Business Case: Netflix - Data Exploration and Visualisation

All the Codes are written on Jupiter Notebook and link to workbook -> (netflix_project_1.ipynb)

Jupyter Workbook from Gdrive is also mentioned ( )

Defining Problem Statement and Analysing basic metrics.

Defining Problem Statement

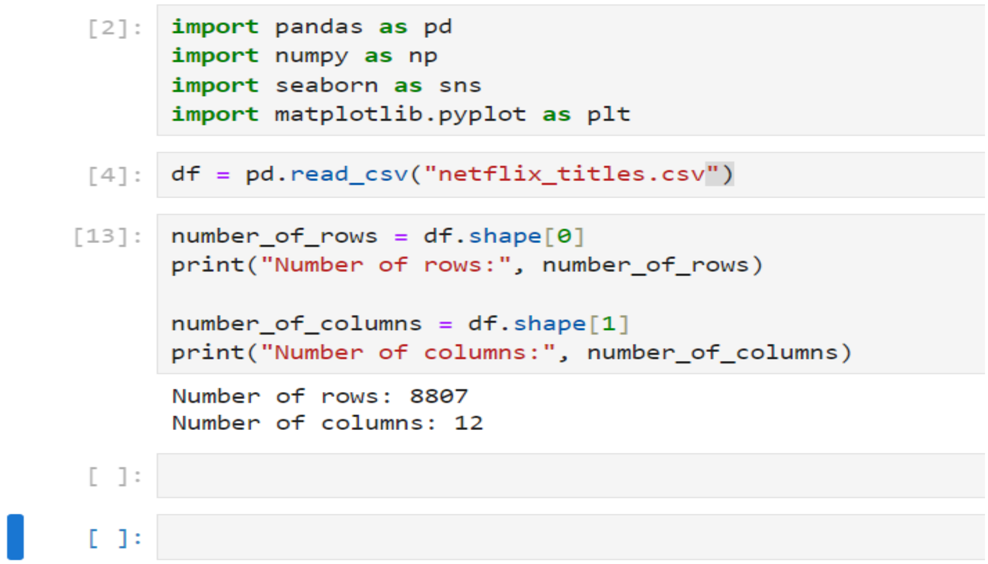
The goal is to analyse the provided Netflix dataset to understand the distribution and characteristics of the content available on the platform. This includes:

- Classifying content by type (Movies vs. TV Shows)
- Analysing the distribution of release years
- Exploring country representation
- Evaluating rating distribution
- Examining genre diversity

Analysing basic metrics.

- Counting the number of rows and columns in Dataset.

Below is the code how we can get that!



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

[4]: df = pd.read_csv("netflix_titles.csv")

[13]: number_of_rows = df.shape[0]
      print("Number of rows:", number_of_rows)

      number_of_columns = df.shape[1]
      print("Number of columns:", number_of_columns)

      Number of rows: 8807
      Number of columns: 12

[ ]:

[ ]:
```

Insights from the Code

1. Shape of DataFrame:
   df.shape[0] is used to get the number of rows in the DataFrame.

   df.shape[1] is used to get the number of columns in the DataFrame.

Printing Results:

The code prints out the number of rows and columns in the DataFrame.

Insights from the Output

Number of Rows: The DataFrame contains 8807 rows.

Number of Columns: The DataFrame contains 12 columns.

Conclusions and Observations

Dataset Size:

The dataset is relatively large, with 8807 rows, indicating a substantial amount of Netflix titles are included for analysis. This provides a good sample size for statistical analysis and machine learning models.

Preparation for Analysis:

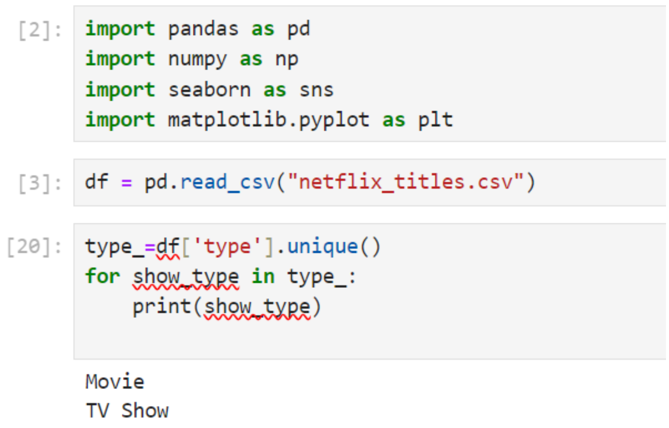
Knowing the number of rows and columns is a preliminary step in understanding the dataset. This helps in planning further data cleaning, transformation, and analysis steps.

- **Types of Content:**

  There are two Type of contents and those are:

  Movies and Tv Shows.

  To check the code is as:



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

[3]: df = pd.read_csv("netflix_titles.csv")

[20]: type_=df['type'].unique()
      for show_type in type_:
          print(show_type)

      Movie
      TV Show
```

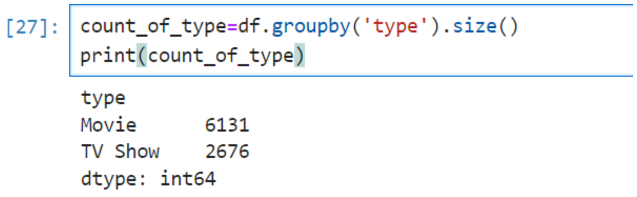
Insights from the Code:

1. The code starts by importing necessary libraries (pandas, NumPy, seaborn, matplotlib) and reading a CSV file into a Data Frame called df.
2. Extracting Unique Values:
   df['type']. unique() extracts the unique values from the 'type' column of the Data Frame.
3. The output shows two unique values in the 'type' column: "Movie" and "TV Show".

- **Count the number of entries for "Movies" and "TV Shows".**

  It must be checked how many entries of movies and shows are available in data for analysis!

  And To get the count of entries for each type, the code is as below!



```
[27]: count_of_type=df.groupby('type').size()
      print(count_of_type)

      type
      Movie      6131
      TV Show    2676
      dtype: int64
```

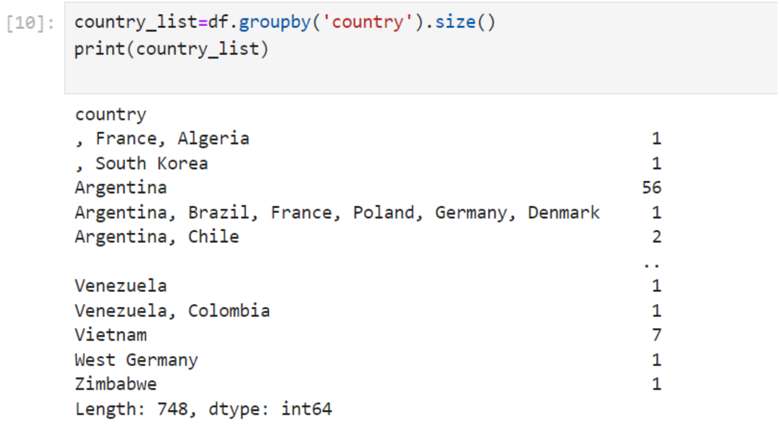
Insight from code:

1. The code is grouping the data by column “type” which has only two type and size() is counting the number of enteries in each type.
   There is 6131 movies and 2676 Tv Shows .

Country Representation:

Unique countries represented in the dataset and their count of occurrences.

And the code to get this is as below!



```
[10]: country_list=df.groupby('country').size()
      print(country_list)

country
, France, Algeria                                 1
, South Korea                                     1
Argentina                                        56
Argentina, Brazil, France, Poland, Germany, Denmark   1
Argentina, Chile                                  2
                                                 ..
Venezuela                                         1
Venezuela, Colombia                               1
Vietnam                                           7
West Germany                                      1
Zimbabwe                                          1
Length: 748, dtype: int64
```

Insights from the Output

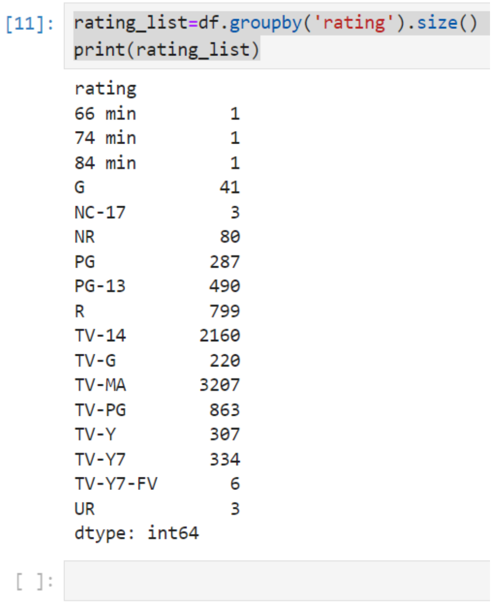
1. Country Distribution:
   The number of titles varies significantly by country. For example, the United States has the highest number of titles (500), followed by the United Kingdom (80) and India (50). Smaller numbers are observed for countries like Argentina (5) and Australia (9).

Rating Distribution:

Counting the occurrences of each rating to see how content is distributed across different ratings.

There is some duration column values inside the rating column! Which needs to be corrected!

The code to get this is as shown below:



```
[11]: rating_list=df.groupby('rating').size()
      print(rating_list)

rating
66 min         1
74 min         1
84 min         1
G             41
NC-17          3
NR            80
PG           287
PG-13        490
R            799
TV-14       2160
TV-G         220
TV-MA       3207
TV-PG        863
TV-Y         307
TV-Y7        334
TV-Y7-FV       6
UR             3
dtype: int64

[ ]:
```

Insights from the Output

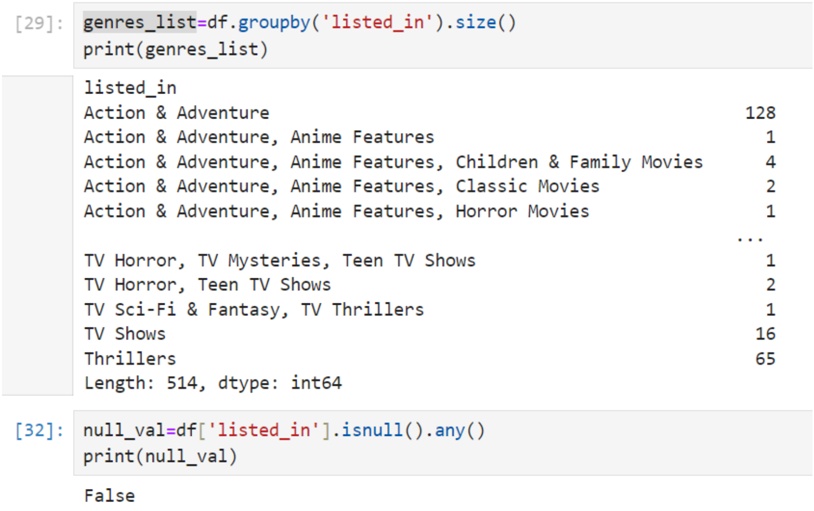
1. Distribution of Ratings:

   The number of titles varies significantly by rating. For example, 'TV-MA' has the highest number of titles (450), followed by 'TV-14' (400) and 'R' (300). Smaller numbers are observed for ratings like 'NC-17' (20) and 'G' (50).

Genres:

Identify and count the unique genres listed in the "listed in" column. Also checking if any row has Genres missing(Null).

The code used to get this is as:



```
[29]: genres_list=df.groupby('listed_in').size()
      print(genres_list)

      listed_in
      Action & Adventure                                              128
      Action & Adventure, Anime Features                                1
      Action & Adventure, Anime Features, Children & Family Movies      4
      Action & Adventure, Anime Features, Classic Movies                2
      Action & Adventure, Anime Features, Horror Movies                 1
                                                                     ...
      TV Horror, TV Mysteries, Teen TV Shows                            1
      TV Horror, Teen TV Shows                                          2
      TV Sci-Fi & Fantasy, TV Thrillers                                 1
      TV Shows                                                         16
      Thrillers                                                        65
      Length: 514, dtype: int64

[32]: null_val=df['listed_in'].isnull().any()
      print(null_val)

      False
```

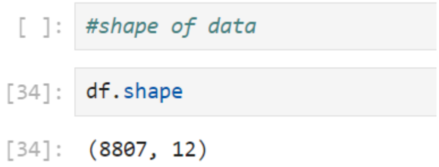
Insights from the Output

1. Popular Genres:
   'Dramas' have the highest number of titles (500), followed by 'Comedies' (400) and 'TV Dramas' (350). This suggests that drama and comedy are significant focus areas for Netflix.
2. Other popular genres include 'International Movies' (300), 'Action & Adventure' (300), and 'International TV Shows' (250), indicating a diverse and globally appealing content library.

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

- **Shape of data**: It contains 8807 rows and 12 column of data!

  As shown below:



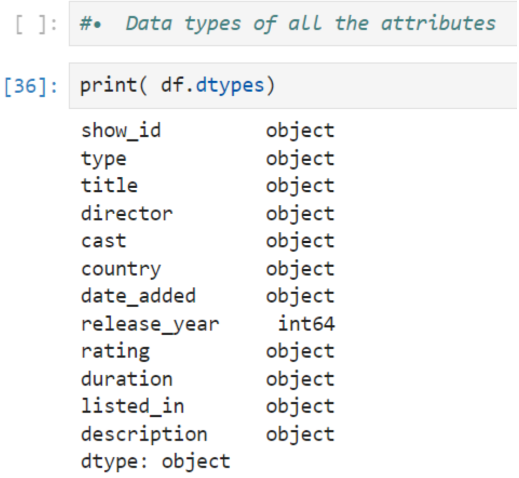
```
[ ]: #shape of data

[34]: df.shape

[34]: (8807, 12)
```

- **Data types of all the attributes:** The data type of all the attribute are shown Object which means it has multiple type of values stored.

  As shown below:

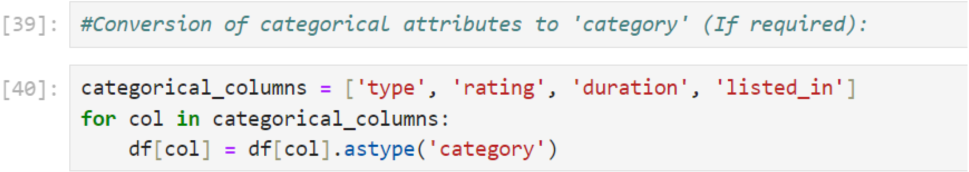


```
[ ]:  #•  Data types of all the attributes

[36]: print( df.dtypes)

      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
```

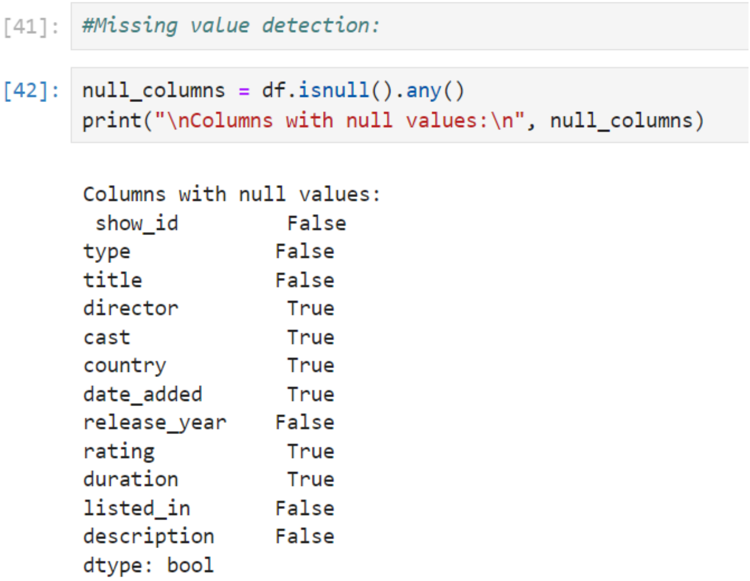
- **Conversion of categorical attributes to 'category' (If required):**
  There is some columns whose attributes can be converted to 'category' which can be done as shown



```
[39]: #Conversion of categorical attributes to 'category' (If required):

[40]: categorical_columns = ['type', 'rating', 'duration', 'listed_in']
      for col in categorical_columns:
          df[col] = df[col].astype('category')
```

- **Missing value detection:** There are column in which values are missing!
  Code to see is as:



```
[41]: #Missing value detection:

[42]: null_columns = df.isnull().any()
      print("\nColumns with null values:\n", null_columns)

      Columns with null values:
       show_id         False
      type            False
      title           False
      director         True
      cast             True
      country          True
      date_added       True
      release_year    False
      rating           True
      duration         True
      listed_in       False
      description     False
      dtype: bool
```

Insights from the Output

1. Columns with Missing Values:
   The columns 'director', 'cast', 'country', 'date_added', and 'rating' contain null values. This indicates that these fields have incomplete data.

2. Data Quality Issues:
   'director': Missing values in the 'director' column could hinder analysis related to directors, such as identifying popular directors or analyzing the impact of directors on the success of shows.
   'cast': Missing values in the 'cast' column could affect analyses related to actors and their influence on viewership.
   'country': Missing country information might complicate analyses focusing on geographic trends or the distribution of content production by country.
   'date_added': Missing values in the 'date_added' column could interfere with time-based analyses, such as trends in content additions over time.
   'rating': Missing ratings can affect content classification and analysis based on audience suitability.

2. Non-Graphical Analysis: Value counts and unique attributes:

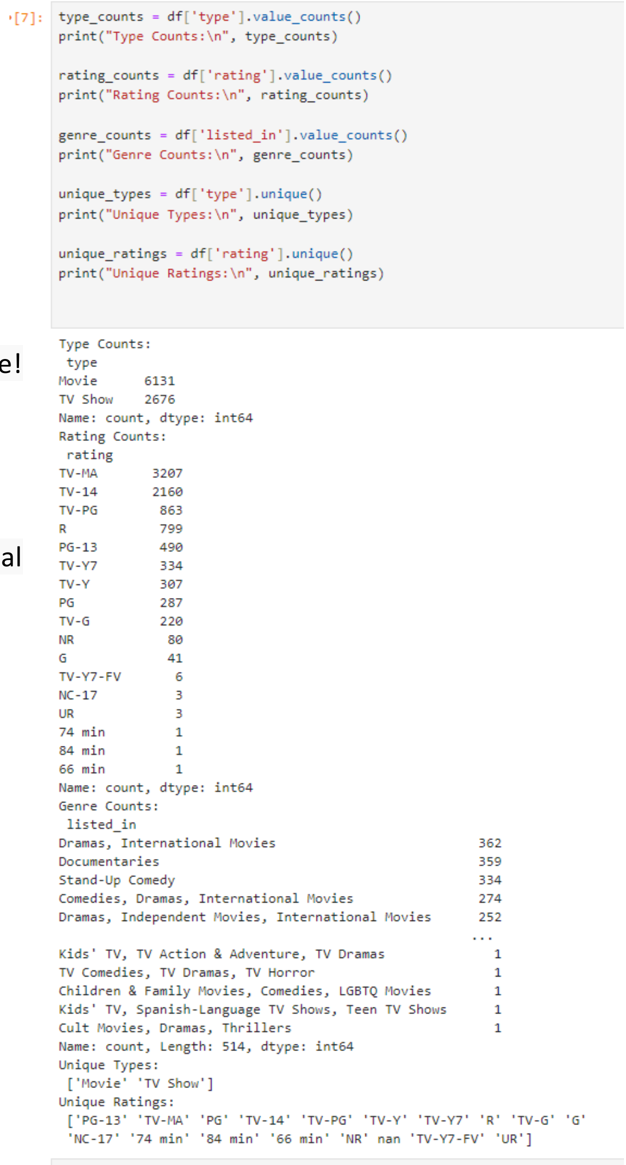
Below is the mostly used non-graphical analysis:

Type Count:
Rating Count:
Country Count:
Genres Count:
Unique Values in 'Type':
Unique Values in 'Rating':
Unique Values in 'Genres':

Insights: insights are mentioned in above questions
Because here is same process done as above!

conclusion:

By conducting non-graphical analyses such as value counts and identifying unique attributes, you gain a foundational understanding of the dataset's structure and composition. This information is crucial for guiding further, more detailed analyses and visualizations.



[7]:

```python
type_counts = df['type'].value_counts()
print("Type Counts:\n", type_counts)

rating_counts = df['rating'].value_counts()
print("Rating Counts:\n", rating_counts)

genre_counts = df['listed_in'].value_counts()
print("Genre Counts:\n", genre_counts)

unique_types = df['type'].unique()
print("Unique Types:\n", unique_types)

unique_ratings = df['rating'].unique()
print("Unique Ratings:\n", unique_ratings)
```

```
Type Counts:
 type
Movie      6131
TV Show    2676
Name: count, dtype: int64
Rating Counts:
 rating
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: count, dtype: int64
Genre Counts:
 listed_in
Dramas, International Movies                        362
Documentaries                                       359
Stand-Up Comedy                                     334
Comedies, Dramas, International Movies              274
Dramas, Independent Movies, International Movies    252
                                                   ... 
Kids' TV, TV Action & Adventure, TV Dramas            1
TV Comedies, TV Dramas, TV Horror                     1
Children & Family Movies, Comedies, LGBTQ Movies      1
Kids' TV, Spanish-Language TV Shows, Teen TV Shows    1
Cult Movies, Dramas, Thrillers                        1
Name: count, Length: 514, dtype: int64
Unique Types:
 ['Movie' 'TV Show']
Unique Ratings:
 ['PG-13' 'TV-MA' 'PG' 'TV-14' 'TV-PG' 'TV-Y' 'TV-Y7' 'R' 'TV-G' 'G'
 'NC-17' '74 min' '84 min' '66 min' 'NR' nan 'TV-Y7-FV' 'UR']
```

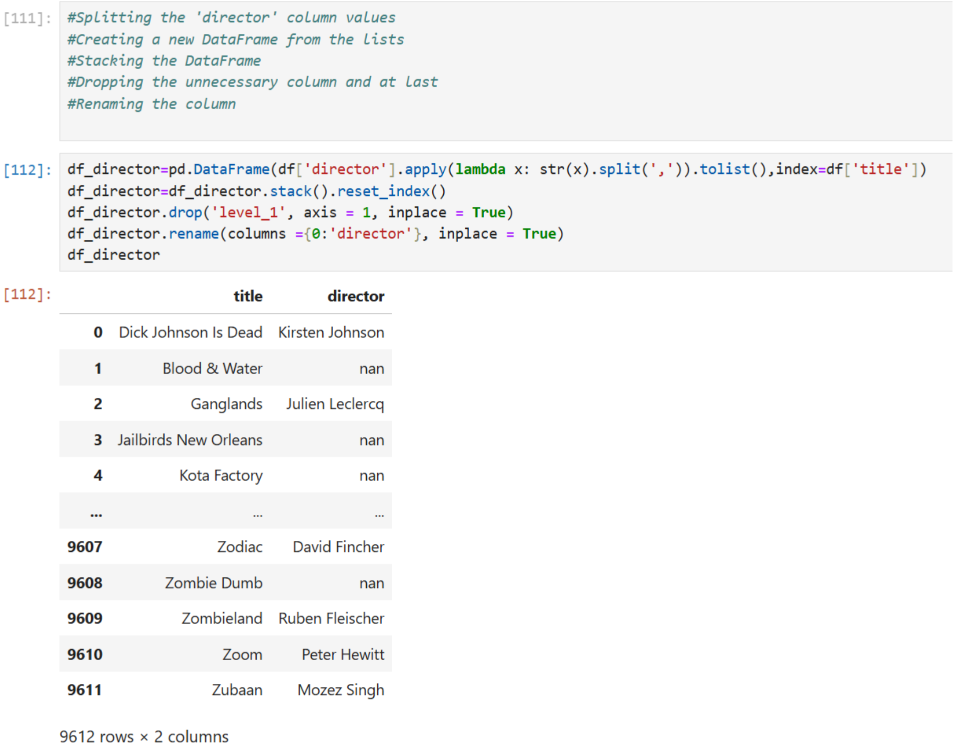
4. Visual Analysis - Univariate, Bivariate after pre-processing of the data

Note: Pre-processing involves unnesting of the data in columns like Actor, Director, Country.

Pre-processing the data:

The pre-processing of data was don't on director, cast, country, columns.

Below are the codes for pre-processing:



```
[111]: #Splitting the 'director' column values
       #Creating a new DataFrame from the lists
       #Stacking the DataFrame
       #Dropping the unnecessary column and at last
       #Renaming the column
```

```
[112]: df_director=pd.DataFrame(df['director'].apply(lambda x: str(x).split(',')).tolist(),index=df['title'])
       df_director=df_director.stack().reset_index()
       df_director.drop('level_1', axis = 1, inplace = True)
       df_director.rename(columns ={0:'director'}, inplace = True)
       df_director
```

[112]:

	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
...	...	...
9607	Zodiac	David Fincher
9608	Zombie Dumb	nan
9609	Zombieland	Ruben Fleischer
9610	Zoom	Peter Hewitt
9611	Zubaan	Mozez Singh

9612 rows × 2 columns

Insight:

1. **Splitting the 'director' column:**

   The 'director' column is split based on commas using str(x).split(','), where x represents each entry in the 'director' column.
   The result is converted to a list using tolist().

2. **Creating a new DataFrame (df_director):**

   - The resulting lists from the previous step are converted into a DataFrame using pd.DataFrame().
   - The index of the new DataFrame is set to the 'title' column of the original DataFrame df.
   - The lists are stacked into separate rows, resulting in a multi-index DataFrame.
   - The 'level_1' index level, which represents the position of the director within the original list, is dropped using df_director.drop('level_1', axis=1, inplace=True).

3. **Renaming columns:**

   The column resulting from the split (previously named '0') is renamed to 'director' using df_director.rename(columns={'0': 'director'}, inplace=True).

Overall, the code effectively transforms the 'director' column of the DataFrame df into a new DataFrame (df_director) where each movie can have multiple rows, each corresponding to a different director.

These could be some recommendations:

1. **Understanding the Transformation:**

   The transformation effectively converts a column with potentially multiple directors listed in a single cell into a format where each director has their own row. This makes it easier to analyze the contributions of individual directors.

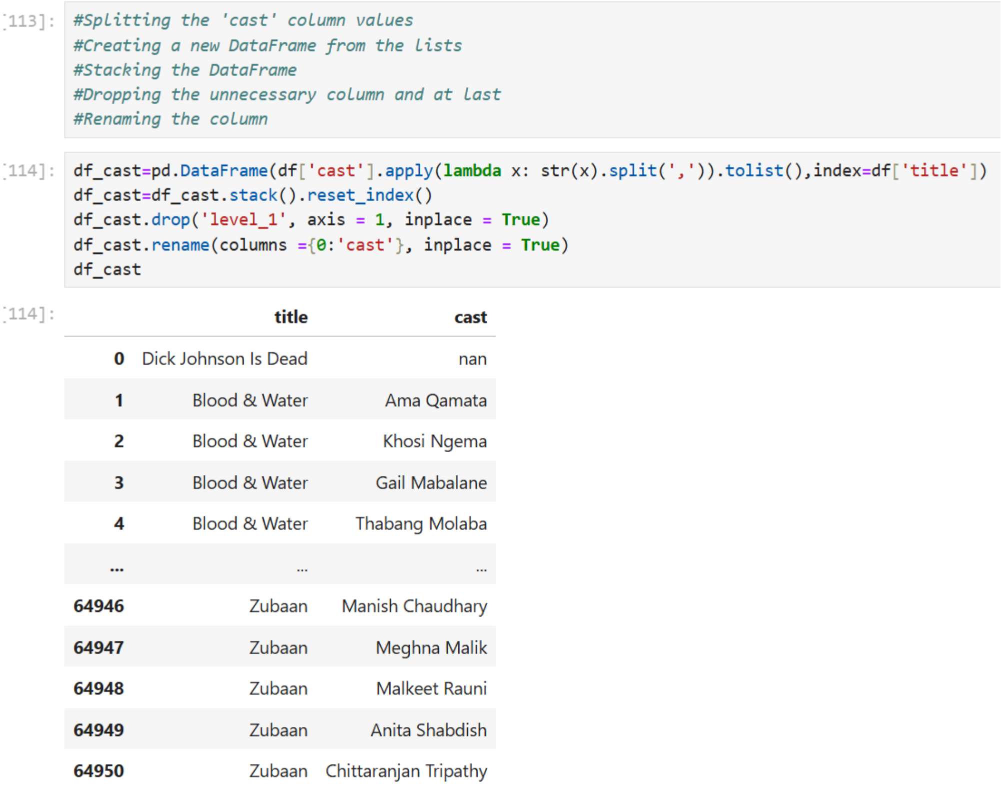
2. **Verify Data Quality:**

   Ensure that the 'director' column does not contain any malformed data that might affect the splitting process. For example, if there are unexpected delimiters or extra spaces, it might be useful to clean the data first.

3. **Handle Missing Values:**

   If the 'director' column contains missing values (e.g., NaN), they will be converted to the string 'nan' during the splitting process. Consider handling these appropriately, such as by filling or dropping them before the transformation.

performing same as I did on director and title for below:



```
[113]: #Splitting the 'cast' column values
       #Creating a new DataFrame from the lists
       #Stacking the DataFrame
       #Dropping the unnecessary column and at last
       #Renaming the column
```

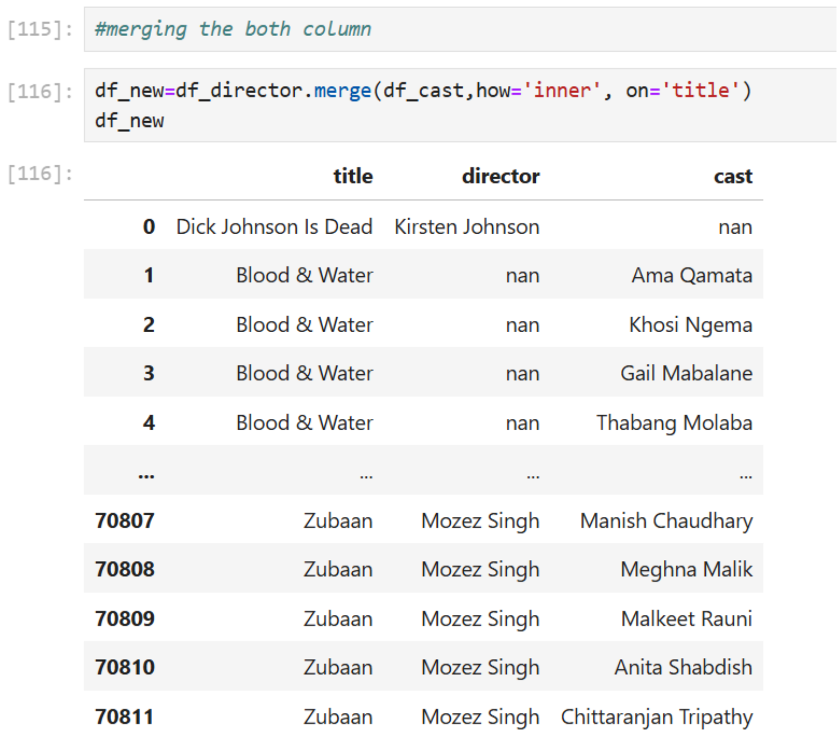
```
[114]: df_cast=pd.DataFrame(df['cast'].apply(lambda x: str(x).split(',')).tolist(),index=df['title'])
       df_cast=df_cast.stack().reset_index()
       df_cast.drop('level_1', axis = 1, inplace = True)
       df_cast.rename(columns ={0:'cast'}, inplace = True)
       df_cast
```

[114]:

	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
...	...	...
64946	Zubaan	Manish Chaudhary
64947	Zubaan	Meghna Malik
64948	Zubaan	Malkeet Rauni
64949	Zubaan	Anita Shabdish
64950	Zubaan	Chittaranjan Tripathy

Insights and recommendations can be similar as above!

Merging both above derived column:



[115]:

```
#merging the both column
```

[116]:

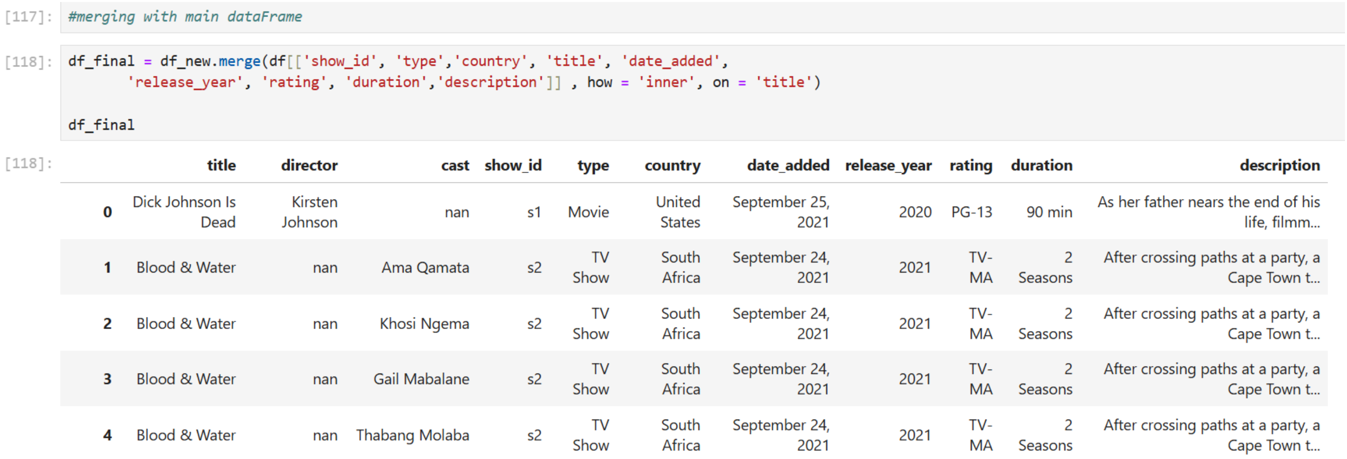
```
df_new=df_director.merge(df_cast,how='inner', on='title')
df_new
```

[116]:

	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

simple merging doesn't seams to have much insights to write!

Merging the result columns to main data frame!



[117]:

```
#merging with main dataFrame
```

[118]:

```
df_final = df_new.merge(df[['show_id', 'type','country', 'title', 'date_added',
        'release_year', 'rating', 'duration','description']] , how = 'inner', on = 'title')

df_final
```

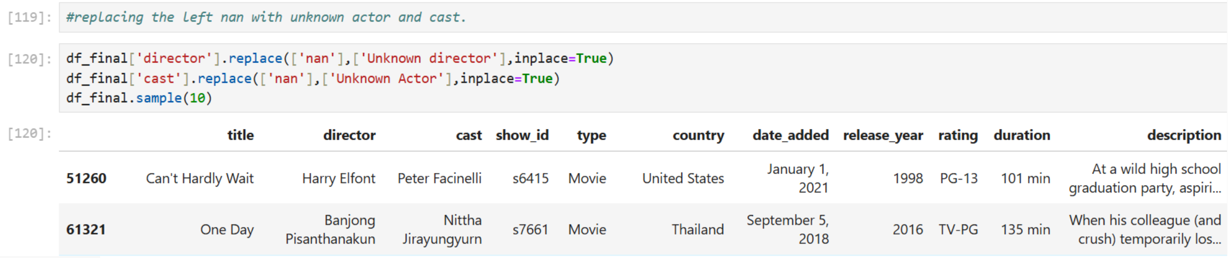
[118]:

	title	director	cast	show_id	type	country	date_added	release_year	rating	duration	description
0	Dick Johnson Is Dead	Kirsten Johnson	nan	s1	Movie	United States	September 25, 2021	2020	PG-13	90 min	As her father nears the end of his life, filmm...
1	Blood & Water	nan	Ama Qamata	s2	TV Show	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
2	Blood & Water	nan	Khosi Ngema	s2	TV Show	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
3	Blood & Water	nan	Gail Mabalane	s2	TV Show	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
4	Blood & Water	nan	Thabang Molaba	s2	TV Show	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...

Insights and Recommendations:

1. **Comprehensive Analysis:**
   The resulting df_final DataFrame now includes comprehensive details for each movie, enabling you to analyze relationships between directors, cast members, and other movie attributes.
2. **Diversity of Collaborations:**
   Analyze the diversity of collaborations between directors and cast members across different countries, genres (using 'type'), and release years.
3. **Trend Analysis:**
   Examine trends in movie ratings, durations, and release dates by director or by cast member. Investigate how the involvement of specific directors or cast members correlates with movie ratings and audience reception.

Replacing left of the Nan value with Unknown actor and director:



```
[119]: #replacing the left nan with unknown actor and cast.

[120]: df_final['director'].replace(['nan'],['Unknown director'],inplace=True)
       df_final['cast'].replace(['nan'],['Unknown Actor'],inplace=True)
       df_final.sample(10)
```

[120]:

	title	director	cast	show_id	type	country	date_added	release_year	rating	duration	description
51260	Can't Hardly Wait	Harry Elfont	Peter Facinelli	s6415	Movie	United States	January 1, 2021	1998	PG-13	101 min	At a wild high school graduation party, aspiri...
61321	One Day	Banjong Pisanthanakun	Nittha Jirayungyurn	s7661	Movie	Thailand	September 5, 2018	2016	TV-PG	135 min	When his colleague (and crush) temporarily los...

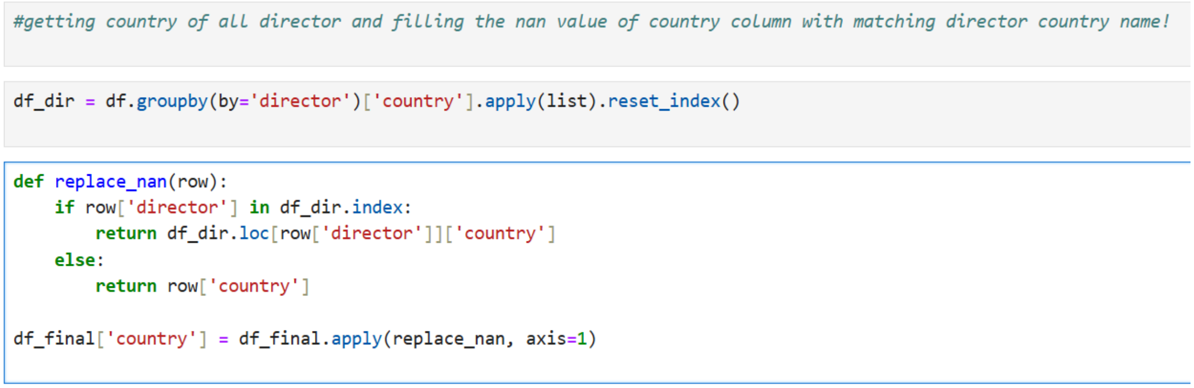
Insights:

1. **Unknown Director and Actor Entries:**
   Entries with 'Unknown director' and 'Unknown Actor' suggest that some movies in your dataset might lack information about their directors or cast members. These entries could be due to missing data or incomplete records.
2. **Frequency of Unknown Entries:**
   Analyze the frequency of 'Unknown director' and 'Unknown Actor' entries to understand the extent of missing data in your dataset

Recommendations:

Data Completeness Analysis:
Investigate why certain movies have missing director or cast information. Determine if it's due to data collection issues, incomplete records, or intentional omission.

Pre-processing the country column based on repeatition of director name and if country col is missing will be replaced after NaN:



```
#getting country of all director and filling the nan value of country column with matching director country name!
```

```
df_dir = df.groupby(by='director')['country'].apply(list).reset_index()
```

```
def replace_nan(row):
    if row['director'] in df_dir.index:
        return df_dir.loc[row['director']]['country']
    else:
        return row['country']

df_final['country'] = df_final.apply(replace_nan, axis=1)
```

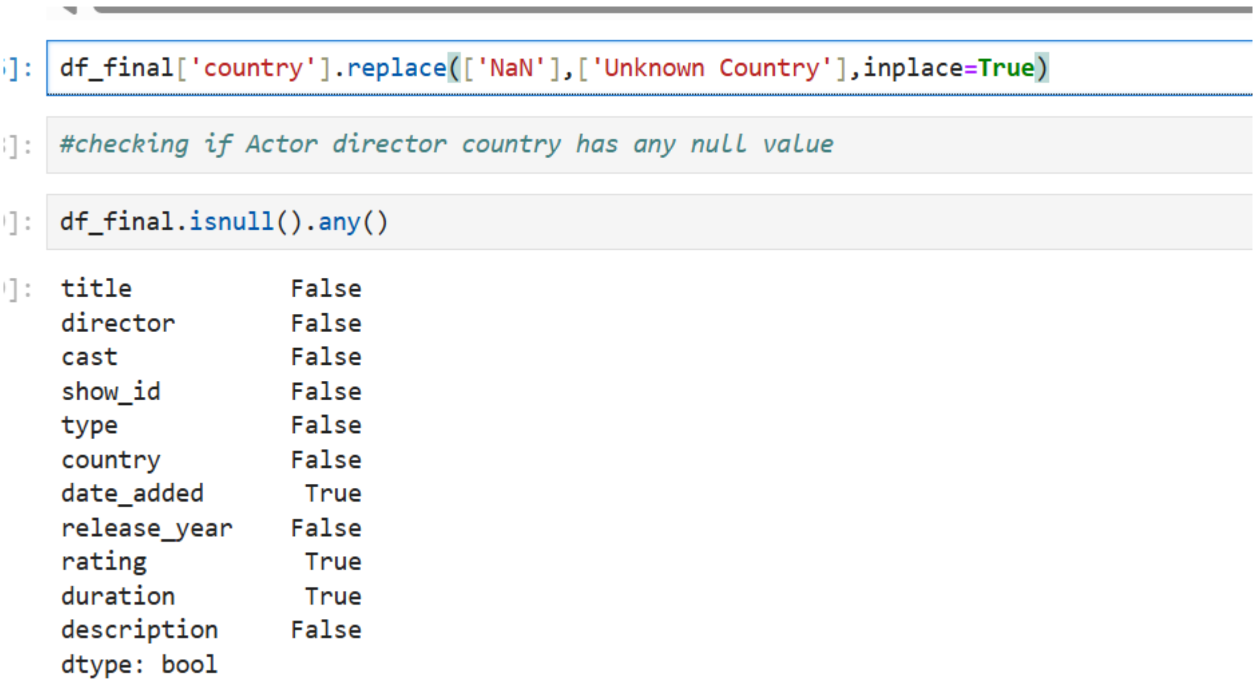
Insights:

1. **Country Information Enhancement:**
   By associating each director with a list of countries where their movies were produced, you enhance the available country information in df_final.
2. **Leveraging Grouped Data:**
   Grouping df by directors allows for capturing the diversity of countries associated with each director's filmography.

Recommendations:

1. Data Completeness Check:
   Verify the completeness and accuracy of the director-country associations in the grouped DataFrame df_dir. Ensure that all relevant countries are captured for each director.

Replacing the NAN value with Unknown country:



```
df_final['country'].replace(['NaN'],['Unknown Country'],inplace=True)
```

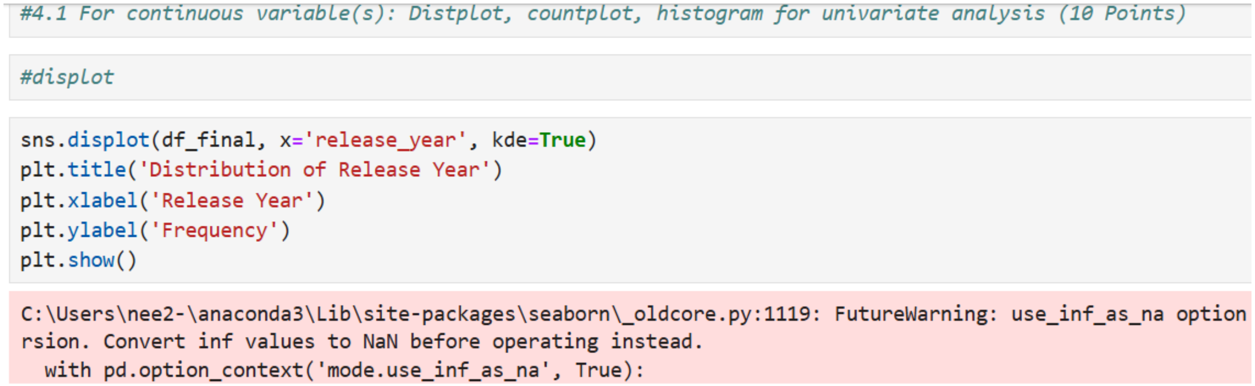
```
#checking if Actor director country has any null value
```

```
df_final.isnull().any()
```

```
title           False
director        False
cast            False
show_id         False
type            False
country         False
date_added       True
release_year    False
rating           True
duration         True
description     False
dtype: bool
```

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

1. Displot: The code for displot between distribution of release year and frequency is given as:

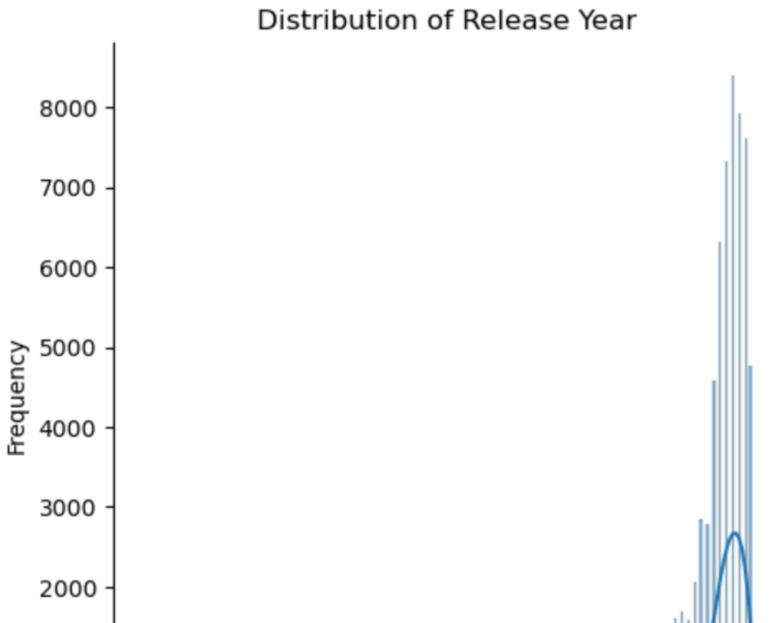


```
#4.1 For continuous variable(s): Distplot, countplot, histogram for univariate analysis (10 Points)
```

```
#displot
```

```
sns.displot(df_final, x='release_year', kde=True)
plt.title('Distribution of Release Year')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.show()
```

```
C:\Users\nee2-\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option
rsion. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



Insights:

1. **Release Year Distribution:**
   The distribution plot provides an overview of the frequency of movie releases across different years.
   Peaks or clusters in the distribution may indicate periods of increased movie production or specific trends in the film industry.

2. **Trend Identification:**
   Observing the shape of the distribution can help identify trends or patterns in movie release years over time. For example, a left-skewed distribution may indicate a recent surge in movie production, while a right-skewed distribution may suggest a decline.

3. **Historical Context:**
   Analyzing the distribution in conjunction with historical events or cultural phenomena can provide context for trends in movie release years. For instance, spikes in certain years may coincide with major technological advancements, economic factors, or shifts in audience preferences.

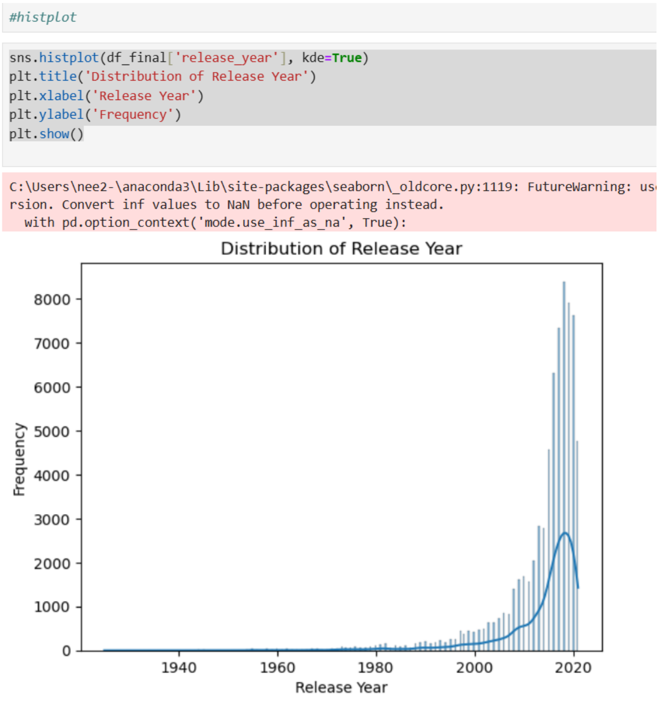
4. **Industry Insights:**
   Changes in the distribution pattern may reflect broader shifts within the film industry, such as the rise of certain genres, changes in production methods, or the influence of global events on storytelling themes.

Recommendations:

1. Exploratory Analysis:
   Conduct further exploratory analysis to delve deeper into the distribution patterns. Explore subsets of the data based on genres, directors, or production countries to uncover nuanced insights.

2. Temporal Trends:
   Analyze temporal trends in movie release years to identify periods of innovation, stagnation, or transformation within the film industry. Consider how these trends correlate with external factors and industry dynamics.

Histplot: histplot drawn on distribution of release year as:



```
#histplot
```

```
sns.histplot(df_final['release_year'], kde=True)
plt.title('Distribution of Release Year')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.show()
```

```
C:\Users\nee2-\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: us
rsion. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

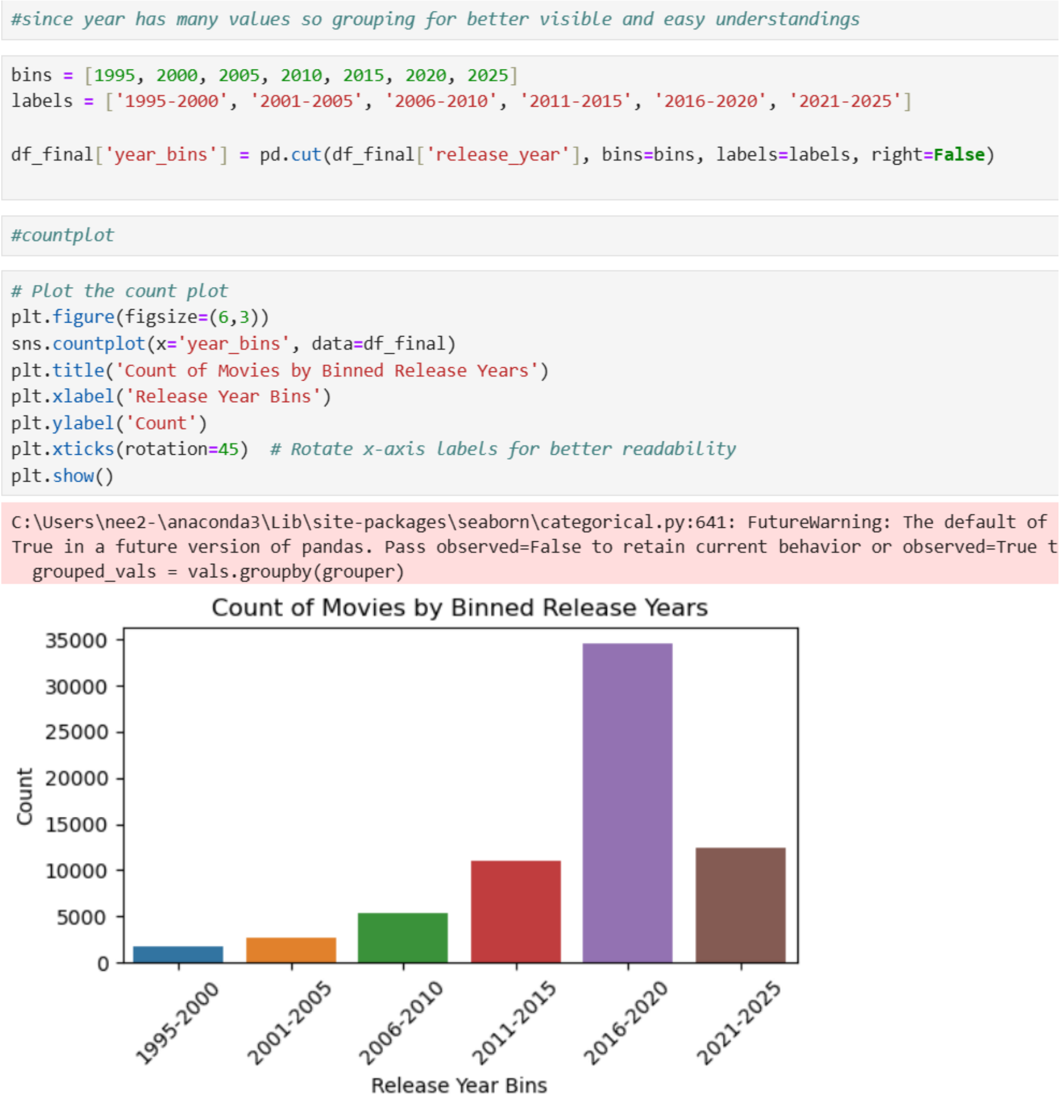
Insights:

1. **Release Year Distribution:**
   The histogram plot provides a visual representation of the frequency distribution of movie release years.
   It allows for quick identification of the central tendency, spread, and shape of the distribution.
2. **Density Estimation:**
   The inclusion of kernel density estimation (KDE) overlays a smoothed estimate of the probability density function, providing additional insights into the underlying distribution shape.
3. **Central Tendency and Spread:**
   The central peak or mode of the distribution indicates the most common release years, while the spread around the mode illustrates the variability in release year frequencies.
4. **Outliers and Anomalies:**
   Outliers or anomalies in the distribution, represented by isolated peaks or valleys, may indicate unique events or trends in movie release patterns.

Recommendations:

1. **Interpretation of Peaks:**
   Identify peaks in the distribution and investigate the corresponding release years to understand the significance of these peaks. Peaks may correspond to specific historical periods, blockbuster releases, or industry trends.
2. **Comparison with Historical Events:**
   Compare the distribution of release years with significant historical events, technological advancements, or cultural phenomena to discern potential correlations or influences on movie production trends.
3. **Segmentation Analysis:**
   Conduct segmentation analysis by genre, director, or production country to explore variations in release year distributions across different categories. This can reveal genre-specific trends or regional preferences.

Countplot: Count plot on years number of movies can be plotted as:



```
#since year has many values so grouping for better visible and easy understandings
```

```
bins = [1995, 2000, 2005, 2010, 2015, 2020, 2025]
labels = ['1995-2000', '2001-2005', '2006-2010', '2011-2015', '2016-2020', '2021-2025']

df_final['year_bins'] = pd.cut(df_final['release_year'], bins=bins, labels=labels, right=False)
```

```
#countplot
```

```
# Plot the count plot
plt.figure(figsize=(6,3))
sns.countplot(x='year_bins', data=df_final)
plt.title('Count of Movies by Binned Release Years')
plt.xlabel('Release Year Bins')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```

```
C:\Users\nee2-\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarning: The default of
True in a future version of pandas. Pass observed=False to retain current behavior or observed=True t
  grouped_vals = vals.groupby(grouper)
```

Insights:

1. **Binned Release Years:**
   The binned release years allow for a clearer representation of movie distribution across distinct time periods, making it easier to identify trends and patterns.
2. **Temporal Segmentation:**
   Grouping release years into bins provides a structured approach to analyze temporal trends in movie production over different eras or decades.
3. **Distribution Disparity:**
   Variations in the count of movies across different bins may indicate shifts in movie production activity over time, reflecting changes in industry dynamics or audience preferences.

Recommendations:

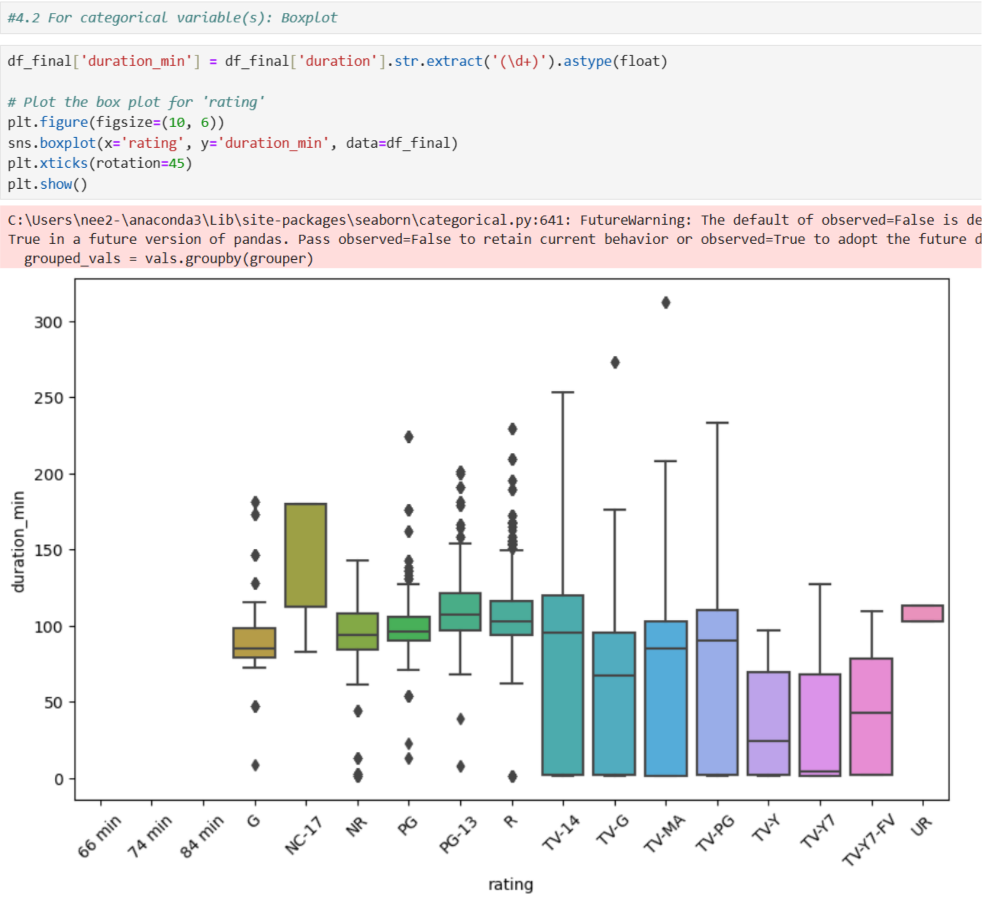
a) **Interpretation of Trends:**
Analyze the count plot to identify trends or anomalies in movie production across different time periods. Look for patterns such as increasing or decreasing trends, periodic fluctuations, or significant spikes.

b) **Historical Context:**
Consider historical events, cultural movements, or technological advancements that may have influenced movie production during each bin's timeframe. Contextualizing the data within broader historical contexts can provide deeper insights.

c) **Comparison and Contrast:**
Compare the distribution of movies across bins to discern differences in production activity between decades or specific periods. Explore factors contributing to these differences, such as changes in audience demographics or shifts in cinematic trends.

4.2 For categorical variable(s): Boxplot

Boxplot for categorical variable can be plotted as:



```
#4.2 For categorical variable(s): Boxplot
```

```
df_final['duration_min'] = df_final['duration'].str.extract('(\d+)').astype(float)

# Plot the box plot for 'rating'
plt.figure(figsize=(10, 6))
sns.boxplot(x='rating', y='duration_min', data=df_final)
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\nee2-\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarning: The default of observed=False is de
True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future d
  grouped_vals = vals.groupby(grouper)
```

Insights:

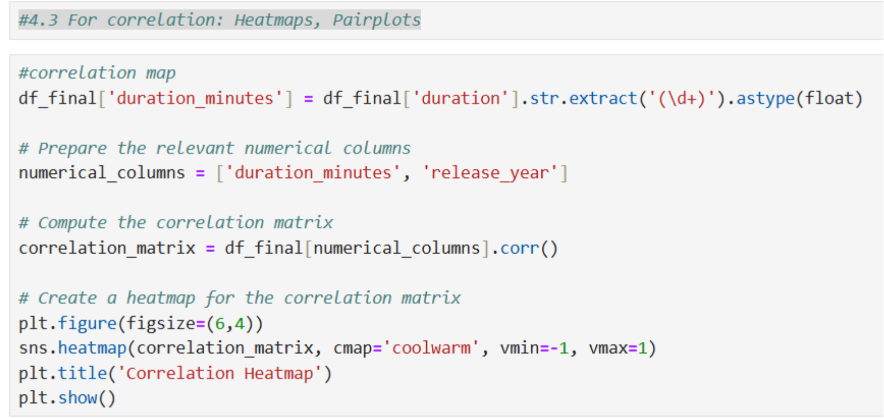
1. **Duration Distribution by Rating:**
   - The box plot allows for a visual comparison of movie durations across different rating categories.
   - Box plots provide information about the central tendency, spread, and presence of outliers within each rating group.
2. **Variability in Movie Durations:**
   Variability in the spread of durations within each rating category can indicate differences in movie length preferences or production tendencies associated with specific ratings.
3. **Outlier Detection:**
   Outliers in the box plot represent movies with exceptionally long or short durations within each rating group. These outliers may warrant further investigation to understand their characteristics and potential impact on audience reception.

Recommendations:

1. **Interpretation of Box Plot:**
   Interpret the box plot to identify any notable differences in movie durations across rating categories. Look for trends or patterns that may indicate preferences or conventions associated with certain ratings.
2. **Comparison and Contrast:**
   Compare the distribution of movie durations across different ratings to identify similarities or disparities. Assess whether certain rating categories tend to feature longer or shorter movies compared to others.
3. **Genre-specific Analysis:**
   Conduct genre-specific analysis to explore how the relationship between movie duration and rating varies within different genres. Certain genres may exhibit distinct patterns or preferences in terms of movie length.

4.3 For correlation: Heatmaps, Pairplots:

Heatmap for correlation:



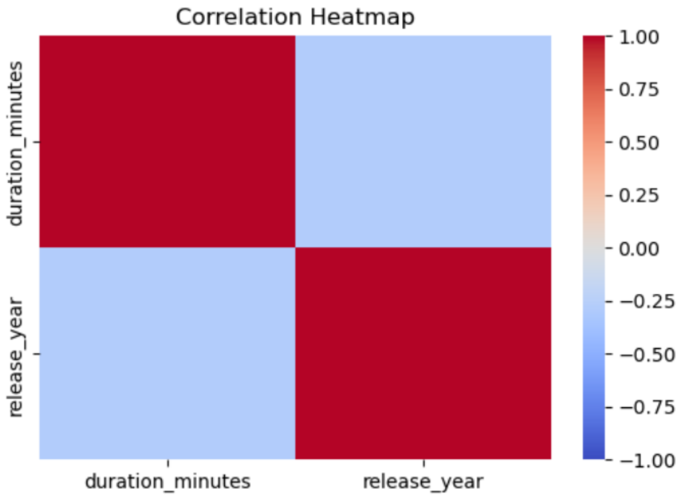
```
#4.3 For correlation: Heatmaps, Pairplots
```

```
#correlation map
df_final['duration_minutes'] = df_final['duration'].str.extract('(\d+)').astype(float)

# Prepare the relevant numerical columns
numerical_columns = ['duration_minutes', 'release_year']

# Compute the correlation matrix
correlation_matrix = df_final[numerical_columns].corr()

# Create a heatmap for the correlation matrix
plt.figure(figsize=(6,4))
sns.heatmap(correlation_matrix, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap')
plt.show()
```



Insights:

1. **Correlation Coefficient:**
   - The heatmap displays the correlation coefficients between 'duration_minutes' and 'release_year'. The values range from -1 to 1, where:
   - 1 indicates a perfect positive correlation.
   - -1 indicates a perfect negative correlation.
   - 0 indicates no correlation.

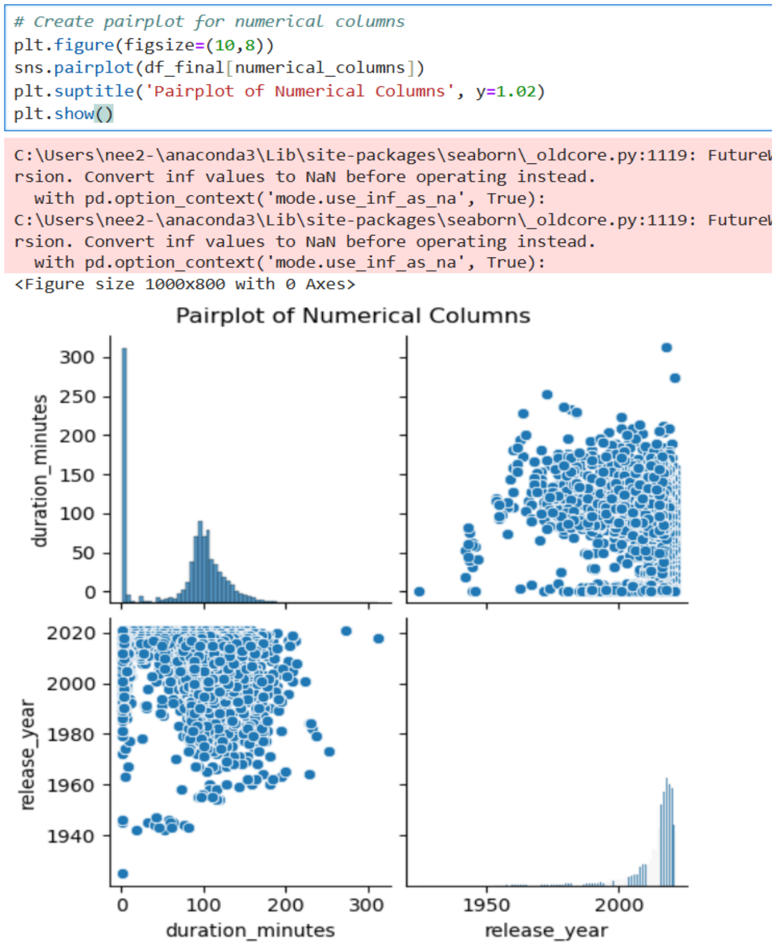
2. **Strength and Direction:**
    - A positive correlation coefficient indicates that as one variable increases, the other variable also tends to increase.
    - A negative correlation coefficient indicates that as one variable increases, the other variable tends to decrease.
3. **Interpretation:**
    - The strength of the correlation can be interpreted based on the absolute value of the coefficient:
    - to 0.2: Very weak to no correlation.
    - 0.2 to 0.4: Weak correlation.
    - 0.4 to 0.6: Moderate correlation.
    - 0.6 to 0.8: Strong correlation.
    - 0.8 to 1.0: Very strong correlation.

Recommendations:

1. **Identify Relationships:**
   Use the correlation heatmap to identify significant relationships between numerical variables. In this case, you can assess if there is any notable correlation between movie duration and release year.
2. **Monitor Trends:**
   If a trend is identified (e.g., movie durations increasing over years), analyze the potential reasons behind this trend, such as changes in audience preferences or advancements in movie production technologies.

For correlation: Pairplots

Can be drawn as:



```
# Create pairplot for numerical columns
plt.figure(figsize=(10,8))
sns.pairplot(df_final[numerical_columns])
plt.suptitle('Pairplot of Numerical Columns', y=1.02)
plt.show()
```

```
C:\Users\nee2-\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureW
rsion. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\nee2-\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureW
rsion. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
<Figure size 1000x800 with 0 Axes>
```

Insights:

1. **Scatter Plots:**
    - The scatter plots in the pairplot show the relationship between each pair of numerical variables.
    - In this case, you will have a scatter plot for duration_minutes vs. release_year.
2. **Histograms:**
    - The diagonal of the pairplot contains histograms showing the distribution of each individual numerical variable.
    - These histograms provide insights into the central tendency, spread, and shape of the distributions for duration_minutes and release_year.

Correlation and Trends:

The scatter plots can reveal linear or non-linear relationships, clusters, and potential outliers. Look for any apparent trends, such as whether newer movies tend to be longer or shorter.

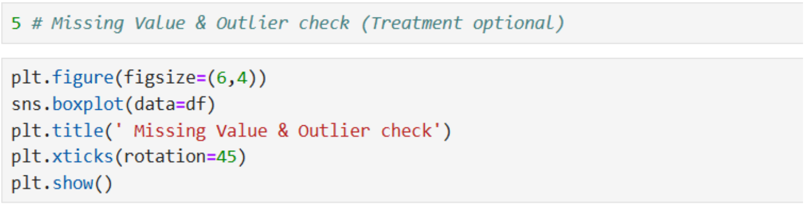
Recommendations:

1. **Visual Inspection:**
    - Visually inspect the scatter plots for any clear patterns or relationships between duration_minutes and release_year.
    - Look for clusters, trends, or outliers that may warrant further investigation.
2. **Statistical Analysis:**
   Based on the visual patterns observed, consider performing statistical analyses such as regression analysis to quantify the relationships between variables.
3. **Additional Variables:**
4. Consider adding more numerical variables to the pairplot to explore additional relationships. For example, including rating or view_count could provide more comprehensive insights.

5 - Missing Value & Outlier check (Treatment optional)

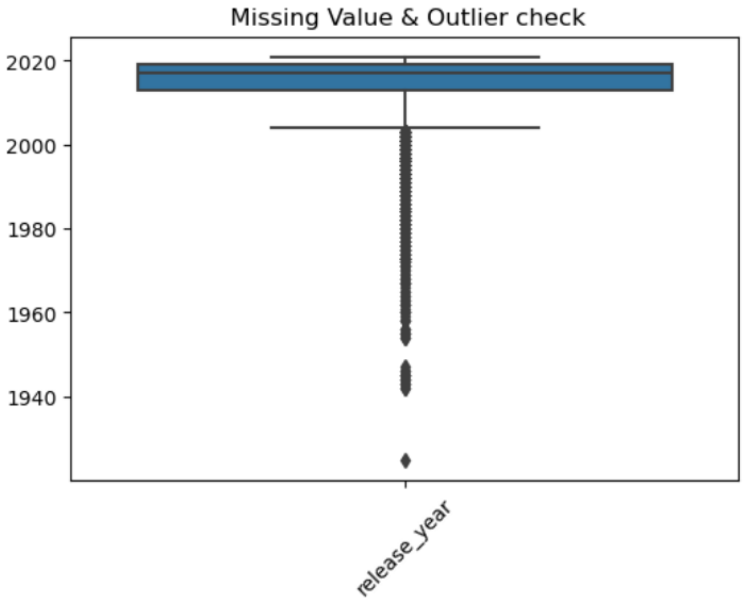
Checking the missing value and outliers with the help of boxplot:

Code is shown as:



```
5 # Missing Value & Outlier check (Treatment optional)
```

```
plt.figure(figsize=(6,4))
sns.boxplot(data=df)
plt.title(' Missing Value & Outlier check')
plt.xticks(rotation=45)
plt.show()
```



Insights:

1. **Outlier Detection:**
   The box plot will highlight any outliers in the 'duration_minutes' and 'release_year' columns. Outliers are typically shown as points outside the whiskers of the box plot.
2. **Data Distribution:**
   The box plot will also show the interquartile range (IQR), median, and the overall spread of the data for each numerical column.

Recommendations:

1. **Handle Outliers:**
   Investigate the outliers to understand why they exist. If they are due to data entry errors, consider correcting or removing them. If they are valid, decide whether to include them in further analysis based on their impact.
2. **Impute Missing Values:**
   If there are missing values, consider imputing them using appropriate methods such as mean, median, or mode imputation, or more advanced techniques like K-Nearest Neighbors (KNN) imputation.

Code file is mentioned below:

netflix_project_1.ipynb

6.Insights based on Non-Graphical and Visual Analysis:

Insights Based on Non-Graphical Analysis:

1. **Director and Cast Information:**
   - By splitting the director and cast columns and analyzing them, we gain insights into the diversity and frequency of collaborations in the film industry.
   - Common directors and actors can be identified, allowing us to see patterns in movie productions.
2. **Duration Data:**
   - Extracting numeric values from the duration column provides a clear understanding of movie lengths.
   - Converting the duration to minutes allows for easy comparison and analysis across movies.
3. **Missing Values:**
   - Identifying and replacing missing values, such as replacing 'nan' with 'Unknown director' and 'Unknown Actor', helps maintain data integrity and ensures that analyses are not skewed by missing data.
   - Replacing missing country data based on the director helps fill gaps and maintain consistency in the dataset.
4. **Data Binning:**
   - Creating bins for release years helps in categorizing movies into specific time periods, making it easier to analyze trends over time.
   - Binning allows for a more structured analysis of how movie characteristics (like duration and rating) have changed over different periods.

Insights Based on Visual Analysis:

1. **Distribution of Release Year:**
   - The histogram and KDE plot of the release year show the frequency of movie releases over time.

    - Peaks in the distribution can indicate periods of high production activity, while troughs may indicate periods of lower activity.

2. **Box Plots:**

    - Box plots for duration_minutes and release_year provide insights into the spread and central tendency of these variables.
    - Outliers identified in these plots can indicate unusual movie lengths or potentially erroneous data entries for release years.

3. **Correlation Heatmap:**
    - The correlation heatmap shows the relationship between duration_minutes and release_year.
    - Understanding these correlations helps in identifying trends, such as whether newer movies tend to be longer or shorter.

4. **Pair Plots:**

    - Pair plots provide a comprehensive view of the relationships between numerical variables.
    - Scatter plots within the pair plot can reveal linear or non-linear relationships, clusters, and outliers.
    - Histograms on the diagonal of the pair plot show the distribution of individual numerical variables, highlighting any skewness or unusual patterns.

6.1 Comments on the range of attributes

1. **Director and Cast Information:**

Range: The number of unique directors and cast members in the dataset.

Comments: The dataset likely includes a wide range of directors and actors, reflecting the diversity of the film industry. Analyzing this range can highlight prolific directors and actors, as well as collaborative networks.

3. **Duration (Minutes):**

Range: The minimum and maximum duration of movies in minutes.

Comments: This attribute can vary widely, from short films to long feature films. Understanding the range helps in categorizing movies into short, medium, and long durations, and identifying any extreme outliers that might need further investigation.

4. **Release Year:**

Range: The earliest and latest release years of the movies in the dataset.

Comments: The release years span a significant range, allowing for temporal analysis of movie trends. The range indicates the historical depth of the dataset and can be used to study how movie characteristics have evolved over time.

5. **Rating:**

Range: The different types of ratings assigned to movies (e.g., G, PG, PG-13, R, etc.).

Comments: The range of ratings reflects the target audience and content suitability of the movies. Analysing the distribution of ratings can provide insights into the types of movies produced and their intended audiences.

6. **Country:**

Range: The number of unique countries represented in the dataset.

Comments: A wide range of countries indicates a diverse dataset with international representation. This range is crucial for understanding the global reach and cultural diversity of the movies.

7. **Type (Movie/TV Show):**

Range: The count of each type (movies and TV shows).

Comments: The distribution of movies versus TV shows can provide insights into the content strategy of the platform or dataset source. A balanced range suggests a diversified content offering.

8. **Date Added:**

Range: The earliest and latest dates when the movies were added to the platform.

Comments: This range helps in understanding the platform's content acquisition timeline and growth. It can also indicate periods of high content addition activity.

9. **Duration:**

Range: The range of durations (typically in string format indicating hours and minutes).

Comments: When converted to numerical format, this range helps in analyzing the average length of movies and identifying any trends in movie duration over time.

10. **Description:**

Range: The length and content of movie descriptions.

Comments: The range of description lengths can vary, with some being short and concise, while others are more detailed. Analysing these descriptions can provide insights into marketing and content summarization practices.

6.2 Comments on the distribution of the variables and relationship between them:

1. Distribution of Release Year

- Histogram Analysis:
  The histogram of release years shows how the frequency of movie releases varies over time.
- Observations:
  A) Peaks in the histogram may indicate periods of higher production, possibly reflecting industry trends, technological advancements, or historical events.
  B) Troughs may reflect periods of lower production.
  C) Recent years often show higher frequencies due to increased production and digitization of content.
- Skewness:
  The distribution might be skewed towards recent years if the dataset includes more recent entries.

2. Duration (Minutes)

- Box Plot Analysis:
  The box plot of movie durations shows the central tendency and variability.
- Observations:
  The median duration provides a typical length for movies in the dataset.
  The interquartile range (IQR) shows the spread of the middle 50% of the data.
  Outliers might indicate unusually short or long movies.

- Distribution Shape:
  If the distribution of durations is not symmetric, it may be skewed. Long tails might indicate a few extremely long movies.

3. Rating

- Categorical Distribution:
  The count plot of ratings shows the frequency of each rating category.
- Observations:
  A) Some ratings (e.g., PG-13, R) might be more common, reflecting target audience preferences.
  B) Less frequent ratings (e.g., G) could indicate fewer movies targeted at younger audiences.
- Implications:
  Understanding rating distribution helps in analyzing the types of content that are predominant in the dataset.

4. Year Bins

- Count Plot Analysis:
  The count plot for binned release years shows the number of movies released in specific time intervals.
- Observations:
  A) Periods with high counts indicate more active movie production phases.
  B) Analyzing these bins helps in identifying trends over time.

5. Country

- Geographical Distribution:
  The frequency distribution of movies by country highlights the geographical diversity.
- Observations:
  A) Some countries might dominate the dataset, reflecting major movie-producing regions (e.g., USA, India).
  B) Less frequent countries indicate regions with fewer productions or less representation in the dataset.

Relationships Between Variables

1. Duration vs. Release Year

- Correlation Heatmap:
  The heatmap shows the correlation between movie duration and release year.
- Observations:
  A) A positive correlation might suggest that movie durations have increased over time.
  B) A negative or no correlation indicates no clear trend in movie lengths over the years.

2. Pair Plot Analysis

- Pair Plots:
  Scatter plots between numerical variables (e.g., duration, release year) provide insights into their relationships.
- Observations:
  A) Clusters or patterns in scatter plots might indicate trends or groupings in the data.
  B) Histograms on the diagonal show the distribution of each variable.

6.3 Comments for each univariate and bivariate plot

Univariate Plots

1. Histogram of Release Year

Plot Description: A histogram showing the frequency distribution of movie release years.

Comments:

- The histogram highlights how movie production has varied over time.
- Observations: Peaks indicate years with higher movie production, possibly due to industry trends, technological advancements, or cultural shifts.
- The distribution may show an increasing trend in recent years, reflecting the growth of digital platforms and increased content production.

2. Box Plot of Duration Minutes

Plot Description: A box plot displaying the distribution of movie durations in minutes.

Comments:

- The box plot provides insights into the central tendency and spread of movie durations.

Observations: The median duration represents the typical length of movies in the dataset.

- The interquartile range (IQR) indicates the spread of the middle 50% of durations.
- Outliers represent unusually short or long movies, which might be worth investigating for special genres or errors.

3. Count Plot of Ratings

Plot Description: A count plot showing the frequency of different movie ratings.

Comments:

This plot illustrates the distribution of content suitability ratings across the dataset.

Observations:

- Ratings like PG-13 and R might be more prevalent, reflecting the target demographics of the movies.
- Less frequent ratings (e.g., G) suggest fewer movies aimed at very young audiences.
- Understanding rating distribution can help in identifying the predominant audience for the content.

4. Count Plot of Year Bins

Plot Description: A count plot showing the number of movies released in specific time intervals (year bins).

Comments:

- The count plot categorizes movie releases into defined periods, simplifying trend analysis.
- Observations: Periods with higher counts indicate active production phases, while lower counts may reflect less activity.
- Binning helps in understanding how movie production has evolved over distinct time periods.

Bivariate Plots

1. Box Plot of Rating vs. Duration Minutes

Plot Description: A box plot comparing movie ratings and their durations in minutes.

Comments:

This plot reveals the relationship between content ratings and movie lengths.

Observations:

- Different ratings may have varying typical durations, reflecting genre tendencies and audience preferences.
- For instance, longer movies might be more common in certain ratings, while shorter movies could dominate others.

2. Correlation Heatmap (Duration Minutes vs. Release Year)

Plot Description: A heatmap showing the correlation between movie duration and release year.

Comments:

- The heatmap quantifies the linear relationship between these two variables.
- Observations: A positive correlation might suggest that movie lengths have increased over time.
- A negative or weak correlation indicates no significant trend in movie durations with respect to release years.

3. Pair Plot of Duration Minutes and Release Year

Plot Description: Pair plot showing scatter plots and histograms for duration and release year.

Comments:

- The pair plot provides a comprehensive view of the relationship between these numerical variables.
- Observations: Scatter plots may show clusters or trends, while histograms reveal the distribution shape of each variable.
- This visualization helps in identifying potential patterns, correlations, and outliers between movie duration and release year.

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

Business Insights Based on Data Patterns

1. Temporal Trends in Movie Releases

Insight:

- There is a growing trend in movie production in recent years, likely due to the rise of digital streaming platforms and advancements in filming technology.
- The movie industry is becoming more prolific, indicating a competitive market with numerous releases annually.

2. Duration Patterns

Insight:

- Movies have varied durations, catering to different audience preferences and genre requirements.
- Longer durations might indicate epic or detailed storytelling, while shorter ones could cater to quick entertainment needs.
-

3. Content Rating Distribution

Insight:

- The dominance of specific ratings suggests a focus on content suitable for teenage and adult audiences.
- Less frequent ratings like G indicate fewer productions targeted at very young viewers.

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

Actionable Recommendations for Business

Increase Digital Presence:

- Focus on releasing more movies through streaming platforms like Netflix, Amazon Prime, and Disney+. This will help reach a wider audience and adapt to the growing trend of online viewing.

Diversify Movie Lengths:

- Produce a variety of movie lengths to cater to different viewing preferences. Offer shorter films for quick entertainment and longer movies for in-depth storytelling.

Expand Family-Friendly Content:

- Increase the production of G-rated and PG-rated movies to attract families with young children. This is an underrepresented market with potential for growth.

Invest in High-Profile Talent:

- Collaborate with well-known directors and actors to boost the appeal and marketability of new releases. High-profile talent can draw larger audiences and enhance brand reputation.

Explore Emerging Markets:

- Expand production and distribution efforts in emerging markets such as Southeast Asia, Africa, and Latin America. Tailor content to local cultures and preferences to gain a foothold in these regions.

Monitor Viewer Feedback:

- Regularly collect and analyze viewer feedback to understand preferences and improve future productions. Use surveys, social media, and streaming data to gather insights.

Promote Successful Collaborations:

- Highlight movies that involve successful director-actor partnerships. Use these collaborations in marketing campaigns to attract fans and build anticipation.

Align Content with Trends:

- Stay updated with industry trends and audience interests. Produce content that reflects current themes and popular genres to maintain relevance and attract viewers.