Business Case: Netflix - Data Exploration & Visualization

About NETFLIX:

Netflix is one of the most popular media and video streaming platforms. They have over 10000 movies or tv shows available on their platform, as of mid-2021, they have over 222M Subscribers globally. This tabular dataset consists of listings of all the movies and tv shows available on Netflix, along with details such as - cast, directors, ratings, release year, duration, etc.

Business Objective:

Netflix wants to identify what types of content (TV Shows/Movies) to produce and how to expand its business across countries, using data-driven insights.

Problem Statement:

Netflix wants to identify the types of content that perform best globally and how to expand its presence across countries. By analyzing historical data on shows/movies, genres, and release trends, we aim to provide actionable insights.

Importing Libraries

In [196…
```
# importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# %matplotlib notebook
%matplotlib inline
```

In [197…
```
df = pd.read_csv('netflix.csv')
df.head()
```

Out[197…

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

In [198…
```
# length of the data
len(df)
```

Out[198…
```
8807
```

In [199…
```
# shape of the data
df.shape
```

Out[199…
```
(8807, 12)
```

In [200…
```
# checking the data types
df.dtypes
```

Out[200…
```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

In [201…
```
# number of unique values in the data
for i in df.columns:
    print(i, ': ', df[i].nunique())
```

```
show_id :  8807
type :  2
title :  8807
director :  4528
cast :  7692
country :  748
date_added :  1767
release_year :  74
rating :  17
duration :  220
listed_in :  514
description :  8775
```

In [202…
```
# Unique values in the type column
df['type'].unique()
```

Out[202…
```
array(['Movie', 'TV Show'], dtype=object)
```

In [203…
```
# checking for missing values in each column
df.isnull().sum()
```

Out[203…
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

In [204…
```
# Count occurrences in 'type' column
df['type'].value_counts()
```

Out[204…
```
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

In [205…
```
# Count occurrences in 'rating' column
df['rating'].value_counts()
```

Out[205…
```
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
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: count, dtype: int64
```

In [206…
```
# sumamry of the datase
df.info()
```

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

In [207…
```
# count of total duplicate rows in the dataset
df.duplicated().sum()
```

Out[207…
```
0
```

In [208…
```
# descriptive statistics for numerical columns
df.describe()
```

Out[208…

	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

In [209…

```
# descriptive statistics for categorical columns
df.describe(include = 'object')
```

Out[209…

	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…
freq	1	6131	1	19	19	2818	109	3207	1793	362	4

In [210…

```
df.select_dtypes(include='object')
```

Out[210…

	show_id	type	title	director	cast	country	date_added	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	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	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...
...	...	...	...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	TV-Y7	2 Seasons	Kids' TV, Korean TV Shows, TV Comedies	While living alone in a spooky town, a young g...
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	R	88 min	Comedies, Horror Movies	Looking to survive in a world taken over by zo...
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

8807 rows × 11 columns

In [211…

```
# Unnesting the director column
constraint1 = df['director'].apply(lambda x: str(x).split(', ')).to_list()
director_df = pd.DataFrame(constraint1, index = df['title'])
director_df = director_df.stack()
director_df = director_df.reset_index()
director_df.rename(columns = {0: 'Directors'}, inplace = True)
director_df.drop(['level_1'], axis = 1, inplace = True)
director_df.head()
```

Out[211…

	title	Directors
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan

In [212…

```
# Unnesting the cast column
constraint2 = df['cast'].apply(lambda x: str(x).split(', ')).to_list()
cast_df = pd.DataFrame(constraint2, index = df['title'])
cast_df = cast_df.stack()
cast_df = cast_df.reset_index()
cast_df.rename(columns = {0: 'Actors'}, inplace = True)
```

```
cast_df.drop(['level_1'], axis = 1, inplace = True)
cast_df.head()
```

Out[212…

	title	Actors
0	Dick Johnson Is Dead	nan
1	Blood & Water	Ama Qamata
2	Blood & Water	Khosi Ngema
3	Blood & Water	Gail Mabalane
4	Blood & Water	Thabang Molaba

In [213…

```
# Unnesting the country column
constraint3 = df['country'].apply(lambda x: str(x).split(', ')).to_list()
country_df = pd.DataFrame(constraint3, index = df['title'])
country_df = country_df.stack()
country_df = country_df.reset_index()
country_df.rename(columns = {0: 'Country'}, inplace = True)
country_df.drop(['level_1'], axis = 1, inplace = True)
country_df.head()
```

Out[213…

	title	Country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	nan
3	Jailbirds New Orleans	nan
4	Kota Factory	India

In [214…

```
# Unnesting the genre column
constraint4 = df['listed_in'].apply(lambda x: str(x).split(', ')).to_list()
genre_df = pd.DataFrame(constraint4, index = df['title'])
genre_df = genre_df.stack()
genre_df = genre_df.reset_index()
genre_df.rename(columns = {0: 'Genre'}, inplace = True)
genre_df.drop(['level_1'], axis = 1, inplace = True)
genre_df.head()
```

Out[214…

	title	Genre
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows
2	Blood & Water	TV Dramas
3	Blood & Water	TV Mysteries
4	Ganglands	Crime TV Shows

In [215…

```
# merging the unnested directors with unnested cast
merged_df1 = cast_df.merge(director_df, on = ['title'], how = 'inner')

# merging the already merged df with the unnested genre
merged_df2 = merged_df1.merge(genre_df, on = ['title'], how = 'inner')

# merging the already merged df with the unnested country
df_new = merged_df2.merge(country_df, on = ['title'], how = 'inner')
df_new.head()
```

Out[215…

	title	Actors	Directors	Genre	Country
0	Dick Johnson Is Dead	nan	Kirsten Johnson	Documentaries	United States
1	Blood & Water	Ama Qamata	nan	International TV Shows	South Africa
2	Blood & Water	Ama Qamata	nan	TV Dramas	South Africa
3	Blood & Water	Ama Qamata	nan	TV Mysteries	South Africa
4	Blood & Water	Khosi Ngema	nan	International TV Shows	South Africa

In [216…

```
# nan in Actors, Directors, Genre, Country
print('nan in Actors: ', (df_new['Actors'] == 'nan').sum())
print('nan in Directors: ',(df_new['Directors'] == 'nan').sum())
print('nan in Genre: ',(df_new['Genre'] == 'nan').sum())
print('nan in Country: ',(df_new['Country'] == 'nan').sum())
```

```
nan in Actors:  2146
nan in Directors:  50643
nan in Genre:  0
nan in Country:  11897
```

In [217…

```
# Replace the nan in Actors and Directors with Unknown
df_new['Actors'] = df_new['Actors'].replace(['nan'], ['Unknown Actor'])
df_new['Directors'] = df_new['Directors'].replace(['nan'], ['Unknown Director'])

# Replace the nan in country with the np.nan
df_new['Country'] = df_new['Country'].replace(['nan'], [np.nan])
df_new.head()
```

Out[217…

	title	Actors	Directors	Genre	Country
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa
3	Blood & Water	Ama Qamata	Unknown Director	TV Mysteries	South Africa
4	Blood & Water	Khosi Ngema	Unknown Director	International TV Shows	South Africa

In [218…

```
df.columns
```

Out[218…

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

In [219…

```
# merging the unnested new_df with the original df
df_final = df_new.merge(df[['show_id', 'type', 'title', 'date_added', 'release_year', 'rating', 'duration']], on = ['title'],
df_final.head()
```

Out[219…

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating	duration
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90 min
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
3	Blood & Water	Ama Qamata	Unknown Director	TV Mysteries	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
4	Blood & Water	Khosi Ngema	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons

In [220…

```
# Check for the null in the df
df_final.isnull().sum()
```

Out[220…

```
title             0
Actors            0
Directors         0
Genre             0
Country       11897
show_id           0
type              0
date_added      158
release_year      0
rating           67
duration          3
dtype: int64
```

In [221…

```
df_final[df_final['duration'].isnull()]
```

Out[221…

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating	duration
126537	Louis C.K. 2017	Louis C.K.	Louis C.K.	Movies	United States	s5542	Movie	April 4, 2017	2017	74 min	NaN
131603	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	Movies	United States	s5795	Movie	September 16, 2016	2010	84 min	NaN
131737	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	Movies	United States	s5814	Movie	August 15, 2016	2015	66 min	NaN

The duration column values are shifted left to the rating column and that is why the duration is NaN

In [222…

```
# Filling the null values in duration with the values in rating column
df_final.loc[df_final['duration'].isnull(), 'duration'] = df_final.loc[df_final['duration'].isnull(), 'duration'].fillna(df_fi
```

```
df_final.isnull().sum()
```

Out[222…

```
title              0
Actors             0
Directors          0
Genre              0
Country        11897
show_id            0
type               0
date_added       158
release_year       0
rating            67
duration           0
dtype: int64
```

In [223…

```
# Filling the null values in rating with 'NR'
# Replacing the values in rating column which contains 'min' with 'NR'
df_final.loc[df_final['rating'].str.contains('min', na=False), 'rating'] = 'NR'
df_final.fillna({'rating': 'NR'}, inplace=True)
df_final.isnull().sum()
```

Out[223…

```
title              0
Actors             0
Directors          0
Genre              0
Country        11897
show_id            0
type               0
date_added       158
release_year       0
rating             0
duration           0
dtype: int64
```

In [224…

```
# Checking for the null values in date_added column
df_final[df_final['date_added'].isnull()].head()
```

Out[224…

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating	duration
136893	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136894	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	TV Comedies	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136895	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	TV Dramas	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136896	A Young Doctor's Notebook and Other Stories	Jon Hamm	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136897	A Young Doctor's Notebook and Other Stories	Jon Hamm	Unknown Director	TV Comedies	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons

In [225…

```
# Checking the release_year of the null values in date_added column
df_final[df_final['date_added'].isnull()]['release_year'].value_counts()
```

Out[225…

```
release_year
2015    40
2008    36
2013    27
2003    24
2012    24
2016     5
2018     1
2010     1
Name: count, dtype: int64
```

In [226…

```
# Filling the null values in date_added with the mode of the respective release_year
for i in df_final[df_final['date_added'].isnull()]['release_year'].unique():
    imp = df_final.loc[df_final['release_year'] == i, 'date_added'].mode()[0]
    # print(i, ': ', imp)
    df_final.loc[df_final['release_year'] == i, 'date_added'] = df_final.loc[df_final['release_year'] == i, 'date_added'].fill
```

In [227…

```
# Checking the null values in the df
df_final.isnull().sum()
```

Out[227…
```
title              0
Actors             0
Directors          0
Genre              0
Country        11897
show_id            0
type               0
date_added         0
release_year       0
rating             0
duration           0
dtype: int64
```

In [228…
```
# Checking the Directors of the null values in country column
df_final[df_final['Country'].isnull()]['Directors'].value_counts()
```

Out[228…
```
Directors
Unknown Director     4927
Hidenori Inoue        153
Suhas Kadav           129
Yoshiyuki Tomino      129
S.S. Rajamouli        120
                     ...
Christopher Guest       1
Chris Howe              1
Paul Dugdale            1
Paul M. Green           1
Storm Theunissen        1
Name: count, Length: 373, dtype: int64
```

In [229…
```
# Checking the Actors of the null values in country column
df_final[df_final['Country'].isnull()]['Actors'].value_counts()
```

Out[229…
```
Actors
Unknown Actor    316
Julie Tejwani     32
Rupa Bhimani      31
Rajesh Kava       22
Toru Furuya       21
                ...
Frank Grillo       1
Mike Epps          1
Wanda Sykes        1
Samin Nosrat       1
Ketan Kava         1
Name: count, Length: 3939, dtype: int64
```

In [230…
```
# Checking the Directors of the null values in country column
# Filling the null values in country with the mode of the respective Directors
for i in df_final[df_final['Country'].isnull()]['Directors'].unique():
  if i in df_final[~df_final['Country'].isnull()]['Directors'].unique():
    imp=df_final[df_final['Directors']==i]['Country'].mode().values[0]
    df_final.loc[df_final['Directors']==i,'Country']=df_final.loc[df_final['Directors']==i,'Country'].fillna(imp)
```

In [231…
```
df_final.isnull().sum()
```

Out[231…
```
title              0
Actors             0
Directors          0
Genre              0
Country         4276
show_id            0
type               0
date_added         0
release_year       0
rating             0
duration           0
dtype: int64
```

In [232…
```
# Filling the null values in country with the mode of the respective Actors
for i in df_final[df_final['Country'].isnull()]['Actors'].unique():
  if i in df_final[~df_final['Country'].isnull()]['Actors'].unique():
    imp=df_final[df_final['Actors']==i]['Country'].mode().values[0]
    df_final.loc[df_final['Actors']==i,'Country']=df_final.loc[df_final['Actors']==i,'Country'].fillna(imp)
```

In [233…
```
df_final.isnull().sum()
```

Out[233…
```
title              0
Actors             0
Directors          0
Genre              0
Country         2069
show_id            0
type               0
date_added         0
release_year       0
rating             0
duration           0
dtype: int64
```

In [234…
```
# Filling the remaining null values in country with 'Unknown Country'
df_final['Country'].fillna('Unknown Country',inplace=True)
df_final.isnull().sum()
```

```
C:\Users\Vasanth Murugan\AppData\Local\Temp\ipykernel_520\1539638812.py:2: FutureWarning: A value is trying to be set on a copy
of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are set
ting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = d
f[col].method(value) instead, to perform the operation inplace on the original object.


  df_final['Country'].fillna('Unknown Country',inplace=True)
```

Out[234…
```
title           0
Actors          0
Directors       0
Genre           0
Country         0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
dtype: int64
```

In [235…
```
df_final['duration'].unique()
```

Out[235…
```
array(['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',
       '179 min', '11 Seasons', '43 min', '200 min', '196 min', '167 min',
       '178 min', '228 min', '18 min', '205 min', '201 min', '191 min'],
      dtype=object)
```

In [236…
```
# Extracting the duration in minutes from duration column
df_final['duration_minutes'] = df_final['duration'].str.extract(r'(\d+)\s*min', expand=False).fillna(0).astype(int)
```

In [237…
```
# Extracting number of seasons from duration column
df_final['seasons'] = df_final['duration'].str.extract(r'(\d+)\s*Season[s]?', expand=False).fillna(0).astype(int)
```

In [238…
```
df_final.sample(20)
```

Out[238…

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating	duration	dura
116049	Swearnet Live	Mike Smith	John Paul Tremblay	Comedies	Canada	s5016	Movie	March 1, 2018	2014	TV-MA	75 min	
1338	InuYasha the Movie 4: Fire on the Mystic Island	Kumiko Watanabe	Toshiya Shinohara	International Movies	Japan	s54	Movie	September 15, 2021	2004	TV-PG	88 min	
32733	Invisible City	Alessandra Negrini	Unknown Director	Crime TV Shows	Brazil	s1340	TV Show	February 5, 2021	2021	TV-MA	1 Season	
121982	The Distinguished Citizen	Nora Navas	Gastón Duprat	Comedies	Argentina	s5311	Movie	September 1, 2017	2016	TV-MA	113 min	
187976	The Breadwinner	Shaista Latif	Nora Twomey	Dramas	India	s8226	Movie	February 20, 2018	2017	PG-13	93 min	
79251	Let It Snow	D'Arcy Carden	Luke Snellin	LGBTQ Movies	United States	s3297	Movie	November 8, 2019	2019	PG-13	93 min	
76294	Teasing Master Takagi-san	Fukushi Ochiai	Unknown Director	Anime Series	Japan	s3174	TV Show	December 6, 2019	2019	TV-PG	1 Season	
170255	Miss Me This Christmas	Brely Evans	Kenny Young	Romantic Movies	United States	s7467	Movie	December 19, 2017	2017	TV-14	85 min	
114059	Amateur	Josh Charles	Ryan Koo	Dramas	United States	s4936	Movie	April 6, 2018	2018	TV-MA	96 min	
110413	The Could've-Gone-All-the-Way Committee	Mayuko Fukuda	Unknown Director	Romantic TV Shows	Japan	s4743	TV Show	August 1, 2018	2018	TV-14	1 Season	
188440	The CEO	Peter King Nzioki Mwania	Kunle Afolayan	Thrillers	Nigeria	s8242	Movie	September 1, 2019	2016	TV-14	109 min	
166698	Lion's Heart	Hassan Hosny	Karim El Sobky	Action & Adventure	Egypt	s7310	Movie	May 9, 2019	2013	TV-14	111 min	
95246	That Thing Called Tadhana	JM de Guzman	Antoinette Jadaone	International Movies	Philippines	s4041	Movie	March 7, 2019	2015	TV-MA	91 min	
152470	Eyyvah Eyyvah 2	Salih Kalyon	Hakan Algül	International Movies	Turkey	s6727	Movie	March 10, 2017	2011	TV-PG	106 min	
105050	Rake	Kate Box	Unknown Director	Crime TV Shows	Australia	s4505	TV Show	October 16, 2018	2018	TV-MA	5 Seasons	
32788	Little Big Women	Chang Han	Joseph Hsu	International Movies	Taiwan	s1341	Movie	February 5, 2021	2020	TV-14	123 min	
76614	V Wars	Bo Martyn	Unknown Director	TV Action & Adventure	United States	s3183	TV Show	December 5, 2019	2019	TV-MA	1 Season	
37274	Incarnate	Aaron Eckhart	Brad Peyton	Horror Movies	United States	s1537	Movie	December 16, 2020	2016	PG-13	87 min	
22715	In Our Mothers' Gardens	Unknown Actor	Shantrelle P. Lewis	Documentaries	United States	s908	Movie	May 7, 2021	2020	TV-MA	85 min	
151836	Enter the Dragon	Bruce Lee	Robert Clouse	Action & Adventure	United States	s6702	Movie	January 1, 2021	1973	R	103 min	

In [239…
```
df_final.drop(['duration'], axis = 1, inplace = True)
df_final.head()
```

Out[239…

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating	duration_minutes	seasons
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90	0
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	0	2
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	0	2
3	Blood & Water	Ama Qamata	Unknown Director	TV Mysteries	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	0	2
4	Blood & Water	Khosi Ngema	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	0	2

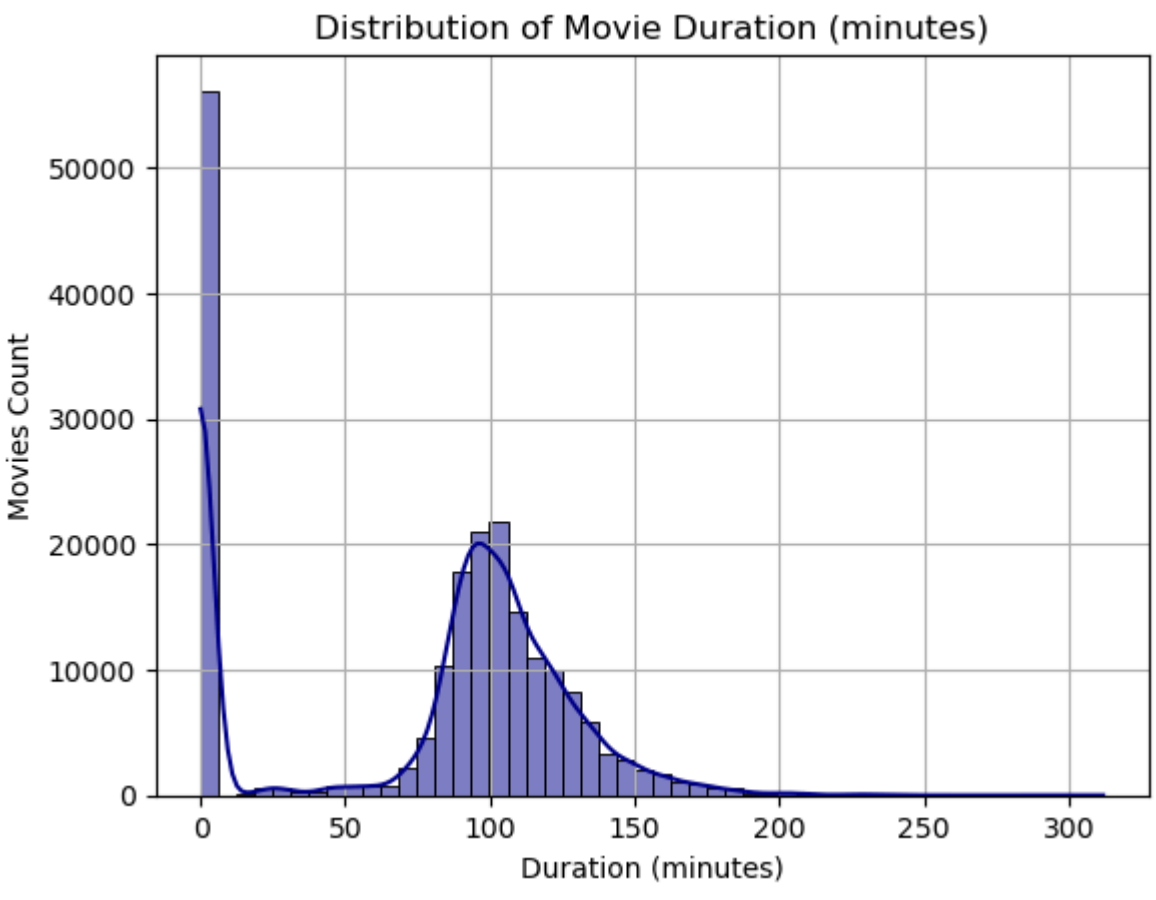
In [240…
```
df_final['duration_minutes'].describe()
```

Out[240…
```
count    201991.000000
mean         77.152789
std          52.269154
min           0.000000
25%           0.000000
50%          95.000000
75%         112.000000
max         312.000000
Name: duration_minutes, dtype: float64
```

In [241…
```
df_final['seasons'].describe()
```

Out[241…
```
count    201991.000000
mean          0.535960
std           1.287839
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max          17.000000
Name: seasons, dtype: float64
```

In [242…
```
# Visualizing the distribution of movie durations
sns.histplot(df_final['duration_minutes'], kde=True, bins=50, color='darkblue')
plt.title('Distribution of Movie Duration (minutes)')
plt.xlabel('Duration (minutes)')
plt.ylabel('Movies Count')
plt.grid(True)
plt.show()
```



Most movies on Netflix have a duration between 80 and 120 minutes, with a sharp drop-off for longer durations.

In [243…
```
# Creating duration bins
bins1 = [-1,1,50,80,100,120,150,200,315]
labels1 = ['<1','1-50','50-80','80-100','100-120','120-150','150-200','200-315']
df_final['duration'] = pd.cut(df_final['duration_minutes'],bins=bins1,labels=labels1)
```

In [244…
```
# Converting the date_added column to datetime format
# Splitting the date_added column into month, year and weekday
```

```
df_final['date_added'] = pd.to_datetime(df_final['date_added'], format= 'mixed')
df_final['month_added'] = df_final['date_added'].dt.month_name()
df_final['year_added'] = df_final['date_added'].dt.year
df_final['weekday_added'] = df_final['date_added'].dt.day_name()
```

In [245…
```
df_final[df_final['title'].str.contains('\(')]
```

Out[245…

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating	duration_minutes
1833	Tughlaq Durbar (Telugu)	Vijay Sethupathi	Delhiprasad Deenadayalan	Comedies	India	s80	Movie	2021-09-11	2021	TV-14	145
1834	Tughlaq Durbar (Telugu)	Vijay Sethupathi	Delhiprasad Deenadayalan	Dramas	India	s80	Movie	2021-09-11	2021	TV-14	145
1835	Tughlaq Durbar (Telugu)	Vijay Sethupathi	Delhiprasad Deenadayalan	International Movies	India	s80	Movie	2021-09-11	2021	TV-14	145
1836	Tughlaq Durbar (Telugu)	Parthiban	Delhiprasad Deenadayalan	Comedies	India	s80	Movie	2021-09-11	2021	TV-14	145
1837	Tughlaq Durbar (Telugu)	Parthiban	Delhiprasad Deenadayalan	Dramas	India	s80	Movie	2021-09-11	2021	TV-14	145
…	…	…	…	…	…	…	…	…	…	…	…
197875	Trikal (Past, Present, Future)	Naseeruddin Shah	Shyam Benegal	Dramas	India	s8629	Movie	2019-12-31	1985	TV-14	134
197876	Trikal (Past, Present, Future)	Naseeruddin Shah	Shyam Benegal	Independent Movies	India	s8629	Movie	2019-12-31	1985	TV-14	134
197877	Trikal (Past, Present, Future)	Kulbhushan Kharbanda	Shyam Benegal	Comedies	India	s8629	Movie	2019-12-31	1985	TV-14	134
197878	Trikal (Past, Present, Future)	Kulbhushan Kharbanda	Shyam Benegal	Dramas	India	s8629	Movie	2019-12-31	1985	TV-14	134
197879	Trikal (Past, Present, Future)	Kulbhushan Kharbanda	Shyam Benegal	Independent Movies	India	s8629	Movie	2019-12-31	1985	TV-14	134

1165 rows × 16 columns

In [246…
```
# Removing the text within parentheses from the title column
df_final['title'] = df_final['title'].str.replace(r"\(.*\)","", regex=True)
df_final.head()
```

Out[246…

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating	duration_minutes	seasons	d
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	2021-09-25	2020	PG-13	90	0	
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	0	2	
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	0	2	
3	Blood & Water	Ama Qamata	Unknown Director	TV Mysteries	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	0	2	
4	Blood & Water	Khosi Ngema	Unknown Director	International TV Shows	South Africa	s2	TV Show	2021-09-24	2021	TV-MA	0	2	

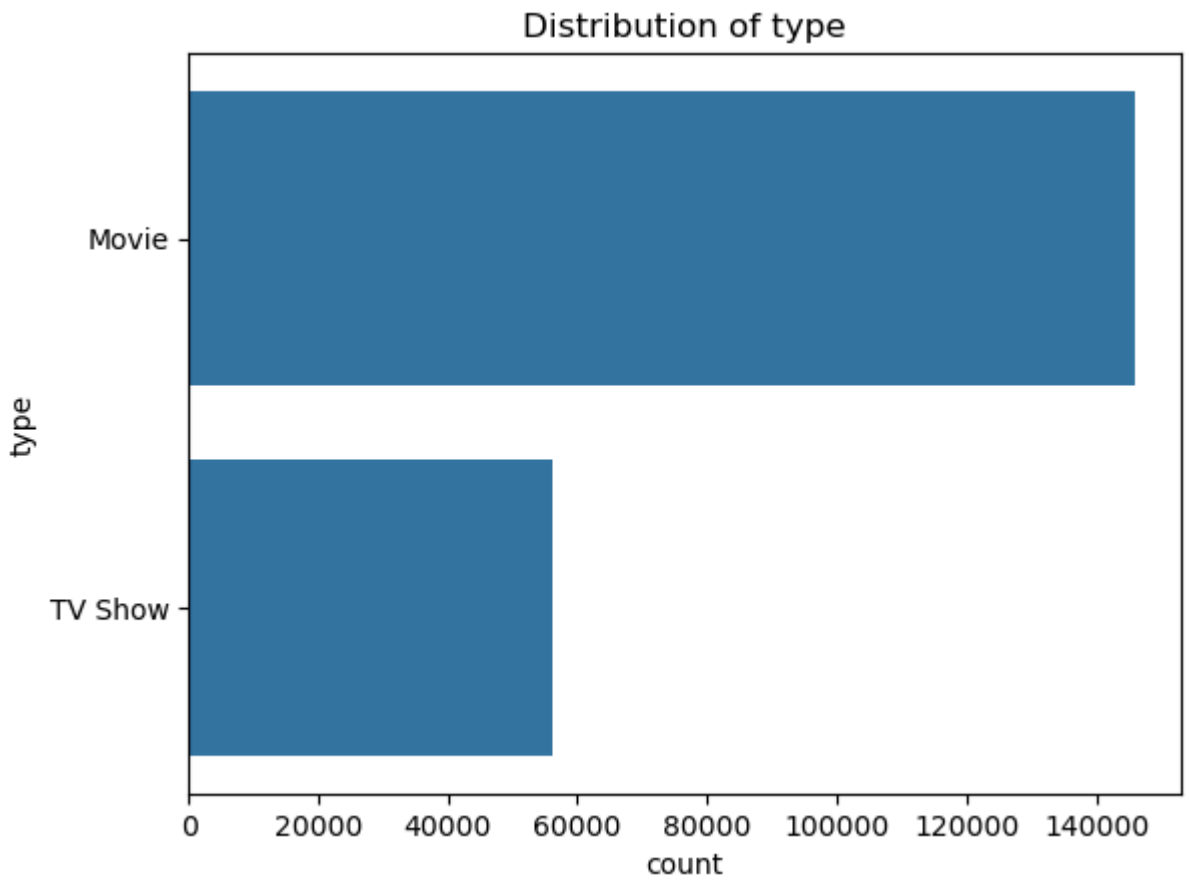
Univariate Analysis

Categorical Columns

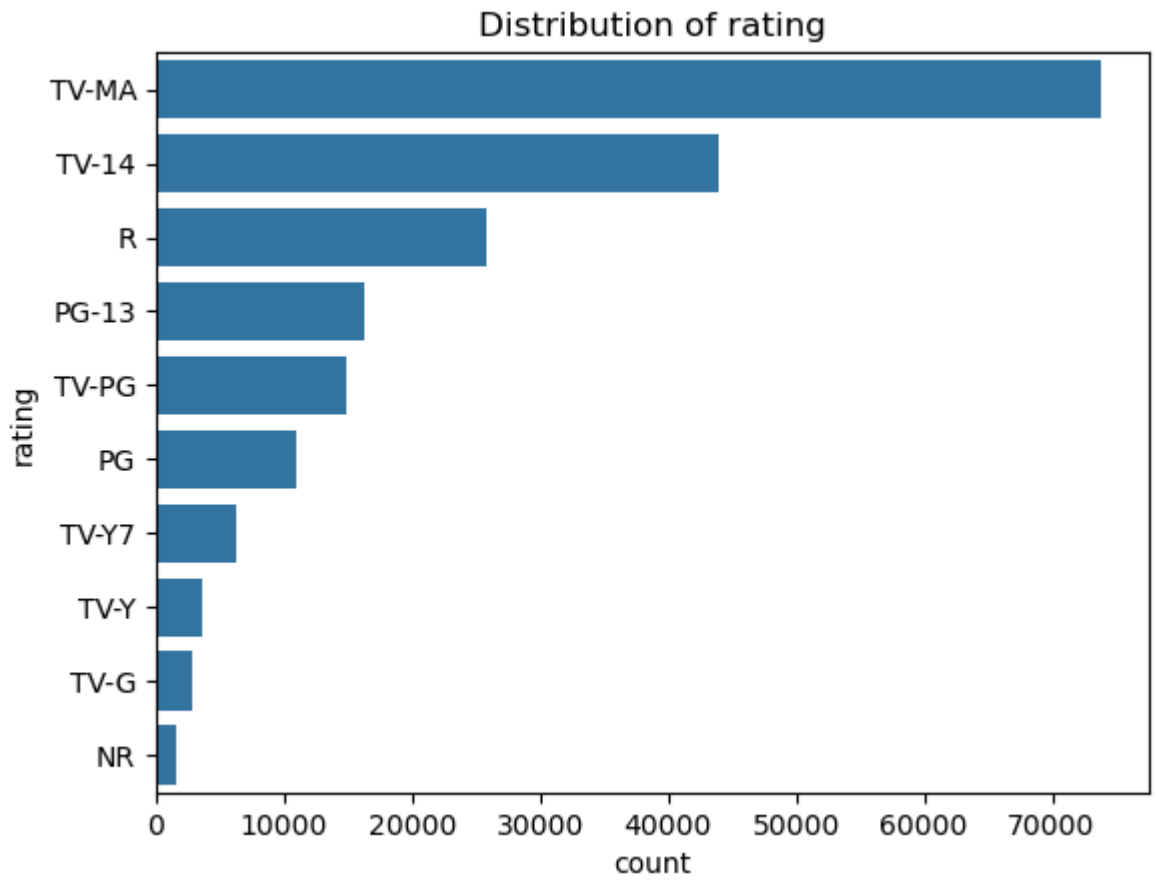
In [247…

```
# Visualizing the categorical columns using countplot for top 10 categories
for col in ['type', 'rating', 'Genre', 'Country', 'month_added', 'year_added', 'weekday_added', 'duration']:
    print(f"\nColumn: {col}")
    print(df_final[col].value_counts())
    sns.countplot(y=df_final[col], order=df_final[col].value_counts().index[:10])
    plt.title(f'Distribution of {col}')
    plt.show()
```

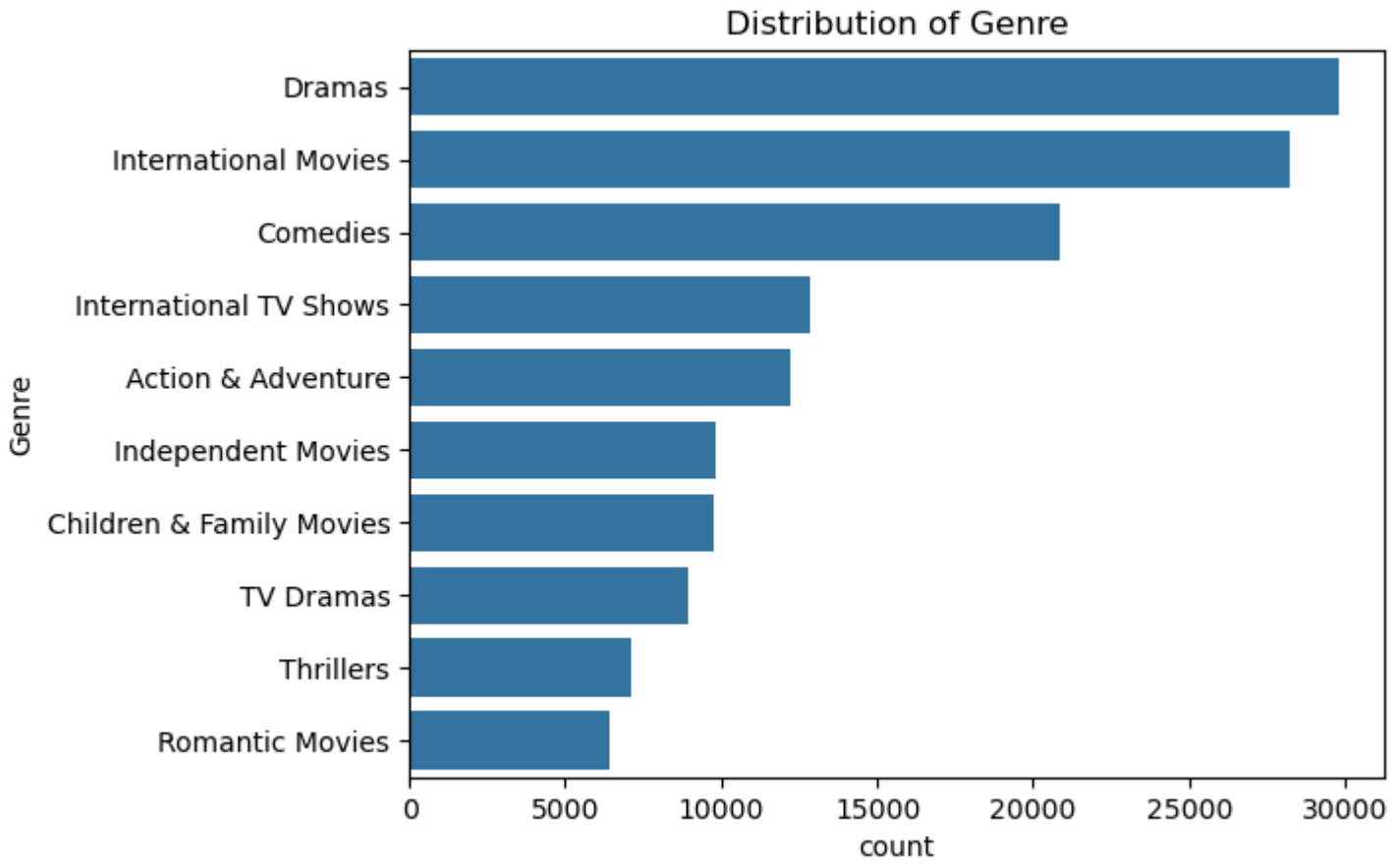
```
Column: type
type
Movie      145843
TV Show     56148
Name: count, dtype: int64
```



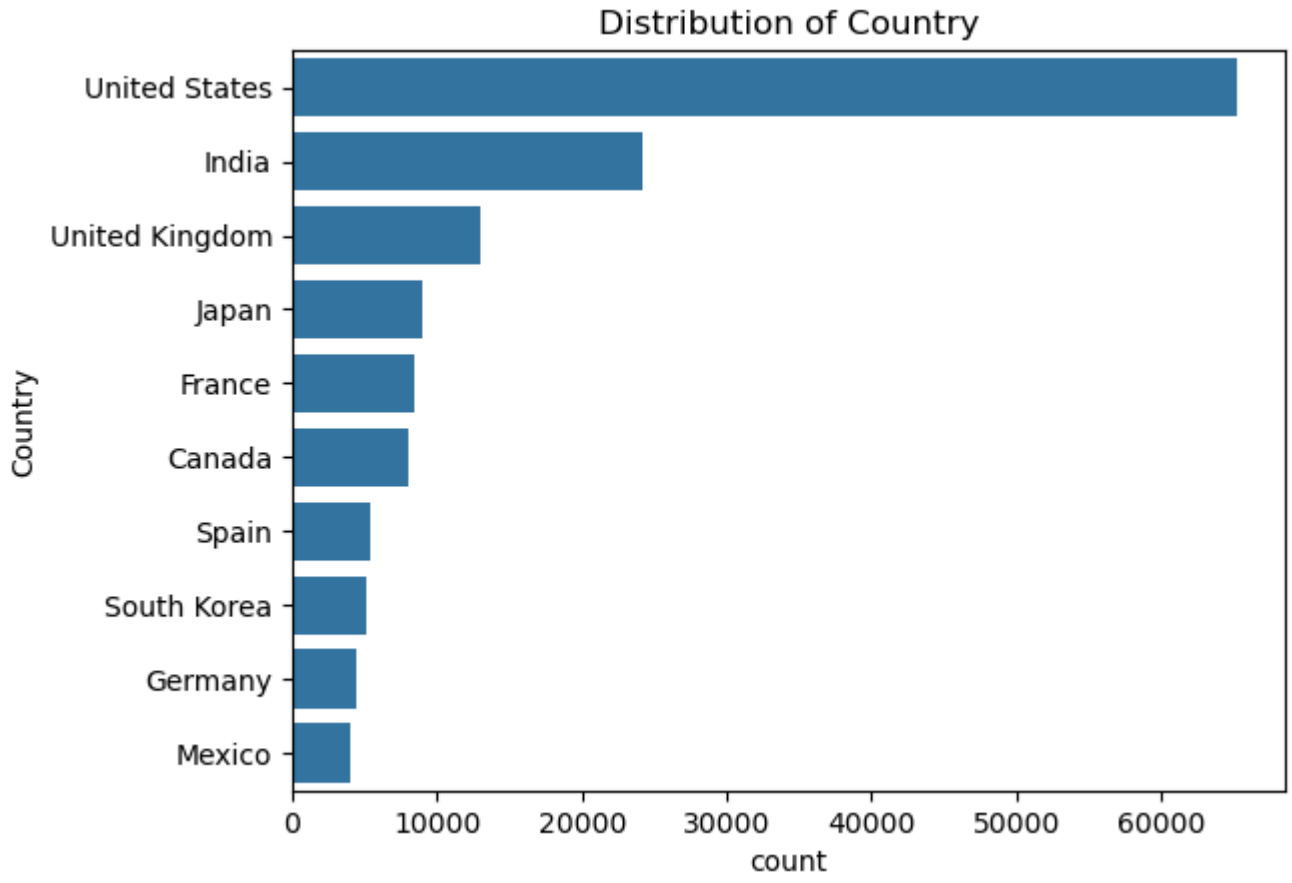

```
Column: rating
rating
TV-MA       73867
TV-14       43931
R           25860
PG-13       16246
TV-PG       14926
PG          10919
TV-Y7        6304
TV-Y         3665
TV-G         2779
NR           1643
G            1530
NC-17         149
TV-Y7-FV       86
UR             86
Name: count, dtype: int64
```



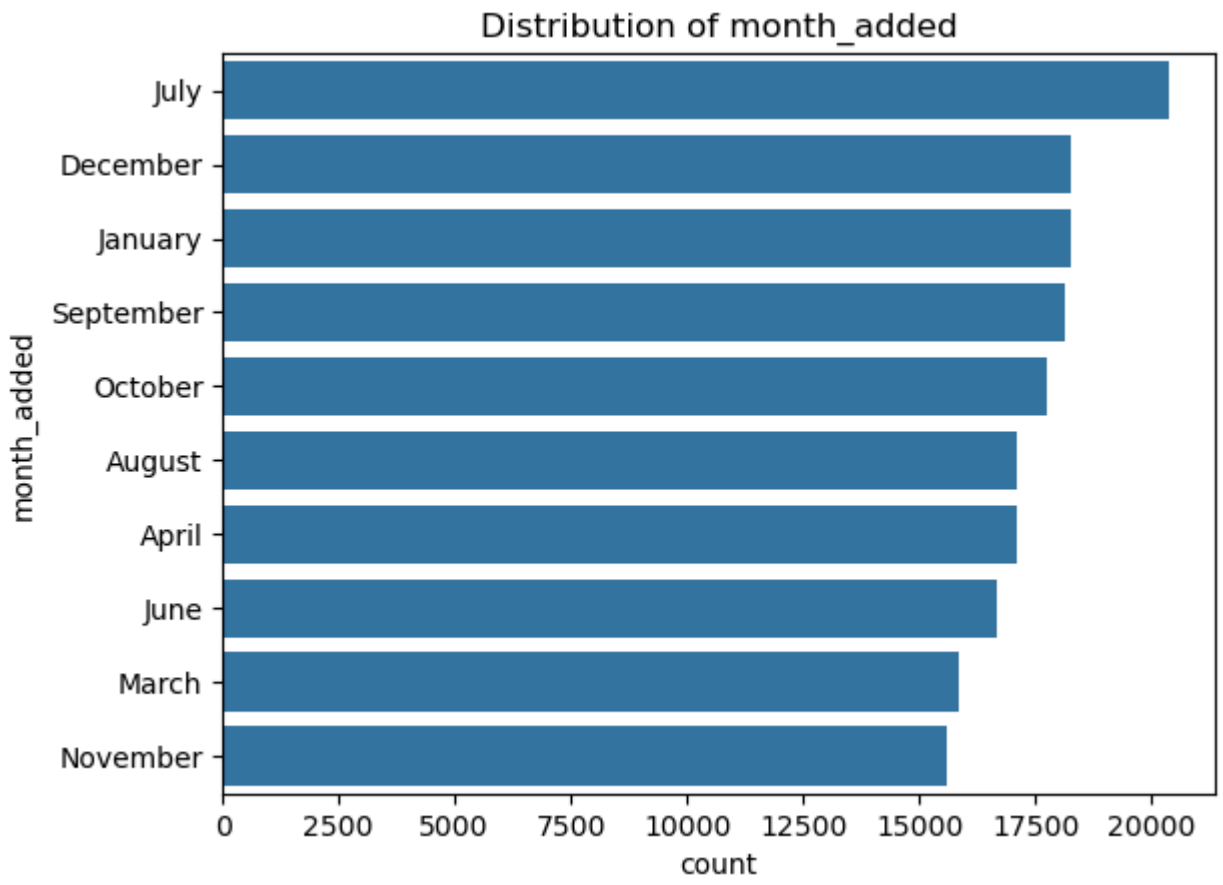
```
Column: Genre
Genre
Dramas                          29775
International Movies            28211
Comedies                        20829
International TV Shows          12845
Action & Adventure              12216
Independent Movies               9834
Children & Family Movies         9771
TV Dramas                        8942
Thrillers                        7107
Romantic Movies                  6412
TV Comedies                      4963
Crime TV Shows                   4733
Horror Movies                    4571
Kids' TV                         4568
Sci-Fi & Fantasy                 4037
Music & Musicals                 3077
Romantic TV Shows                3049
Documentaries                    2407
Anime Series                     2313
TV Action & Adventure            2288
Spanish-Language TV Shows        2126
British TV Shows                 1808
Sports Movies                    1531
Classic Movies                   1434
TV Mysteries                     1281
Korean TV Shows                  1122
Cult Movies                      1077
TV Sci-Fi & Fantasy              1045
Anime Features                   1045
TV Horror                         941
Docuseries                        845
LGBTQ Movies                      838
TV Thrillers                      768
Teen TV Shows                     742
Reality TV                        735
Faith & Spirituality              719
Stand-Up Comedy                   540
Movies                            412
TV Shows                          337
Classic & Cult TV                 272
Stand-Up Comedy & Talk Shows      268
Science & Nature TV               157
Name: count, dtype: int64
```



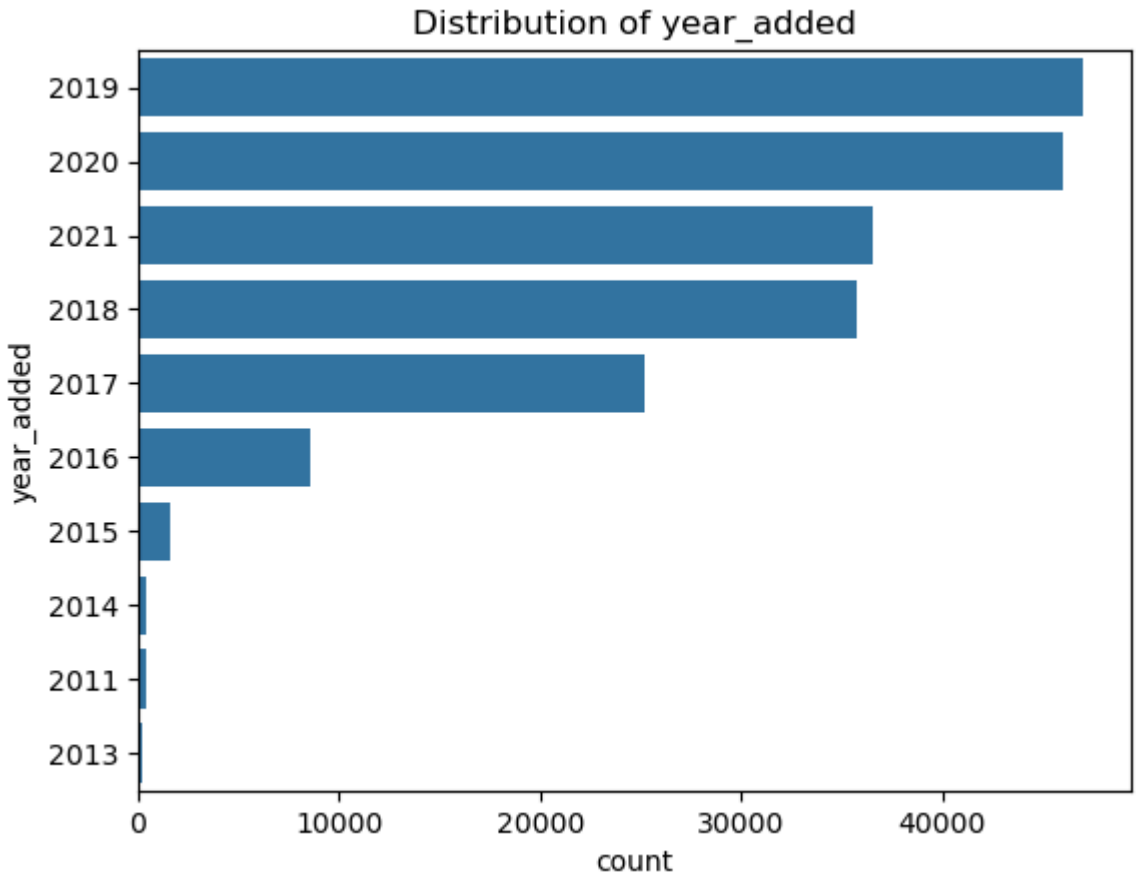

```
Column: Country
Country
United States     65250
India             24121
United Kingdom    13023
Japan              9053
France             8371
                  ...
Samoa                 2
Nicaragua             1
United States,        1
Kazakhstan            1
Uganda                1
Name: count, Length: 128, dtype: int64
```



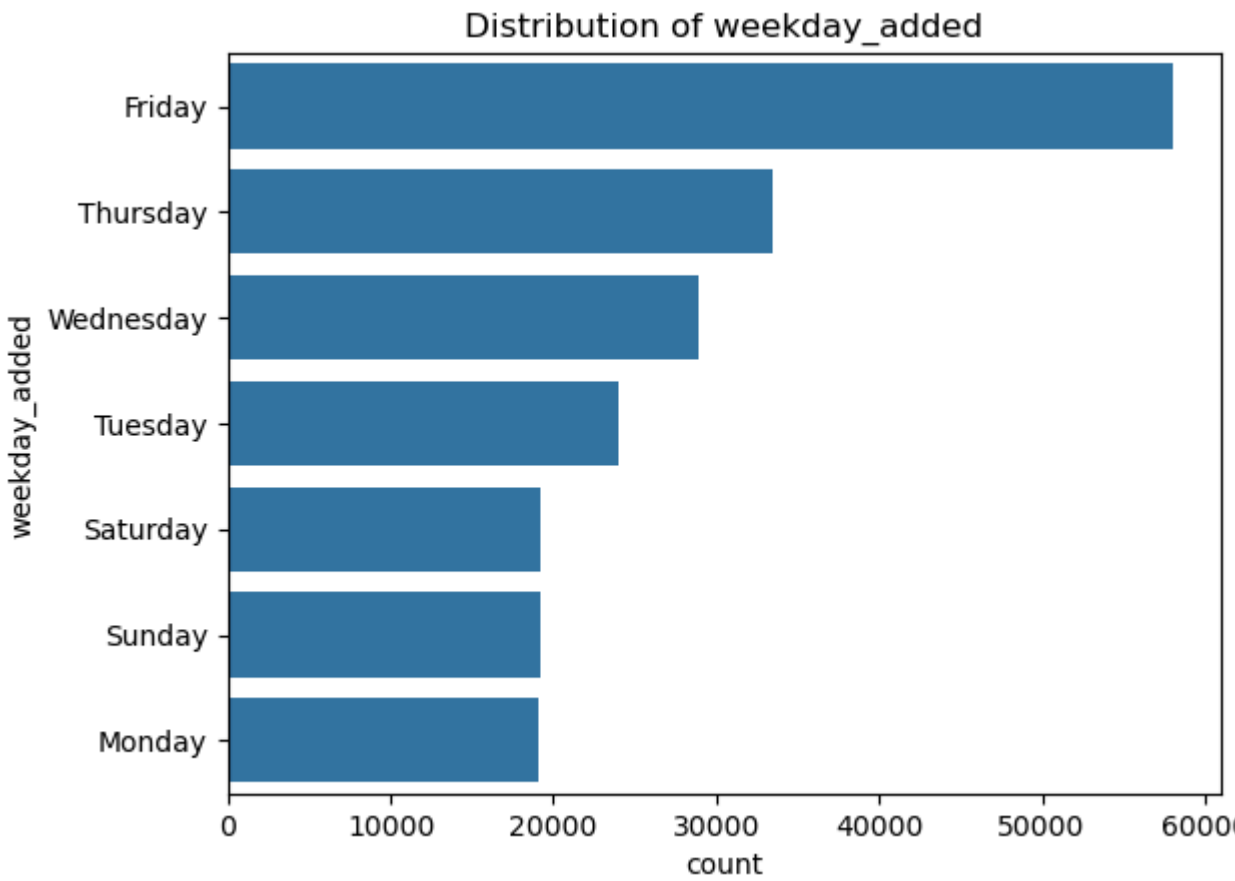

```
Column: month_added
month_added
July          20358
December      18266
January       18254
September     18143
October       17769
August        17110
April         17108
June          16659
March         15841
November      15596
May           13827
February      13060
Name: count, dtype: int64
```



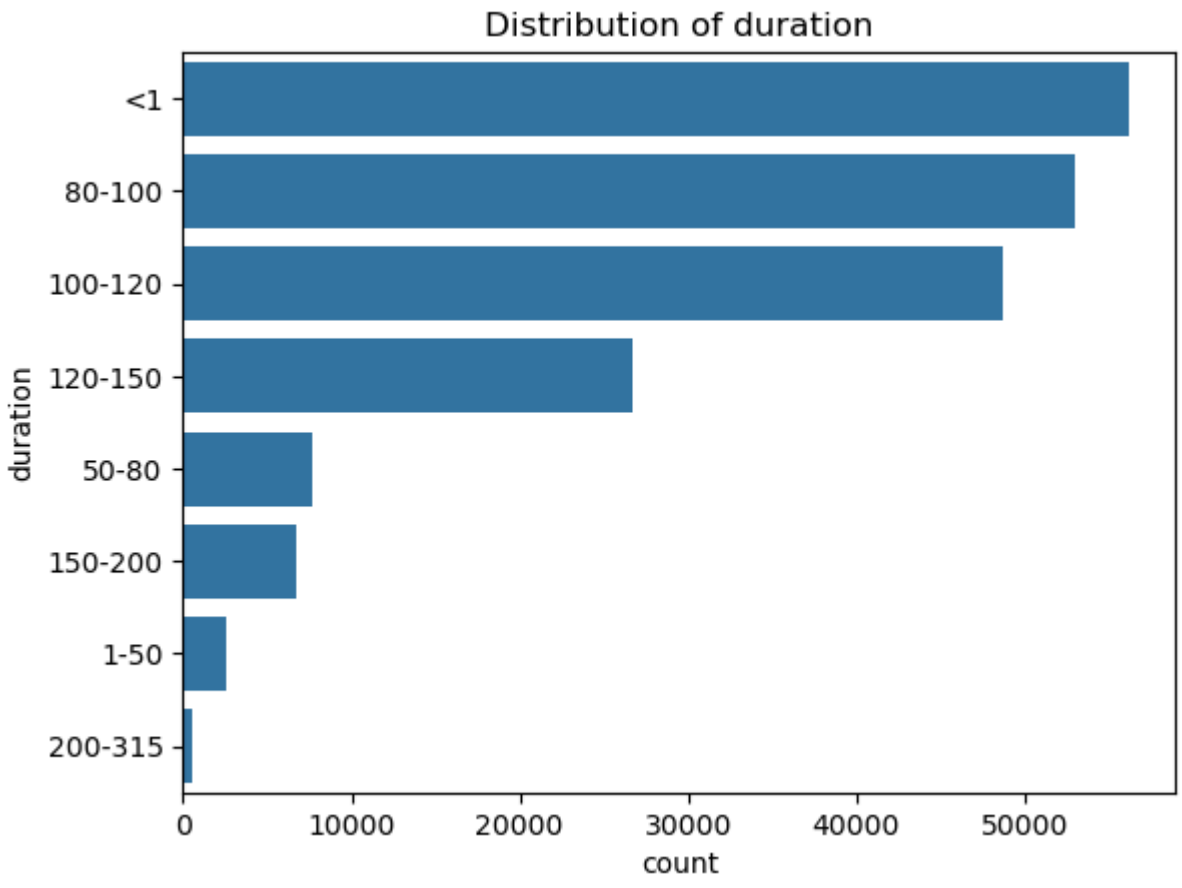

```
Column: year_added
year_added
2019    47033
2020    46025
2021    36541
2018    35785
2017    25233
2016     8614
2015     1560
2014      450
2011      438
2013      207
2012       36
2009       30
2010       20
2008       19
Name: count, dtype: int64
```




```
Column: weekday_added
weekday_added
Friday       58071
Thursday     33415
Wednesday    28958
Tuesday      24026
Saturday     19252
Sunday       19246
Monday       19023
Name: count, dtype: int64
```




```
Column: duration
duration
<1         56148
80-100     52937
100-120    48724
120-150    26691
50-80       7700
150-200     6737
1-50        2530
200-315      524
Name: count, dtype: int64
```



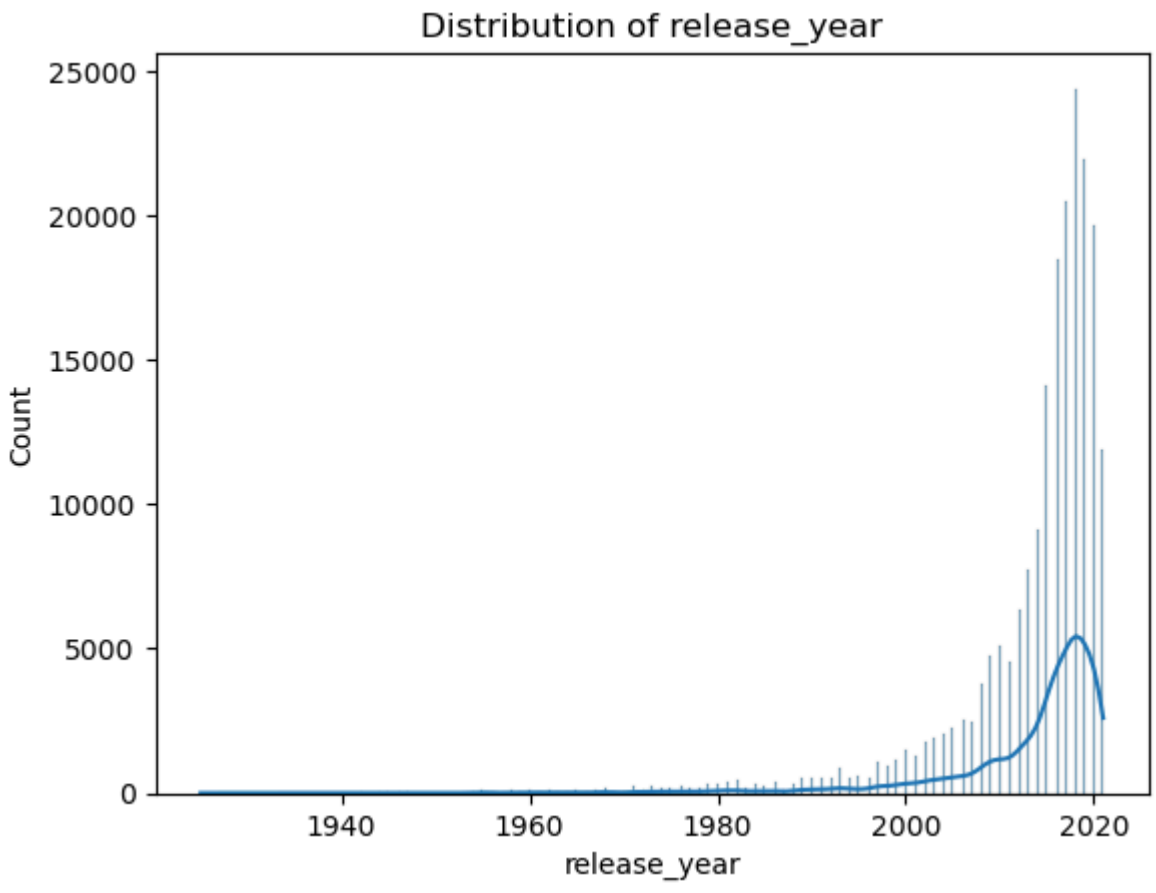
Inferences

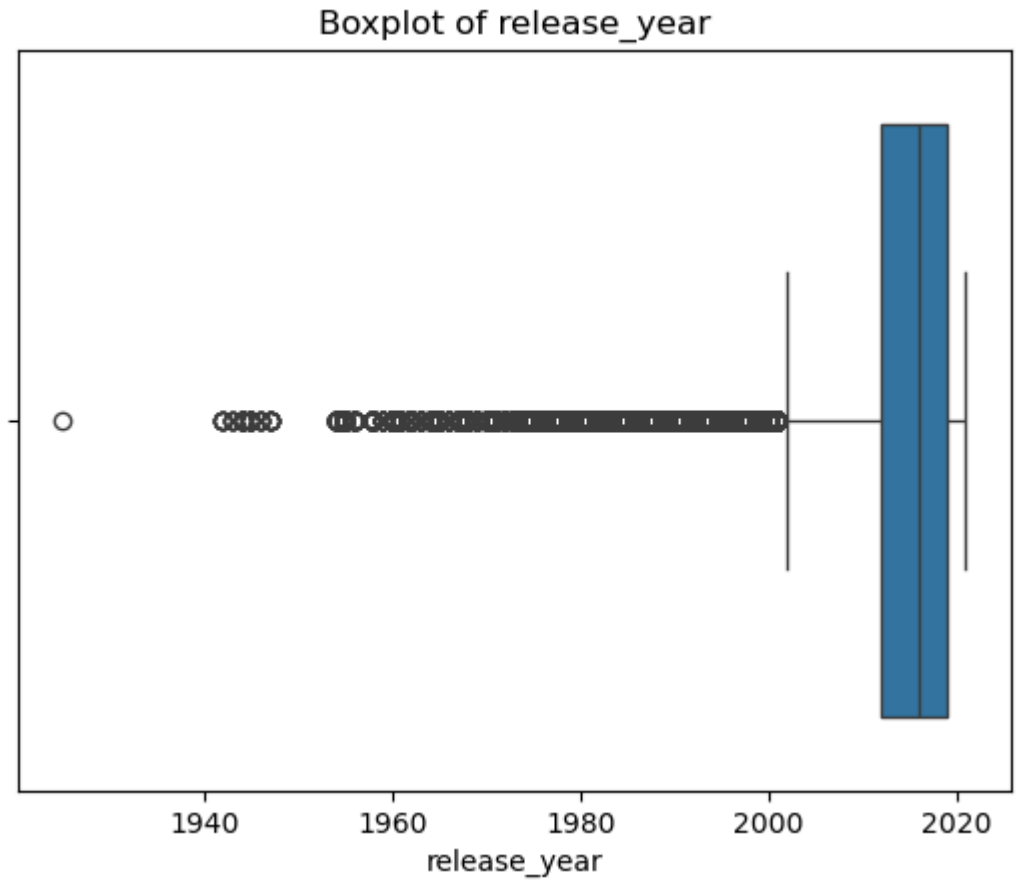
- Movies are more common than TV Shows.
- 'TV-MA' and 'TV-14' are the most common ratings, indicating a focus on mature audiences.
- The most frequent genres are Dramas, International Movies, and Comedies.
- The United States, India, and the United Kingdom are the top content-producing countries.
- Most content is added in January and March, with a steady addition throughout the year.
- Fridays see the highest content additions, suggesting a weekend release strategy.
- Most content has a duration of 1-2 hours for movies or 1-2 seasons for TV Shows.

Numerical Columns

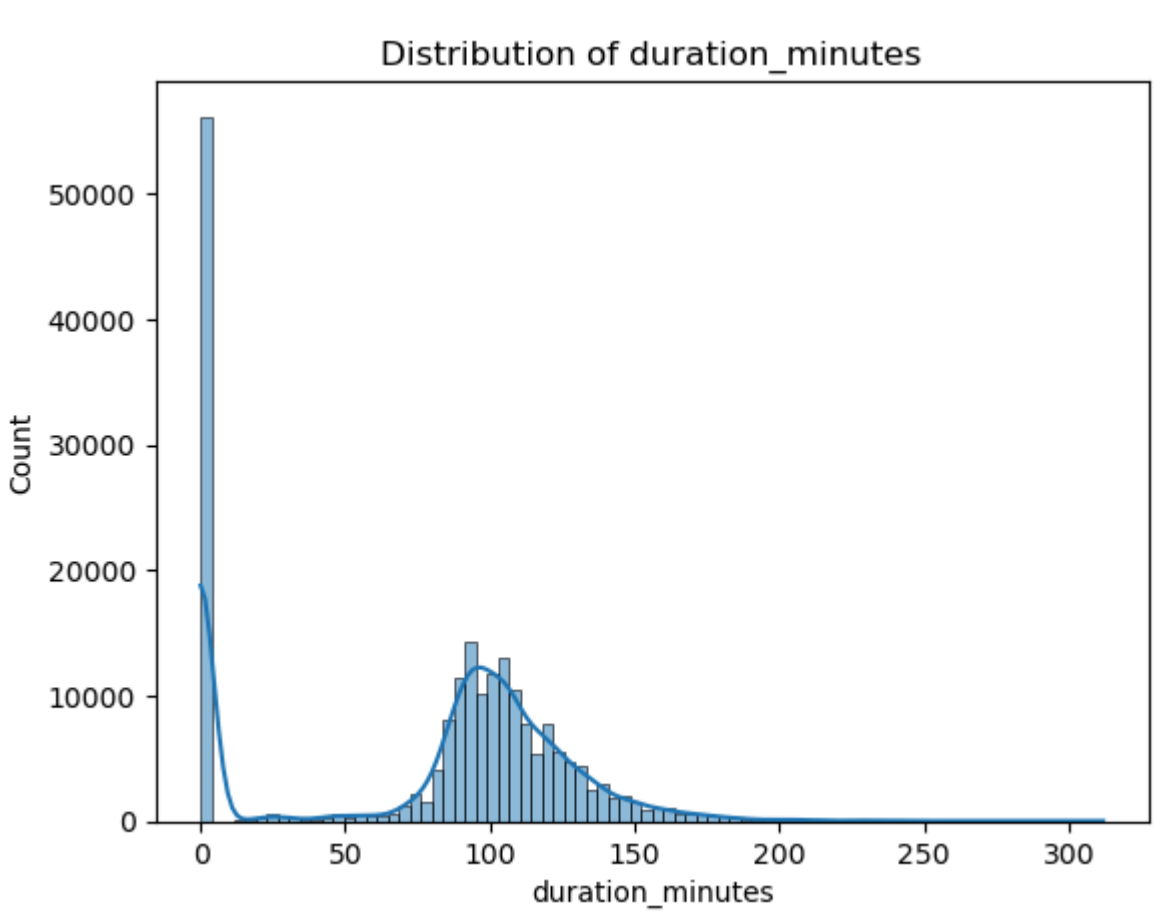
In [248…

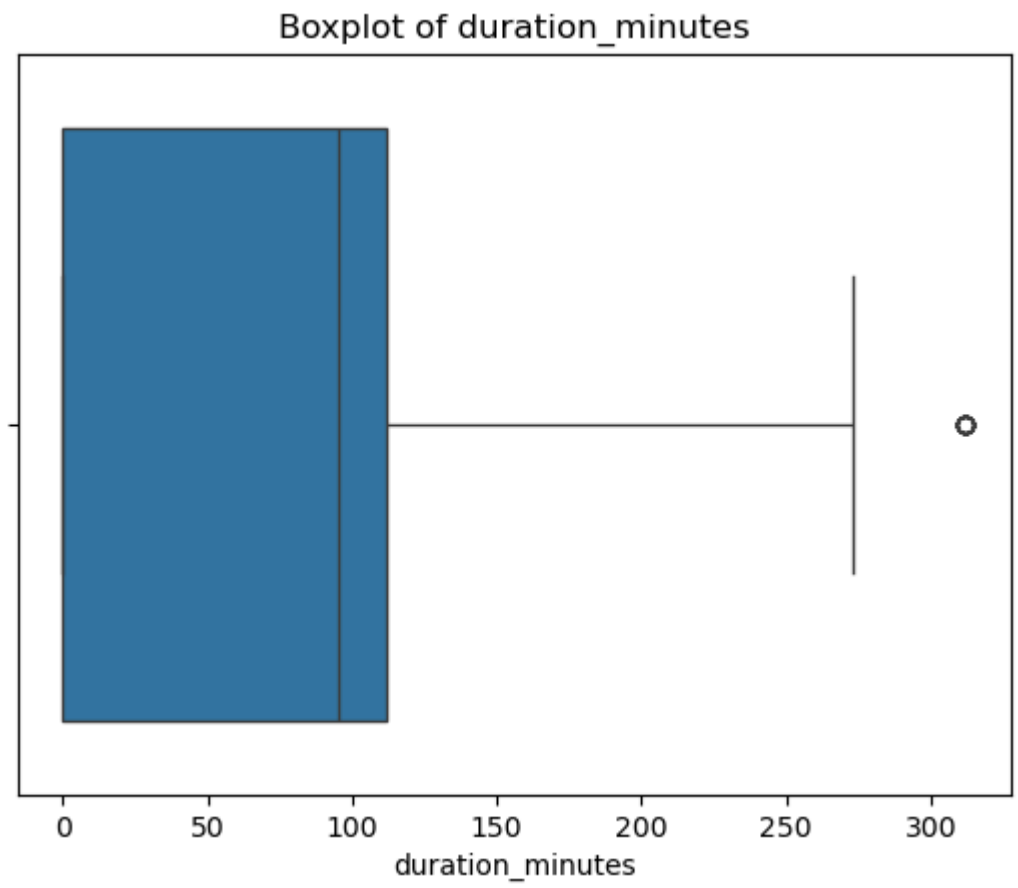
```
num_cols = ['release_year', 'duration_minutes', 'seasons']

for col in num_cols:
    sns.histplot(df_final[col], kde=True)
    plt.title(f'Distribution of {col}')
    plt.show()

    sns.boxplot(x=df_final[col])
    plt.title(f'Boxplot of {col}')
    plt.show()
```

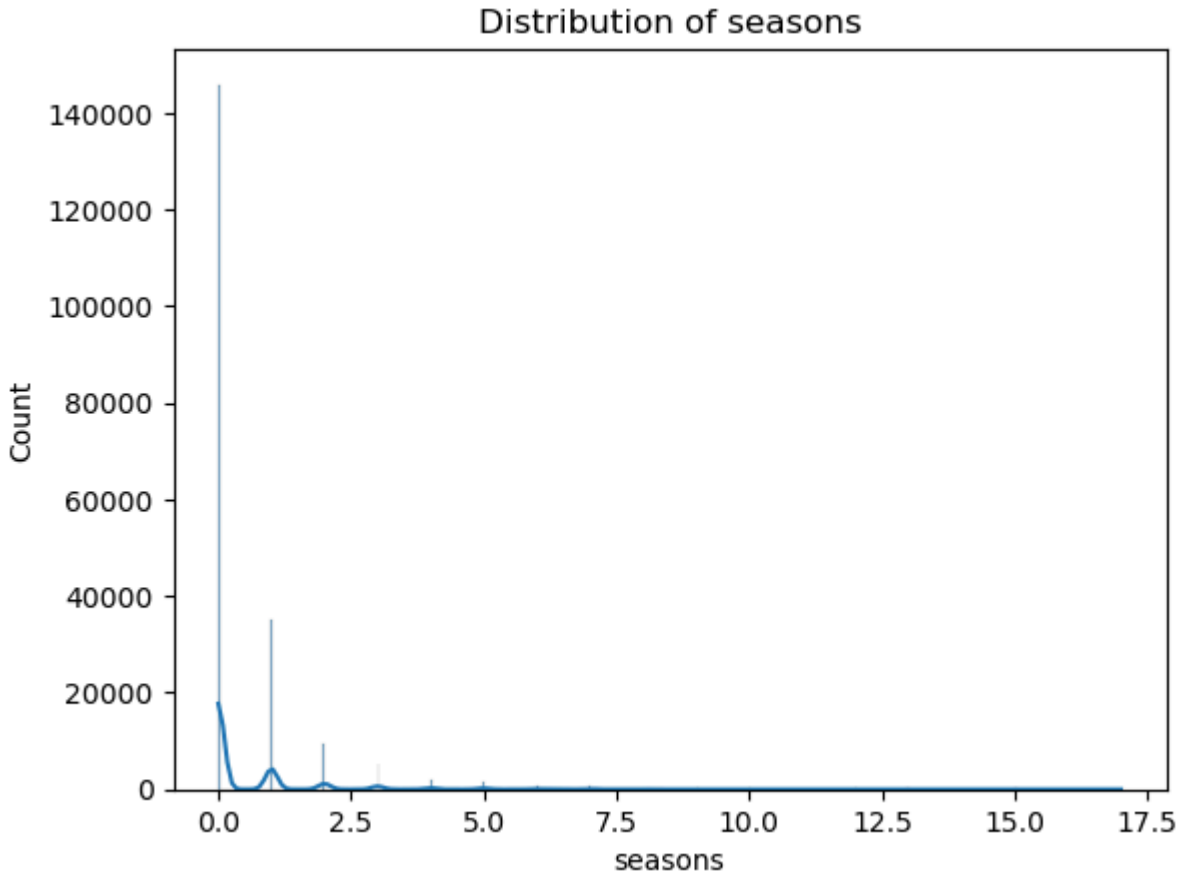


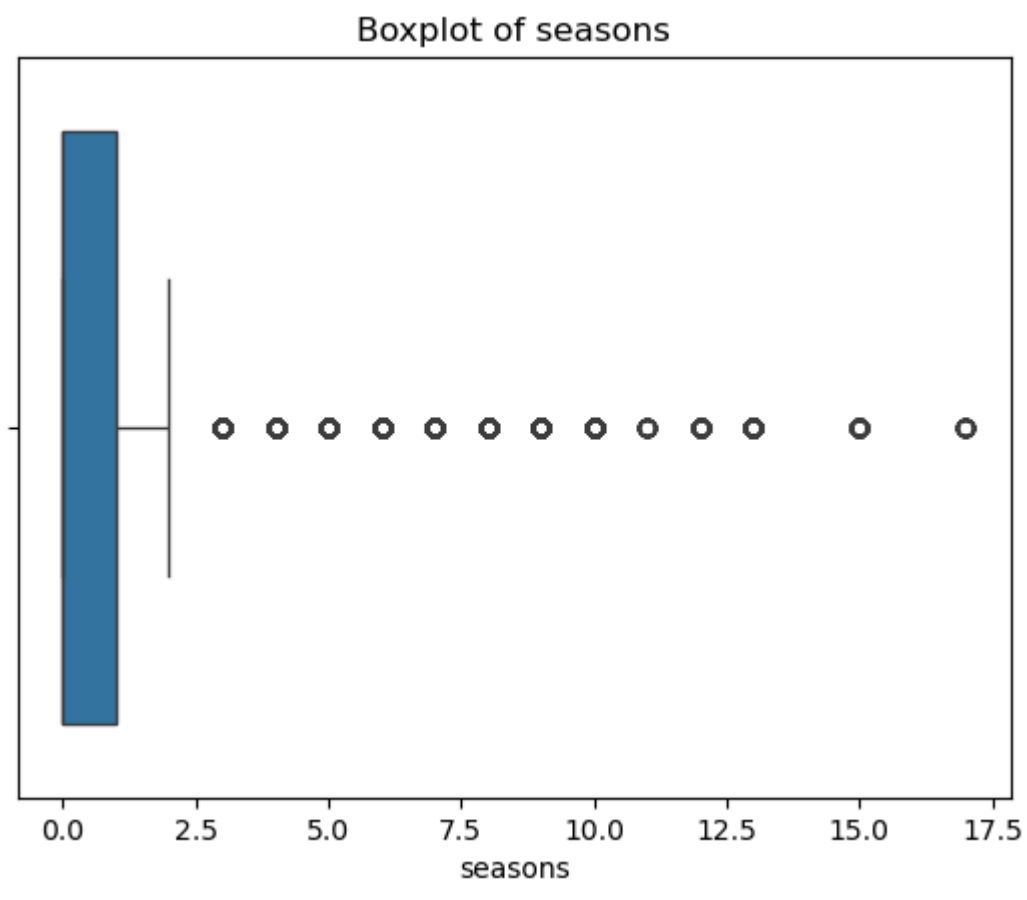


Distribution of duration_minutes





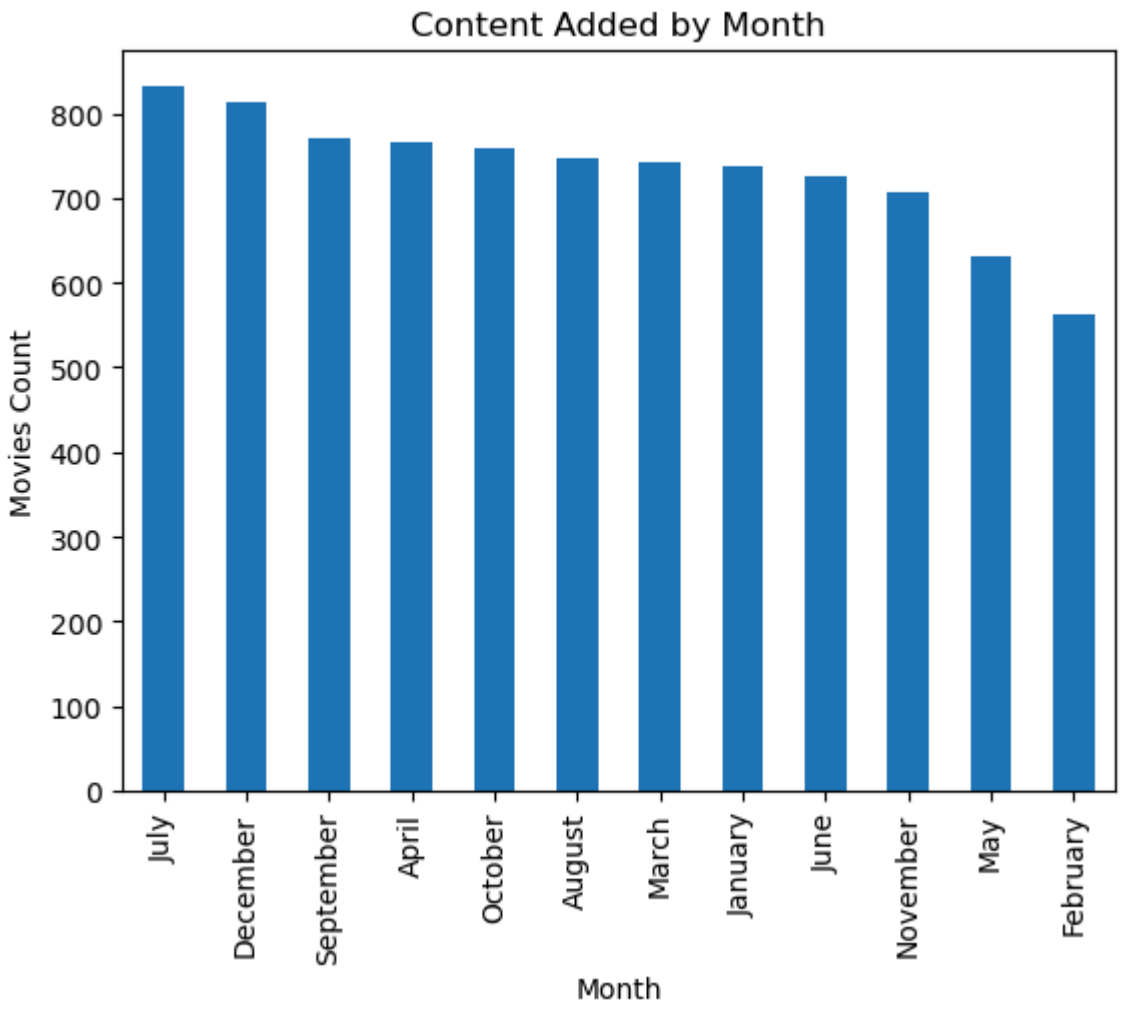




Inferences

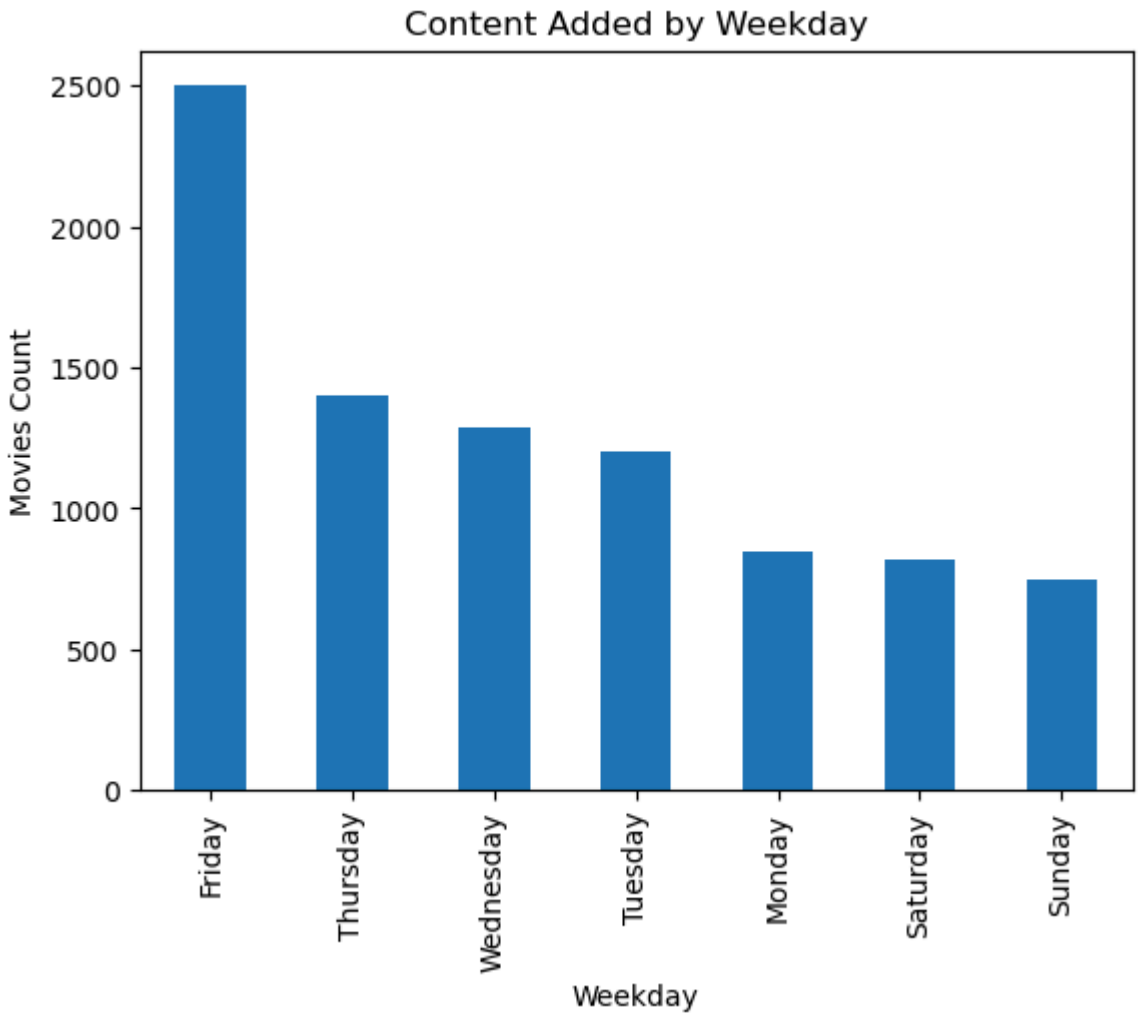
- The number of releases has increased over the years, peaking in recent years.
- Most TV Shows have 1-2 seasons.

In [ ]:
```
# Visualizing the content added by month
df_final.groupby('month_added').agg({'title':'nunique'}).reset_index().sort_values(by='title', ascending=False).plot(kind='bar
plt.title('Content Added by Month')
plt.ylabel('Movies Count')
plt.xlabel('Month')
plt.show()
```



Most content is released in July and December to capitalize on peak audience engagement during holidays and seasonal breaks.

In [ ]:
```
# Visualizing the content added by weekday
df_final.groupby('weekday_added').agg({'title':'nunique'}).reset_index().sort_values(by='title', ascending=False).plot(kind='b
plt.title('Content Added by Weekday')
plt.ylabel('Movies Count')
plt.xlabel('Weekday')
plt.show()
```



New content is primarily released on Fridays to capture weekend audiences.

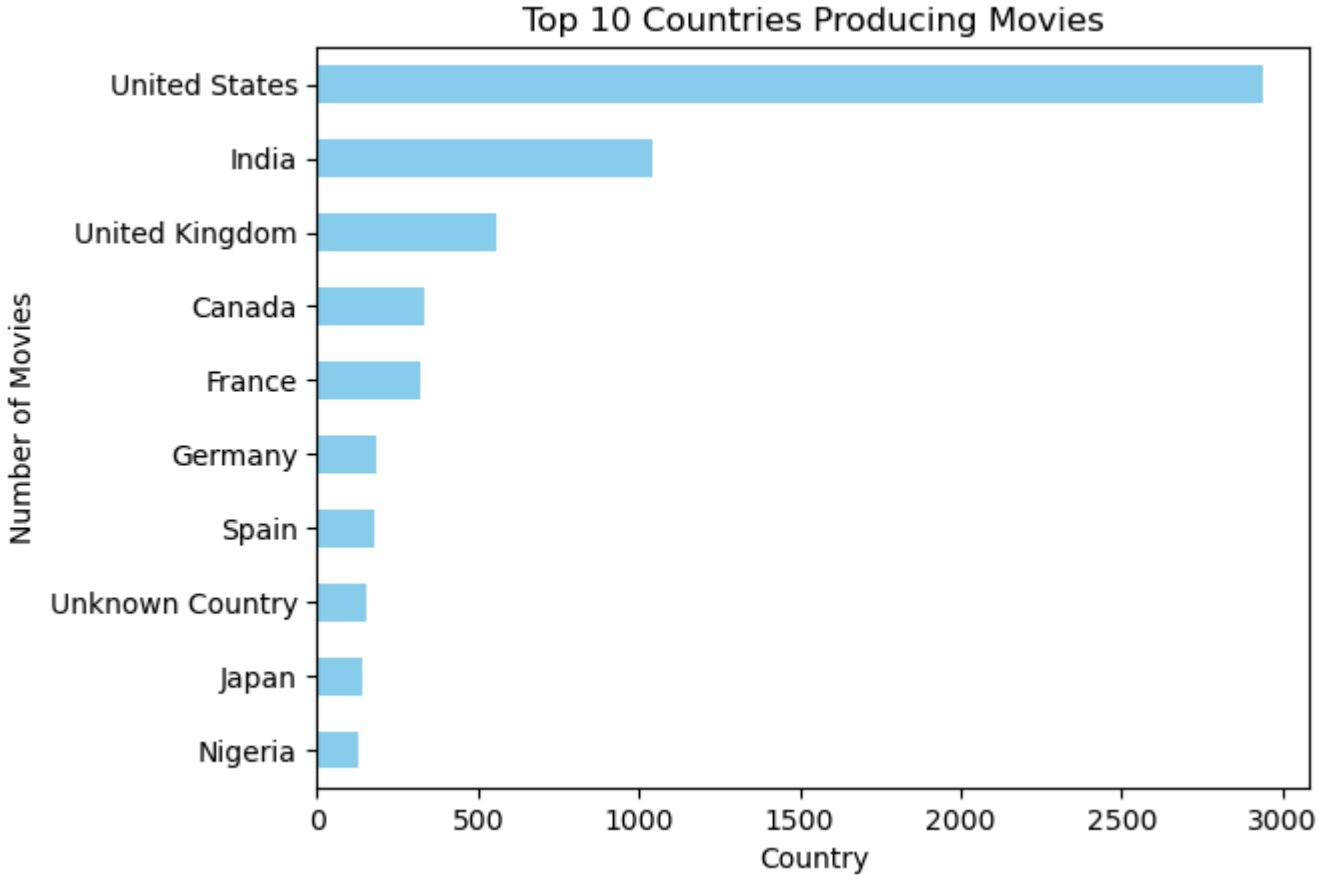
In [336…
```
# top 10 countries producing movies
top_10_movie_country = (df_final[df_final['type'] == 'Movie']
                        .groupby(['Country'])
                        .agg({'title': 'nunique'})
                        .reset_index().sort_values(by='title', ascending=False)[:10])
top_10_movie_country.reset_index(drop=True, inplace=True)
top_10_movie_country
```

Out[336…

	Country	title
0	United States	2937
1	India	1040
2	United Kingdom	556
3	Canada	334
4	France	318
5	Germany	187
6	Spain	176
7	Unknown Country	156
8	Japan	138
9	Nigeria	129

In [337…

```
# Visualizing the top 10 countries producing movies
top_10_movie_country.sort_values(by='title', ascending=True).plot(kind='barh', x='Country', y='title', legend=False, color='sk
plt.title('Top 10 Countries Producing Movies')
plt.xlabel('Country')
plt.ylabel('Number of Movies')
plt.show()
```



United States, India and United Kingdom are the leading movie content creators.

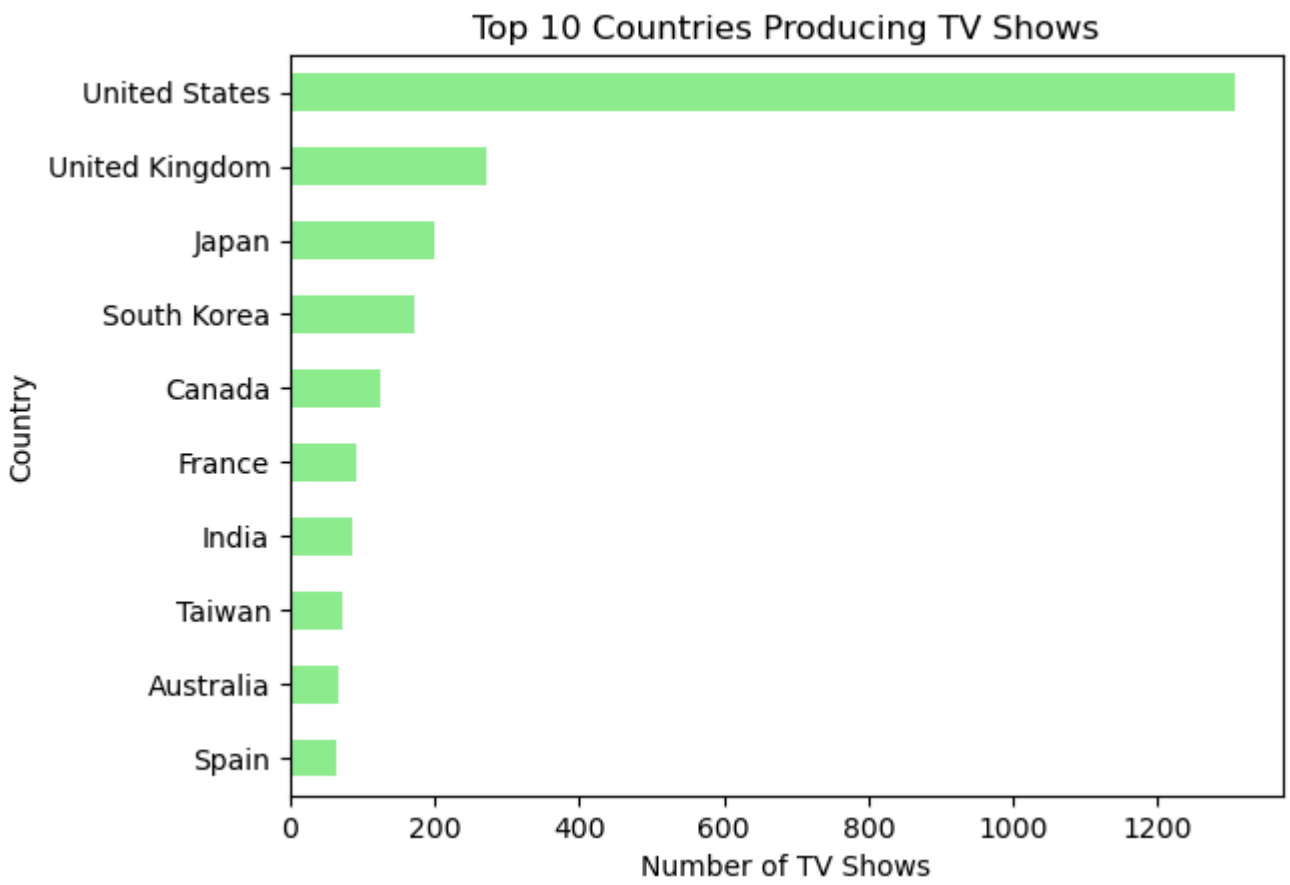
In [338…

```
# top 10 countries producing TV Shows
top_10_tvshow_country = (df_final[df_final['type'] == 'TV Show']
                         .groupby(['Country'])
                         .agg({'title': 'nunique'})
                         .reset_index().sort_values(by='title', ascending=False)[:10])
top_10_tvshow_country.reset_index(drop=True, inplace=True)
top_10_tvshow_country
```

Out[338…

	Country	title
0	United States	1308
1	United Kingdom	273
2	Japan	200
3	South Korea	171
4	Canada	126
5	France	91
6	India	86
7	Taiwan	72
8	Australia	66
9	Spain	63

In [339…
```
# Visualizing the top 10 countries producing TV Shows
top_10_tvshow_country.sort_values(by='title', ascending=True).plot(kind='barh', x='Country', y='title', legend=False, color='l
plt.title('Top 10 Countries Producing TV Shows')
plt.xlabel('Number of TV Shows')
plt.ylabel('Country')
plt.show()
```



United States, United Kingdom, Japan and South Korea are the leading TV Shows content creators.

In [255…
```
df_final.columns
```

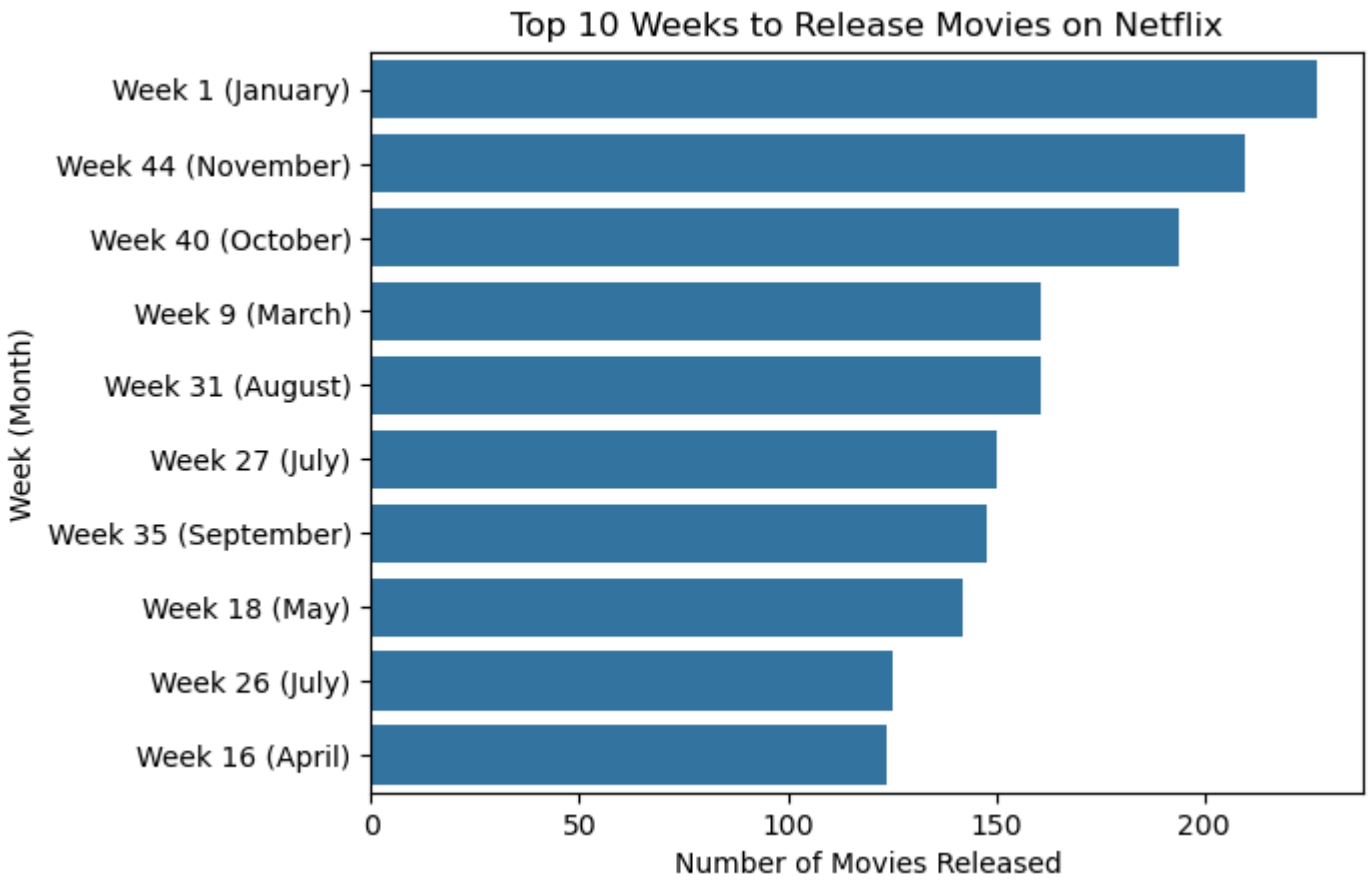
Out[255…
```
Index(['title', 'Actors', 'Directors', 'Genre', 'Country', 'show_id', 'type',
       'date_added', 'release_year', 'rating', 'duration_minutes', 'seasons',
       'duration', 'month_added', 'year_added', 'weekday_added'],
      dtype='object')
```

In [342…
```
# Best week to release Movies
week_movie = df_final['date_added'].dt.isocalendar().week
best_week_movie = df_final[df_final['type'] == 'Movie'].groupby([week_movie, 'month_added']).agg({'title': 'nunique'}).reset_i
best_week_movie
```

Out[342…

	week	month_added	title
1	1	January	227
59	44	November	210
54	40	October	194
11	9	March	161
41	31	August	161
35	27	July	150
47	35	September	148
24	18	May	142
33	26	July	125
20	16	April	124

In [343…
```
# Top 10 Weeks to Release Movies on Netflix
best_week_movie['label'] = 'Week ' + best_week_movie['week'].astype(str) + ' (' + best_week_movie['month_added'] + ')'

best_week_movie = best_week_movie.sort_values(by='title', ascending=False)

sns.barplot(
    data=best_week_movie,
    x='title',
    y='label'
)
plt.title('Top 10 Weeks to Release Movies on Netflix')
plt.xlabel('Number of Movies Released')
plt.ylabel('Week (Month)')
plt.show()
```



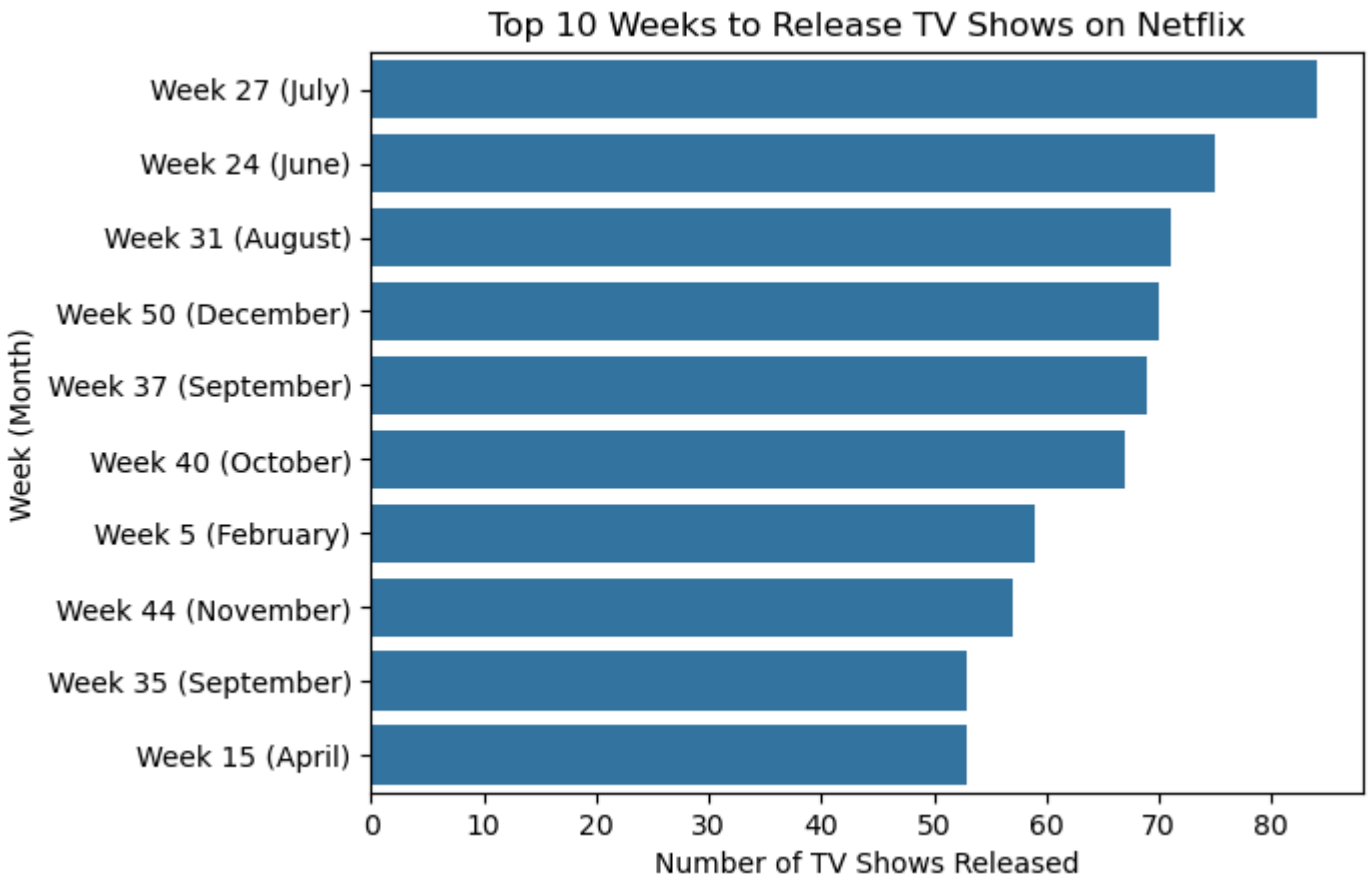
The first week of January and weeks in November, October and March are the best for movie releases.

In [345…
```
# Best week to release TV Shows
week_tvshow = df_final['date_added'].dt.isocalendar().week
best_week_tvshow = df_final[df_final['type'] == 'TV Show'].groupby([week_tvshow, 'month_added']).agg({'title': 'nunique'}).res
best_week_tvshow
```

Out[345…

	week	month_added	title
35	27	July	84
31	24	June	75
41	31	August	71
66	50	December	70
49	37	September	69
53	40	October	67
5	5	February	59
58	44	November	57
47	35	September	53
19	15	April	53

In [346…
```
# Top 10 Weeks to Release TV Shows on Netflix
best_week_tvshow['label'] = 'Week ' + best_week_tvshow['week'].astype(str) + ' (' + best_week_tvshow['month_added'] + ')'
best_week_tvshow = best_week_tvshow.sort_values(by='title', ascending=False)

sns.barplot(
    data=best_week_tvshow,
    x='title',
    y='label'
)
plt.title('Top 10 Weeks to Release TV Shows on Netflix')
plt.xlabel('Number of TV Shows Released')
plt.ylabel('Week (Month)')
plt.show()
```



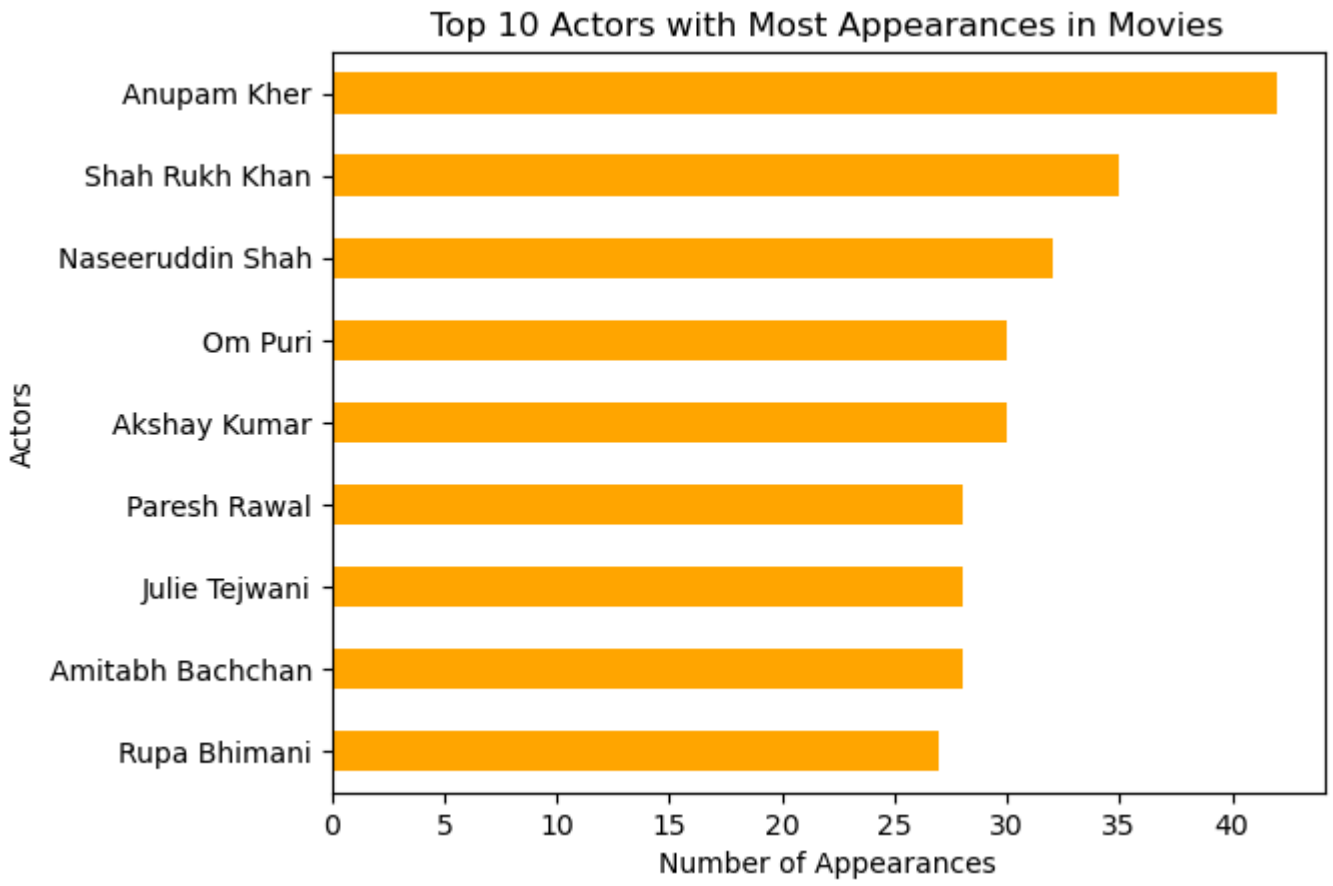
TV shows are most frequently released in the weeks of July, June, and August.

In [348…
```
# Top 10 Actors with most appearances in movies
top_10_actors_movie = (df_final[df_final['type'] == 'Movie']
                       .groupby(['Actors'])
                       .agg({'title': 'nunique'})
                       .reset_index().sort_values(by='title', ascending=False)[1:10])
top_10_actors_movie.reset_index(drop=True, inplace=True)
top_10_actors_movie
```

Out[348…

	Actors	title
0	Anupam Kher	42
1	Shah Rukh Khan	35
2	Naseeruddin Shah	32
3	Akshay Kumar	30
4	Om Puri	30
5	Amitabh Bachchan	28
6	Julie Tejwani	28
7	Paresh Rawal	28
8	Rupa Bhimani	27

In [349…
```
# Visualizing the top 10 actors with most appearances in movies
top_10_actors_movie.sort_values(by='title', ascending=True).plot(kind='barh', x='Actors', y='title', legend=False, color='oran
plt.title('Top 10 Actors with Most Appearances in Movies')
plt.xlabel('Number of Appearances')
plt.ylabel('Actors')
plt.show()
```



Certain actors, such as Anupam Kher, Shah Rukh Khan and Naseeruddin Shah, have a high number of appearances, indicating their popularity or frequent collaborations.

In [271…

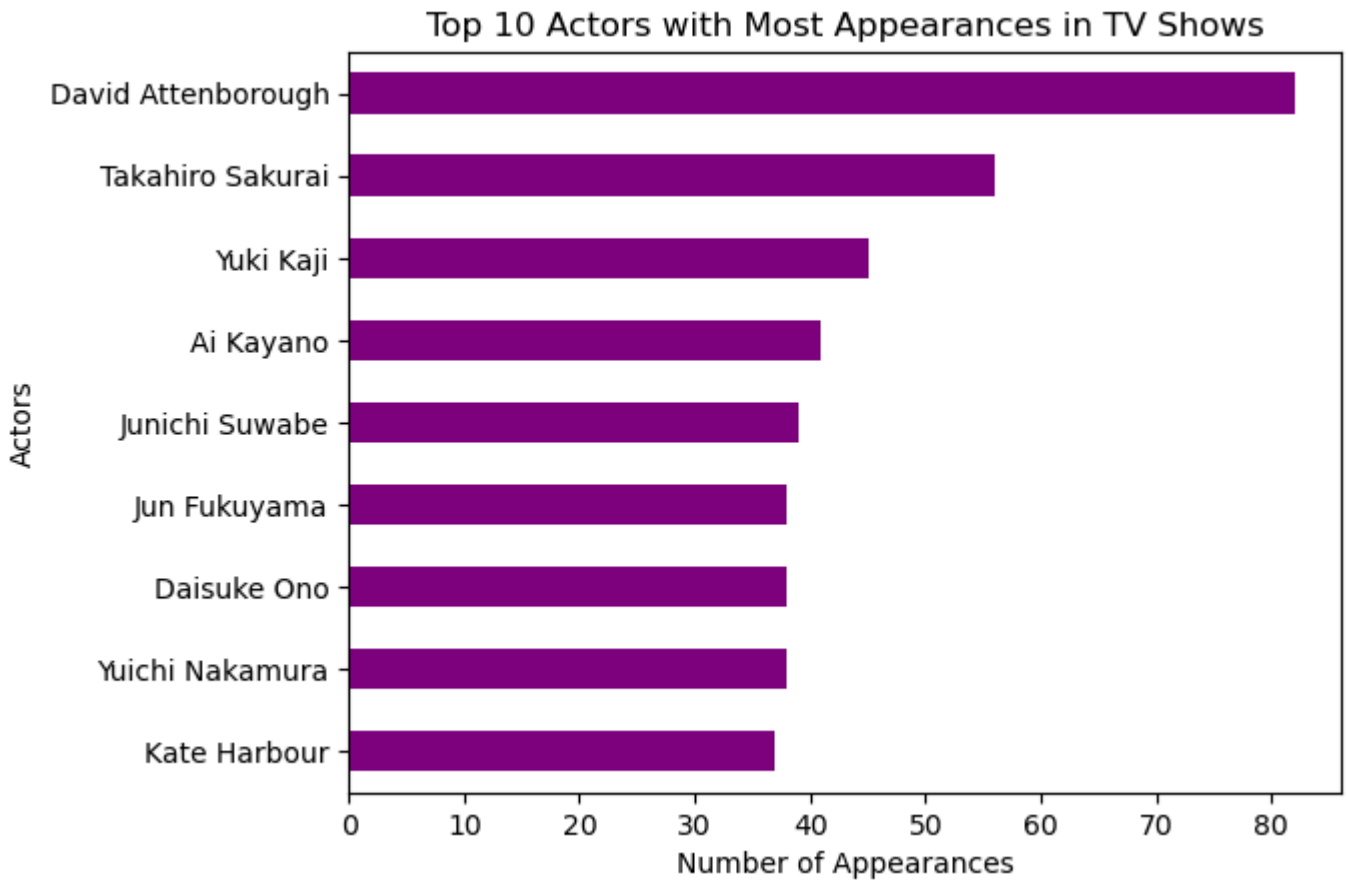
```
# Top 10 Actors with most appearances in TV Shows
top_10_actors_tvshow = (df_final[df_final['type'] == 'TV Show']
                        .groupby(['Actors'])
                        .agg({'title': 'count'})
                        .reset_index().sort_values(by='title', ascending=False)[1:10])
top_10_actors_tvshow.reset_index(drop=True, inplace=True)
top_10_actors_tvshow
```

Out[271…

	Actors	title
0	David Attenborough	82
1	Takahiro Sakurai	56
2	Yuki Kaji	45
3	Ai Kayano	41
4	Junichi Suwabe	39
5	Yuichi Nakamura	38
6	Daisuke Ono	38
7	Jun Fukuyama	38
8	Kate Harbour	37

In [272…

```
# Visualizing the top 10 actors with most appearances in TV Shows
top_10_actors_tvshow.sort_values(by='title', ascending=True).plot(kind='barh', x='Actors', y='title', legend=False, color='pur
plt.title('Top 10 Actors with Most Appearances in TV Shows')
plt.xlabel('Number of Appearances')
plt.ylabel('Actors')
plt.show()
```



David Attenborough, Takahiro Sakurai, and Yuki Kaji are prominent figures in the TV show industry.

In [361…

```
# Top Genres for Movies using word cloud
from wordcloud import WordCloud
from collections import Counter
top_10_genre = (df_final[df_final['type'] == 'Movie']
                .groupby(['Genre'])
                .agg({'title': 'nunique'})
                .reset_index().sort_values(by='title', ascending=False))
top_10_genre.reset_index(drop=True, inplace=True)
top_10_genre
```

Out[361…

	Genre	title
0	International Movies	2738
1	Dramas	2418
2	Comedies	1673
3	Documentaries	869
4	Action & Adventure	854
5	Independent Movies	756
6	Children & Family Movies	639
7	Romantic Movies	615
8	Thrillers	573
9	Music & Musicals	372
10	Horror Movies	353
11	Stand-Up Comedy	343
12	Sci-Fi & Fantasy	243
13	Sports Movies	219
14	Classic Movies	116
15	LGBTQ Movies	102
16	Cult Movies	71
17	Anime Features	71
18	Faith & Spirituality	65
19	Movies	57

In [369…

```
# Visualizing the top genres using word cloud
genre_counts = Counter(df_final['Genre'])
wordcloud = WordCloud(background_color='white').generate_from_frequencies(genre_counts)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Top Genres for Movies')
plt.show()
```

International Movies, Dramas and Comedies are the most prominent genres.

In [ ]:
```
# Difference between release date and date added to Netflix in days
df_final['release_date'] = pd.to_datetime(df_final['release_year'].astype(str) + '-01-01')
df_final['date_difference'] = (df_final['release_date'] - df_final['date_added']).dt.days
```

In [ ]:
```
# Top 10 movies with the longest gap between release date and date added to Netflix
valid_movies = df_final[(df_final['type'] == 'Movie') & (df_final['date_difference'] >= 0)]
valid_movies = valid_movies.sort_values(by='date_difference', ascending=False).head(10)
valid_movies
```

Out[ ]:

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rating	duration_minutes	sea
123789	Hans Teeuwen: Real Rancour	Hans Teeuwen	Doesjka van Hoogdalem	Stand-Up Comedy	Netherlands	s5395	Movie	2017-07-01	2018	TV-MA	86	
160534	Incoming	Dominic Power	Eric Zaragosa	Sci-Fi & Fantasy	United States	s7064	Movie	2018-10-26	2019	TV-MA	89	
160503	Incoming	Vahldin Prelic	Eric Zaragosa	Action & Adventure	Serbia	s7064	Movie	2018-10-26	2019	TV-MA	89	
160512	Incoming	Michelle Lehane	Eric Zaragosa	Action & Adventure	United States	s7064	Movie	2018-10-26	2019	TV-MA	89	
160511	Incoming	Michelle Lehane	Eric Zaragosa	Action & Adventure	Serbia	s7064	Movie	2018-10-26	2019	TV-MA	89	
160510	Incoming	Lukas Loughran	Eric Zaragosa	Sci-Fi & Fantasy	United States	s7064	Movie	2018-10-26	2019	TV-MA	89	
160509	Incoming	Lukas Loughran	Eric Zaragosa	Sci-Fi & Fantasy	Serbia	s7064	Movie	2018-10-26	2019	TV-MA	89	
160508	Incoming	Lukas Loughran	Eric Zaragosa	Action & Adventure	United States	s7064	Movie	2018-10-26	2019	TV-MA	89	
160507	Incoming	Lukas Loughran	Eric Zaragosa	Action & Adventure	Serbia	s7064	Movie	2018-10-26	2019	TV-MA	89	
160506	Incoming	Vahldin Prelic	Eric Zaragosa	Sci-Fi & Fantasy	United States	s7064	Movie	2018-10-26	2019	TV-MA	89	

In [310…
```
# Mode of the delay in adding content to Netflix
mode_delay = valid_movies['date_difference'].mode()[0]
print("Most common delay in adding content to Netflix (in days):", mode_delay)
```

```
Most common delay in adding content to Netflix (in days): 67
```

There can be a significant delay (sometimes years) between a movie's release and its addition to Netflix, but the most common delay is around 67 days.

Recommendations:

1. **Prioritize movies over TV shows**, as movies are more prevalent on the platform and align better with current content distribution and viewer behavior.
2. **Focus on mature-rated content**, especially titles rated 'TV-MA' and 'TV-14', which dominate the catalog and suggest strong demand from adult audiences.
3. **Invest more heavily in high-performing genres** such as Dramas, International Movies, and Comedies, as these are the most frequently produced and likely resonate with global audiences.
4. **Favor short-form, easily consumable content** — particularly movies with a duration of 1–2 hours and TV shows limited to 1–2 seasons — to increase engagement and completion rates.

5. **Strengthen production and acquisition partnerships** in top content-producing countries: United States, India, and United Kingdom for movies; and United States, United Kingdom, Japan, and South Korea for TV shows.
6. **Leverage the popularity of frequently featured actors**, such as Anupam Kher, Shah Rukh Khan, Naseeruddin Shah in films, and David Attenborough, Takahiro Sakurai, and Yuki Kaji in TV shows, to boost viewer interest and loyalty.
7. **Align major content releases with high-activity months** like January and March, where platform data shows a higher volume of content additions and likely stronger viewer engagement.
8. **Increase the volume of releases in July and December**, capitalizing on holidays and seasonal breaks when audiences are more likely to binge-watch new and existing content.
9. **Maintain and optimize the Friday release strategy**, as this day shows the highest volume of content additions and aligns well with weekend viewing patterns.
10. **Reduce the delay between original release and Netflix availability**, with a target of around 67 days — the most common current gap — to keep content fresh and competitive.
11. **Acquire and promote high-quality older titles**, especially those with proven popularity, to extend long-tail value and attract viewers seeking reliable, curated content.