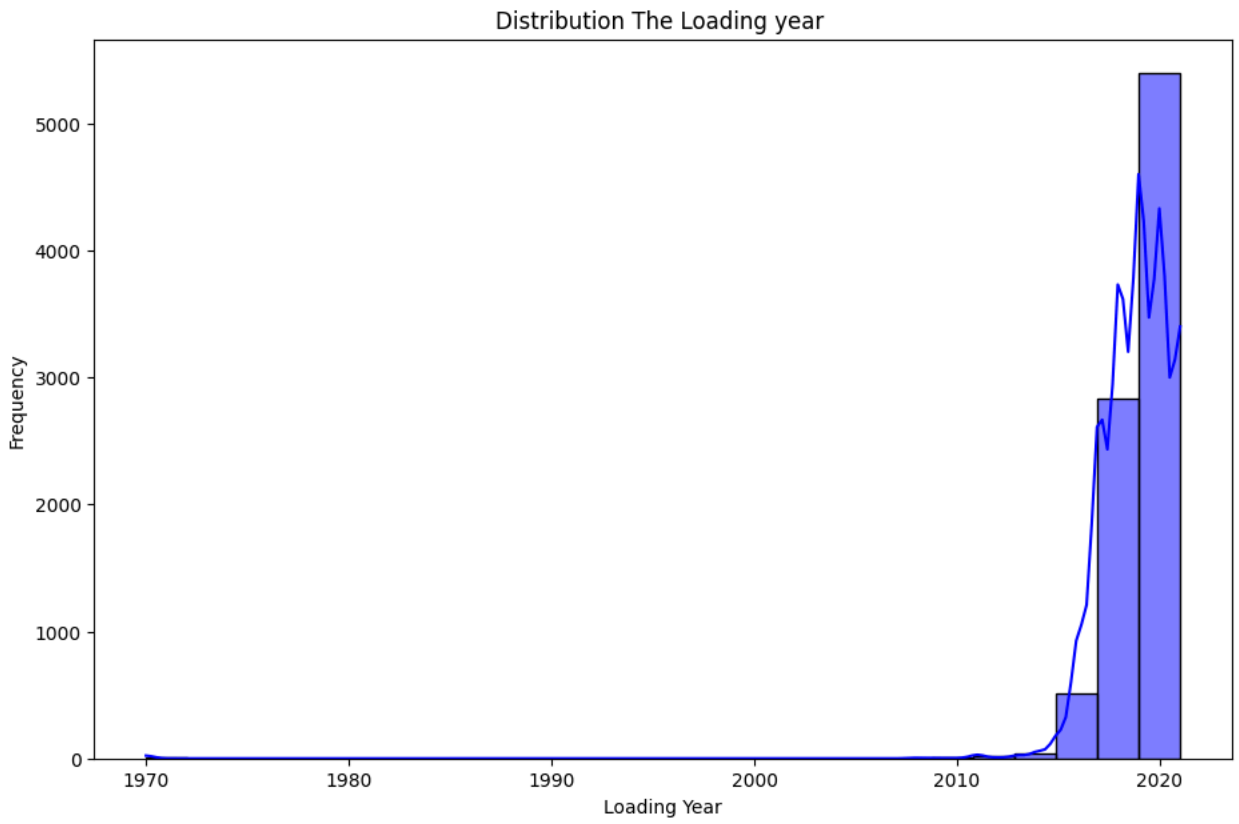
Comment: This pair plot for Netflix data provides a visual representation of pairwise relationships between numerical variables.

```
df['loading_year'].value_counts().reset_index(name='count')
```

	loading_year	count
0	2019	2016
1	2020	1879
2	2018	1649
3	2021	1498
4	2017	1188
5	2016	429
6	2015	82
7	2014	24
8	2011	13
9	2013	11
10	1970	10
11	2012	3
12	2009	2
13	2008	2
14	2010	1

Histogram

```
plt.figure(figsize=(11,7))
sns.histplot(data=df, x='loading_year', bins=25, kde=True, color='blue')
plt.title("Distribution The Loading year ")
plt.xlabel('Loading Year')
plt.ylabel('Frequency')
plt.show()
```



comments: The histogram shows a significant increase in content production in recent years, with a peak around 2018-2020. This reflects Netflix's ramped-up production and acquisition strategy during this period. Older content is less prevalent, suggesting a focus on more contemporary releases.

```
country_counts = df['country'].value_counts()
print(country_counts.head(10))
```

```
country
United States     2818
India              972
Unknown            831
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Name: count, dtype: int64
```

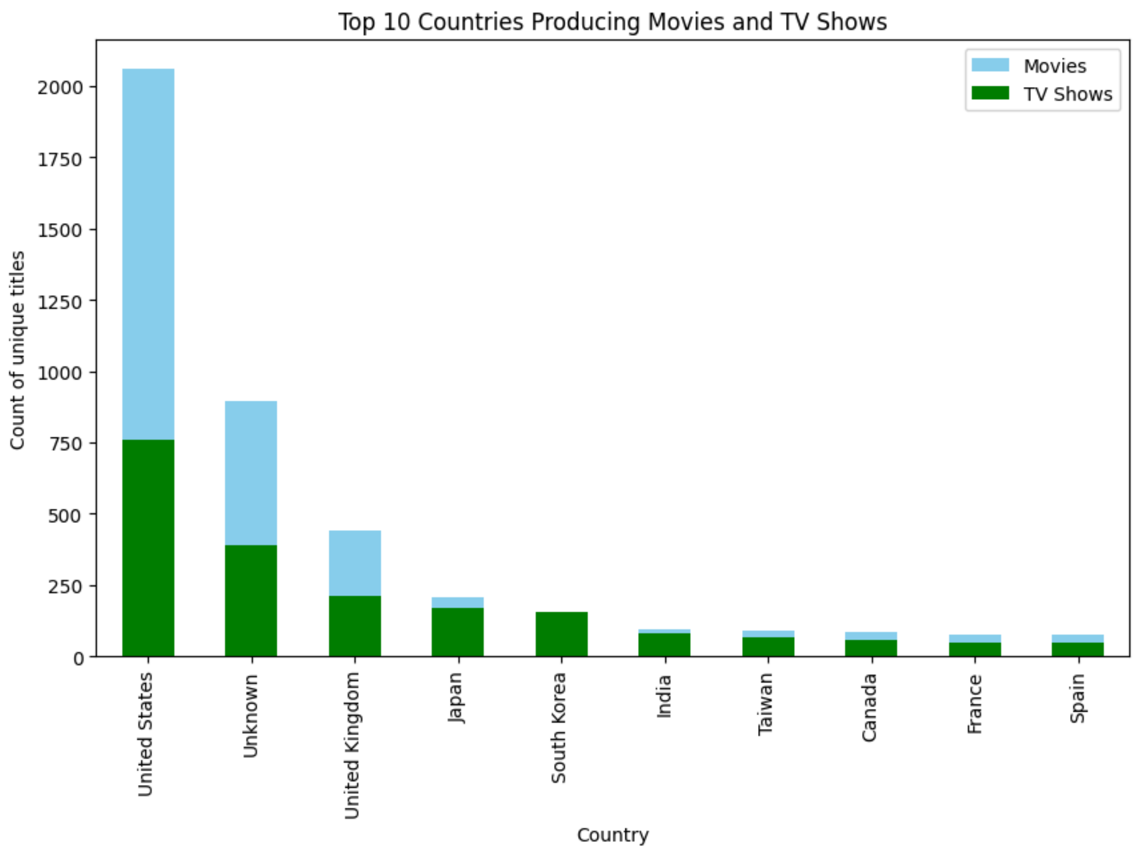
```
movies = df[df['type'] == 'Movie']
movies_per_country = movies.groupby('country')['title'].nunique()
top_10_movie_countries = movies_per_country.sort_values(ascending=False).head(10)
print(top_10_movie_countries)
```

```
country
United States     2058
India              893
Unknown            440
United Kingdom     206
Canada             122
Spain               97
Egypt               92
Nigeria             86
Indonesia           77
Japan               76
Name: title, dtype: int64
```

```
tv_shows = df[df['type'] == 'TV Show']
tv_shows_per_country = tv_shows.groupby('country')['title'].nunique()
top_10_tv_countries = tv_shows_per_country.sort_values(ascending=False).head(10)
print(top_10_tv_countries)
```

```
country
United States     760
Unknown           391
United Kingdom    213
Japan             169
South Korea       158
India              79
Taiwan             68
Canada             59
France             49
Spain              48
Name: title, dtype: int64
```

```
import matplotlib.pyplot as plt

top_10_movie_countries.plot(kind='bar', color='skyblue', label='Movies', figsize=(10, 6))
top_10_tv_countries.plot(kind='bar', color='green', label='TV Shows')
plt.title('Top 10 Countries Producing Movies and TV Shows')
plt.xlabel('Country')
plt.ylabel('Count of unique titles')
plt.legend()
plt.show()
```



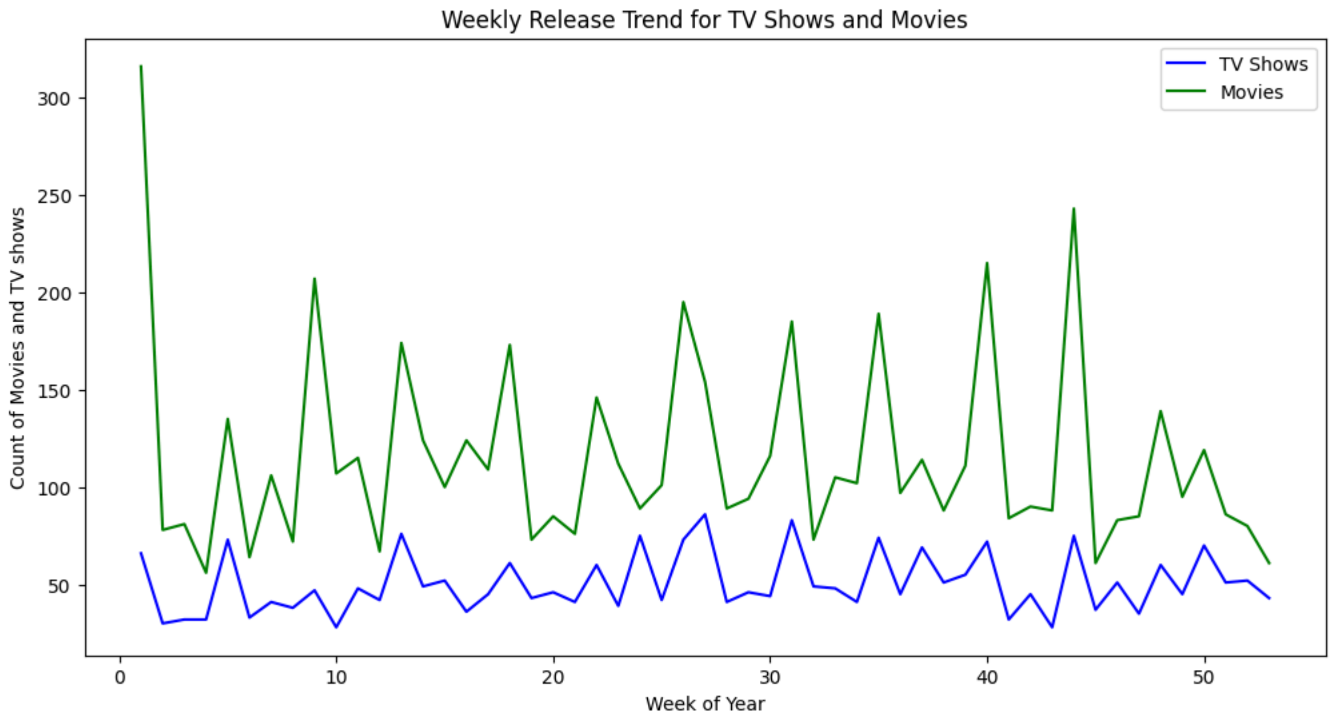
Comment: The U.S. leads in both TV Show and Movie production, reflecting its dominance in global entertainment. Countries like India and South Korea feature prominently, likely due to their burgeoning entertainment industries and Netflix's investment in regional content.

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

df['week'] = df['date_added'].dt.isocalendar().week
df['month'] = df['date_added'].dt.month

tv_shows = df[df['type'] == 'TV Show']
movies = df[df['type'] == 'Movie']

tv_shows_weekly = tv_shows.groupby('week').size()
movies_weekly = movies.groupby('week').size()
print(tv_shows_weekly)
print(movies_weekly)
```

```
week
1     66
2     30
3     32
4     32
5     73
6     33
7     41
8     38
9     47
10    28
11    48
12    42
13    76
14    49
15    52
16    36
17    45
18    61
19    43
20    46
21    41
22    60
23    39
24    75
25    42
```

```
26    73
27    86
28    41
29    46
30    44
31    83
32    49
33    48
34    41
35    74
36    45
37    69
38    51
39    55
40    72
41    32
42    45
43    28
44    75
45    37
46    51
47    35
48    60
49    45
50    70
51    51
52    52
53    43
dtype: int64
week
1     316
```

```
plt.figure(figsize=(12,6))
sns.lineplot(x=tv_shows_weekly.index, y=tv_shows_weekly.values, label='TV Shows', color='blue')
sns.lineplot(x=movies_weekly.index, y=movies_weekly.values, label='Movies', color='green')
plt.title('Weekly Release Trend for TV Shows and Movies')
plt.xlabel('Week of Year')
plt.ylabel('Count of Movies and TV shows')
plt.legend()
plt.show()
```

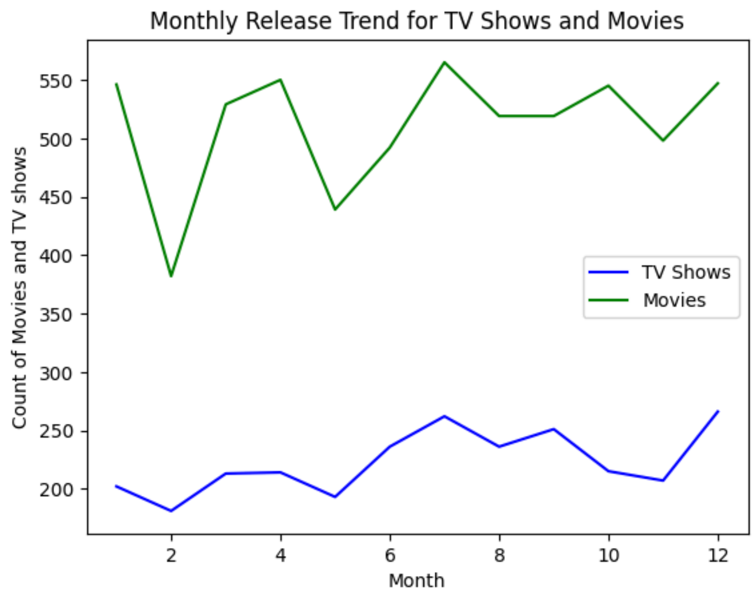


comment: The trend reveals specific weeks where there is a noticeable increase in movie releases. These spikes could correspond to major holidays, festivals, or events when audience demand is higher (e.g., Christmas, summer holidays). Weeks with consistent releases indicate Netflix's strategy to maintain audience engagement by providing a steady flow of content. Some weeks may show low or no releases, possibly reflecting off-peak periods where new releases may have less impact or coincide with limited production timelines.

```
df['month'] = df['date_added'].dt.month

tv_shows_monthly = tv_shows.groupby('month').size()
movies_monthly = movies.groupby('month').size()
print(tv_shows_monthly)
print(movies_monthly)
```

```
month
1     202
2     181
3     213
4     214
5     193
6     236
7     262
8     236
9     251
10    215
11    207
12    266
dtype: int64
month
1     546
2     382
3     529
4     550
5     439
6     492
7     565
8     519
9     519
10    545
11    498
12    547
dtype: int64
```

```
sns.lineplot(x=tv_shows_monthly.index, y=tv_shows_monthly.values, label='TV Shows', color='blue')
sns.lineplot(x=movies_monthly.index, y=movies_monthly.values, label='Movies', color='green')
plt.title('Monthly Release Trend for TV Shows and Movies')
plt.xlabel('Month')
plt.ylabel('Count of Movies and TV shows')
plt.legend()
plt.show()
```



comment: The trend shows specific months, like December and July, with higher movie releases. These months likely coincide with holidays and vacation periods, targeting maximum viewership. Some months, such as February or September, might exhibit fewer releases. This could reflect production cycles or periods where audience engagement is relatively lower. The overall trend may indicate that Netflix aligns content strategies with seasonal behaviors, focusing on peak times like winter breaks and summer vacations to release blockbuster movies.

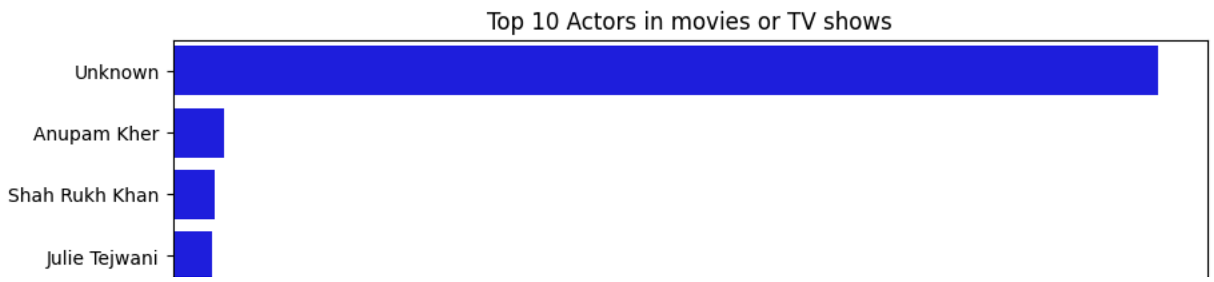
```
top_actors = cast_df.groupby('unnested_cast')['title'].nunique().sort_values(ascending=False).head(10)
print(top_actors)
```

```
unnested_cast
Unknown              825
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
Name: title, dtype: int64
```

```
directors = df[['title', 'director']]
directors = directors.assign(director=directors['director'].str.split(',')).explode('director')
directors['director'] = directors['director'].str.strip()

top_directors = directors.groupby('director')['title'].nunique().sort_values(ascending=False).head(10)
print(top_directors)
```

```
director
Unknown                 2634
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
Name: title, dtype: int64
```

```
#plotting top 10 Actors
plt.figure(figsize=(10, 6))
sns.barplot(x=top_actors.values, y=top_actors.index, color="blue")
plt.title('Top 10 Actors in movies or TV shows')
plt.xlabel('Count of unique titles')
plt.ylabel('Actor')
plt.show()
```


Top 10 Actors in movies or TV shows
Unknown
Anupam Kher
Shah Rukh Khan
Julie Tejwani