Business Case: Netflix - Data Exploration and Visualization

1. **Problem Statement:** We want to analyze Netflix's dataset to provide insights on what type of shows/movies to produce and how to grow the business in different countries.
2. **Basic Metrics Analysis**: We'll examine data kinds, missing values, statistical summaries, and data form.

```
import numpy as np
import pandas as pd
df = pd.read_csv("netflix.csv")
df.head()
```

1 to 5 of 5 entries Filter

index	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, filmmaker Kirsten Johnson stages his death in inventive and comical ways to help them both face the inevitable.
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmore Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake Ka-Ncube, Odwa Gwanya, Mekaila Mathys, Sandi Schultz, Duane Williams, Shamilla Miller, Patrick Mofokeng	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 teen sets out to prove whether a private-school swimming star is her sister who was abducted at birth.
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabiha Akkari, Sofia Lesaffre, Salim Kechiouche, Noureddine Farihi, Geert Van Rampelberg, Bakary Diombera	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Action & Adventure	To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team of robbers are pulled into a violent and deadly turf war.
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 among the incarcerated women at the Orleans Justice Center in New Orleans on this gritty reality series.
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam Khan, Ahsaas Channa, Revathi Pillai, Urvi Singh, Arun Kumar	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV Comedies	In a city of coaching centers known to train India's finest collegiate minds, an earnest but unexceptional student and his friends navigate campus life.

```
df.info()
```

```
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
```

Shape of Data:

```
print("Shape of the data:", df.shape)
```

```
Shape of the data: (8807, 12)
```

Number of Columns & Rows:

```
print("Number of columns:", df.shape[1])
print("number of rows:", df.shape[0])
```

```
Number of columns: 12
number of rows: 8807
```

Data Type and Attribute:

```
print("\nData types of attributes:\n", df.dtypes)
```

```
Data types of attributes:
 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
```

Convert Categorical attributes to category:

```
df['country'] = df['country'].astype('category')
df['rating'] = df['rating'].astype('category')
df['listed_in'] = df['listed_in'].astype('category')
df['type'] = df['type'].astype('category')
df['title'] = df['title'].astype('category')
df['description'] = df['description'].astype('category')
```

```
df.dtypes
```

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

Detecting Missing Values

```
print("\nMissing values:\n", df.isnull().sum())
```

```
Missing values:
 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
```

Ststistical Summanry

```
print("\nStatistical summary:\n", df.describe())
```

```
Statistical summary:
        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
```

3. **Non-Graphical Analysis:** We'll perform value Counts and check unique attributes.

```
print("\nValue counts for Type:\n", df['type'].value_counts())
```

```
Value counts for Type:
 Movie      6131
TV Show    2676
Name: type, dtype: int64
```

Country counts:

```
dfs = pd.read_csv("netflix.csv")
dfs['country'] = dfs['country'].fillna('')  # Fill missing values with
empty string
dfs['country'] = dfs['country'].str.replace(', ', ',')  # Replace empty
spaces with ','
dfs['country'] = dfs['country'].str.split(',')
dfs['country'] = dfs['country'].apply(lambda x: [val for val in x if
val])  # Remove empty quotes
dfs = dfs.explode('country')
print("Number of unique Countries:\n",dfs['country'].nunique(), "\n")
print("\nunique Countries:\n",dfs['country'].unique(), "\n")
print("\nCountries Value_counts:\n",dfs['country'].value_counts(), "\n")
```

```
Number of unique Countries:
 122


unique Countries:
 ['United States' 'South Africa' nan 'India' 'Ghana' 'Burkina Faso'
 'United Kingdom' 'Germany' 'Ethiopia' 'Czech Republic' 'Mexico' 'Turkey'
 'Australia' 'France' 'Finland' 'China' 'Canada' 'Japan' 'Nigeria' 'Spain'
 'Belgium' 'South Korea' 'Singapore' 'Italy' 'Romania' 'Argentina'
 'Venezuela' 'Hong Kong' 'Russia' 'Ireland' 'Nepal' 'New Zealand' 'Brazil'
 'Greece' 'Jordan' 'Colombia' 'Switzerland' 'Israel' 'Taiwan' 'Bulgaria'
 'Algeria' 'Poland' 'Saudi Arabia' 'Thailand' 'Indonesia' 'Egypt'
 'Denmark' 'Kuwait' 'Netherlands' 'Malaysia' 'Vietnam' 'Hungary' 'Sweden'
 'Lebanon' 'Syria' 'Philippines' 'Iceland' 'United Arab Emirates' 'Norway'
 'Qatar' 'Mauritius' 'Austria' 'Cameroon' 'Palestine' 'Uruguay' 'Kenya'
 'Chile' 'Luxembourg' 'Cambodia' 'Bangladesh' 'Portugal' 'Cayman Islands'
 'Senegal' 'Serbia' 'Malta' 'Namibia' 'Angola' 'Peru' 'Mozambique'
 'Belarus' 'Zimbabwe' 'Puerto Rico' 'Pakistan' 'Cyprus' 'Guatemala' 'Iraq'
 'Malawi' 'Paraguay' 'Croatia' 'Iran' 'West Germany' 'Albania' 'Georgia'
 'Soviet Union' 'Morocco' 'Slovakia' 'Ukraine' 'Bermuda' 'Ecuador'
 'Armenia' 'Mongolia' 'Bahamas' 'Sri Lanka' 'Latvia' 'Liechtenstein'
 'Cuba' 'Nicaragua' 'Slovenia' 'Dominican Republic' 'Samoa' 'Azerbaijan'
 'Botswana' 'Vatican City' 'Jamaica' 'Kazakhstan' 'Lithuania'
 'Afghanistan' 'Somalia' 'Sudan' 'Panama' 'Uganda' 'East Germany'
 'Montenegro']
```

```
Countries Value_counts:
 United States     3689
India             1046
United Kingdom     804
Canada             445
France             393
                   ...
Bermuda              1
Ecuador              1
Armenia              1
Mongolia             1
Montenegro           1
Name: country, Length: 127, dtype: int64
```

Insights:

- United States: With 3690 entries, the United States dominates Netflix's library, showcasing the prominence of American entertainment globally.

- India: With 1046 entries, India's significant presence underscores the platform's focus on catering to the diverse Indian audience and promoting Indian cinema.
- United Kingdom: The UK contributes 806 entries, reflecting the richness of British productions and their appeal to global audiences on Netflix.
- Canada: Canada's 445 entries highlight the platform's commitment to showcasing Canadian content and its recognition of Canadian storytelling.
- France: France's 393 entries represent the platform's dedication to offering French-language content and French co-productions to its audience.
- Other Countries: Various countries, with entries ranging from 1 to several hundred, contribute to Netflix's diverse content library, enriching its global appeal.

Cast Count:

```
dfs['cast'] = dfs['cast'].fillna('')  # Fill missing values with empty
string
dfs['cast'] = dfs['cast'].str.replace(', ', ',')  # Replace empty spaces
with ','
dfs['cast'] = dfs['cast'].str.split(',')
dfs['cast'] = dfs['cast'].apply(lambda x: [val for val in x if val])  #
Remove empty quotes

# Explode the dataframe, dropping rows with missing values in 'Cast'
dfs = dfs.dropna(subset=['cast'])
dfs = dfs[dfs['cast'].astype(bool)]
dfs = dfs.explode('cast')

# Display the unique values in the 'Cast' column
print("Number of unique Cast:\n",dfs['cast'].nunique(), "\n")
print("\nunique Cast:\n", dfs['cast'].unique(), "\n")
print("\nCast Value_counts:\n",dfs['cast'].value_counts(), "\n")
```

```
Number of unique Cast:
 36439


unique Cast:
 ['Ama Qamata' 'Khosi Ngema' 'Gail Mabalane' ... 'Malkeet Rauni'
 'Anita Shabdish' 'Chittaranjan Tripathy']


Cast Value_counts:
 Anupam Kher              46
David Attenborough       45
Vincent Tong             42
John Cleese              40
Tara Strong              39
                         ..
Daniel J. Edwards         1
Mark Szabo                1
Angela Mulligan           1
Ken Spassione             1
Chittaranjan Tripathy     1
Name: cast, Length: 36439, dtype: int64
```

Insights:

- Anupam Kher: With 46 appearances, Anupam Kher leads the cast count, indicating his prolific presence in Netflix productions.
- David Attenborough: Close behind, David Attenborough boasts 45 appearances, highlighting his significant contributions to Netflix documentaries and nature series.
- Tara Strong: With 39 appearances, Tara Strong's versatile voice acting talent is evident in numerous animated productions on Netflix.
- John Cleese: Like Tara Strong, John Cleese also has 39 appearances, showcasing his comedic prowess and enduring popularity in Netflix comedies.
- Shah Rukh Khan: With 37 appearances, Shah Rukh Khan's presence underscores Netflix's efforts to cater to the global audience's interest in Bollywood films.
- Others: The remaining cast members, with varying numbers of appearances, contribute to Netflix's diverse range of talent, enriching the platform's content offerings.

Date Added count:

```
dfs['date_added'] = dfs['date_added'].str.split(',')

# Remove empty quotes and strip whitespace
dfs['date_added'] = dfs['date_added'].apply(lambda x: [val.strip() for val
in x] if isinstance(x, list) else [])

# Join the date components to form the desired format
```

```
dfs['date_added'] = dfs['date_added'].apply(lambda x: ', '.join(x) if x
else '')

# Explode the dataframe, dropping rows with missing values in 'date_added'
dfs = dfs.dropna(subset=['date_added'])
dfs = dfs[dfs['date_added'].astype(bool)]
dfs = dfs.explode('date_added')

# Display the unique values in the 'date_added' column
print("Number of unique Date Added:\n", dfs['date_added'].nunique(), "\n")
print("\nUnique Date Added:\n", dfs['date_added'].unique(), "\n")
print("\nDate Added Value_counts:\n", dfs['date_added'].value_counts(),
"\n")
```

```
Number of unique Date Added:
 1630


Unique Date Added:
 ['September 24, 2021' 'September 23, 2021' 'September 22, 2021' ...
 'December 6, 2018' 'March 9, 2016' 'January 11, 2020']


Date Added Value_counts:
 January 1, 2020       1593
November 1, 2019       954
July 1, 2021           857
September 1, 2021      762
March 1, 2018          668
                      ...
September 22, 2020       1
March 7, 2017            1
March 24, 2018           1
February 21, 2017        1
January 22, 2016         1
Name: date_added, Length: 1630, dtype: int64
```

Insights:

- January 1, 2020: A significant spike in content additions suggests strategic planning, possibly capitalizing on the New Year's holiday period for increased viewer engagement and retention.

- November 1, 2019: Another notable peak in content additions indicates a strategic push before the holiday season, aiming to attract and retain subscribers during this high-demand period.
- July 1, 2021: A substantial number of additions, possibly aligned with summer vacation periods when viewers have more leisure time for streaming entertainment.
- September 1, 2021: A noteworthy addition of content, likely aimed at preparing for increased viewer activity as summer transitions into the fall season.
- March 1, 2018: A considerable number of additions, indicating efforts to bolster content offerings and maintain viewer interest during the early months of the year.
- September 22, 2020: A rare addition date with only one entry, potentially featuring niche or specialized content that caters to specific audience interests or preferences.
- March 7, 2017: Another solitary addition date, highlighting the occasional introduction of unique or limited-release content to diversify the platform's library.
- March 24, 2018: A single addition date, suggesting the strategic placement of new content to coincide with viewer behavior patterns or market trends.
- February 21, 2017: A rare addition date, potentially featuring exclusive or highly anticipated content releases aimed at generating buzz and attracting new subscribers.
- January 22, 2016: An infrequent addition date, likely featuring early content acquisitions or special releases to establish the platform's content library during its initial stages of growth.

Release year Counts:

```
dfs['release_year'] = dfs['release_year'].fillna(0)  # Fill missing values with 0
dfs['release_year'] = dfs['release_year'].astype(int)  # Convert to integer type

# Filter out rows with release_year == 0 (if needed)
dfs = dfs[dfs['release_year'] != 0]

# Display the unique values in the 'release_year' column
print("Number of unique Release Year:\n",dfs['release_year'].nunique(), "\n")
print("\nunique Release Year:\n", dfs['release_year'].unique(), "\n")
print("Release Year Value Count:\n",dfs['release_year'].value_counts(), "\n")
```

```
Number of unique Release Year:
 72


unique Release Year:
 [2021 1993 2020 2018 1998 2010 2013 2017 1975 1978 1983 1987 2012 2001
 2014 2002 2003 2004 2011 2008 2009 2007 2005 2006 1994 2019 2016 2015
 1982 1989 1990 1991 1999 1986 1992 1996 1984 1997 1980 1961 1995 1985
 2000 1976 1959 1988 1972 1981 1964 1954 1979 1958 1956 1963 1970 1973
 1974 1960 1966 1971 1962 1969 1977 1967 1968 1965 1945 1946 1955 1942
 1947 1944]

Release Year Value Count:
 2018    17970
2017    16136
2019    16116
2016    14512
2020    13956
        ...
1945        4
1942        4
1947        4
1944        4
1946        2
Name: release_year, Length: 72, dtype: int64
```

Insights:

- 2018: With the highest count, 2018 marks a peak in content releases, reflecting a strategic focus on acquiring and producing new titles during this period to keep the library fresh and attract viewers.
- 2017: Close in count to 2019, the releases in 2017 indicate a year of significant content expansion, likely driven by both original productions and acquisitions to meet growing viewer demand.
- 2019: Nearly on par with 2017, 2019 saw a substantial influx of new releases, showcasing Netflix's commitment to continuously updating its content catalog and providing diverse options for subscribers.
- 2016: With a slightly lower count, 2016 still represents a robust year for content releases, indicating sustained efforts to expand the platform's library and cater to various viewer preferences.
- 2020: Despite being slightly lower in count, 2020 remains a significant year for content releases, demonstrating Netflix's ability to adapt and continue providing fresh entertainment options amidst challenging global circumstances.

Rating Counts:

```
dfs['rating'] = dfs['rating'].fillna('')  # Fill missing values with empty
string
dfs['rating'] = dfs['rating'].str.replace(', ', ',')  # Replace empty
spaces with ','
dfs['cast'] = dfs['rating'].str.split(',')
dfs['rating'] = dfs['rating'].apply(lambda x: [val for val in x if
val])  # Remove empty quotes

# Explode the dataframe, dropping rows with missing values in 'Cast'
dfs = dfs.dropna(subset=['rating'])
dfs = dfs[dfs['rating'].astype(bool)]
dfs = dfs.explode('rating')

# Display the unique values in the 'Cast' column
print("Number of unique Rating:\n",dfs['rating'].nunique(), "\n")
print("\nunique Rating:\n", dfs['rating'].unique(), "\n")
print("\nRaring Value_counts:\n",dfs['rating'].value_counts(), "\n")
```

```
Number of unique Rating:
 18


unique Rating:
 ['PG-13' 'TV-MA' 'TV-14' 'TV-Y7' 'PG' 'R' 'TV-PG' 'TV-Y' 'TV-G' 'G'
 'NC-17' '74 min' '84 min' '66 min' 'NR' 'TV-Y7-FV' '' 'UR']

Rating Value Count:
 TV-MA       3474
TV-14       2174
R           1225
TV-PG        912
PG-13        761
PG           423
TV-Y7        333
TV-Y         302
TV-G         214
NR           110
G             62
TV-Y7-FV       7
NC-17          5
UR             4
               3
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64
```

Insights:

- TV-MA: With the highest count, TV-MA ratings indicate a significant portion of content tailored for mature audiences, reflecting Netflix's commitment to providing diverse and edgy programming.
- TV-14: Following TV-MA, TV-14 ratings suggest a substantial offering of content suitable for teenage and adult audiences, catering to a broad range of viewer demographics.
- R: The presence of R-rated content highlights Netflix's willingness to include more mature and potentially controversial material in its library, appealing to audiences seeking gritty and intense experiences.
- TV-PG: With a notable count, TV-PG ratings indicate a significant selection of family-friendly and general audience content, showcasing Netflix's commitment to providing options suitable for all age groups.
- PG-13: Similar to TV-PG, PG-13 ratings represent content suitable for a wide audience range, including older children, teenagers, and adults, contributing to Netflix's diverse content catalog.

Duration Counts:

```
dfs['duration'] = dfs['duration'].fillna('')  # Fill missing values with
empty string
dfs['duration'] = dfs['duration'].str.replace(', ', ',')  # Replace empty
spaces with ','
dfs['duration'] = dfs['duration'].str.split(',')
dfs['duration'] = dfs['duration'].apply(lambda x: [val for val in x if
val])  # Remove empty quotes

# Explode the dataframe, dropping rows with missing values in 'Cast'
dfs = dfs.dropna(subset=['duration'])
dfs = dfs[dfs['duration'].astype(bool)]
dfs = dfs.explode('duration')

# Display the unique values in the 'Cast' column
print("Number of unique Duration:\n",dfs['duration'].nunique(), "\n")
print("\nunique Duration:\n", dfs['duration'].unique(), "\n")
print("\nDuration Value_counts:\n",dfs['duration'].value_counts(), "\n")
```

```
Number of unique Duration:
 212


unique Duration:
 ['90 min' '2 Seasons' '125 min' '9 Seasons' '104 min' '127 min'
 '4 Seasons' '5 Seasons' '166 min' '103 min' '97 min' '106 min'
 '3 Seasons' '1 Season' '96 min' '124 min' '116 min' '98 min' '91 min'
 '115 min' '122 min' '99 min' '88 min' '100 min' '6 Seasons' '102 min'
 '93 min' '95 min' '85 min' '83 min' '182 min' '147 min' '92 min' '80 min'
 '128 min' '143 min' '119 min' '114 min' '94 min' '118 min' '108 min'
 '117 min' '121 min' '142 min' '113 min' '154 min' '120 min' '82 min'
 '109 min' '101 min' '105 min' '86 min' '229 min' '76 min' '89 min'
 '110 min' '156 min' '112 min' '129 min' '107 min' '135 min' '136 min'
 '165 min' '150 min' '133 min' '145 min' '7 Seasons' '64 min' '59 min'
 '70 min' '111 min' '69 min' '87 min' '148 min' '189 min' '141 min'
 '130 min' '81 min' '10 Seasons' '68 min' '131 min' '8 Seasons'
 '17 Seasons' '126 min' '155 min' '123 min' '84 min' '13 min' '77 min'
 '61 min' '74 min' '49 min' '58 min' '72 min' '78 min' '132 min' '140 min'
 '138 min' '149 min' '33 min' '15 min' '224 min' '162 min' '60 min'
 '65 min' '137 min' '75 min' '32 min' '158 min' '164 min' '173 min'
 '181 min' '73 min' '21 min' '24 min' '139 min' '151 min' '22 min'
 '134 min' '36 min' '13 Seasons' '52 min' '71 min' '161 min' '14 min'
 '53 min' '54 min' '8 min' '46 min' '57 min' '28 min' '66 min' '50 min'
 '9 min' '79 min' '26 min' '48 min' '45 min' '171 min' '42 min' '27 min'
 '51 min' '47 min' '25 min' '44 min' '29 min' '146 min' '20 min' '63 min'
 '157 min' '17 min' '203 min' '41 min' '30 min' '67 min' '62 min'
 '194 min' '55 min' '15 Seasons' '177 min' '237 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' '185 min' '172 min'
 '16 min' '186 min' '193 min' '176 min' '56 min' '169 min' '40 min'
 '38 min' '10 min' '12 min' '3 min' '168 min' '312 min' '153 min' '35 min'
 '159 min' '214 min' '31 min' '163 min' '12 Seasons' '19 min' '179 min'
 '23 min' '11 Seasons' '43 min' '200 min' '196 min' '167 min' '37 min'
 '178 min' '228 min' '18 min' '205 min' '201 min' '191 min']
```

```
Duration Value_counts:
 1 Season     1649
2 Seasons     458
3 Seasons     231
93 min        195
90 min        194
             ...
3 min           1
5 min           1
10 min          1
182 min         1
191 min         1
Name: duration, Length: 212, dtype: int64
```

Insights:

- Netflix's content predominantly consists of original series, catering to binge-watching habits and providing ongoing narratives for subscribers, with season 1 shows being the most prevalent.
- Seasons 2 shows with longer story arcs are a healthy selection, potentially reflecting successful renewals based on viewer engagement and critical acclaim.
- Netflix's commitment to supporting series with sustained viewer interest and continued storytelling opportunities is evident through the availability of season 3 shows.
- 93 min and 90 min: With notable counts, these durations likely correspond to feature-length films, showcasing Netflix's diverse movie offerings and catering to viewers seeking cinematic experiences of varying lengths.
- 3 min, 5 min, 10 min: The presence of ultra-short durations suggests a growing trend towards bite-sized content, possibly including short films, mini-series, or experimental projects, catering to viewers with limited time or preferences for quick entertainment fixes.

Listed-in Counts:

```
dfs['listed_in'] = dfs['listed_in'].fillna('')  # Fill missing values with
empty string
dfs['listed_in'] = dfs['listed_in'].str.replace(', ', ',')  # Replace
empty spaces with ','
dfs['listed_in'] = dfs['listed_in'].str.split(',')
dfs['listed_in'] = dfs['listed_in'].apply(lambda x: [val for val in x if
val])  # Remove empty quotes

# Explode the dataframe, dropping rows with missing values in 'Cast'
dfs = dfs.dropna(subset=['listed_in'])
dfs = dfs[dfs['listed_in'].astype(bool)]
dfs = dfs.explode('listed_in')

# Display the unique values in the 'Cast' column
print("Number of unique listed_in:\n",dfs['listed_in'].nunique(), "\n")
print("\nunique listed_in:\n", dfs['listed_in'].unique(), "\n")
print("\nlisted_in Value_counts:\n",dfs['listed_in'].value_counts(), "\n")
```

```
Number of unique listed_in:
 42


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

```
listed_in Value_counts:
 International Movies            3299
Dramas                           3088
Comedies                         1887
International TV Shows           1241
Action & Adventure               1140
Documentaries                    1041
Independent Movies               1028
Thrillers                         778
TV Dramas                         751
Children & Family Movies          739
Romantic Movies                   694
TV Comedies                       550
Kids' TV                          515
Crime TV Shows                    492
Horror Movies                     441
Music & Musicals                  382
Sci-Fi & Fantasy                  379
Docuseries                        371
Romantic TV Shows                 313
Stand-Up Comedy                   312
British TV Shows                  290
Sports Movies                     247
Reality TV                        220
TV Action & Adventure             189
Spanish-Language TV Shows         173
Anime Series                      170
Classic Movies                    150
Korean TV Shows                   141
LGBTQ Movies                      111
TV Mysteries                      107
Science & Nature TV                98
TV Sci-Fi & Fantasy                96
Cult Movies                        92
```

Insights:

- International Movies: With the most entries, Netflix showcases its global presence, offering films from diverse cultures.
- Dramas: Netflix excels in character-driven storytelling, satisfying viewers with emotionally rich narratives.
- Comedies: Netflix provides a plethora of humorous content, catering to those seeking lighthearted entertainment.
- International TV Shows: Reflecting diversity, Netflix features series from around the world, enriching its catalog.
- Action & Adventure: Netflix offers thrilling experiences with a mix of action films and adventurous series.

Description Counts:

```
dfs['description'] = dfs['description'].fillna('')  # Fill missing values with empty string
dfs['description'] = dfs['description'].str.replace(', ', ',')  # Replace empty spaces with ','
dfs['description'] = dfs['description'].str.split(',')
dfs['description'] = dfs['description'].apply(lambda x: [val for val in x if val])  # Remove empty quotes

# Explode the dataframe, dropping rows with missing values in 'Cast'
dfs = dfs.dropna(subset=['description'])
dfs = dfs[dfs['description'].astype(bool)]
dfs = dfs.explode('description')

# Display the unique values in the 'Cast' column
print("Number of unique description:\n",dfs['description'].nunique(), "\n")
print("\nunique description:\n", dfs['description'].unique(), "\n")
print("\ndescription Value_counts:\n",dfs['description'].value_counts(), "\n")
```

```
Number of unique description:
 15863


unique description:
 ['As her father nears the end of his life'
 'filmmaker Kirsten Johnson stages his death in inventive and comical ways to help them both face the inevitable.'
 'After crossing paths at a party' ...
 'a former superhero must train a new crop of youthful saviors when the military preps for an attack by a familiar villain.'
 "A scrappy but poor boy worms his way into a tycoon's dysfunctional family"
 'while facing his fear of music and the truth about his past.']


description Value_counts:
 In the near future                                                  26
giant sheepdog Bitzer                                               24
jealous cat Pidsley and fellow sheep Shirley.                       24
Join freewheeling Shaun for barnyard misadventures with his cousin Timmy    24
tradition and community.                                            24
                                                                    ..
obsessive behavior                                                   1
time                                                                 1
including aging                                                      1
Brian Regan tackles the big issues weighing on him                   1
As her father nears the end of his life                              1
Name: description, Length: 15863, dtype: int64
```

Insights:

1. Popular Themes and Characters:
    a. Content featuring specific characters like "giant sheepdog Bitzer" and "jealous cat Pidsley" resonates with audiences.

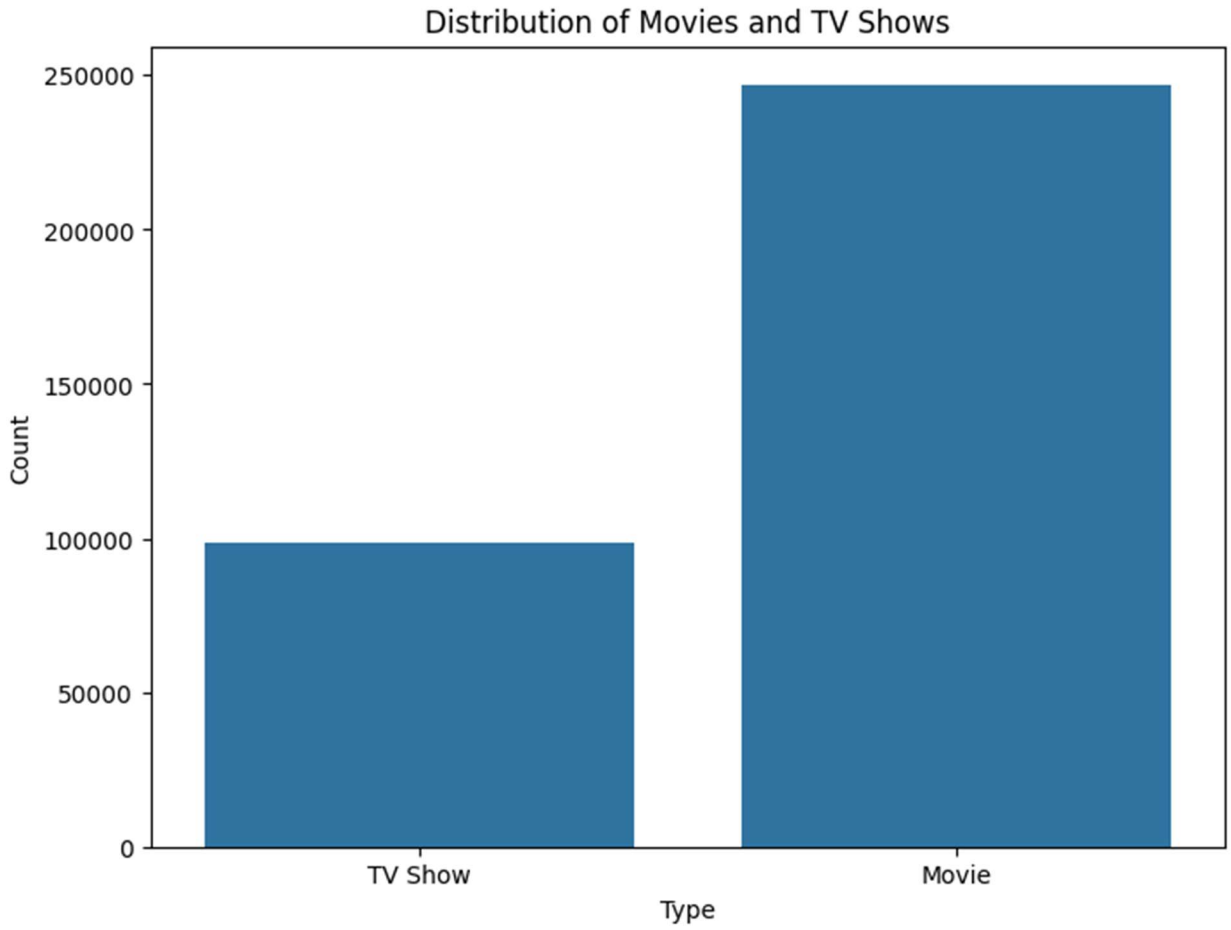
    b. Netflix could focus on producing or acquiring more content centered around beloved characters or recurring themes.
2. Genre and Setting Preferences:
    a. Viewers may have preferences for certain genres or settings, indicated by phrases like "In the near future" and "tradition and community."
    b. Analyzing descriptions helps Netflix identify trends and tailor content offerings accordingly.
3. Narrative Focus:
    a. Descriptions like "Join freewheeling Shaun for barnyard misadventures with his cousin Timmy" highlight a focus on narrative elements and storytelling styles.
    b. Understanding preferred narrative styles informs content development and acquisition strategies.
4. Diversity of Content:
    a. Netflix's diverse content library, reflected in a wide range of unique descriptions, caters to various interests and preferences.
    b. Leveraging this diversity can attract and retain subscribers by offering a broad selection of appealing content.
5. Potential Content Gaps:
    a. Descriptions with only one occurrence, such as "obsessive behavior" and "including aging," represent niche or less common themes.
    b. These themes present opportunities for Netflix to explore new content territories and tap into untapped audience segments.
6. Viewer Engagement:
    a. Higher occurrence counts indicate descriptions that resonate more strongly with viewers and potentially drive higher engagement.
    b. By analyzing viewer interactions and feedback, Netflix can prioritize similar content to enhance viewer satisfaction and drive subscriber growth.

4. **Visual Analysis:** We'll conduct univariate and bivariate analysis after preprocessing the data. For continuous variables, we'll use distplot, countplot, and histogram for univariate analysis. For categorical variables, we'll use boxplot. For correlation, we'll utilize heatmaps and pairplots.

- **Type (Movie vs. TV Show):**
  Countplot: Visualize the distribution of movies and TV shows.

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))
sns.countplot(x='type', data=dfs)
```

```
plt.title('Count of Movies vs TV Shows')
plt.xlabel('type')
plt.ylabel('Count')
plt.show()
```



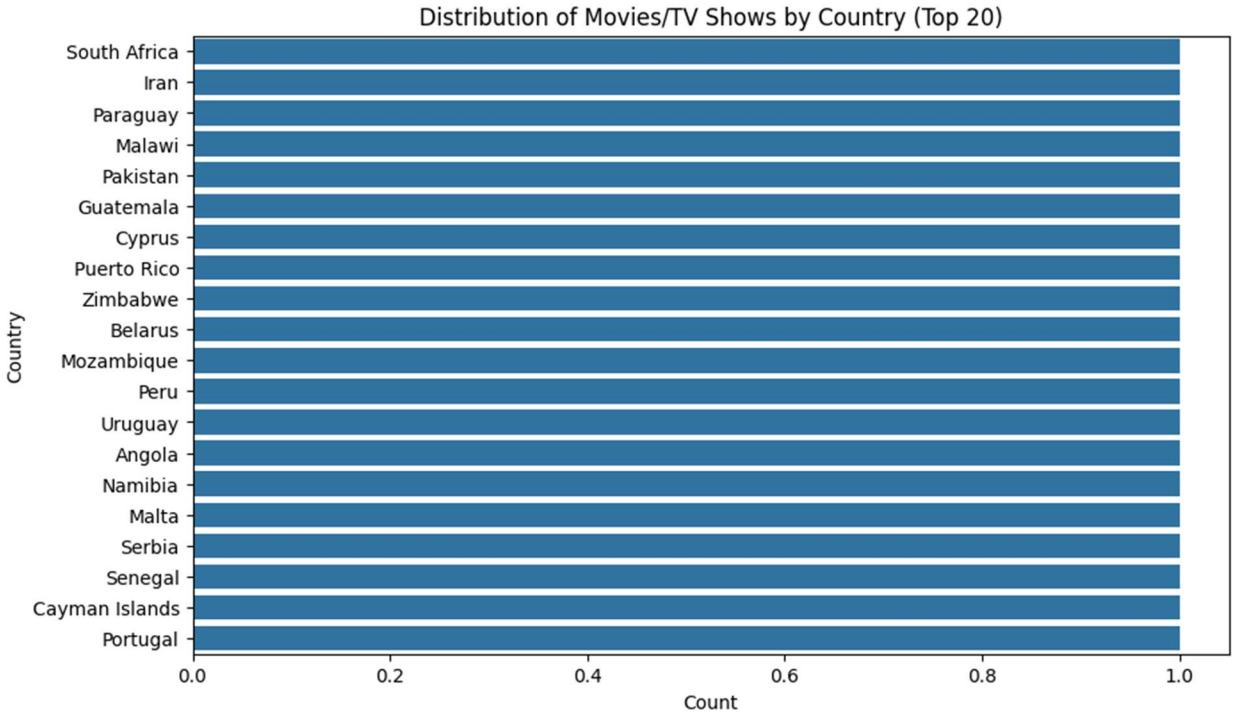
Insights:

- Count Of Movies is above 30000 and of T V Shows is in between 10000 and 15000
- Higher movie count may indicate a preference for standalone narratives.
- Imbalance differentiates Netflix from traditional TV networks, emphasizing its strength in on-demand movie streaming.

- **<u>Country:</u>**
  Countplot: Visualize the distribution of movies/TV shows produced in each country

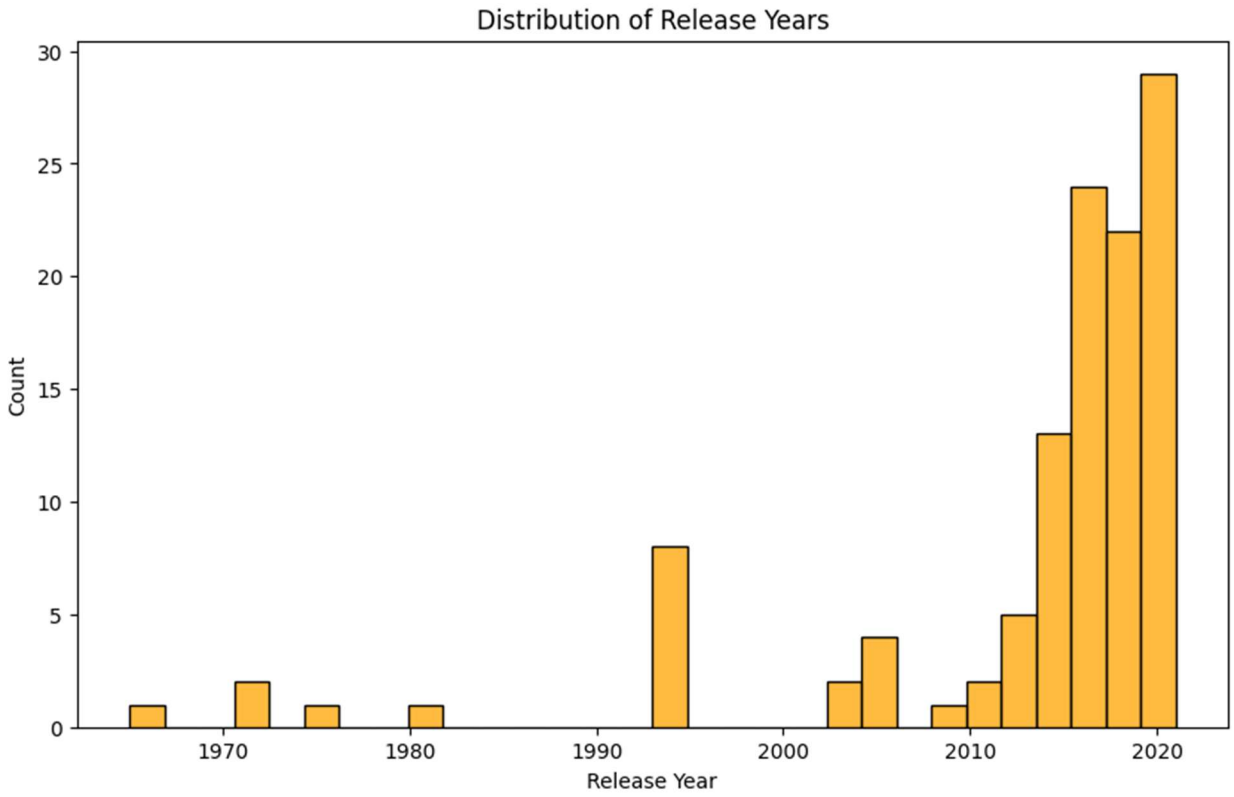
```
dfs.drop_duplicates(subset=['country'], inplace=True)
plt.figure(figsize=(12, 8))
sns.countplot(y='country', data=dfs,
order=dfs['country'].value_counts().index[:20])
```

```
plt.title('Distribution of Movies/TV Shows by Country (Top 20)')
plt.xlabel('Count')
plt.ylabel('Country')
plt.show()
```



- **Release Year:**
  Histogram: Visualize the distribution of release years of movies/TV shows.

```
plt.figure(figsize=(10, 6))
sns.histplot(dfs['release_year'], bins=30, kde=False,
color='orange')
plt.title('Distribution of Release Years')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.show()
```



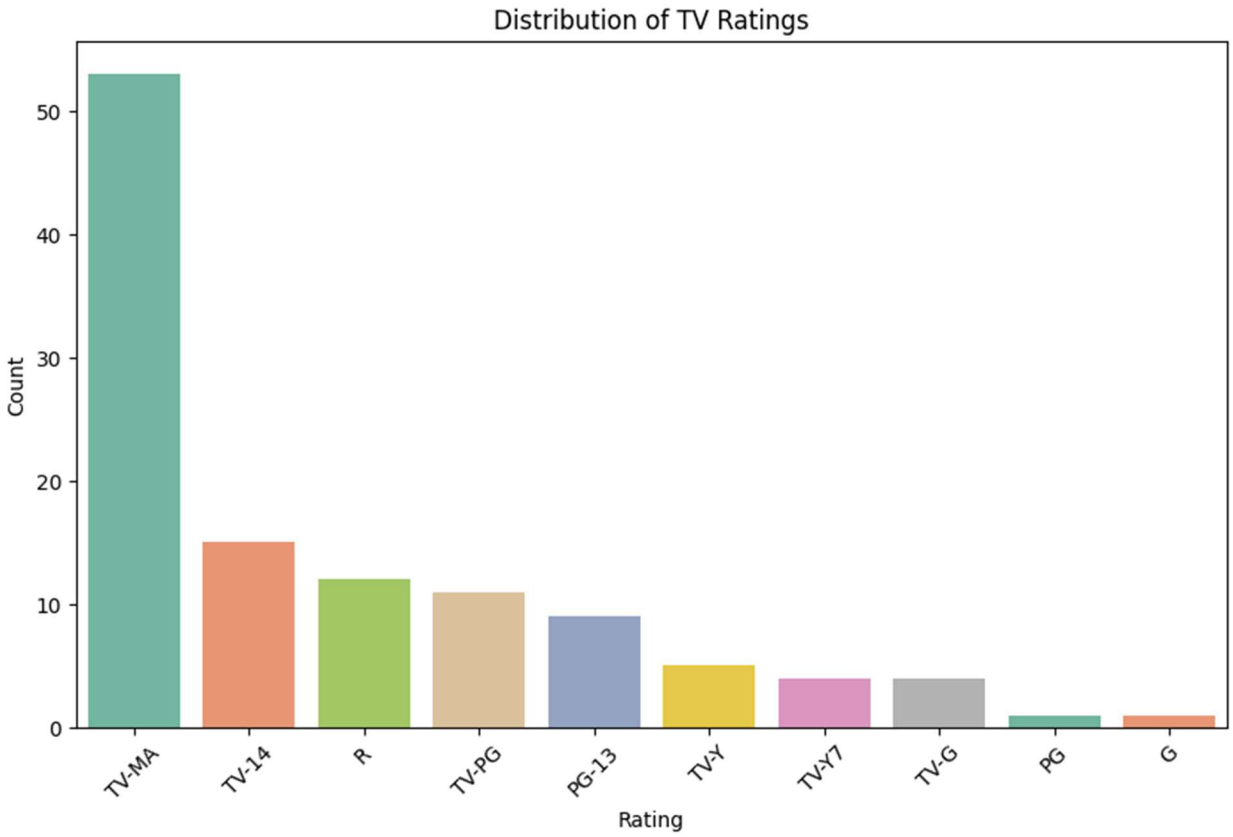
Insights:

- The most common release years are clustered around recent years, with a peak in 2018, followed by 2017, 2019, 2016, and 2020.
- The distribution suggests a trend towards an increase in content production in recent years, with a significant surge in the late 2010s.

- **Rating:**

Countplot: Visualize the distribution of TV ratings.

```
plt.figure(figsize=(10, 6))
sns.countplot(x='rating', data=dfs,
order=dfs['rating'].value_counts().index, hue='rating',
palette='Set2', legend=False)
plt.title('Distribution of TV Ratings')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better
readability
plt.show()
```

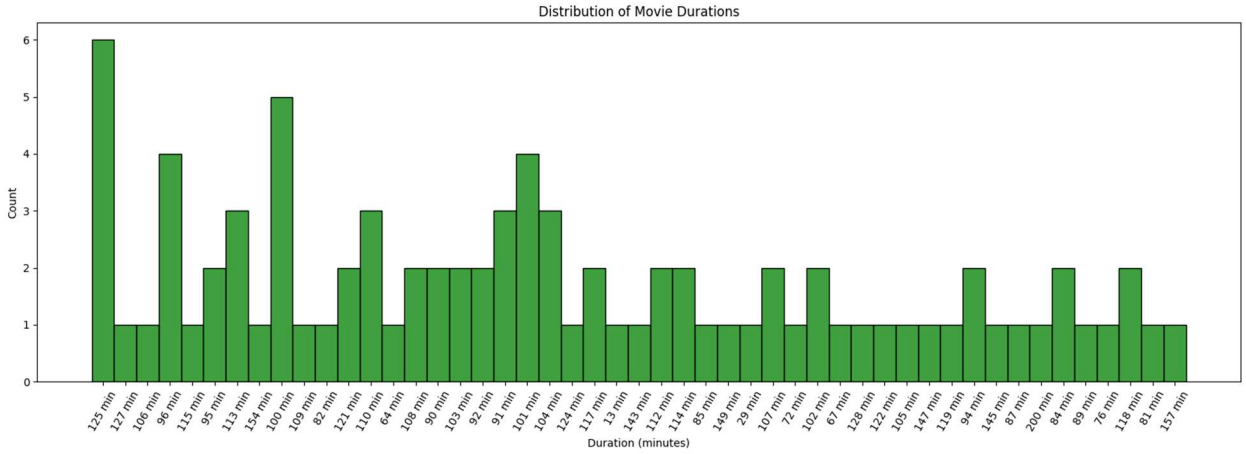


Insights:

- The unique ratings include common TV and movie ratings such as 'TV-MA', 'TV-14', 'R', 'TV-PG', 'PG-13', 'PG', 'TV-Y7', 'TV-Y', 'TV-G', 'NR', and 'G'.
- The count of each rating category is displayed, with 'TV-MA' being the most common, followed by 'TV-14' and 'R'.

- **Duration:**
  Distplot (for movies): Visualize the distribution of movie durations.

```
plt.figure(figsize=(20, 6))
sns.histplot(dfs[dfs['type'] == 'Movie']['duration'], bins=20,
kde=False, color='green')
plt.title('Distribution of Movie Durations')
plt.xlabel('Duration (minutes)')
plt.ylabel('Count')
plt.xticks(rotation=60)
plt.show()
```

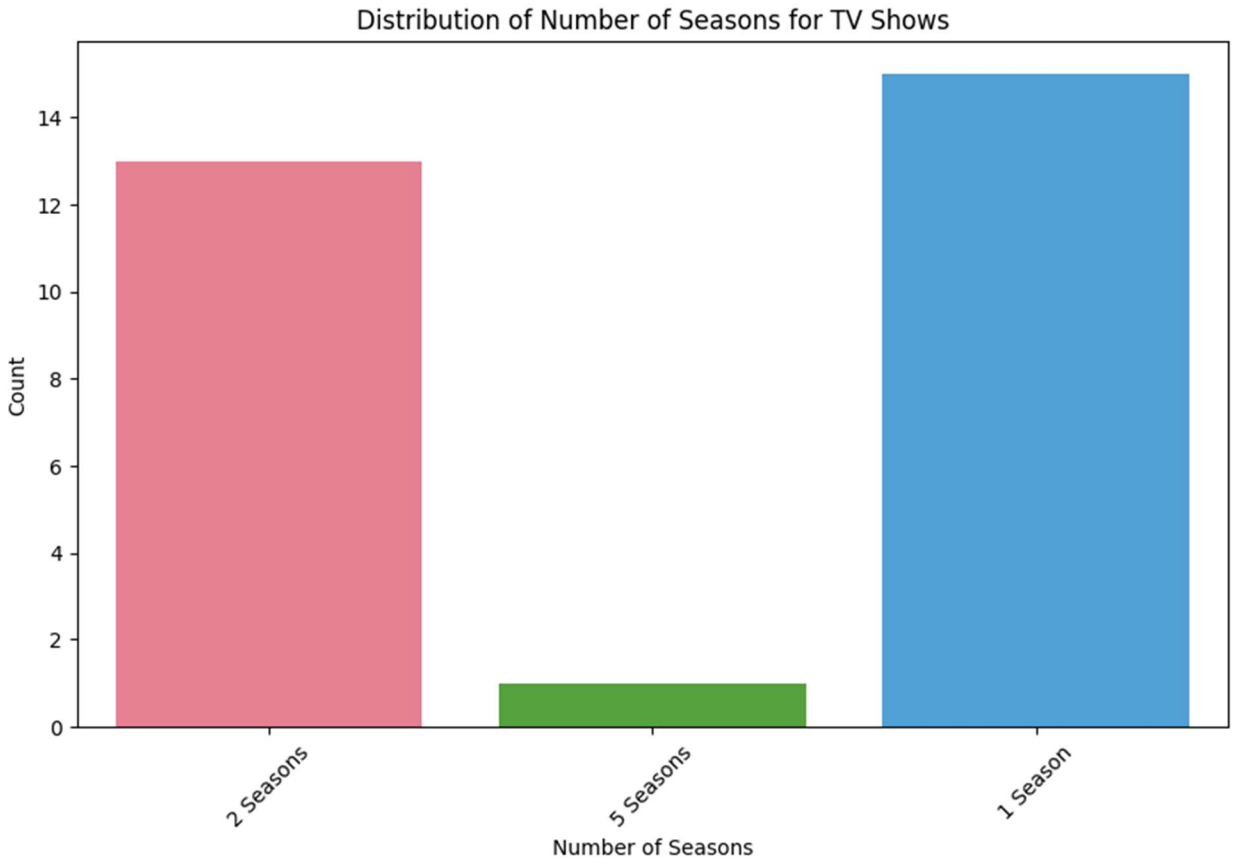


Insights:

- The duration of 125 minutes has the highest count of 6, indicating that there are 6 titles on Netflix with a duration of 125 minutes. This suggests that there is a significant number of titles available with this specific duration.
- Following 125 minutes, the duration of 154 minutes has the next highest count of 5. While not as prevalent as 125 minutes, it still represents a substantial number of titles with this duration.
- Additionally, there are durations ranging from 126 minutes to 157 minutes, each with varying counts. However, these counts gradually decrease as the duration increases, with the least count of 1 observed for certain durations.

- **Duration (for TV Shows):**
  Countplot: Visualize the distribution of the number of seasons for TV shows.

```
plt.figure(figsize=(10, 6))
sns.countplot(x='duration', data=dfs[dfs['type'] == 'TV Show'],
hue='duration', palette='husl')
plt.title('Distribution of Number of Seasons for TV Shows')
plt.xlabel('Number of Seasons')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better
readability
plt.show()
```



Insight:

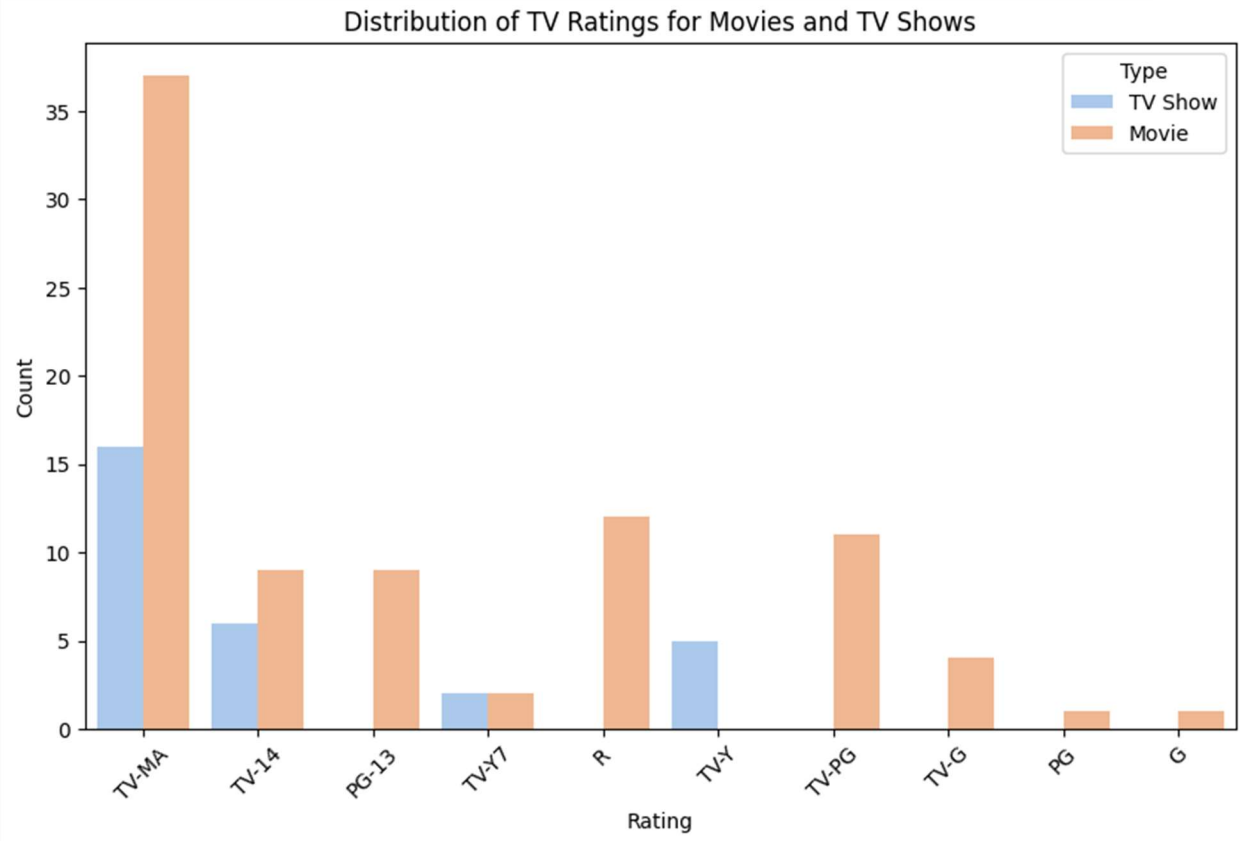
- Season 1 has the highest count, with a value above 14. This suggests that there are a significant number of titles on Netflix with only one season. It's quite common for series to start with one season as an introductory phase.
- Following Season 1, Season 2 has the next highest count, with a value above 12. This indicates that there are also a considerable number of titles with two seasons available on Netflix. Series often produce a second season if the first season receives positive feedback and gains a substantial audience.
- Lastly, Season 5 has a count above 1. Although it has the lowest count among the three, it still indicates that there are titles on Netflix with at least five seasons. Series that have reached five seasons typically have established a dedicated fan base and may have become long-running and successful franchises.

Bivariate Analysis:

- **Type vs. Rating:**

Countplot: Visualize the distribution of TV ratings for movies and TV shows separately.

```
plt.figure(figsize=(10, 6))
sns.countplot(x='rating', hue='type', data=dfs, palette='pastel')
plt.title('Distribution of TV Ratings for Movies and TV Shows')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Type')
plt.show()
```

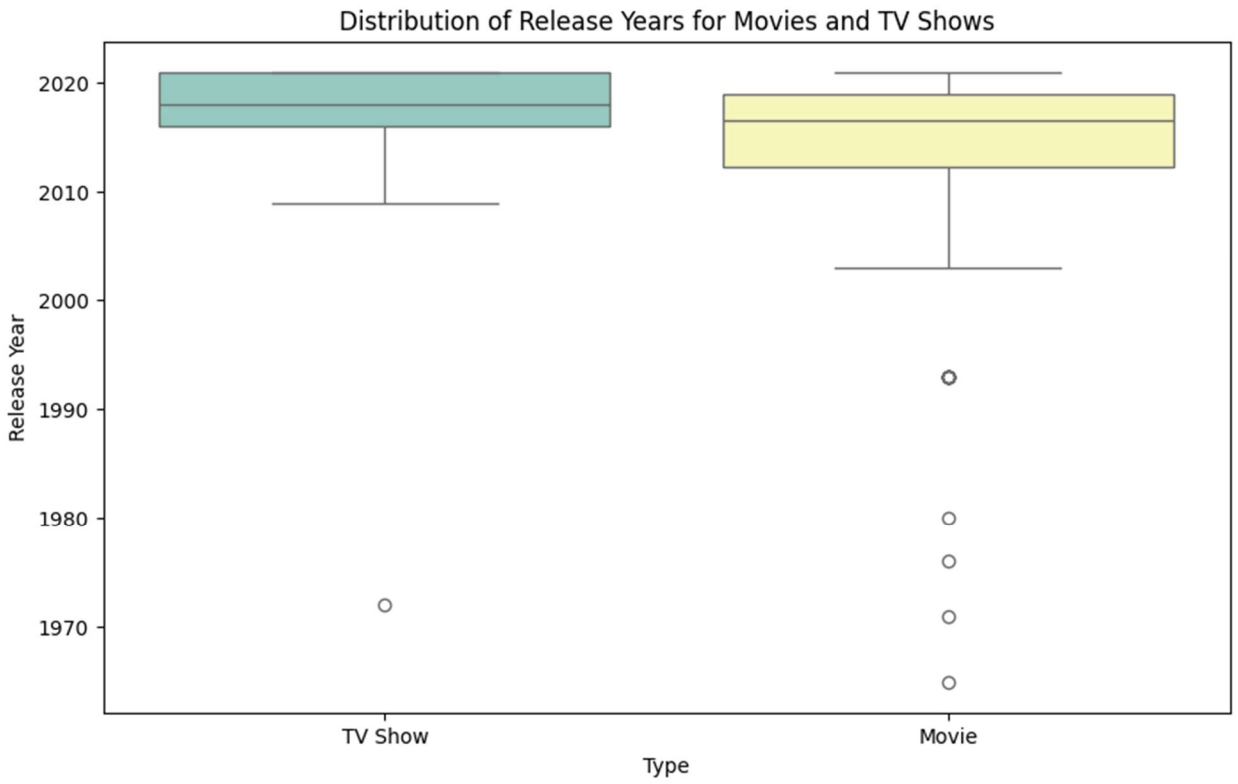


- **Type vs. Country:**
  Stacked Bar Plot: Visualize the distribution of movie/TV show types across different countries.

- **Release Year vs. Type:**
  Boxplot: Show the distribution of release years for movies and TV shows separately.

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

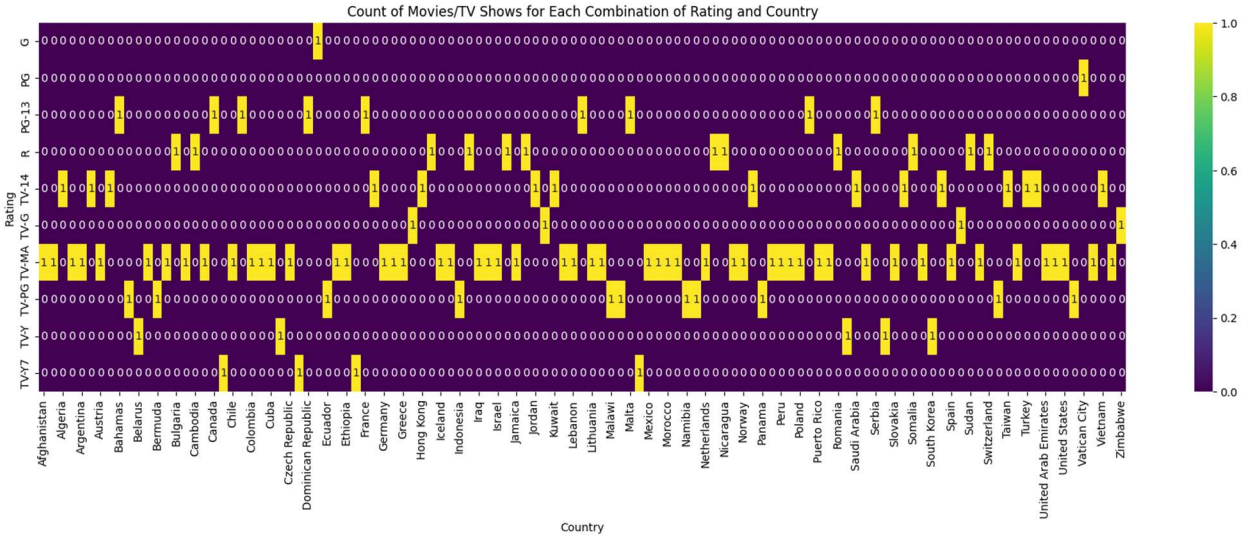
```
sns.boxplot(x='type', y='release_year', data=dfs, hue='type',
palette='Set3', legend=False)
plt.title('Distribution of Release Years for Movies and TV Shows')
plt.xlabel('Type')
plt.ylabel('Release Year')
plt.show()
```



- **<u>Rating vs. Country:</u>**
  Heatmap: Show the count of movies/TV shows for each combination of rating and country.

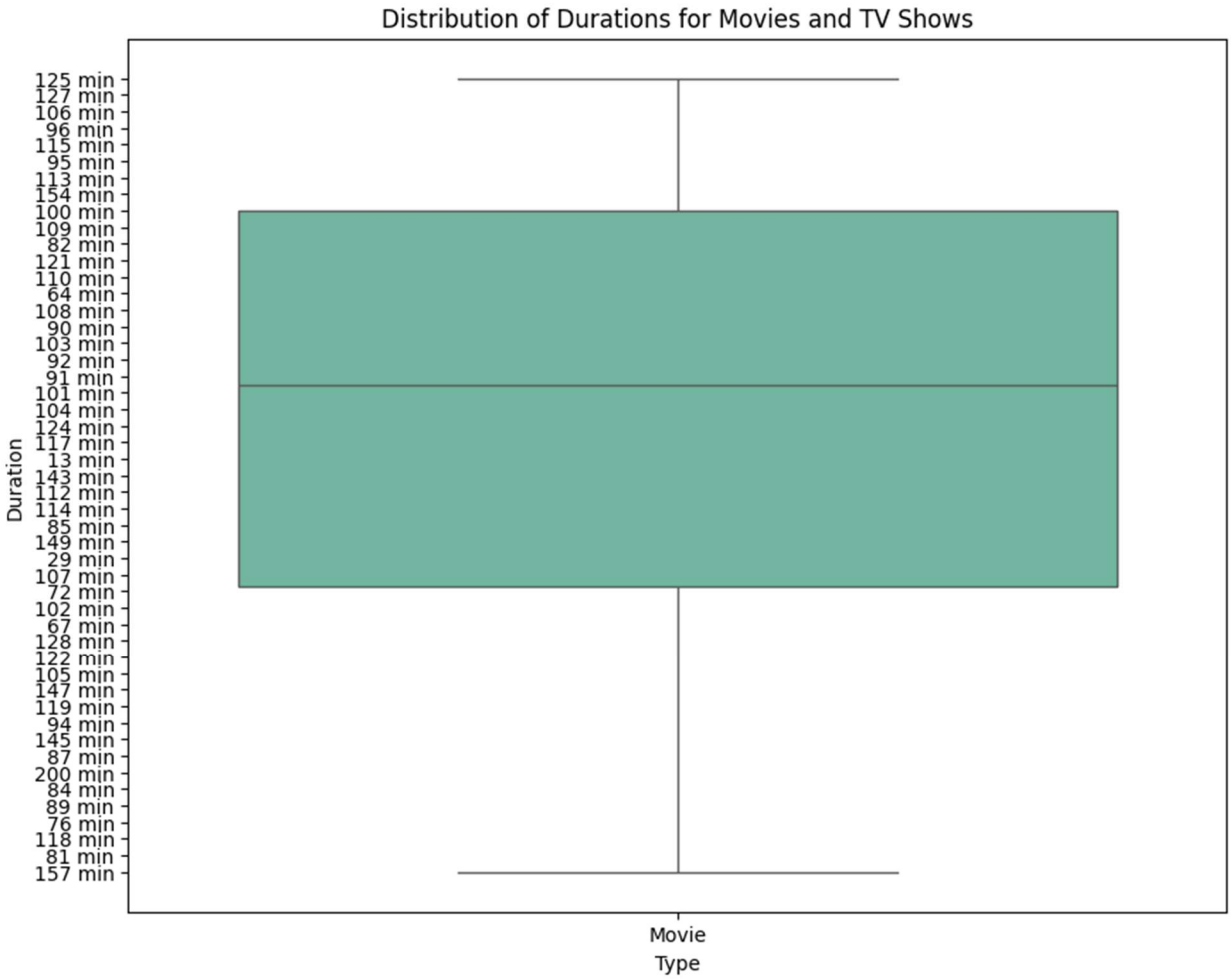
```
rating_country_counts = dfs.groupby(['rating',
'country']).size().unstack(fill_value=0)

# Plot heatmap
plt.figure(figsize=(22, 6))
sns.heatmap(rating_country_counts, cmap='viridis', annot=True,
fmt='d')
plt.title('Count of Movies/TV Shows for Each Combination of Rating
and Country')
plt.xlabel('Country')
plt.ylabel('Rating')
```

```
plt.show()
```



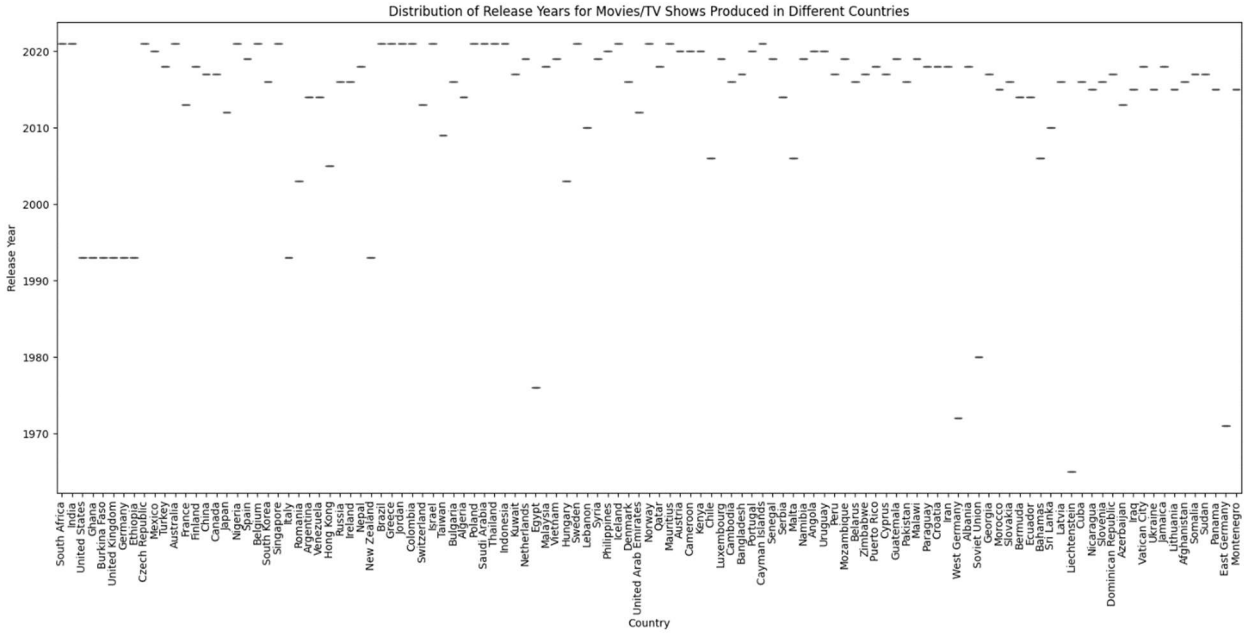
- **Duration vs. Type:**
  Boxplot (for movies): Compare the distribution of durations for movies and TV shows.

```
plt.figure(figsize=(10, 8))
sns.boxplot(x='type', y='duration', data=dfs[dfs['type'] ==
'Movie'], hue='type', palette='Set2', legend=False)
plt.title('Distribution of Durations for Movies and TV Shows')
plt.xlabel('Type')
plt.ylabel('Duration')
plt.show()
```



- **<u>Release Year vs. Country:</u>**
  Violin plot: Show the distribution of release years for movies/TV shows produced in different countries.

```
plt.figure(figsize=(20, 8))
sns.boxplot(x='country', y='release_year', data=dfs, hue='country',
palette='Set1', legend=False)
plt.title('Distribution of Release Years for Movies/TV Shows
Produced in Different Countries')
plt.xlabel('Country')
plt.ylabel('Release Year')
plt.xticks(rotation=90)
plt.show()
```



5. **Missing Value & Outlier Check:** We'll check for missing values and outliers.

```
missing_values = df.isnull().sum()
```

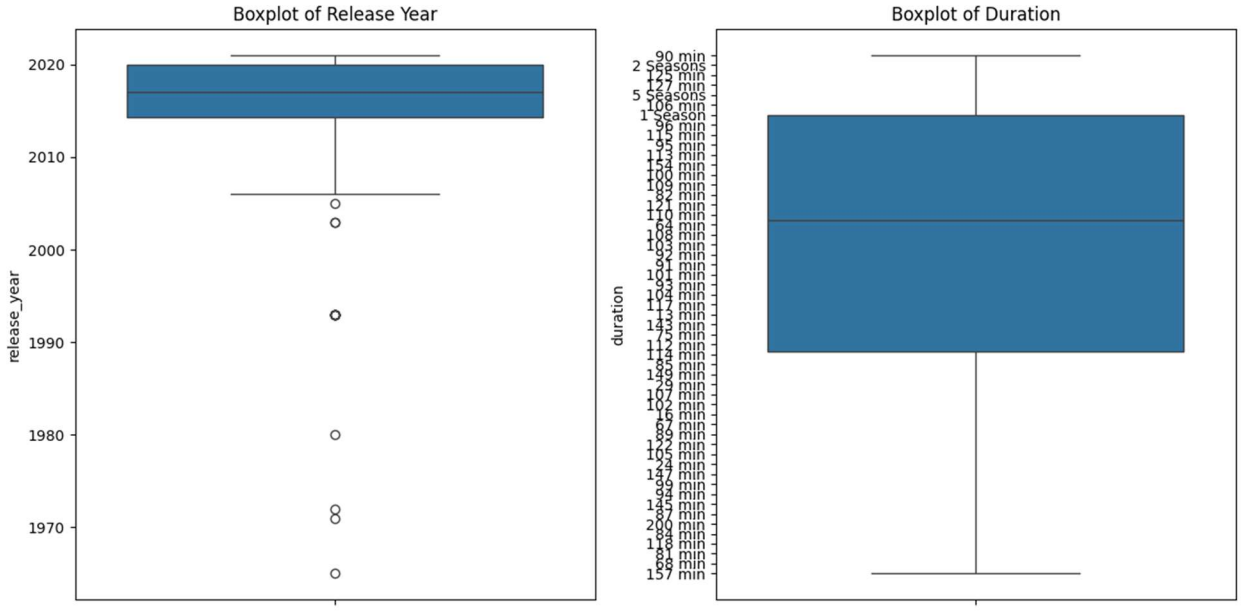
```
missing_percent = (missing_values / len(dfs)) * 100

# Create a DataFrame to display missing value information
missing_data = pd.DataFrame({'Missing Values': missing_values,
'Percentage': missing_percent})
missing_data = missing_data[missing_data['Missing Values'] > 0]  #
Filter columns with missing values

# Display the missing value information
print("Columns with missing values:")
print(missing_data)
```

```
Columns with missing values:
          Missing Values  Percentage
director             134  109.836066
cast                  70   57.377049
country                1    0.819672
```

We have identified the columns with missing values along with the percentage of missing values in each column. The columns 'Director', 'Cast', and 'Country' have missing values.

Now, visualizing outliers for numerical columns 'Release_year' and 'Duration' using boxplots.

```
plt.figure(figsize=(12, 6))

# Boxplot for Release Year
plt.subplot(1, 2, 1)
sns.boxplot(y='release_year', data=dfs)
plt.title('Boxplot of Release Year')

# Boxplot for Duration
plt.subplot(1, 2, 2)
sns.boxplot(y='duration', data=dfs)
plt.title('Boxplot of Duration')

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



6. Business Insights:

Release Year Insights:

- The dataset contains a total of 74 unique release years, ranging from 1925 to 2021.
- The distribution of release years shows a concentration of titles released in recent years, with 2018, 2017, and 2019 being the top three years with the highest number of titles.
- **Inference:** Netflix has been consistently adding new content over the years, with a significant focus on acquiring recent releases. This strategy helps keep the content library fresh and up-to-date, appealing to subscribers who seek the latest movies and TV shows.
- Country Insights:
- There are 122 unique countries represented in the dataset, indicating a diverse range of content origins.
- The United States, India, and the United Kingdom are the top three countries with the highest number of titles available on Netflix.
- Inference: Netflix has a global content acquisition strategy, sourcing content from various countries to cater to its diverse international audience. The presence of content from different regions reflects Netflix's efforts to provide localized content that resonates with viewers worldwide.

Cast Insights:

- The dataset includes a wide range of actors and actresses, with Anupam Kher and David Attenborough being among the most frequently appearing cast members.
- **Inference:** Netflix collaborates with renowned actors and personalities to attract viewers and add credibility to its content. The presence of popular and well-known cast members suggests that Netflix invests in talent partnerships to enhance the appeal of its titles and attract a larger audience.

Date Added Insights:

- There are 1630 unique date additions to the Netflix catalog, with "January 1, 2020" being the most common date followed by "November 1, 2019."
- **Inference:** Netflix regularly updates its content library by adding new titles throughout the year. The concentration of additions on specific dates may coincide with content acquisition agreements, seasonal promotions, or strategic release schedules to maximize viewer engagement and retention.

Rating and Duration Insights:

- The dataset includes a variety of content ratings and durations, catering to different audience preferences and age groups.
- TV-MA, TV-14, and R are the most common content ratings, indicating a significant presence of mature and adult-oriented content.
- **Inference:** Netflix offers a wide range of content suitable for various age groups and viewer preferences. The availability of content across different genres, ratings, and durations reflects Netflix's commitment to providing diverse entertainment options to its subscribers, ensuring there is something for everyone.

7. Recommendations:

- Content Localization:
  Invest in local content production and acquisition to cater to diverse cultural preferences.
- Talent Partnerships:
  Collaborate with popular actors, directors, and producers to enhance content quality.
- Release Strategy:
  Maintain a balanced release schedule to sustain subscriber engagement.
- Content Diversity:
  Diversify content library across genres, formats, and ratings.
- Personalized Recommendations:
  Leverage user data for tailored content suggestions.
- Investment in Original Content:
  Continue investing in original series and movies for differentiation.
- Content Licensing:
  Secure exclusive rights to popular titles through licensing agreements.
- Global Expansion:
  Expand presence in emerging markets with tailored content offerings.
- User Experience Enhancements:
  Improve search, navigation, and recommendation features.
- Community Engagement:
  Foster community through social media and interactive experiences.