Project Report

Netflix Data Cleaning, Analysis and Visualization

jayraj Raulji
Unified Mentor
15/11/24 to 15/12/24

Table of Contents

1. Introduction:-

Netflix is one of the most popular streaming platforms worldwide. This project focuses on cleaning, analyzing, and visualizing Netflix data to find useful insights. By studying the data, we can better understand trends and patterns in Netflix's content and usage.

2. Data Preprocessing:-

Step:-1 Import Required Libraries

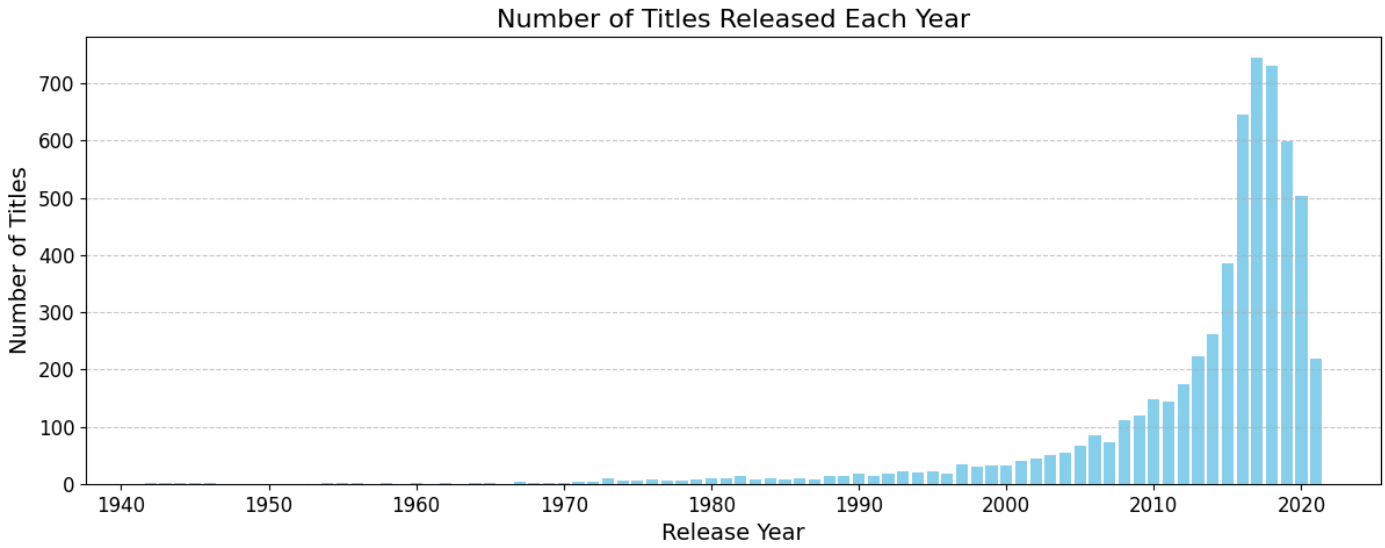
```
import pandas as pd
import matplotlib.pyplot as plt
```

Step 2: Load the Dataset

```
# Load data
df = pd.read_csv('/Users/jayraj/Desktop/study/gitdemo/code-demo/Netflix_dataanalyzing/netflix1.csv')

# Review data
print(df.columns, df.head())
print(df.shape, df.info())
```

```
jayraj@jayrajs-MacBook-Air Netflix_dataanalyzing % /usr/local/bin/python3 /Users/jayraj/Desktop/study/gitdemo/code-demo/Netflix_dataanalyzing/netflix.py
Index(['show_id', 'type', 'title', 'director', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in'],
      dtype='object')   show_id     type                          title         director        country date_added  release_year rating  duration
ed_in
0      s1    Movie              Dick Johnson Is Dead  Kirsten Johnson  United States  9/25/2021          2020  PG-13    90 min                                       Documentaries
1      s3  TV Show                         Ganglands  Julien Leclercq         France  9/24/2021          2021  TV-MA  1 Season  Crime TV Shows, International TV Shows, TV Act...
2      s6  TV Show                      Midnight Mass    Mike Flanagan  United States  9/24/2021          2021  TV-MA  1 Season                  TV Dramas, TV Horror, TV Mysteries
3     s14    Movie  Confessions of an Invisible Girl    Bruno Garotti         Brazil  9/22/2021          2021  TV-PG    91 min                  Children & Family Movies, Comedies
4      s8    Movie                           Sankofa     Haile Gerima  United States  9/24/2021          1993  TV-MA   125 min   Dramas, Independent Movies, International Movies
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8790 entries, 0 to 8789
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8790 non-null   object
 1   type          8790 non-null   object
 2   title         8790 non-null   object
 3   director      8790 non-null   object
 4   country       8790 non-null   object
 5   date_added    8790 non-null   object
 6   release_year  8790 non-null   int64
 7   rating        8790 non-null   object
 8   duration      8790 non-null   object
 9   listed_in     8790 non-null   object
dtypes: int64(1), object(9)
memory usage: 686.8+ KB
(8790, 10) None
show_id         0
type            0
title           0
director        0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
dtype: int64
```

```
# Count and sort release years
release_year_counts =
df['release_year'].value_counts().sort_index()
plt.figure(figsize=(15, 5))
plt.bar(release_year_counts.index,
release_year_counts.values, color='skyblue')
plt.title('Number of Titles Released Each Year',
fontsize=16)
plt.xlabel('Release Year', fontsize=14)
plt.ylabel('Number of Titles', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



Step 3: Data Cleaning

```
# Clean and format data
# Replace 'Not Given' with NaN and drop rows with critical
missing information
df.replace({'Not Given': None}, inplace=True)
df.dropna(subset=['director', 'country', 'duration'],
inplace=True)

# Check for missing values
print(df.isnull().sum())

# Drop duplicates

df.drop_duplicates(inplace=True)
# Convert 'date_added' to datetime
```

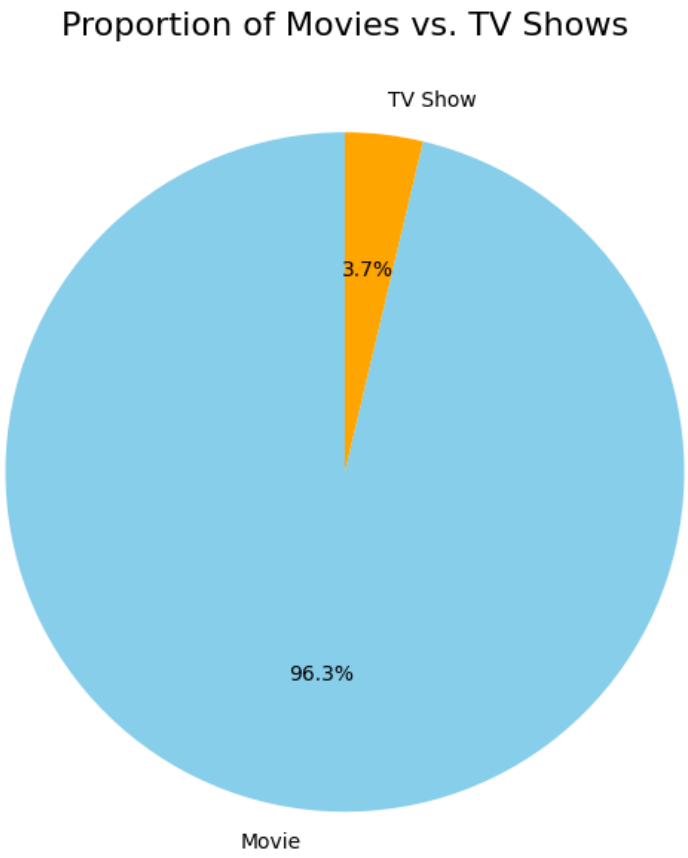
```
df['date_added'] = pd.to_datetime(df['date_added'],
errors='coerce')

# Extract 'duration_value' and 'duration_unit' using raw
strings
df['duration_value'] =
df['duration'].str.extract(r'(\d+)').astype(float)
df['duration_unit'] = df['duration'].str.extract(r'([a-zA-Z]
+)')

# Analyze data
print(df['director'].value_counts())
print(df['country'].value_counts())
print(df['type'].value_counts())
```

```
director
Rajiv Chilaka              20
Alastair Fothergill        18
Raúl Campos, Jan Suter     18
Marcus Raboy               16
Suhas Kadav                16
                           ..
Wagner de Assis             1
White Trash Tyler           1
Yeung Yat-Tak               1
Rai Yuvraj Bains            1
Mozez Singh                 1
Name: count, Length: 4286, dtype: int64
country
United States     2401
India              975
United Kingdom     406
Canada             189
France             156
                   ...
Zimbabwe             1
Mozambique           1
Namibia              1
Mauritius            1
Croatia              1
Name: count, Length: 78, dtype: int64
type
Movie      5696
TV Show     219
```

Step 4: Exploratory Data Analysis (EDA)

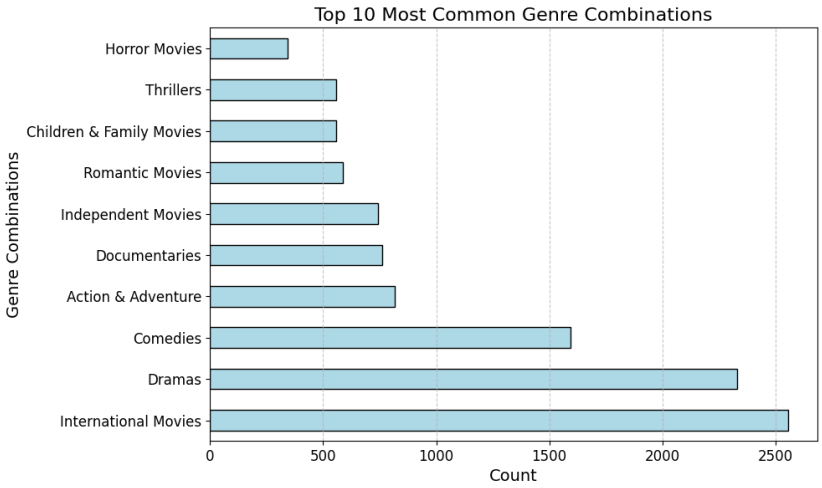
1.Content Type Distribution (Movies vs. TV Shows)

```
# Compare Movies and TV Shows
type_counts = df['type'].value_counts()
type_counts.plot(kind='pie', autopct='%1.1f%%',
startangle=90, colors=['skyblue', 'orange'], figsize=(8, 8))
plt.title('Proportion of Movies vs. TV Shows', fontsize=16)
plt.ylabel('')
plt.show()
```



2. Most Common Genres

```
# Analyze popular genres
genre_counts =
df['listed_in'].str.split(',').explode().str.strip().value_c
ounts()
genre_counts.head(10).plot(kind='barh', figsize=(10, 6),
color='lightblue', edgecolor='black')
plt.title('Top 10 Popular Genres', fontsize=16)
plt.xlabel('Count', fontsize=14)
plt.ylabel('Genres', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

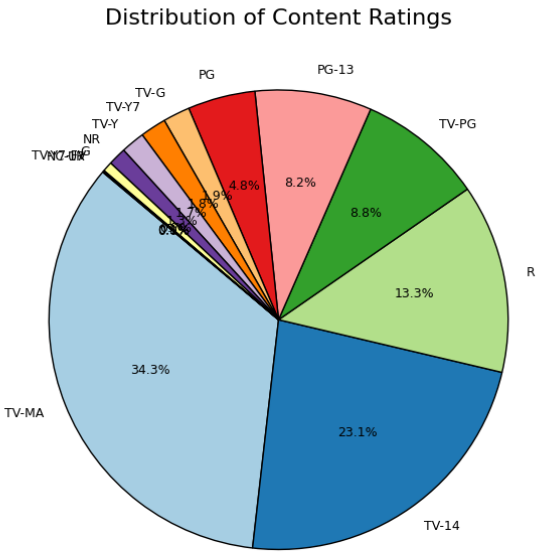


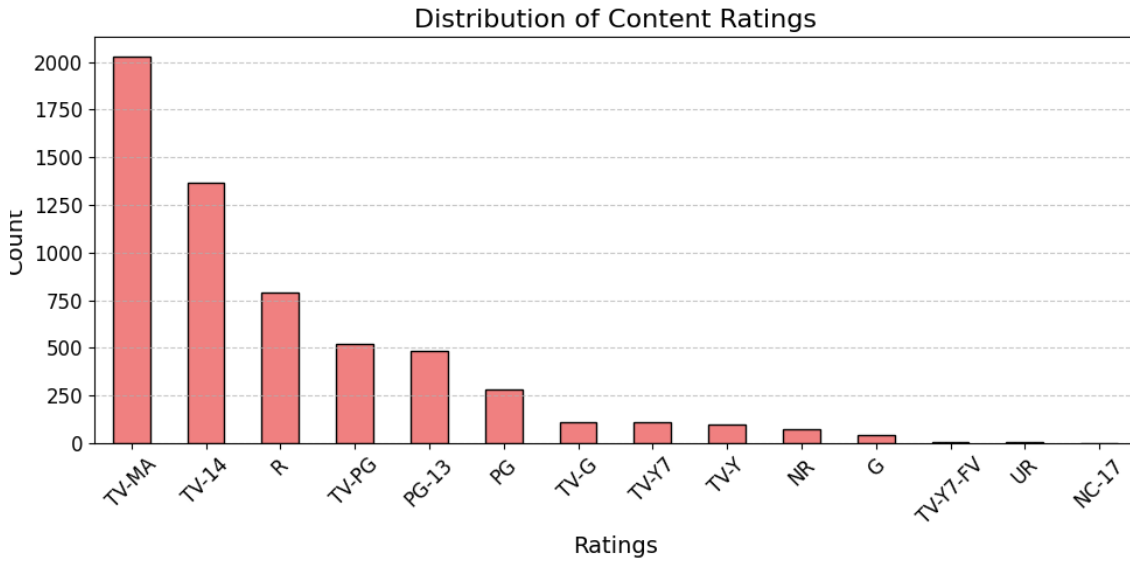
3.Visual representation of rating frequency of movies and TV Shows on Netflix

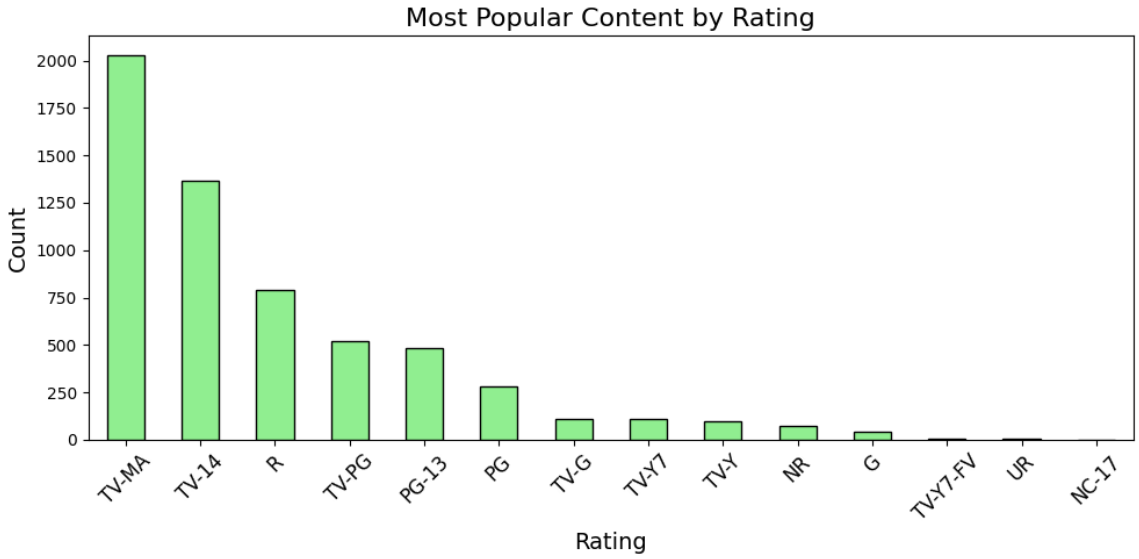
```
# Analyze Ratings Distribution
rating_counts = df[‘rating'].value_counts()
print(rating_counts)
rating_counts.plot(kind='bar', figsize=(10, 5),
color='lightcoral', edgecolor='black')
plt.title('Distribution of Content Ratings', fontsize=16)
plt.xlabel('Ratings', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.xticks(fontsize=12, rotation=45)
plt.yticks(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

```
rating_counts.plot.pie(
    figsize=(10, 10), autopct='%1.1f%%', startangle=140,
colors=plt.cm.Paired.colors,
    wedgeprops={'edgecolor': 'black'},
textprops={'fontsize': 9}
)
plt.title('Distribution of Content Ratings', fontsize=16)
plt.ylabel('')
plt.show()
```

rating	
TV-MA	2029
TV-14	1368
R	787
TV-PG	521
PG-13	486
PG	281
TV-G	112
TV-Y7	108
TV-Y	99
NR	75
G	41
TV-Y7-FV	3
UR	3
NC-17	2





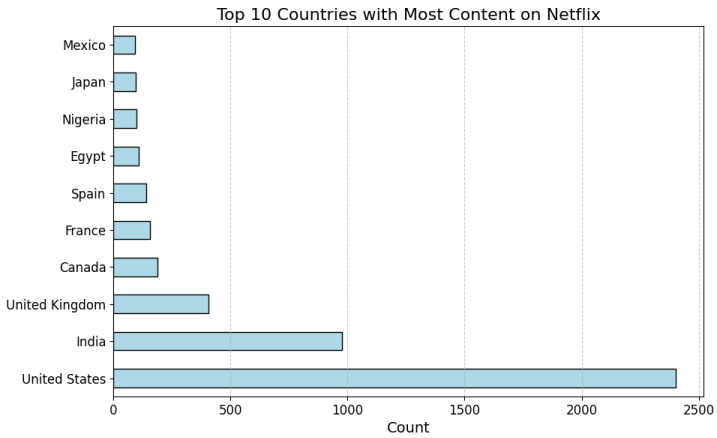



```
#Converting date_added column to datetime.
df['date_added']=pd.to_datetime(df['date_added'])
print(df.describe())
```

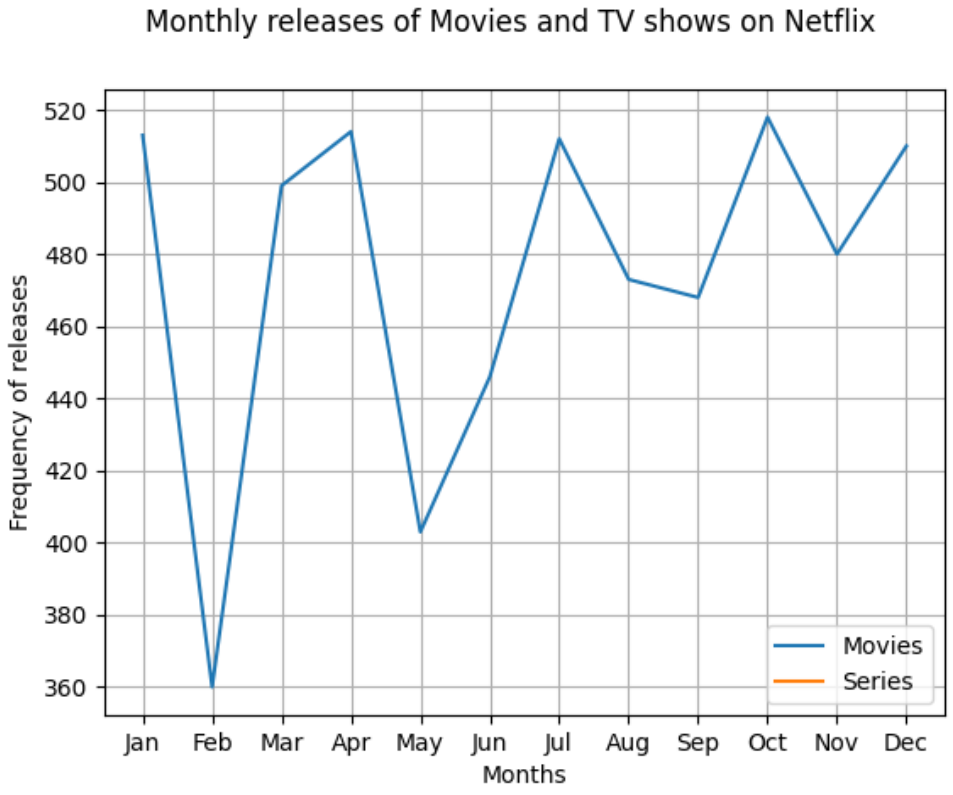
```
                      date_added  release_year  duration_value   year_added
count                       5915   5915.000000     5915.000000  5915.000000
mean   2019-04-25 03:18:24.649197056   2013.014708       97.127473  2018.817244
min              2008-01-01 00:00:00   1942.000000        1.000000  2008.000000
25%              2018-03-30 12:00:00   2012.000000       86.000000  2018.000000
50%              2019-06-12 00:00:00   2016.000000       98.000000  2019.000000
75%              2020-07-03 00:00:00   2018.000000      114.000000  2020.000000
max              2021-09-25 00:00:00   2021.000000      253.000000  2021.000000
std                              NaN      9.693770       32.276934     1.557241
```

```
country_counts = df['country'].value_counts().head(10)
print(country_counts)
country_counts.plot(kind='barh', figsize=(10, 6),
color='lightblue', edgecolor='black')
plt.title('Top 10 Countries with Most Content on Netflix',
fontsize=16)
plt.xlabel('Count', fontsize=14)
plt.ylabel('Countries', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

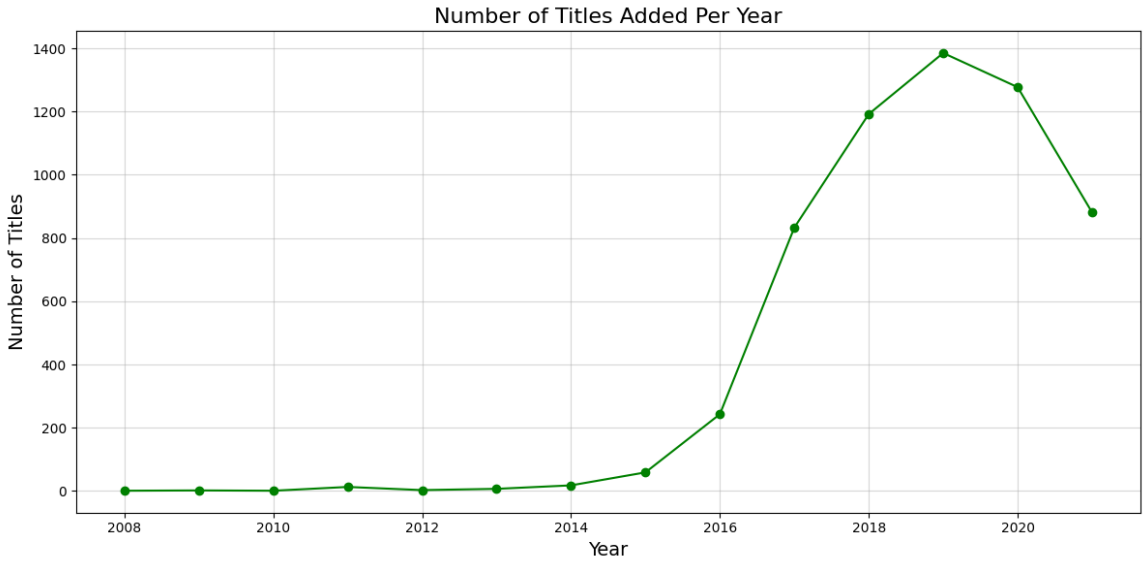
```
country
United States     2401
India              975
United Kingdom     406
Canada             189
France             156
Spain              140
Egypt              107
Nigeria            100
Japan               96
Mexico              93
Name: count, dtype: int64
```



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

#Monthly releases of Movies and TV shows on Netflix
monthly_movie_release=df[df['type']=='Movie']
['month'].value_counts().sort_index()
monthly_series_release=df[df['type']=='TVShow']
['month'].value_counts().sort_index()
plt.plot(monthly_movie_release.index,
monthly_movie_release.values, label='Movies')
plt.plot(monthly_series_release.index,
monthly_series_release.values,label='Series')
plt.xlabel("Months")
plt.ylabel("Frequency of releases")
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May',
'Jun', 'Jul', 'Aug',
'Sep', 'Oct', 'Nov', 'Dec'])
plt.legend()
plt.grid(True)
plt.suptitle("Monthly releases of Movies and TV shows on
Netflix")
plt.show()
```

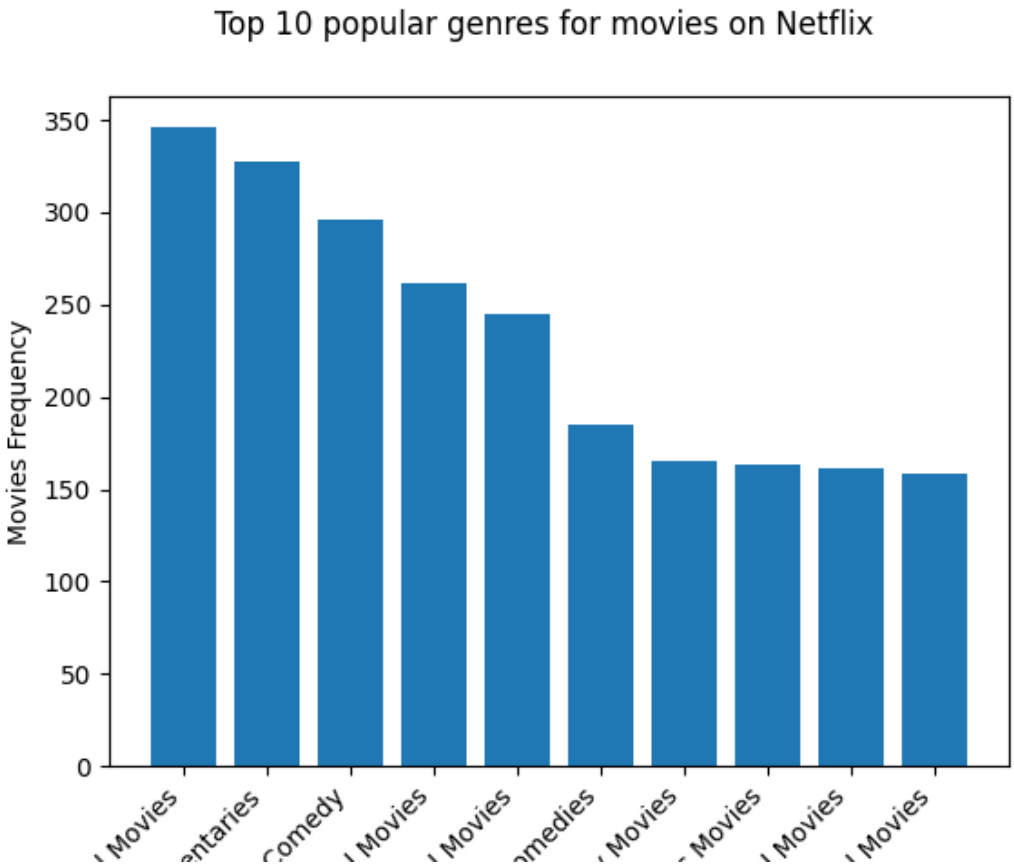


```
# Analyze Content Added Over Time
df['year_added'] = df['date_added'].dt.year
content_added_per_year =
df['year_added'].value_counts().sort_index()

content_added_per_year.plot(kind='line', marker='o',
figsize=(12, 6), color='green')
plt.title('Number of Titles Added Per Year', fontsize=16)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Number of Titles', fontsize=14)
plt.grid(alpha=0.5)
plt.tight_layout()
plt.show()
```



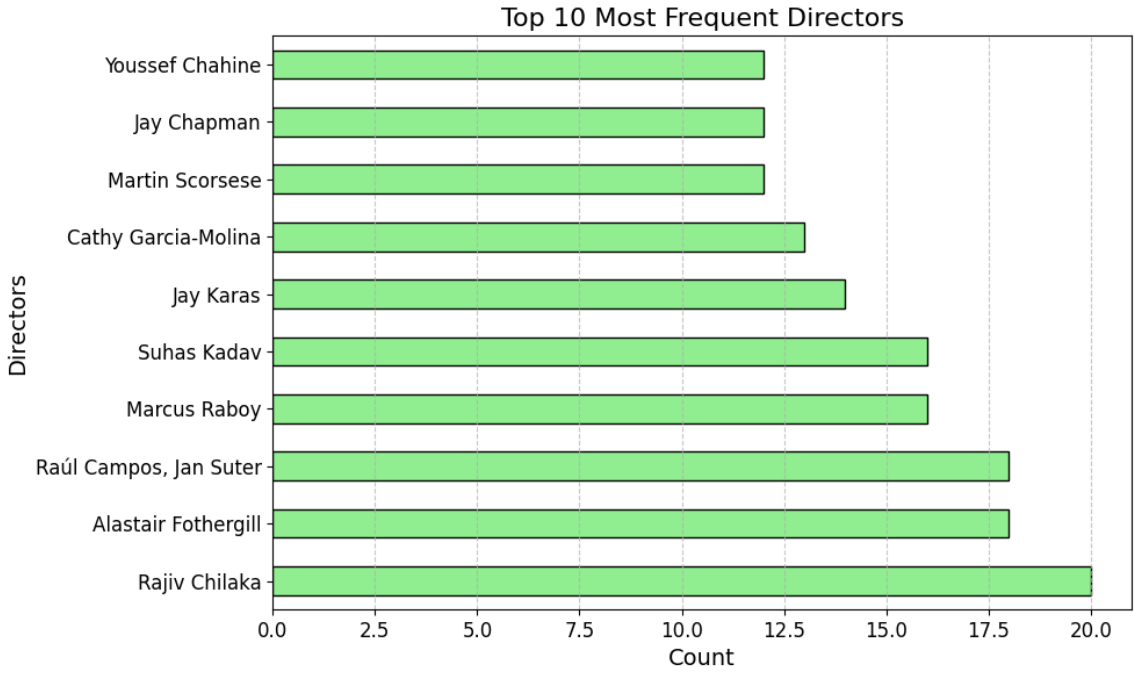

```
#Top 10 popular movie genres
popular_movie_genre=df[df['type']=='Movie'].groupby("listed_
in").size().sort_values(ascending=False)[:10]
popular_series_genre=df[df['type']=='TVShow'].groupby("liste
d_in").size().sort_values(ascending=False)[:10]
plt.bar(popular_movie_genre.index,
popular_movie_genre.values)
plt.xticks(rotation=45, ha='right')
plt.xlabel("Genres")
plt.ylabel("Movies Frequency")
plt.suptitle("Top 10 popular genres for movies on Netflix")
plt.show()
```

```
Dramas, International Movies                          346
Documentaries                                         328
Stand-Up Comedy                                       296
Comedies, Dramas, International Movies                262
Dramas, Independent Movies, International Movies      245
Children & Family Movies, Comedies                    185
Children & Family Movies                              165
Dramas, International Movies, Romantic Movies         163
Documentaries, International Movies                   161
Comedies, International Movies                        158
```

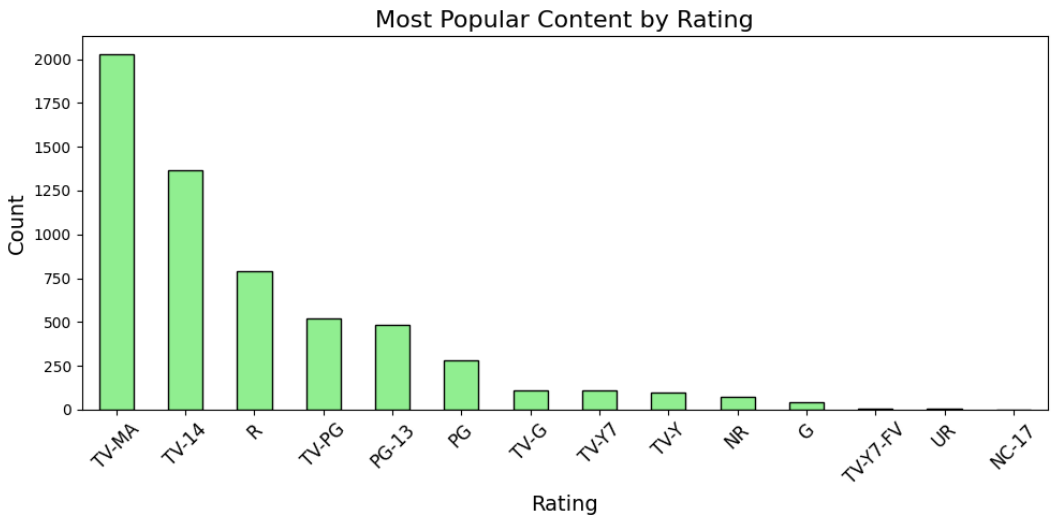



```
# Explore Most Frequent Directors
director_counts = df['director'].value_counts().head(10)
print(director_counts)
director_counts.plot(kind='barh', figsize=(10, 6),
color='lightgreen', edgecolor='black')
plt.title('Top 10 Most Frequent Directors', fontsize=16)
plt.xlabel('Count', fontsize=14)
plt.ylabel('Directors', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

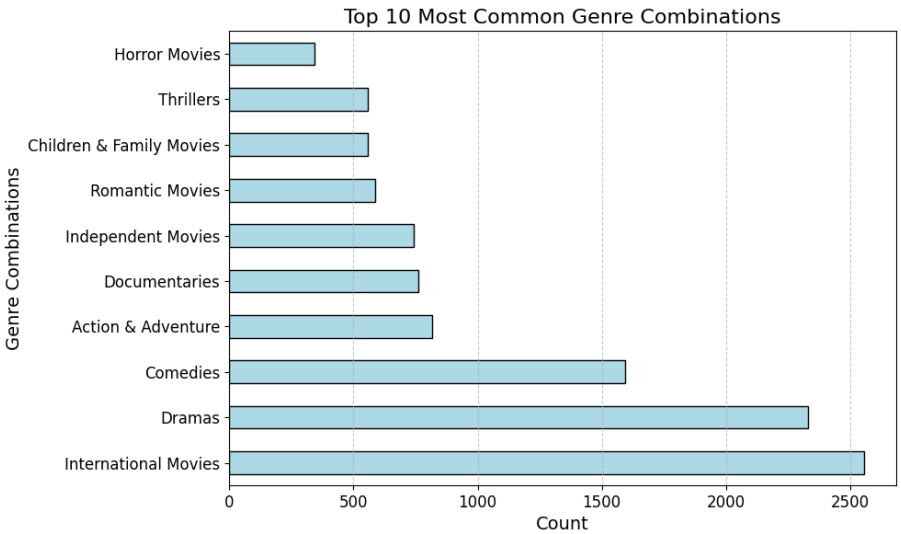
```
director
Rajiv Chilaka              20
Alastair Fothergill        18
Raúl Campos, Jan Suter     18
Marcus Raboy               16
Suhas Kadav                16
Jay Karas                  14
Cathy Garcia-Molina        13
Martin Scorsese            12
Jay Chapman                12
Youssef Chahine            12
```




```
# Identify the Most Popular Content by Rating
popular_content_by_rating = df.groupby('rating')
['title'].count().sort_values(ascending=False)
popular_content_by_rating.plot(kind='bar', figsize=(10, 5),
color='lightgreen', edgecolor='black')
plt.title('Most Popular Content by Rating', fontsize=16)
plt.xlabel('Rating', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.xticks(fontsize=12, rotation=45)
plt.tight_layout()
plt.show()
```




```
# Find Most Common Combinations of Genres
genre_combinations =
df['listed_in'].str.split(',').explode().str.strip()
genre_combinations_counts =
genre_combinations.value_counts().head(10)
genre_combinations_counts.plot(kind='barh', figsize=(10, 6),
color='lightblue', edgecolor='black')
plt.title('Top 10 Most Common Genre Combinations',
fontsize=16)
plt.xlabel('Count', fontsize=14)
plt.ylabel('Genre Combinations', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



```
# Analyze Average Duration for Movies and TV Shows
movie_duration = df[df['type'] == 'Movie']
['duration_value'].mean()
tv_show_seasons = df[df['type'] == 'TV Show']
['duration_value'].mean()
print(f"Average Movie Duration: {movie_duration:.2f}
minutes")
print(f"Average TV Show Seasons: {tv_show_seasons:.2f}
seasons")
```

```
Average Movie Duration: 100.80 minutes
Average TV Show Seasons: 1.68 seasons
```

Step 5: Conclusion and Insights

In this project, we:

1. Cleaned the data by handling missing values, removing duplicates, and converting data types.

2. Explored the data through various visualizations such as bar plots and word clouds.

3. Analyzed content trends over time, identified popular genres, and highlighted top directors.

Github - Link :-

https://github.com/Jayraj2201/code-demo/tree/cd45b1c720b8d4a610771fabf3e743b2e721131f/Netflix_dataanalyzing