Netflix_Business_case_Study

May 29, 2024

[ ]:
```
#Q1. Business Case: Netflix - Data Exploration and Visualisation
```

[ ]:
```
# 1. Defining Problem Statement and Analysing basic metrics
# Problem Statement

# Netflix, as a leading streaming platform, continuously aims to enhance its␣
 ↪content library and subscriber base.
# The goal of this analysis is to provide data-driven insights into the type of␣
 ↪shows and movies that Netflix should produce and
# how to expand its business in different countries. The analysis will focus on␣
 ↪understanding the content distribution, popular genres, ratings,
# and trends over time.

# Basic Metrics Analysis

# To begin, we will explore the dataset, understand its structure, and analyze␣
 ↪some basic metrics.

# Steps:
# 1.Loading the Data:
# 2.Exploring the Data:
# 3.Handling Missing Values:
```

[ ]:
```
#Import Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[ ]:
```
#1.Loading Dataset
df = pd.read_csv('/content/netflix_dataset.csv')
```

[ ]:
```
#2.Exploring the Data:
# Check the shape of the DataFrame
print(f"Shape of the DataFrame: {df.shape}")

# Check for missing values
```

```python
print("Missing values in each column:\n", df.isnull().sum())

# Get basic statistics of the DataFrame
print("Basic statistics:\n", df.describe(include='all'))
```

```
Shape of the DataFrame: (8807, 12)
Missing values in each column:
 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
Basic statistics:
        show_id   type                 title       director  \
count      8807   8807                  8807           6173
unique     8807      2                  8807           4528
top          s1  Movie  Dick Johnson Is Dead  Rajiv Chilaka
freq          1   6131                     1             19
mean        NaN    NaN                   NaN            NaN
std         NaN    NaN                   NaN            NaN
min         NaN    NaN                   NaN            NaN
25%         NaN    NaN                   NaN            NaN
50%         NaN    NaN                   NaN            NaN
75%         NaN    NaN                   NaN            NaN
max         NaN    NaN                   NaN            NaN

                      cast        country       date_added  release_year  \
count                 7982           7976             8797   8807.000000
unique                7692            748             1767           NaN
top     David Attenborough  United States  January 1, 2020           NaN
freq                    19           2818              109           NaN
mean                   NaN            NaN              NaN   2014.180198
std                    NaN            NaN              NaN      8.819312
min                    NaN            NaN              NaN   1925.000000
25%                    NaN            NaN              NaN   2013.000000
50%                    NaN            NaN              NaN   2017.000000
75%                    NaN            NaN              NaN   2019.000000
max                    NaN            NaN              NaN   2021.000000
```

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

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

[ ]:
```
#There are 8807 rows and 12 columns in the dataframe
# There are a total of 4307 null values across the en re dataset with 2634␣
 ↪missing points under "director", 825 under "cast",
# 831 under "country", 11 under "date_added", 4 under "ra ng" and 3 under␣
 ↪"dura on ". We will have to handle all null data points
# before we can dive into EDA and modelling.
```

[ ]:
```
df.sample(5) #df.head(5)
```

[ ]:
```
     show_id   type                                      title  \
4052   s4053  Movie                    BNK48: Girls Don't Cry
2471   s2472  Movie                                 Uncut Gems
7191   s7192  Movie                    Kickboxer: Retaliation
4683   s4684  Movie  Maria Bamford: The Special Special Special
8459   s8460  Movie                                  The Plan

                      director  \
4052  Nawapol Thamrongrattanarit
2471   Josh Safdie, Benny Safdie
7191          Dimitri Logothetis
4683                Jordan Brady
```

```
8459                    Keerthi

                                                     cast        country  \
4052                                                  NaN       Thailand
2471  Adam Sandler, LaKeith Stanfield, Kevin Garnett…  United States
7191  Alain Moussi, Jean-Claude Van Damme, Mike Tyso…  United States
4683      Maria Bamford, Wayne Federman, Jackie Kashian  United States
8459  Anant Nag, Koustubh Jayakumar, Hemanth, Shreer…          India

          date_added  release_year rating duration  \
4052    March 1, 2019          2018  TV-14  108 min
2471     May 25, 2020          2019      R  135 min
7191   April 26, 2018          2017      R  110 min
4683  August 25, 2018          2012  TV-MA   50 min
8459    March 1, 2018          2015  TV-MA  124 min

                                          listed_in  \
4052  Documentaries, International Movies, Music & M…
2471                                Dramas, Thrillers
7191                             Action & Adventure
4683                               Stand-Up Comedy
8459           Dramas, International Movies, Thrillers

                                          description
4052  Members of the Thai idol girl group BNK48 open…
2471  With his debts mounting and angry collectors c…
7191  Sloan's vow to never return to Thailand is cut…
4683  Spend an evening with gleeful, oh-so-awkward M…
8459  After being sent to a remote prison, three you…
```

[ ]:
```
# To seperate the list with commas in single cell to individual(for easy
 ↪access) [director,cast,country,listed_in]
df_director_r = pd.DataFrame(df['director'].apply(lambda x: str(x).split(',')).
 ↪tolist(), index =df['title'])

df_director = df_director_r.stack().reset_index()
df_director.drop('level_1', axis = 1, inplace = True)
df_director.rename(columns ={0:'director'}, inplace = True)
df_director.head()
```

[ ]:
```
                   title         director
0   Dick Johnson Is Dead  Kirsten Johnson
1          Blood & Water              nan
2              Ganglands  Julien Leclercq
3  Jailbirds New Orleans              nan
4           Kota Factory              nan
```

[ ]:
```
df_cast_r = pd.DataFrame(df['cast'].apply(lambda x: str(x).split(',')).
 ↪tolist(), index =df['title'])

df_cast = df_cast_r.stack().reset_index()
df_cast.drop('level_1', axis = 1, inplace = True)
df_cast.rename(columns ={0:'cast'}, inplace = True)
df_cast.head()
```

[ ]:
```
                  title             cast
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
```

[ ]:
```
df_country_r = pd.DataFrame(df['country'].apply(lambda x: str(x).split(',')).
 ↪tolist(),index = df['title'])

df_country = df_country_r.stack().reset_index()
df_country.drop('level_1',axis = 1 ,inplace = True)
df_country.rename(columns = {0:'country'},inplace = True)
df_country.head()
```

[ ]:
```
                   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
```

[ ]:
```
df_listed_in_r = pd.DataFrame(df['listed_in'].apply(lambda x: str(x).
 ↪split(',')).tolist(),index = df['title'])

df_listed_in  = df_listed_in_r.stack().reset_index()
df_listed_in .drop('level_1',axis = 1 ,inplace = True)
df_listed_in .rename(columns = {0:'listed_in'},inplace = True)
df_listed_in .head()
```

[ ]:
```
                  title              listed_in
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
```

[ ]:
```
# Merging director and cast col
df_new = df_director.merge(df_cast, how ='inner', on ='title')
```

```
df_new
```

[ ]:
```
                    title         director                    cast
0      Dick Johnson Is Dead  Kirsten Johnson                     nan
1             Blood & Water              nan              Ama Qamata
2             Blood & Water              nan              Khosi Ngema
3             Blood & Water              nan            Gail Mabalane
4             Blood & Water              nan          Thabang Molaba
…                       …              …                       …
70807               Zubaan      Mozez Singh        Manish Chaudhary
70808               Zubaan      Mozez Singh             Meghna Malik
70809               Zubaan      Mozez Singh            Malkeet Rauni
70810               Zubaan      Mozez Singh           Anita Shabdish
70811               Zubaan      Mozez Singh   Chittaranjan Tripathy

[70812 rows x 3 columns]
```

[ ]:
```
#Merging listed_in & country col
df_new_2 = df_listed_in.merge(df_country, how ='inner', on ='title')
df_new_2
```

[ ]:
```
                    title                listed_in        country
0      Dick Johnson Is Dead             Documentaries  United States
1             Blood & Water    International TV Shows   South Africa
2             Blood & Water                TV Dramas   South Africa
3             Blood & Water             TV Mysteries   South Africa
4                 Ganglands            Crime TV Shows            nan
…                       …                        …           …
23759                  Zoom  Children & Family Movies  United States
23760                  Zoom                 Comedies  United States
23761               Zubaan                   Dramas          India
23762               Zubaan      International Movies          India
23763               Zubaan          Music & Musicals          India

[23764 rows x 3 columns]
```

[ ]:
```
df.columns
```

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

[ ]:
```
# Merging df_new(cast & director) with rest of the cols - df_final
df_final = df_new.merge(df[['show_id', 'type', 'title', 'date_added',
       'release_year', 'rating', 'duration','description']] , how = 'inner', on␣
 ↪= 'title')
```

```
df_final.sample(10)
```

[ ]:

	title	director	cast	show_id	type	date_added	release_year	rating	duration	description
57886	La Bamba	Luis Valdez	Lou Diamond Phillips	s7253	Movie	January 1, 2020	1987	PG-13	109 min	The plane crash that killed Buddy Holly also t…
52807	Dil Hai Tumhaara	Kundan Shah	Dilip Joshi	s6611	Movie	April 1, 2018	2002	TV-14	176 min	The sophisticated son of a powerful businessma…
25379	No Game No Life	nan	Mamiko Noto	s2966	TV Show	February 1, 2020	2014	TV-MA	1 Season	Legendary gamer siblings Sora and Shiro are tr…
43350	The Incredible Jessica James	Jim Strouse	Lakeith Stanfield	s5369	Movie	July 28, 2017	2017	TV-MA	84 min	Burned by a bad breakup, a struggling New York…
34991	Horrid Henry	nan	Lizzie Waterworth	s4203	TV Show	January 11, 2019	2019	TV-Y7	2 Seasons	To his family's frustration, Henry is skilled …
27946	Suffragette	Sarah Gavron	Carey Mulligan	s3262	Movie	November 16, 2019	2015	PG-13	107 min	At the beginning of the 20th century, circumst…
52938	Don Quixote: The Ingenious Gentleman of La Mancha	Dave Dorsey	Vera Cherny	s6625	Movie	January 5, 2018	2015	TV-14	83 min	In this modern adaptation of a Spanish classic…
17390	The Devil All The Time	Antonio Campos	Robert Pattinson	s1996	Movie	September 16, 2020	2020	R	139 min	Sinister characters converge around a young ma…
63442	S.W.A.T.	Clark Johnson	Domenick Lombardozzi	s7913	Movie	January 1, 2021	2003	PG-13	117 min	
29143	Insatiable	nan	Michael Provost	s3431	TV Show	October 11, 2019	2019	TV-MA	2 Seasons	

```
63442  A veteran cop is tasked with drafting and trai…
29143  A bullied teenager turns to beauty pageants as…
```

[ ]:
```
# merging df_new_2(listed_in & country) with df_final - df_final_2
df_final_2 = df_new_2.merge(df_final, how ='inner', on ='title')
df_final_2
```

[ ]:
```
                      title               listed_in        country  \
0       Dick Johnson Is Dead           Documentaries  United States
1              Blood & Water  International TV Shows   South Africa
2              Blood & Water  International TV Shows   South Africa
3              Blood & Water  International TV Shows   South Africa
4              Blood & Water  International TV Shows   South Africa
…                        …                   …              …
202060               Zubaan        Music & Musicals          India
202061               Zubaan        Music & Musicals          India
202062               Zubaan        Music & Musicals          India
202063               Zubaan        Music & Musicals          India
202064               Zubaan        Music & Musicals          India

              director                    cast show_id     type  \
0       Kirsten Johnson                     nan      s1    Movie
1                  nan             Ama Qamata      s2  TV Show
2                  nan             Khosi Ngema      s2  TV Show
3                  nan           Gail Mabalane      s2  TV Show
4                  nan          Thabang Molaba      s2  TV Show
…                  …                       …     …      …
202060      Mozez Singh        Manish Chaudhary   s8807    Movie
202061      Mozez Singh            Meghna Malik   s8807    Movie
202062      Mozez Singh           Malkeet Rauni   s8807    Movie
202063      Mozez Singh          Anita Shabdish   s8807    Movie
202064      Mozez Singh   Chittaranjan Tripathy   s8807    Movie

                date_added  release_year rating   duration  \
0       September 25, 2021          2020  PG-13     90 min
1       September 24, 2021          2021  TV-MA  2 Seasons
2       September 24, 2021          2021  TV-MA  2 Seasons
3       September 24, 2021          2021  TV-MA  2 Seasons
4       September 24, 2021          2021  TV-MA  2 Seasons
…                       …           …    …         …
202060      March 2, 2019          2015  TV-14    111 min
202061      March 2, 2019          2015  TV-14    111 min
202062      March 2, 2019          2015  TV-14    111 min
202063      March 2, 2019          2015  TV-14    111 min
202064      March 2, 2019          2015  TV-14    111 min

                                                description
```

```
0       As her father nears the end of his life, filmm…
1       After crossing paths at a party, a Cape Town t…
2       After crossing paths at a party, a Cape Town t…
3       After crossing paths at a party, a Cape Town t…
4       After crossing paths at a party, a Cape Town t…
…                                                     …
202060  A scrappy but poor boy worms his way into a ty…
202061  A scrappy but poor boy worms his way into a ty…
202062  A scrappy but poor boy worms his way into a ty…
202063  A scrappy but poor boy worms his way into a ty…
202064  A scrappy but poor boy worms his way into a ty…

[202065 rows x 12 columns]
```

[ ]:
```
#3.Handling Missing Values
df_final_2['cast'].replace (['nan'], ['Unknown Actor'], inplace = True)
df_final_2['director'].replace (['nan'], ['Unknown Director'], inplace = True)
df_final_2['country'].replace (['nan'], ['Unknown Country'], inplace = True)
df_final_2['date_added'].fillna('Unknown Date', inplace=True)
df_final_2['rating'].fillna('Unknown Rating', inplace=True)
```

[ ]:
```
#checking for nan in duration col
df_final_2 [df_final_2['duration'].isnull()]
```

[ ]:
```
                                    title listed_in        country  \
126582                     Louis C.K. 2017    Movies  United States
131648                 Louis C.K.: Hilarious    Movies  United States
131782  Louis C.K.: Live at the Comedy Store    Movies  United States

          director        cast show_id   type          date_added  \
126582  Louis C.K.  Louis C.K.   s5542  Movie       April 4, 2017
131648  Louis C.K.  Louis C.K.   s5795  Movie  September 16, 2016
131782  Louis C.K.  Louis C.K.   s5814  Movie     August 15, 2016

        release_year  rating duration  \
126582          2017  74 min      NaN
131648          2010  84 min      NaN
131782          2015  66 min      NaN

                                                description
126582  Louis C.K. muses on religion, eternal love, gi…
131648  Emmy-winning comedy writer Louis C.K. brings h…
131782  The comic puts his trademark hilarious/thought…
```

[ ]:
```
# 3 rows in Duration is NaN,but its located in rating,so to shift this we do :
df_final_2["duration"].fillna(df_final_2[df_final_2["duration"].
 ↪isnull()]["rating"],inplace = True)
```

[ ]:
```
df_final_2
```

[ ]:
```
                        title               listed_in        country  \
0       Dick Johnson Is Dead           Documentaries  United States
1              Blood & Water  International TV Shows   South Africa
2              Blood & Water  International TV Shows   South Africa
3              Blood & Water  International TV Shows   South Africa
4              Blood & Water  International TV Shows   South Africa
…                        …                   …            …
202060               Zubaan        Music & Musicals          India
202061               Zubaan        Music & Musicals          India
202062               Zubaan        Music & Musicals          India
202063               Zubaan        Music & Musicals          India
202064               Zubaan        Music & Musicals          India

                director                    cast show_id     type  \
0        Kirsten Johnson           Unknown Actor      s1    Movie
1       Unknown Director              Ama Qamata      s2  TV Show
2       Unknown Director            Khosi Ngema       s2  TV Show
3       Unknown Director           Gail Mabalane      s2  TV Show
4       Unknown Director          Thabang Molaba      s2  TV Show
…                    …                       …     …      …
202060       Mozez Singh        Manish Chaudhary   s8807    Movie
202061       Mozez Singh            Meghna Malik   s8807    Movie
202062       Mozez Singh           Malkeet Rauni   s8807    Movie
202063       Mozez Singh          Anita Shabdish   s8807    Movie
202064       Mozez Singh   Chittaranjan Tripathy   s8807    Movie

                date_added  release_year rating   duration  \
0       September 25, 2021          2020  PG-13     90 min
1       September 24, 2021          2021  TV-MA  2 Seasons
2       September 24, 2021          2021  TV-MA  2 Seasons
3       September 24, 2021          2021  TV-MA  2 Seasons
4       September 24, 2021          2021  TV-MA  2 Seasons
…                      …           …    …        …
202060       March 2, 2019          2015  TV-14    111 min
202061       March 2, 2019          2015  TV-14    111 min
202062       March 2, 2019          2015  TV-14    111 min
202063       March 2, 2019          2015  TV-14    111 min
202064       March 2, 2019          2015  TV-14    111 min

                                             description
0       As her father nears the end of his life, filmm…
1       After crossing paths at a party, a Cape Town t…
2       After crossing paths at a party, a Cape Town t…
3       After crossing paths at a party, a Cape Town t…
4       After crossing paths at a party, a Cape Town t…
```

```
…                                                       …
202060  A scrappy but poor boy worms his way into a ty…
202061  A scrappy but poor boy worms his way into a ty…
202062  A scrappy but poor boy worms his way into a ty…
202063  A scrappy but poor boy worms his way into a ty…
202064  A scrappy but poor boy worms his way into a ty…

[202065 rows x 12 columns]
```

[ ]:
```
#checking for nan values
df_final_2.isnull().sum()
```

[ ]:
```
title           0
listed_in       0
country         0
director        0
cast            0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
description     0
dtype: int64
```

[ ]:
```
# 2. Observations on the shape of data, data types of all the attributes,␣
 ↪conversion of categorical attributes to 'category',
#   missing value detection, statistical summary

# Check the shape of the DataFrame
print(f"Shape of the DataFrame: {df.shape}")
# Check data types of all columns
print(df_final.dtypes)
```

```
Shape of the DataFrame: (8807, 12)
title           object
director        object
cast            object
show_id         object
type            object
date_added      object
release_year     int64
rating          object
duration        object
description     object
dtype: object
```

[ ]:
```
# Convert categorical columns to 'category' data type
categorical_columns = ['type', 'director', 'cast', 'country', 'rating',␣
 ↪'listed_in']

for col in categorical_columns:
    df_final_2[col] = df_final_2[col].astype('category')

# Verify the changes
print(df_final_2.dtypes)
```

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

[ ]:
```
# Statistical summary for numerical and categorical columns
print(df.describe(include='all'))
```

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

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

```
min                   NaN           NaN              NaN   1925.000000
25%                   NaN           NaN              NaN   2013.000000
50%                   NaN           NaN              NaN   2017.000000
75%                   NaN           NaN              NaN   2019.000000
max                   NaN           NaN              NaN   2021.000000

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

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

[ ]:
```python
# 3. Non-Graphical Analysis: Value Counts and Unique Attributes
# (1) Value Counts for Each Categorical Variable
#     Value counts for 'type'

print("Value counts for 'type':")
print(df_final_2['type'].value_counts())

# Value counts for 'rating'
print("\nValue counts for 'rating':")
print(df_final_2['rating'].value_counts())

# Value counts for 'country'
print("\nValue counts for 'country':")
print(df_final_2['country'].value_counts())
```

```python
# Value counts for 'listed_in'
print("\nValue counts for 'listed_in':")
print(df_final_2['listed_in'].value_counts())

# Value counts for 'director'
print("\nValue counts for 'director':")
print(df_final_2['director'].value_counts())

# Value counts for 'cast'
print("\nValue counts for 'cast':")
print(df_final_2['cast'].value_counts())
```

```
Value counts for 'type':
type
Movie      145917
TV Show     56148
Name: count, dtype: int64

Value counts for 'rating':
rating
TV-MA             73915
TV-14             43957
R                 25860
PG-13             16246
TV-PG             14926
PG                10919
TV-Y7              6304
TV-Y               3665
TV-G               2779
NR                 1573
G                  1530
NC-17               149
UR                   86
TV-Y7-FV             86
Unknown Rating       67
74 min                1
84 min                1
66 min                1
Name: count, dtype: int64

Value counts for 'country':
country
United States      49868
India              22139
Unknown Country    11897
United Kingdom      9733
 United States      9482
```

```
                    …
 Palestine              2
 Ukraine                2
 Nicaragua              1
 Uganda                 1
 Kazakhstan             1
Name: count, Length: 198, dtype: int64

Value counts for 'listed_in':
listed_in
 International Movies    27141
Dramas                   19657
Comedies                 13894
Action & Adventure       12216
 Dramas                  10149
                         …
 Stand-Up Comedy            24
Romantic Movies             20
TV Sci-Fi & Fantasy          7
LGBTQ Movies                 5
Sports Movies                3
Name: count, Length: 73, dtype: int64

Value counts for 'director':
director
Unknown Director       50643
Martin Scorsese          419
Youssef Chahine          409
Cathy Garcia-Molina      356
Steven Spielberg         355
                        …
Robb Dipple                1
James Moll                 1
 Todd Wider                1
 Toby Trackman             1
Alex Stapleton             1
Name: count, Length: 5121, dtype: int64

Value counts for 'cast':
cast
Unknown Actor        2149
 Alfred Molina        160
 Salma Hayek          130
 Frank Langella       128
 John Rhys-Davies     125
                     …
 Quincy Jones III       1
Martin Maloney          1
```

```
Martin Matte            1
Martin Scorsese         1
 Jim Morrison           1
Name: count, Length: 39297, dtype: int64
```

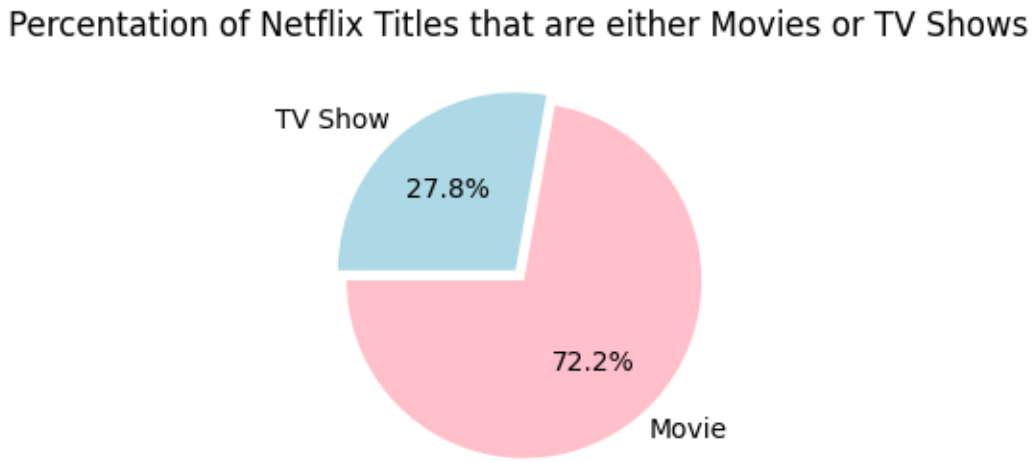
[ ]:
```
#It can be seen that netflix has more movies than TV shows
#TV-MA rating was highest
#Most movies/shows were produced more in US
#There were more of International movies
#Most movies/shows were directed by Martin Scorsese
#Most movies/shows were cast by Alfred Molina
```

[ ]:
```
#(2) Unique Values for Each Categorical Variable
# Unique values for 'type'
print("\nUnique values for 'type':")
print(df_final_2['type'].nunique())

# Unique values for 'rating'
print("\nUnique values for 'rating':")
print(df_final_2['rating'].nunique())

# Unique values for 'country'
print("\nUnique values for 'country':")
print(df_final_2['country'].nunique())

# Unique values for 'listed_in'
print("\nUnique values for 'listed_in':")
print(df_final_2['listed_in'].nunique())

# Unique values for 'director'
print("\nUnique values for 'director':")
print(df_final_2['director'].nunique())

# Unique values for 'cast'
print("\nUnique values for 'cast':")
print(df_final_2['cast'].nunique())
```

```
Unique values for 'type':
2

Unique values for 'rating':
18

Unique values for 'country':
198

Unique values for 'listed_in':
```

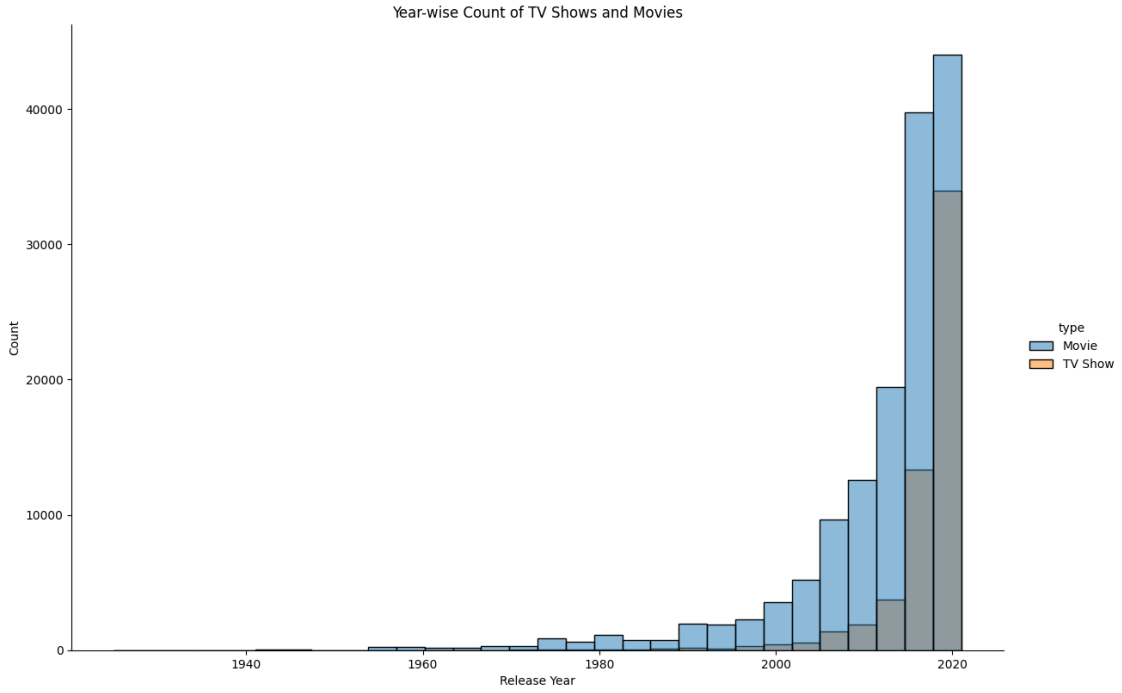
```
73

Unique values for 'director':
5121

Unique values for 'cast':
39297
```

[ ]:
```
# 4. Visual Analysis - Univariate, Bivariate after pre-processing of the data
#4.1 For continuous variable(s):
# Create the pie chart
plt.figure(figsize=(6,3))
plt.title("Percentation of Netflix Titles that are either Movies or TV Shows")
g=plt.pie(df_final_2.type.value_counts(),explode=(0.05,0.025),
labels=df_final_2.type.value_counts().index, colors=['pink',␣
 ↪'lightblue'],autopct='%1.1f%%',
startangle=180)
```



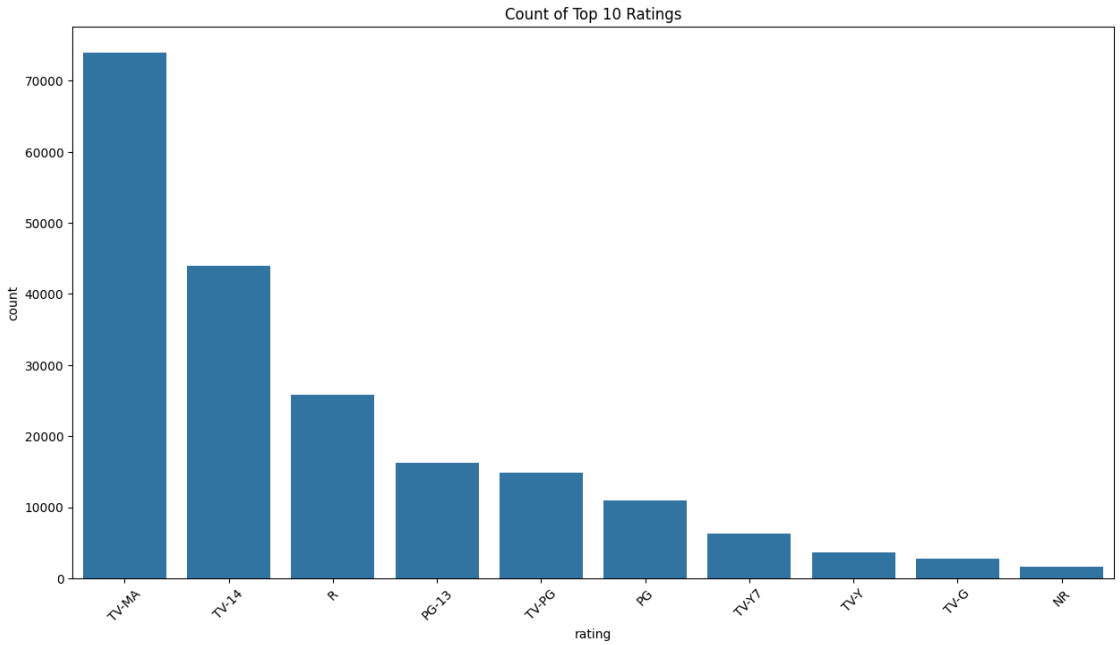
[ ]:
```
#INSIGHTS
# There are far more movie titles (72.8%) that TV shows titles (27.2%) in terms␣
 ↪of title.
```

[ ]:
```
# Distribution plot for release_year by type by DISPLOT
sns.displot(df_final_2, x='release_year', hue='type', kind='hist', height=8,␣
 ↪aspect=1.5, bins=30)
plt.title('Year-wise Count of TV Shows and Movies')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.show()
```



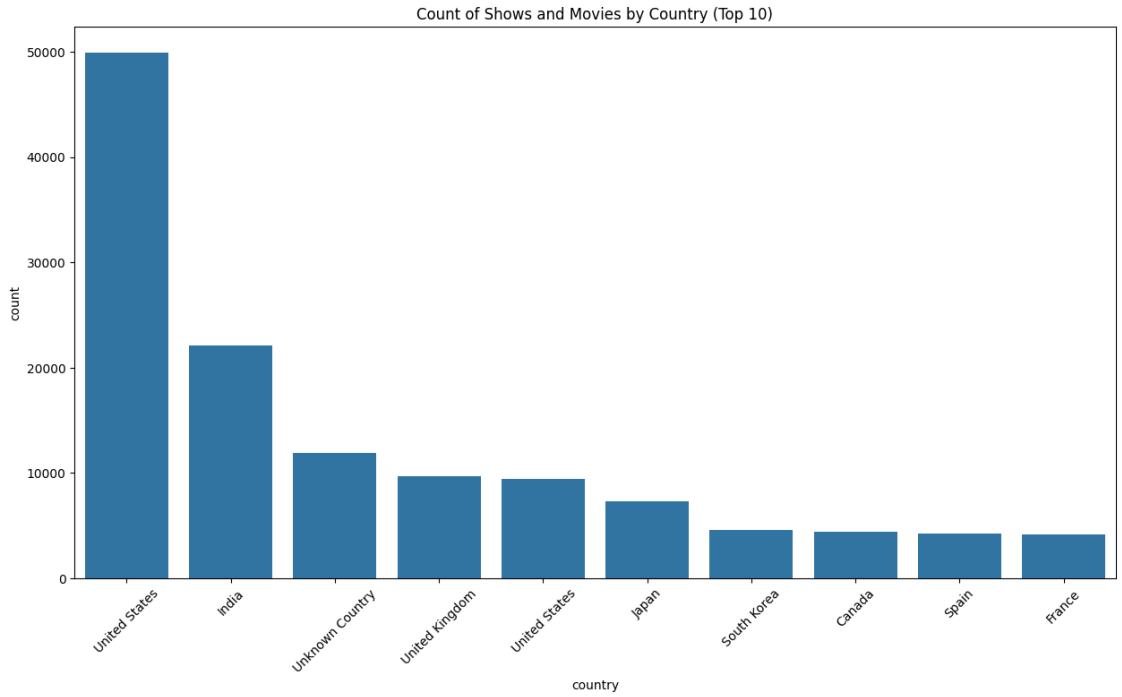
[ ]:
```
# We can observe that every year movies were released more than TV shows.
# The movies released in kept increasing with time and highest in 2020
```

[ ]:
```
# Get the top 10 most frequent ratings
top_10_ratings = df_final_2['rating'].value_counts().nlargest(10).index

# Filter the dataframe to include only the top 10 ratings
df_top_10_ratings = df_final_2[df_final_2['rating'].isin(top_10_ratings)]

# Create the count plot for the top 10 ratings
plt.figure(figsize=(15, 8))
sns.countplot(data=df_top_10_ratings, x='rating', order=top_10_ratings)
plt.title('Count of Top 10 Ratings')
plt.xticks(rotation=45)
plt.show()
```



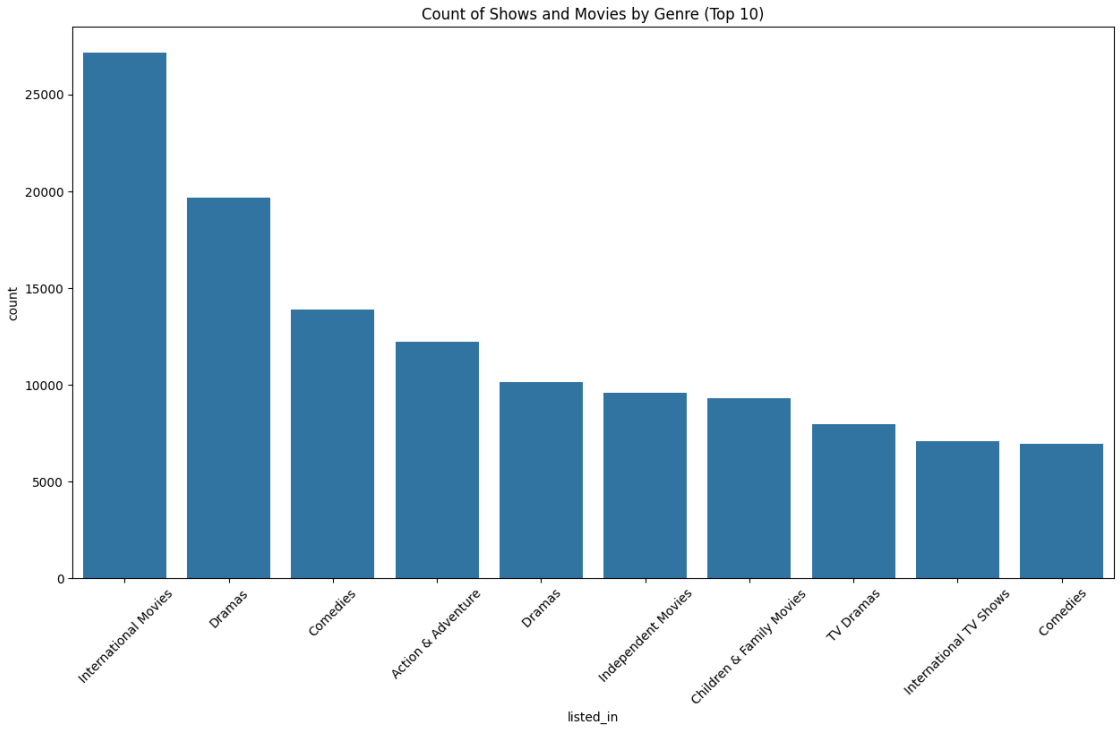
[ ]:
```
# we can see that movies/shows has more TV-MA rating
```

[ ]:
```
# Top 10 countries by count
top_countries = df_final_2['country'].value_counts().nlargest(10).index

# Filter the dataframe to include only the top 10 countries
filtered_countries_df = df_final_2[df_final_2['country'].isin(top_countries)]

# Plot count plot for top 10 countries
plt.figure(figsize=(15, 8))
sns.countplot(data=filtered_countries_df, x='country', order=top_countries)
plt.title('Count of Shows and Movies by Country (Top 10)')
plt.xticks(rotation=45)
plt.show()
```



[ ]:
```
#Most movies/shows were produced more in US
```

[ ]:
```
#Count Plot for Listed Genres (Top 10)
# Top 10 genres by count
top_genres = df_final_2['listed_in'].value_counts().nlargest(10).index

# Filter the dataframe to include only the top 10 genres
filtered_genres_df = df_final_2[df_final_2['listed_in'].isin(top_genres)]

# Plot count plot for top 10 genres
plt.figure(figsize=(15, 8))
sns.countplot(data=filtered_genres_df, x='listed_in', order=top_genres)
plt.title('Count of Shows and Movies by Genre (Top 10)')
plt.xticks(rotation=45)
plt.show()
```



[ ]:
```
#There were more of International movies
```

[ ]:
```
# Comparison of TV Shows vs. Movies

#Movies Produced by Country (Top 10)
#Group by country and count the number of unique movie titles
# Group by country and filter for MOVIES

movies_by_country = df_final_2[df_final_2['type'] == 'Movie'].
 ↪groupby('country')['title'].nunique().nlargest(10)
print(movies_by_country,"movies_by_country")
```

```
country
United States      2364
India               927
Unknown Country     440
 United States      388
United Kingdom      382
Canada              187
 France             155
 United Kingdom     152
France              148
 Canada             132
Name: title, dtype: int64 movies_by_country
```

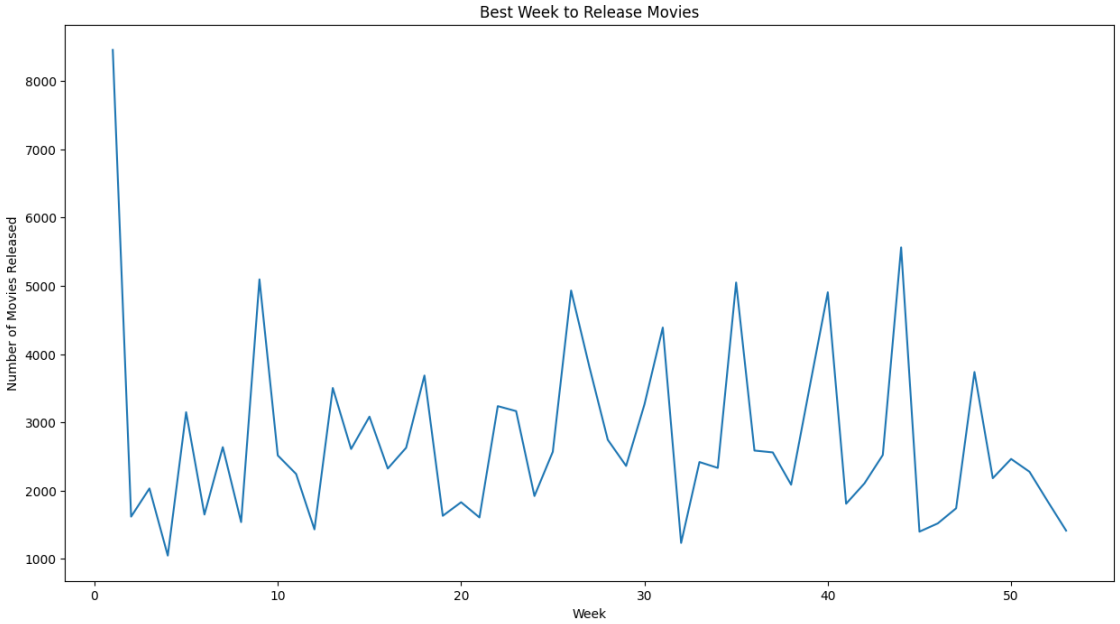
[ ]:
```
#TV Shows Produced by Country (Top 10)
#Group by country and count the number of unique TV show titles.
# Group by country and filter for TV SHOWS

tv_shows_by_country = df_final_2[df_final_2['type'] == 'TV Show'].
 ↪groupby('country')['title'].nunique().nlargest(10)
print(tv_shows_by_country,"tv_shows_by_country")
```

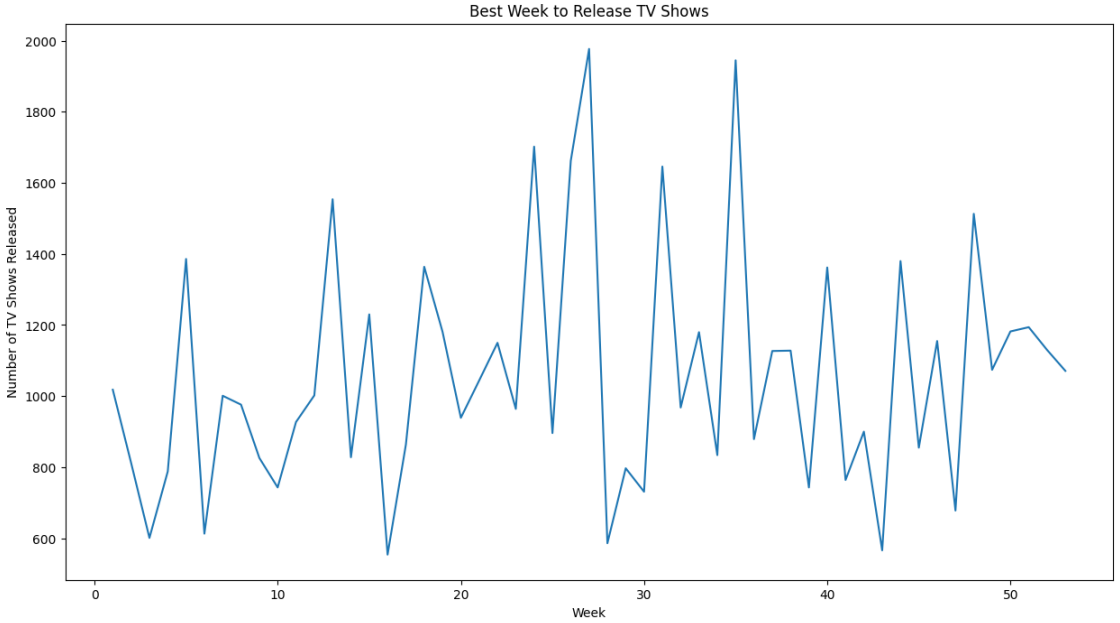
```
country
United States      847
Unknown Country    391
United Kingdom     246
Japan              174
South Korea        164
 United States      91
Canada              84
India               81
Taiwan              70
France              64
Name: title, dtype: int64 tv_shows_by_country
```

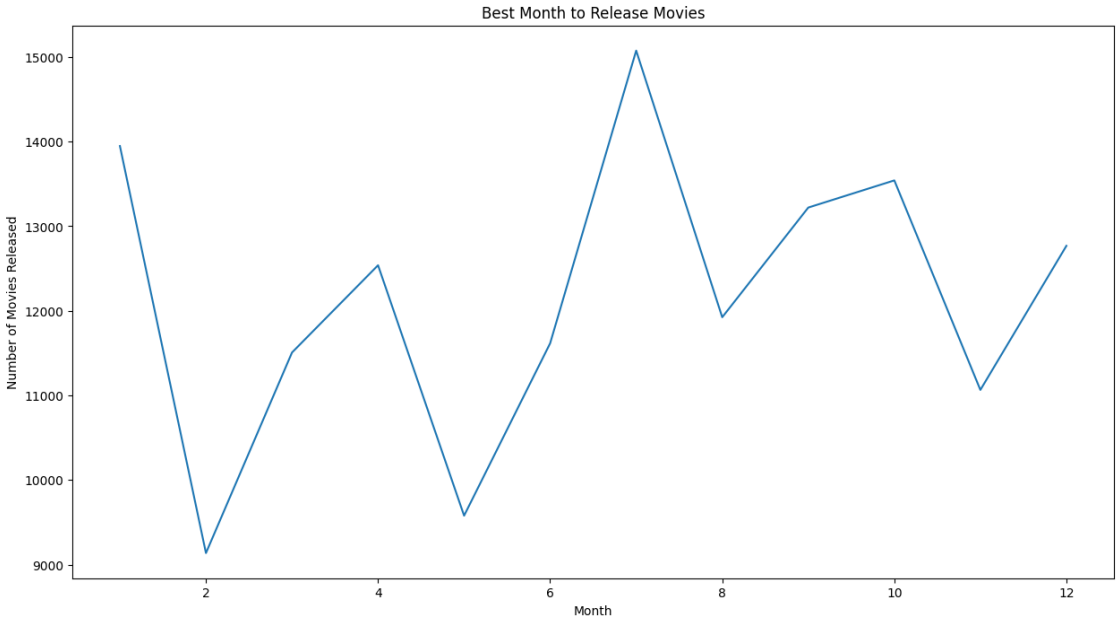
[ ]:
```
#Visualizing Best Week and Month to Release Content

#We can observe that every year movies were released more than TV shows.
# The movies released in kept increasing with time and highest in 2020
#Best Time to Launch a TV Show
#Best Week to Release Content
#Create a new column for the week and group by it to count the number of␣
 ↪releases.

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

# Replace "Unknown Date" with NaN
df_final_2['date_added'] = df_final_2['date_added'].replace("Unknown Date", np.
 ↪nan)

# Drop rows with NaN in 'date_added'
df_final_2 = df_final_2.dropna(subset=['date_added'])

# Strip any leading/trailing spaces in 'date_added'
df_final_2['date_added'] = df_final_2['date_added'].str.strip()

# Convert to datetime
```

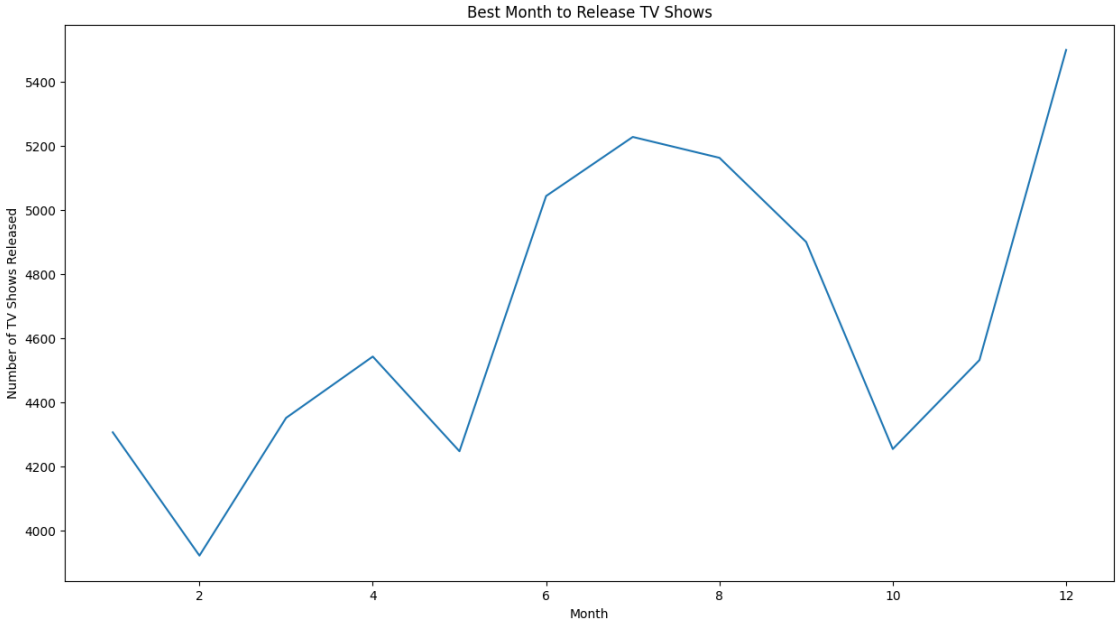
```
df_final_2['date_added'] = pd.to_datetime(df_final_2['date_added'], format='%B␣
 ↪%d, %Y')

# Create the 'week' and 'month' columns
df_final_2['week'] = df_final_2['date_added'].dt.isocalendar().week
df_final_2['month'] = df_final_2['date_added'].dt.month

# Group by week for movies
best_week_movies = df_final_2[df_final_2['type'] == 'Movie'].
 ↪groupby('week')['title'].count()

# Group by week for TV shows
best_week_tv_shows = df_final_2[df_final_2['type'] == 'TV Show'].
 ↪groupby('week')['title'].count()

# Group by month for movies
best_month_movies = df_final_2[df_final_2['type'] == 'Movie'].
 ↪groupby('month')['title'].count()

# Group by month for TV shows
best_month_tv_shows = df_final_2[df_final_2['type'] == 'TV Show'].
 ↪groupby('month')['title'].count()

# Plot for best week (Movies)
plt.figure(figsize=(15, 8))
sns.lineplot(x=best_week_movies.index, y=best_week_movies.values)
plt.title('Best Week to Release Movies')
plt.xlabel('Week')
plt.ylabel('Number of Movies Released')
plt.show()

# Plot for best week (TV Shows)
plt.figure(figsize=(15, 8))
sns.lineplot(x=best_week_tv_shows.index, y=best_week_tv_shows.values)
plt.title('Best Week to Release TV Shows')
plt.xlabel('Week')
plt.ylabel('Number of TV Shows Released')
plt.show()

# Plot for best month (Movies)
plt.figure(figsize=(15, 8))
sns.lineplot(x=best_month_movies.index, y=best_month_movies.values)
plt.title('Best Month to Release Movies')
plt.xlabel('Month')
plt.ylabel('Number of Movies Released')
plt.show()
```

```
# Plot for best month (TV Shows)
plt.figure(figsize=(15, 8))
sns.lineplot(x=best_month_tv_shows.index, y=best_month_tv_shows.values)
plt.title('Best Month to Release TV Shows')
plt.xlabel('Month')
plt.ylabel('Number of TV Shows Released')
plt.show()
```









[ ]:
```
# Best Week to release:
# Movies: Week 1 (Beginning of the year)
# TV Shows: Week 27 and week 45
# Best Month to release:
# Movies: January and July
```

```
# TV Shows: July and December
```

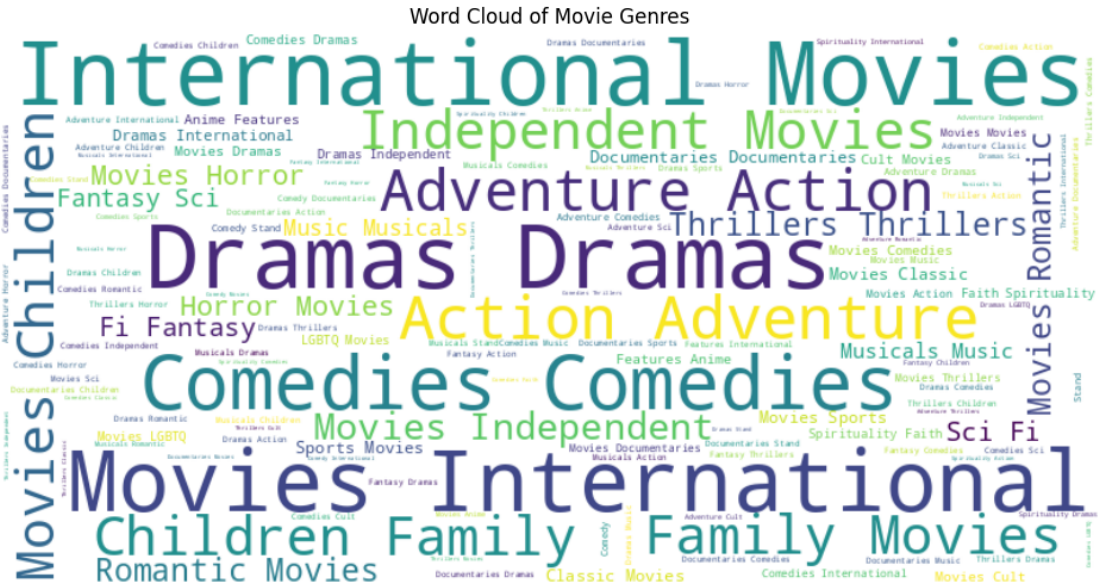
[ ]:
```
#Top 10 Actors Who Have Appeared in the Most Movies or TV Shows

# Grouping by actor and counting unique titles for movies
actor_movie_counts = df_final_2[df_final_2['type'] == 'Movie'].
 ↪groupby('cast')['title'].nunique()

# Grouping by actor and counting unique titles for TV shows
actor_tv_counts = df_final_2[df_final_2['type'] == 'TV Show'].
 ↪groupby('cast')['title'].nunique()

# Concatenating movie and TV show counts for each actor
actor_counts = actor_movie_counts.add(actor_tv_counts, fill_value=0)

# Selecting the top 10 actors
top_10_actors = actor_counts.sort_values(ascending=False).head(10)

# Displaying the top 10 actors
print("Top 10 Actors Who Have Appeared in the Most Movies or TV Shows:")
print(top_10_actors)
```

```
Top 10 Actors Who Have Appeared in the Most Movies or TV Shows:
cast
Unknown Actor        825
 Anupam Kher          39
 Rupa Bhimani         31
 Takahiro Sakurai     30
 Julie Tejwani        28
 Om Puri              27
 Rajesh Kava          26
Shah Rukh Khan        26
 Boman Irani          25
 Paresh Rawal         25
Name: title, dtype: int64
```

[ ]:
```
#Anupam Kher has appeared in the most movies or TV shows.
```

[ ]:
```
#Creating a Word Cloud for Movie Genres

from wordcloud import WordCloud

# Concatenate all genres into a single string
genres_text = ' '.join(df_final_2[df_final_2['type'] == 'Movie']['listed_in'])

# Generate the word cloud
```

```
wordcloud = WordCloud(width=800, height=400, background_color='white').
 ↪generate(genres_text)

# Display the word cloud
plt.figure(figsize=(12, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.title('Word Cloud of Movie Genres')
plt.axis('off')
plt.show()
```



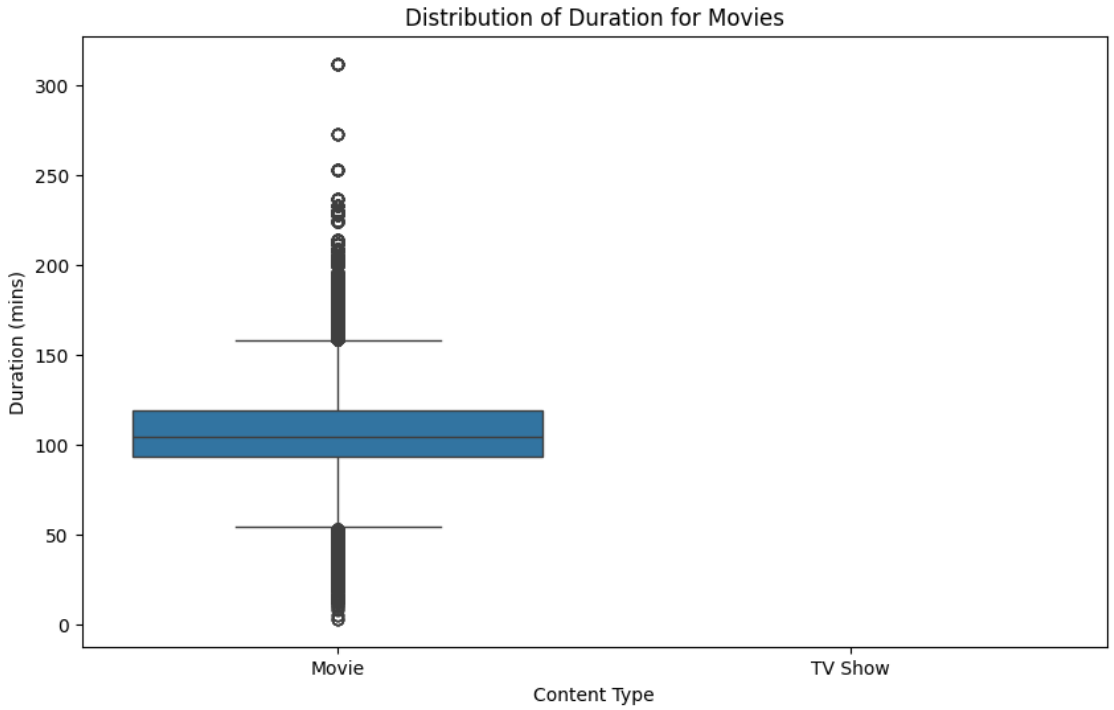
[ ]:
```
#International movies are the most popular genre in Netflix
```

[ ]:
```
#Finding the Time Difference between Release Year and Date Added

# Sample DataFrame definition
# Assuming df_final_2 is already defined

# Replace "Unknown Date" with NaN
df_final_2['date_added'] = df_final_2['date_added'].replace("Unknown Date", np.
 ↪nan)

# Drop rows with NaN in 'date_added'
df_final_2 = df_final_2.dropna(subset=['date_added'])

# Convert 'date_added' column to datetime if not already in datetime format
if df_final_2['date_added'].dtype != '<M8[ns]':
```

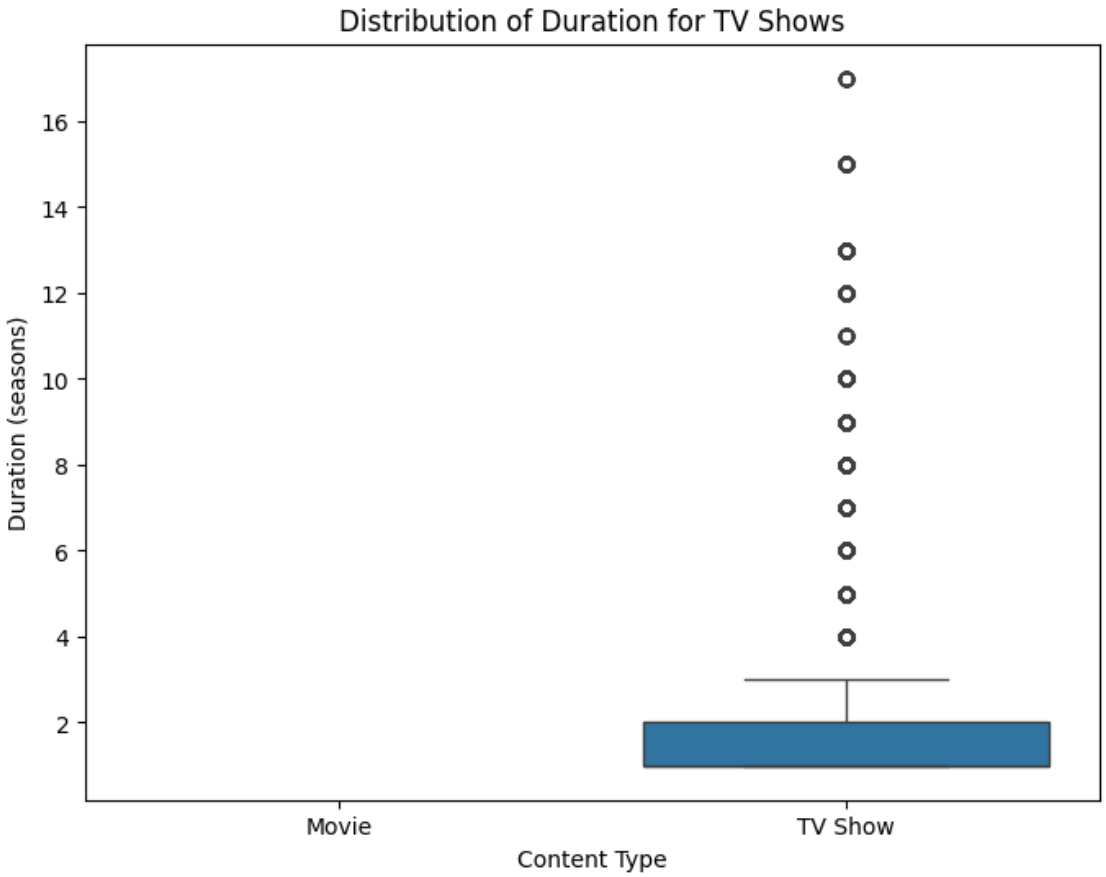
```
    df_final_2['date_added'] = pd.to_datetime(df_final_2['date_added'],␣
 ↪format='%B %d, %Y')

# Convert 'release_year' to datetime with only the year part
df_final_2['release_year'] = pd.to_datetime(df_final_2['release_year'].
 ↪astype(str), format='%Y')

# Calculate the time difference in days
df_final_2['days_to_add'] = (df_final_2['date_added'] -␣
 ↪df_final_2['release_year']).dt.days

# Get the mode of the time difference
mode_days_to_add = df_final_2['days_to_add'].mode()[0]

# Display the mode of the time difference
print("Mode of Days to Add After Release to Netflix:", mode_days_to_add, "days")
```

```
Mode of Days to Add After Release to Netflix: 547 days
```

[ ]:
```
#4.2 For categorical variable(s): Boxplot
#Duration Distribution for Movies and TV Shows

netflix_movies_df = df_final_2[df_final_2.type.str.contains("Movie")].copy()
netflix_movies_df['duration'] = netflix_movies_df['duration'].str.
 ↪extract('(\d+)', expand=False).astype(int)

# Creating a boxplot for movie duration
plt.figure(figsize=(10, 6))
sns.boxplot(data=netflix_movies_df, x='type', y='duration')
plt.xlabel('Content Type')
plt.ylabel('Duration (mins)')
plt.title('Distribution of Duration for Movies')
plt.show()
```



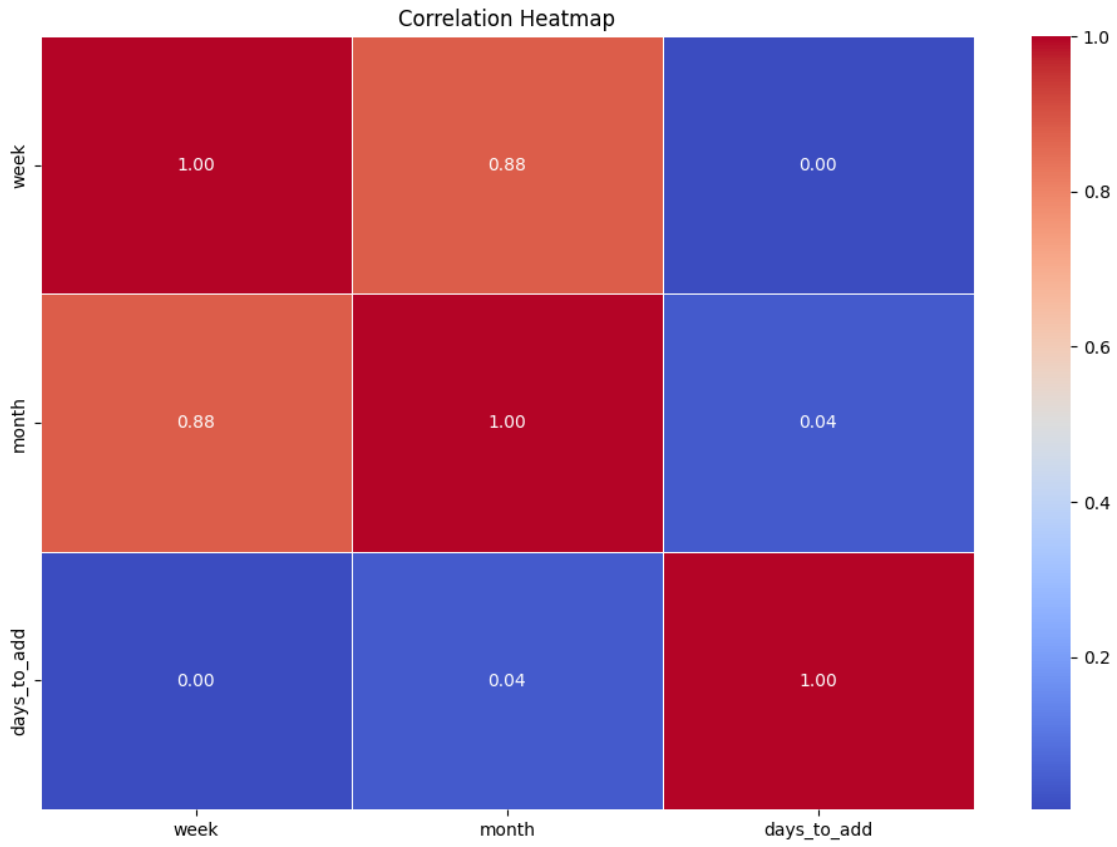
[ ]:
```
netflix_shows_df = df_final_2[df_final_2.type.str.contains("TV Show")].copy()
netflix_shows_df['duration'] = netflix_shows_df['duration'].str.
 ↪extract('(\d+)', expand=False).astype(int)

# Creating a boxplot for TV shows duration
plt.figure(figsize=(8, 6))
sns.boxplot(data=netflix_shows_df, x='type', y='duration')
plt.xlabel('Content Type')
plt.ylabel('Duration (seasons)')
plt.title('Distribution of Duration for TV Shows')
plt.show()
```

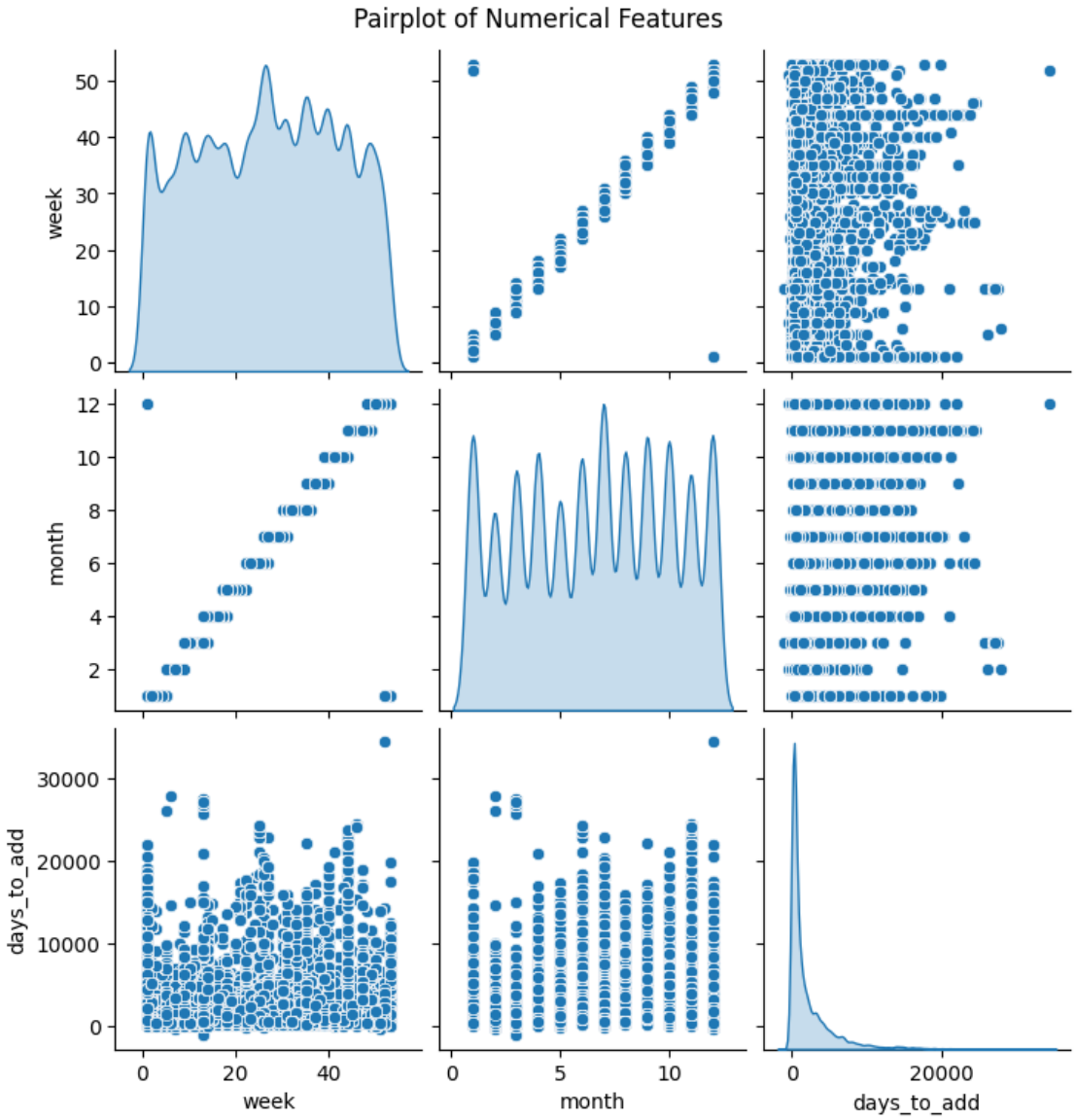


[ ]:
```
#Analysing the movie box plot, we can see that most movies fall within a␣
 ↪reasonable duration
# range, with few outliers exceedingly approximately 2.5 hours. This suggests␣
 ↪that most
# movies on Netflix are designed to fit within a standard viewing time.
# For TV shows, the box plot reveals that most shows have one to four seasons,␣
 ↪with very few
# outliers having longer durations. This aligns with the earlier trends,␣
 ↪indicating that Netflix
# focuses on shorter series formats.
```

[ ]:
```
#4.3 For Correlation:
#Heatmap and Pairplot

# Remove non-numeric columns
numeric_df = df_final_2.select_dtypes(include=np.number)

# Calculate the correlation matrix
correlation_matrix = numeric_df.corr()
```

```
# Plot the heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",␣
 ↪linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()

# Select relevant numerical columns for pairplot
numerical_columns = ['release_year', 'duration_numeric', 'days_to_add']

# Update column selection based on existing numerical columns
numerical_columns = df_final_2.select_dtypes(include=np.number).columns

# Plot the pairplot
sns.pairplot(df_final_2[numerical_columns], diag_kind='kde')
plt.suptitle('Pairplot of Numerical Features', y=1.02)
plt.show()
```



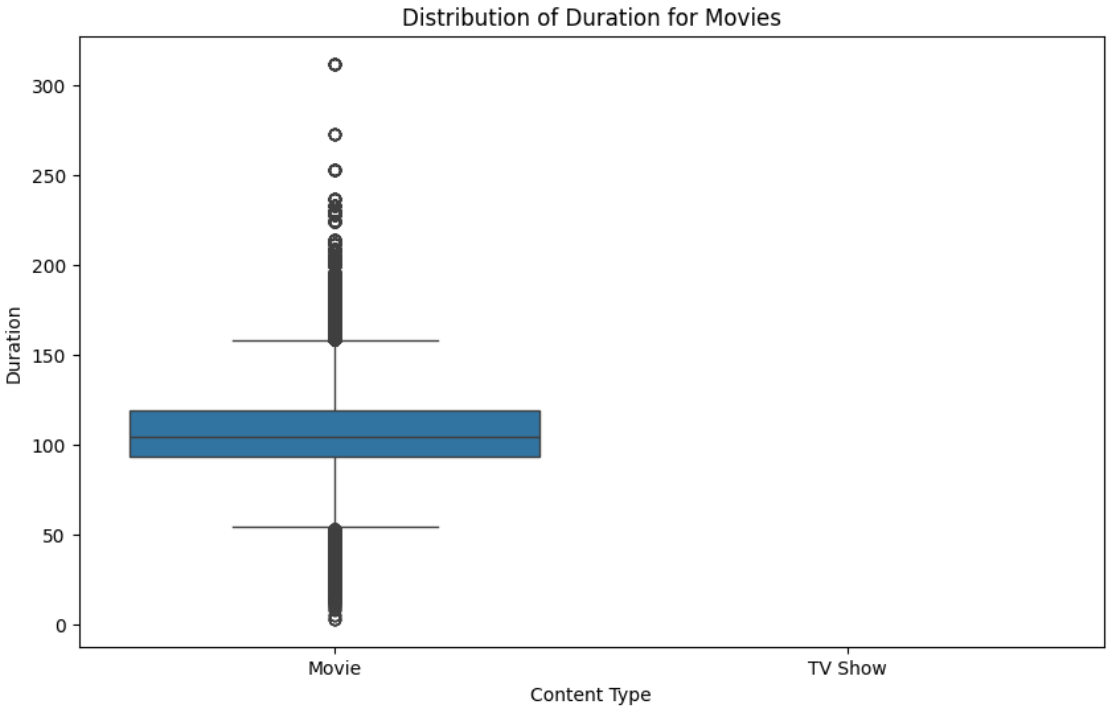


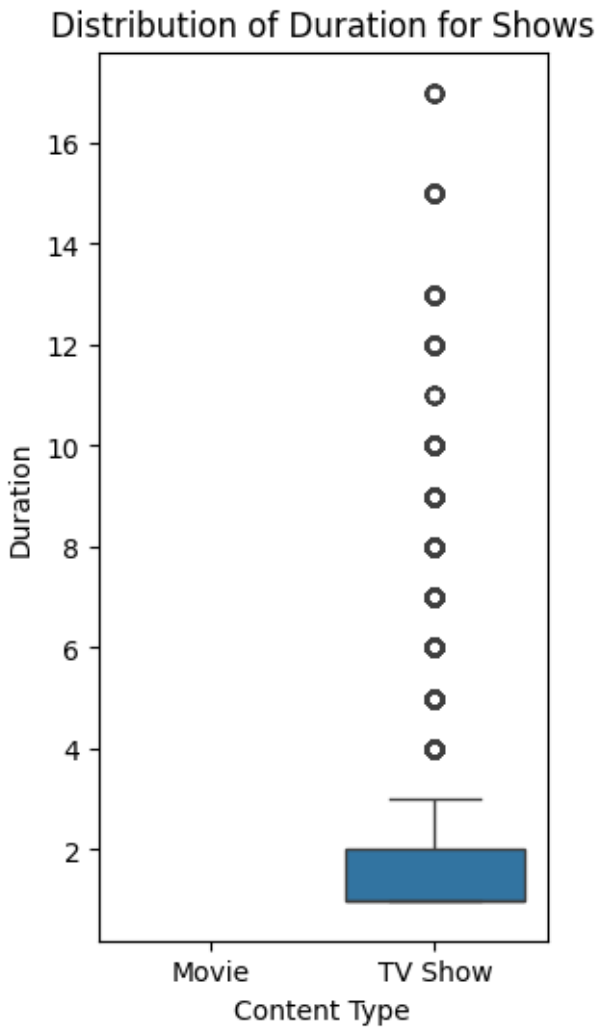
[ ]:
```
#5. Missing Value Check  & Outlier check
#We'll first check for missing values in the dataset and then decide how to␣
 ↪handle them.

# Check for missing values
missing_values = df_final_2.isnull().sum()
print("Missing Values:")
print(missing_values)

# Handle missing values (if necessary)
# Example: Replace missing values in 'rating' with the mode
# df_final_2['rating'].fillna(df_final_2['rating'].mode()[0], inplace=True)
```

```
Missing Values:
title           0
listed_in       0
country         0
director        0
cast            0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
description     0
week            0
month           0
days_to_add     0
dtype: int64
```

[ ]:
```
#  What is an outlier?
# In a random sampling from a population, an outlier is defined as an␣
 ↪observation that deviates abnormally from the standard data. In simple␣
 ↪words, an
# outlier is used to define those data values which are far away from the␣
 ↪general valuesin a dataset. An outlier can be broken down into out-of-line␣
 ↪data.
# For example, let us consider a row of data [10,15,22,330,30,45,60]. In this␣
 ↪dataset, we can easily conclude that 330 is way off from the rest of the
# values in the dataset, thus 330 is an outlier. It was easy to figure out the␣
 ↪outlier in such a small dataset, but when the dataset is huge,
# we need various methods to determine whether a certain value is an outlier or␣
 ↪necessary information.

# Why do we need to treat outliers?
# Outliers can lead to vague or misleading predictions while using machine␣
 ↪learning models. Specific models like linear regression, logistic regression,
# and support vector machines are susceptible to outliers. Outliers decrease␣
 ↪the mathematical power of these models, and thus the output of the models
# becomes unreliable. However, outliers are highly subjective to the dataset.␣
 ↪Some outliers may portray extreme changes in the data as well.

# Visual Detection
# Box plots are a simple way to visualize data through quantiles and detect␣
 ↪outliers. IQR(Interquartile Range) is the basic mathematics behind boxplots.
# The top and bottom whiskers can be understood as the boundaries of data, and␣
 ↪any data lying outside it will be an outlier.
```

[ ]:
```
# For categorical variable(s): Boxplot
# Duration Distribution for Movies and TV Shows Analysing the duration␣
 ↪distribution for movies and TV shows allows us to understand the typical␣
 ↪length
# of content available on Netflix. We can create box plots to visualize these␣
 ↪distributions and identify outliers or standard durations.

# Creating a boxplot for movie duration
netflix_movies_df = df_final_2[df_final_2.type.str.contains("Movie")].copy()
netflix_movies_df['duration'] = netflix_movies_df['duration'].str.
 ↪extract('(\d+)',

expand=False).astype(int)


plt.figure(figsize=(10, 6))

sns.boxplot(data=netflix_movies_df, x='type', y='duration')

plt.xlabel('Content Type')
plt.ylabel('Duration')

plt.title('Distribution of Duration for Movies')

plt.show()

# Creating a boxplot for TV show duration
netflix_shows_df = df_final_2[df_final_2.type.str.contains("TV Show")].copy()
netflix_shows_df['duration'] = netflix_shows_df['duration'].str.extract('(\d+)',

expand=False).astype(int)


plt.figure(figsize=(3, 6))

sns.boxplot(data=netflix_shows_df, x='type', y='duration')

plt.xlabel('Content Type')
plt.ylabel('Duration')

plt.title('Distribution of Duration for Shows')

plt.show()
```


Distribution of Duration for Movies
Duration
300
250
200
150
100
50
0
Movie
TV Show
Content Type



[ ]:
```
# Analysing the movie box plot, we can see that most movies fall within a␣
 ↪reasonable duration range, with few outliers exceedingly
# approximately 2.5 hours. This suggests that most movies on Netflix are␣
 ↪designed to fit within a standard viewing time.
# For TV shows, the box plot reveals that most shows have one to four seasons,␣
 ↪with very few outliers having longer durations.
# This aligns with the earlier trends, indicating that Netflix focuses on␣
 ↪shorter series formats.
```

[ ]:
```
# Business Insights :

# With the help of this article, we have been able to learn about-
# 1. Quantity: Our analysis revealed that Netflix had added more movies than TV␣
 ↪shows,
```

```
# aligning with the expectation that movies dominate their content library.
# 2. Content Addition: July emerged as the month when Netflix adds the most␣
 ↪content,
# closely followed by December, indicating a strategic approach to content␣
 ↪release.
# 3. Genre Correlation: Strong positive associations were observed between␣
 ↪various
# genres, such as TV dramas and international TV shows, romantic and␣
 ↪international
# TV shows, and independent movies and dramas. These correlations provide␣
 ↪insights
# into viewer preferences and content interconnections.
# 4. Movie Lengths: The analysis of movie durations indicated a peak around the␣
 ↪1960s,
# followed by a stabilization around 100 minutes, highlighting a trend in movie␣
 ↪lengths
# over time.
# 5. TV Show Episodes: Most TV shows on Netflix have one season, suggesting a
# preference for shorter series among viewers.
# 6. Common Themes: Words like love, life, family, and adventure were␣
 ↪frequently found
# in titles and descriptions, capturing recurring themes in Netflix content.
# 7. Rating Distribution: The distribution of ratings over the years offers␣
 ↪insights into the
# evolving content landscape and audience reception.
# 8. Data-Driven Insights: Our data analysis journey showcased the power of␣
 ↪data in
# unravelling the mysteries of Netflix's content landscape, providing valuable␣
 ↪insights
# for viewers and content creators.
# 9. Continued Relevance: As the streaming industry evolves, understanding these
# patterns and trends becomes increasingly essential for navigating the dynamic
# landscape of Netflix and its vast library.
# 10. Happy Streaming: We hope this blog has been an enlightening and␣
 ↪entertaining
# journey into the world of Netflix, and we encourage you to explore the␣
 ↪captivating
# stories within its ever-changing content offerings. Let the data guide your␣
 ↪streaming
# adventures!
```

[ ]:
```
# RECOMMENDATIONS
# 1. Netflix has to focus on TV Shows also because there are people who will␣
 ↪like to see
# tv shows rather than movies
```

```
# 2. By approaching the top director we can plan some more movies/tv shows in␣
 ↪order to
# increase the popularity
# 3. Not only reaching top director we can also see the director with less no␣
 ↪of movies
# and having high rating as there may be some financial
# 4. issues or anything so inorder to get good content netflix can reach to␣
 ↪them and netflix
# can produce the movie and give the director a chance.
# 5.We have seen most no of international movies genre so need to give priority␣
 ↪to other
# geners like hooro,comedy..etc
# 6. In TV Shows we may focus on thriller genre which will be helpfull for␣
 ↪having more no
# of seasons
# 7. Most of the movies released in ott is in a year 2019 so we need to go on␣
 ↪increasing
# this value in order to attract people by showing that
# 8. getting subscription is usefull as netflix is releasing more movies per␣
 ↪year
# 9. Mainly the release in ott should focus on the festival holidays, year end␣
 ↪and week
# ends which is to be mainly focussed
# 10. Some movies can be released directly into ott which has some positive␣
 ↪talk which
# may help in improving subscriptions
# 11. Should focus on a actor who has immense following and make use of it by␣
 ↪doing a
# TV Shows or web series
# 12. Advertisement in the country which has very less movies released should be
# increased and attract people of that country by making their native TV Shows
```