Netflix Case Study

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

```
netflix_data = pd.read_csv('netflix.csv')
```

```
netflix_data
```

	show_id	type	title	director	cast	country	date_added	release_ye
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	20
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	20
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	20
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	20
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	20
...	...	...	...	...	...	...	...	
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	20
8803	s8804	TV	Zombie	NaN	NaN	NaN	July 1, 2019	20

Next steps: Generate code with `netflix_data` View recommended plots New interactive sheet

1. (a)

1. Un-nesting the columns a. Un-nest the columns those have cells with multiple comma separated values by creating multiple rows

```
def unnest_column(df, col):
    df = df.dropna(subset=[col]).copy()
    df[col] = df[col].str.split(', ')
    return df.explode(col).reset_index(drop=True)


netflix_data = pd.read_csv('netflix.csv')


columns_to_unnest = ['director', 'cast', 'country', 'listed_in']


for col in columns_to_unnest:
    netflix_data = unnest_column(netflix_data, col)
netflix_data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rat:
0	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba	United States	September 24, 2021	1993	TV-
1	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba	United States	September 24, 2021	1993	TV-

```
netflix_data.shape
```

```
(143102, 12)
```

2. (a) 2. Handling null values a. For categorical variables with null values, update those rows as unknown_column_name. Example : Replace missing value with Unknown Actor for missing value in Actors column.

```
def fill_missing_values(df, columns):
    for col in columns:
        unknown_placeholder = f"Unknown {col.capitalize()}"
        df[col].fillna(unknown_placeholder, inplace=True)
    return df


categorical_columns = ['director', 'cast', 'country', 'rating']
unnested_netflix_data_fixed = netflix_data.copy()
for col in columns_to_unnest:
    unnested_netflix_data_fixed = unnest_column(unnested_netflix_data_fixed, col)
unnested_netflix_data_fixed = fill_missing_values(unnested_netflix_data_fixed, categorica


unnested_netflix_data_fixed.isnull().sum(), unnested_netflix_data_fixed.head()
```

```
(show_id        0
 type           0
 title          0
 director       0
 cast           0
 country        0
 date_added     0
 release_year   0
 rating         0
 duration       3
 listed_in      0
 description    0
 dtype: int64,
   show_id   type    title      director          cast        country  \
 0      s8  Movie  Sankofa  Haile Gerima  Kofi Ghanaba  United States
 1      s8  Movie  Sankofa  Haile Gerima  Kofi Ghanaba  United States
 2      s8  Movie  Sankofa  Haile Gerima  Kofi Ghanaba  United States
 3      s8  Movie  Sankofa  Haile Gerima  Kofi Ghanaba          Ghana
 4      s8  Movie  Sankofa  Haile Gerima  Kofi Ghanaba          Ghana

            date_added  release_year rating duration             listed_in  \
 0  September 24, 2021          1993  TV-MA  125 min                Dramas
 1  September 24, 2021          1993  TV-MA  125 min    Independent Movies
 2  September 24, 2021          1993  TV-MA  125 min  International Movies
 3  September 24, 2021          1993  TV-MA  125 min                Dramas
 4  September 24, 2021          1993  TV-MA  125 min    Independent Movies

                                          description
 0  On a photo shoot in Ghana, an American model s...
 1  On a photo shoot in Ghana, an American model s...
 2  On a photo shoot in Ghana, an American model s...
 3  On a photo shoot in Ghana, an American model s...
 4  On a photo shoot in Ghana, an American model s...  )
```

```
def fill_missing_numerical(df, columns):
    for col in columns:
        df[col].fillna(0, inplace=True)
    return df
numerical_columns = netflix_data.select_dtypes(include=['float64', 'int64']).columns

netflix_data = fill_missing_numerical(netflix_data, numerical_columns)

netflix_data.isnull().sum()
netflix_data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rat
0	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba	United States	September 24, 2021	1993	TV-
1	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba	United States	September 24, 2021	1993	TV-

What does 'good' look like?

1. Find the counts of each categorical variable both using graphical and non- graphical analysis.

a. For Non-graphical Analysis:

```
categorical_columns = ['type', 'director', 'cast', 'country', 'rating', 'listed_in']

for col in categorical_columns:
    print(f"\nValue counts for column: {col}")
    print(netflix_data[col].value_counts())
```

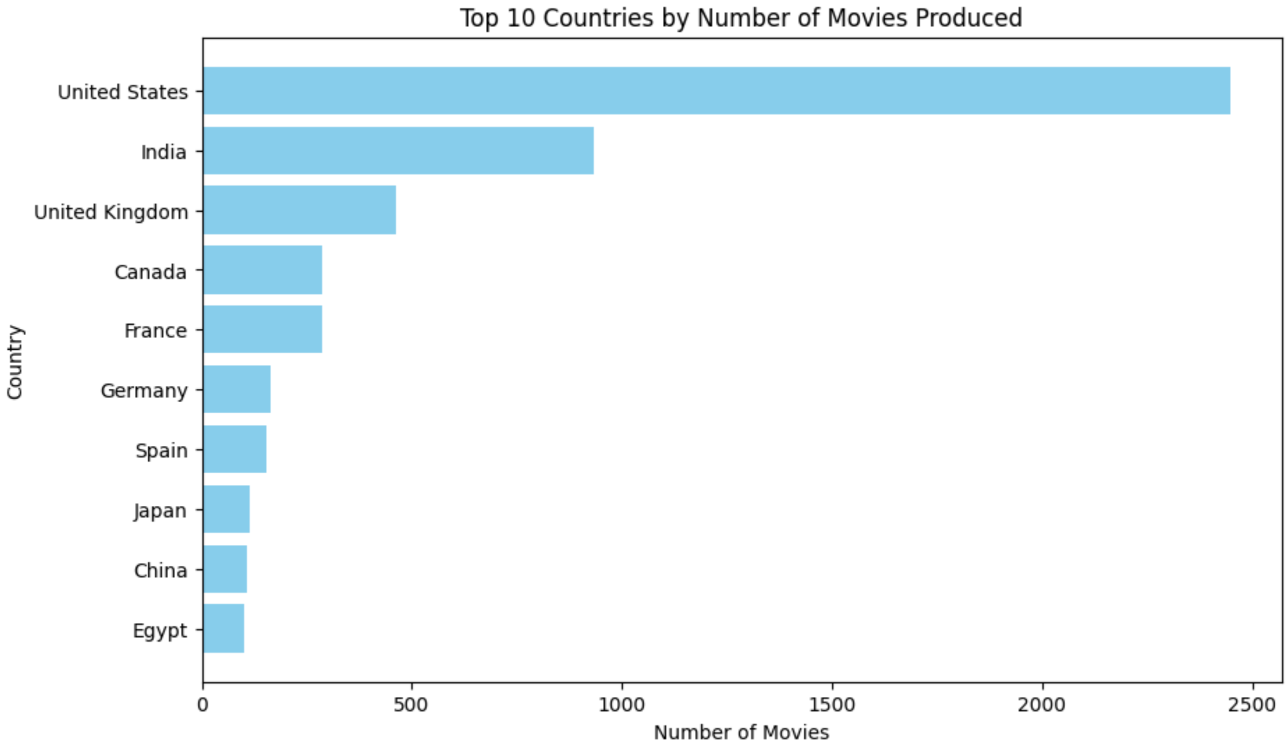
```
Music & Musicals                 2717
Documentaries                    1492
International TV Shows           1450
Classic Movies                   1407
Sports Movies                    1389
Cult Movies                      1071
TV Dramas                        1023
Anime Features                    934
LGBTQ Movies                      769
Faith & Spirituality              699
Crime TV Shows                    547
Stand-Up Comedy                   443
TV Action & Adventure             308
TV Shows                          286
TV Comedies                       265
Kids' TV                          236
Romantic TV Shows                 232
Movies                            231
British TV Shows                  226
Spanish-Language TV Shows         174
Anime Series                      128
TV Mysteries                      123
TV Horror                         119
Korean TV Shows                   111
Docuseries                         87
TV Thrillers                       78
TV Sci-Fi & Fantasy                51
Teen TV Shows                      48
Classic & Cult TV                  29
Stand-Up Comedy & Talk Shows       25
Reality TV                          7
Science & Nature TV                 7
Name: count, dtype: int64
```

2. Comparison of tv shows vs. movies.

a. Find the number of movies produced in each country and pick the top 10 countries.

```
movies_data = netflix_data[netflix_data['type'] == 'Movie']

movies_per_country = movies_data.groupby('country')['title'].nunique().reset_index()

top_10_countries_movies = movies_per_country.sort_values(by='title', ascending=False).hea

top_10_countries_movies.columns = ['Country', 'Number of Movies']

print(top_10_countries_movies)

plt.figure(figsize=(10, 6))
plt.barh(top_10_countries_movies['Country'], top_10_countries_movies['Number of Movies'],
plt.gca().invert_yaxis()
plt.title('Top 10 Countries by Number of Movies Produced')
plt.xlabel('Number of Movies')
plt.ylabel('Country')
plt.show()
```

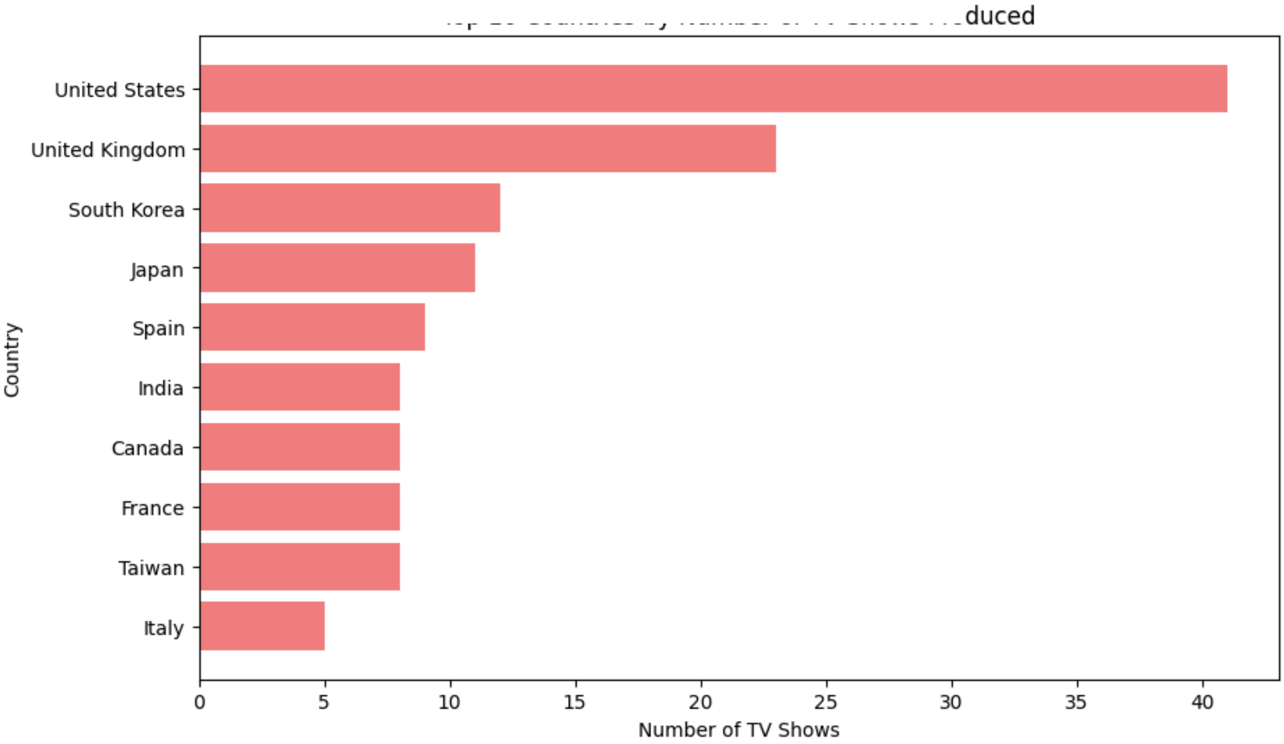
```
            Country  Number of Movies
106   United States              2447
41            India               932
104  United Kingdom               461
18           Canada               287
32           France               285
34          Germany               165
94            Spain               152
49            Japan               113
21            China               107
29            Egypt                99
```



b. Find the number of Tv-Shows produced in each country and pick the top 10 countries.

```
tv_shows_data = netflix_data[netflix_data['type'] == 'TV Show']

tv_shows_per_country = tv_shows_data.groupby('country')['title'].nunique().reset_index()

top_10_countries_tv_shows = tv_shows_per_country.sort_values(by='title', ascending=False)

top_10_countries_tv_shows.columns = ['Country', 'Number of TV Shows']

print(top_10_countries_tv_shows)

plt.figure(figsize=(10, 6))
plt.barh(top_10_countries_tv_shows['Country'], top_10_countries_tv_shows['Number of TV Sh
plt.gca().invert_yaxis()
plt.title('Top 10 Countries by Number of TV Shows Produced')
plt.xlabel('Number of TV Shows')
plt.ylabel('Country')
plt.show()
```

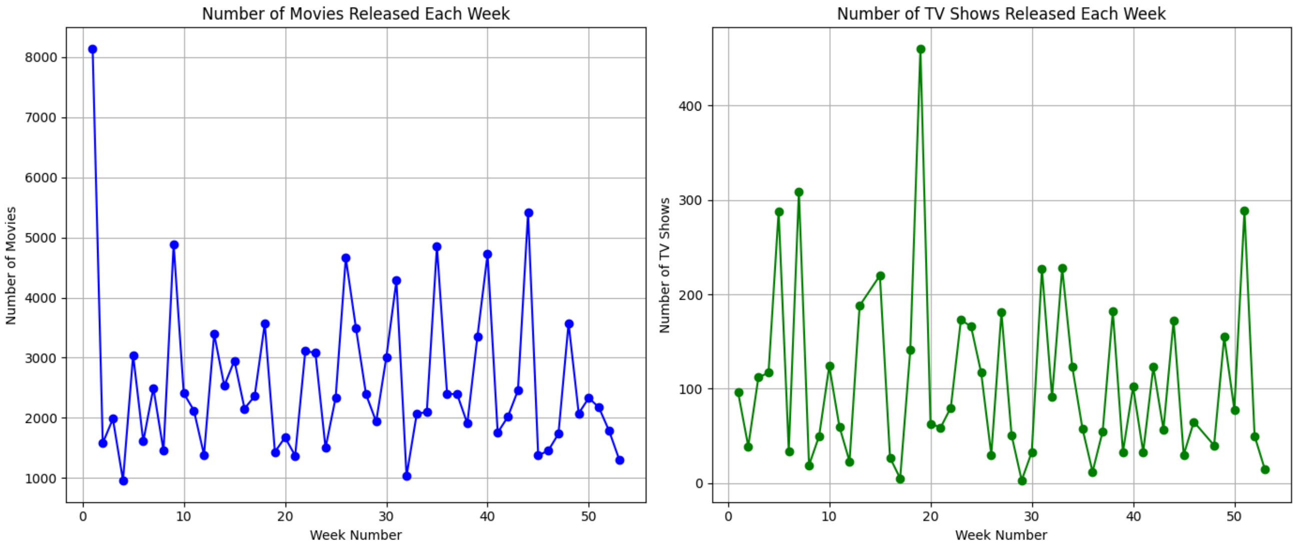
```
           Country  Number of TV Shows
33   United States                  41
32  United Kingdom                  23
27     South Korea                  12
16           Japan                  11
28           Spain                   9
11           India                   8
4           Canada                   8
8           France                   8
29          Taiwan                   8
15           Italy                   5
```



3. What is the best time to launch a TV show?

a. Find which is the best week to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies

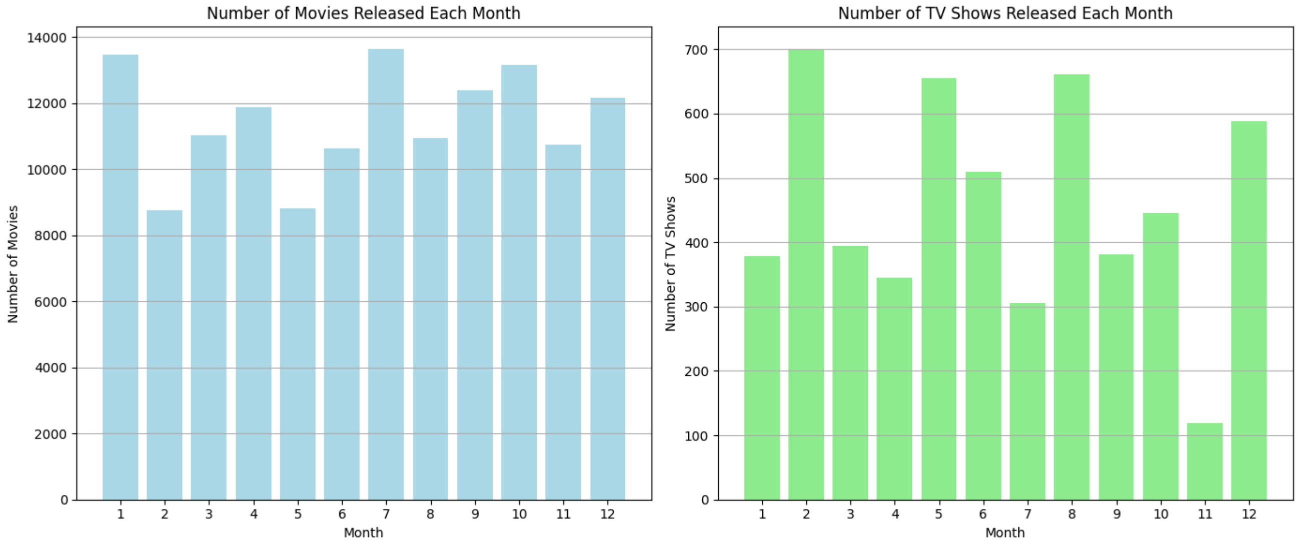
```
netflix_data['date_added'] = pd.to_datetime(netflix_data['date_added'], errors='coerce')

netflix_data['week'] = netflix_data['date_added'].dt.isocalendar().week

movies_data = netflix_data[netflix_data['type'] == 'Movie']
movies_per_week = movies_data.groupby('week')['title'].count().reset_index()
movies_per_week.columns = ['Week', 'Number of Movies']

tv_shows_data = netflix_data[netflix_data['type'] == 'TV Show']
tv_shows_per_week = tv_shows_data.groupby('week')['title'].count().reset_index()
tv_shows_per_week.columns = ['Week', 'Number of TV Shows']

plt.figure(figsize=(14, 6))
plt.subplot(1, 2, 1)
plt.plot(movies_per_week['Week'], movies_per_week['Number of Movies'], marker='o', color=
plt.title('Number of Movies Released Each Week')
plt.xlabel('Week Number')
plt.ylabel('Number of Movies')
plt.grid(True)

plt.subplot(1, 2, 2)
plt.plot(tv_shows_per_week['Week'], tv_shows_per_week['Number of TV Shows'], marker='o',
plt.title('Number of TV Shows Released Each Week')
plt.xlabel('Week Number')
plt.ylabel('Number of TV Shows')
plt.grid(True)

plt.tight_layout()
plt.show()

best_week_movies = movies_per_week.loc[movies_per_week['Number of Movies'].idxmax()]
best_week_tv_shows = tv_shows_per_week.loc[tv_shows_per_week['Number of TV Shows'].idxmax

print(f"The best week to release a movie is Week {best_week_movies['Week']} with {best_we
print(f"The best week to release a TV show is Week {best_week_tv_shows['Week']} with {bes
```



b. Find which is the best month to release the Tv-show or the movie. Do the analysis separately for Tv-shows and Movies

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

netflix_data['month'] = netflix_data['date_added'].dt.month

movies_data = netflix_data[netflix_data['type'] == 'Movie']
movies_per_month = movies_data.groupby('month')['title'].count().reset_index()
movies_per_month.columns = ['Month', 'Number of Movies']

tv_shows_data = netflix_data[netflix_data['type'] == 'TV Show']
tv_shows_per_month = tv_shows_data.groupby('month')['title'].count().reset_index()
tv_shows_per_month.columns = ['Month', 'Number of TV Shows']

plt.figure(figsize=(14, 6))
plt.subplot(1, 2, 1)
plt.bar(movies_per_month['Month'], movies_per_month['Number of Movies'], color='lightblue
plt.title('Number of Movies Released Each Month')
plt.xlabel('Month')
plt.ylabel('Number of Movies')
plt.xticks(range(1, 13))
plt.grid(axis='y')
plt.subplot(1, 2, 2)
plt.bar(tv_shows_per_month['Month'], tv_shows_per_month['Number of TV Shows'], color='lig
plt.title('Number of TV Shows Released Each Month')
plt.xlabel('Month')
plt.ylabel('Number of TV Shows')
plt.xticks(range(1, 13))
plt.grid(axis='y')

plt.tight_layout()
plt.show()

best_month_movies = movies_per_month.loc[movies_per_month['Number of Movies'].idxmax()]
best_month_tv_shows = tv_shows_per_month.loc[tv_shows_per_month['Number of TV Shows'].idx

print(f"The best month to release a movie is {best_month_movies['Month']} with {best_mont
print(f"The best month to release a TV show is {best_month_tv_shows['Month']} with {best_
```

```
The best week to release a movie is Week 1 with 8136 releases.
The best week to release a TV show is Week 19 with 460 releases.
```



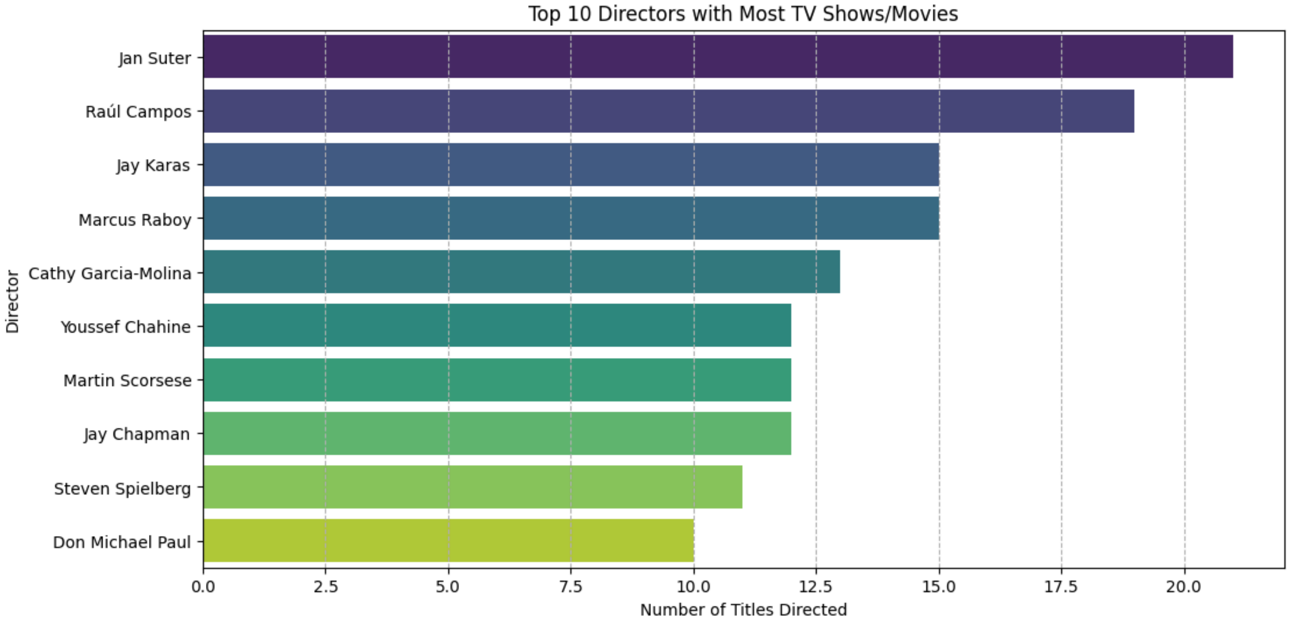
```
The best month to release a movie is 7.0 with 13623.0 releases.
The best month to release a TV show is 2.0 with 700.0 releases.
```

4. Analysis of actors/directors of different types of shows/movies.

a. Identify the top 10 directors who have appeared in most movies or TV shows.

```
directors_data = netflix_data.dropna(subset=['director'])

top_directors = directors_data.groupby('director')['title'].nunique().reset_index()
top_directors.columns = ['Director', 'Number of Titles']

top_10_directors = top_directors.sort_values(by='Number of Titles', ascending=False).head

plt.figure(figsize=(12, 6))
sns.barplot(x='Number of Titles', y='Director', data=top_10_directors, palette='viridis')
plt.title('Top 10 Directors with Most TV Shows/Movies')
plt.xlabel('Number of Titles Directed')
plt.ylabel('Director')
plt.grid(axis='x', linestyle='--')
plt.show()

print(top_10_directors)
```

```
<ipython-input-22-1e79ef5b0bae>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(x='Number of Titles', y='Director', data=top_10_directors, palette='vir
```



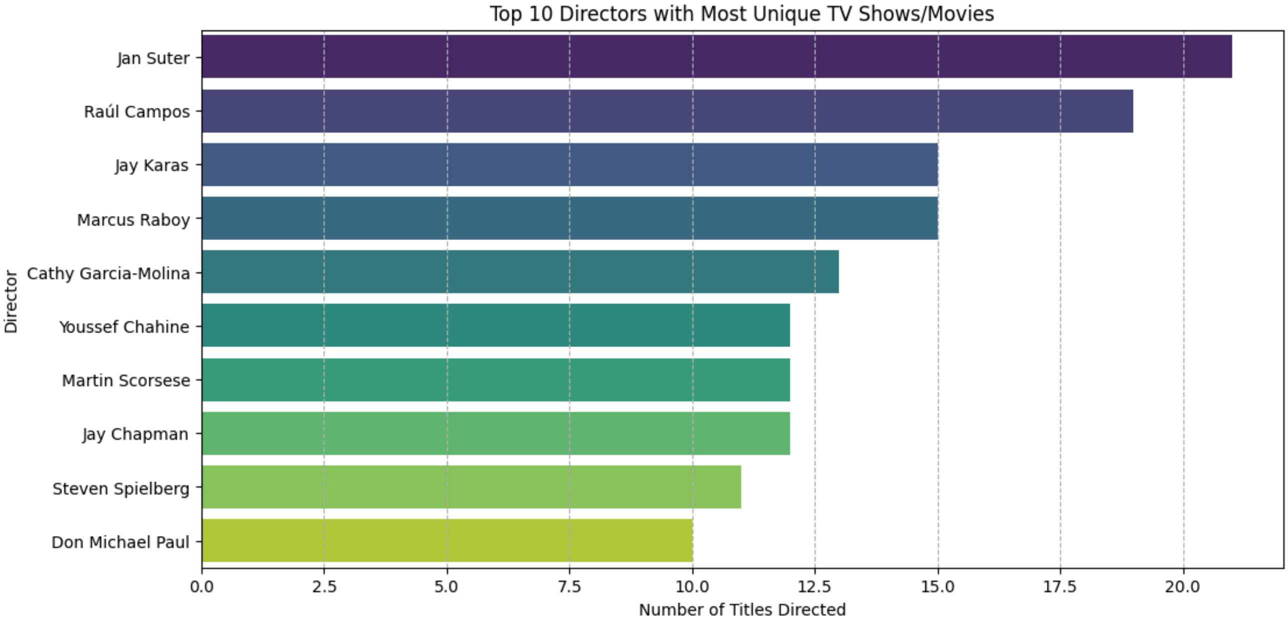
```
                 Director  Number of Titles
1650            Jan Suter                21
3283          Raúl Campos                19
1690            Jay Karas                15
2477         Marcus Raboy                15
656   Cathy Garcia-Molina                13
4263      Youssef Chahine                12
2539      Martin Scorsese                12
1687          Jay Chapman                12
3821     Steven Spielberg                11
1054     Don Michael Paul                10
```

b. Identify the top 10 directors who have appeared in most movies or TV shows.

```
netflix_data['director'] = netflix_data['director'].fillna('')
netflix_data['director'] = netflix_data['director'].apply(lambda x: x.split(', '))

directors_exploded = netflix_data.explode('director')
```

```
directors_count = directors_exploded.groupby('director')['title'].nunique().reset_index()
directors_count.columns = ['Director', 'Number of Titles']

directors_count = directors_count[directors_count['Director'] != '']

top_10_directors = directors_count.sort_values(by='Number of Titles', ascending=False).he

plt.figure(figsize=(12, 6))
sns.barplot(x='Number of Titles', y='Director', data=top_10_directors, palette='viridis')
plt.title('Top 10 Directors with Most Unique TV Shows/Movies')
plt.xlabel('Number of Titles Directed')
plt.ylabel('Director')
plt.grid(axis='x', linestyle='--')
plt.show()

print(top_10_directors)
```

```
<ipython-input-23-85d68d261e86>:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(x='Number of Titles', y='Director', data=top_10_directors, palette='vir
```

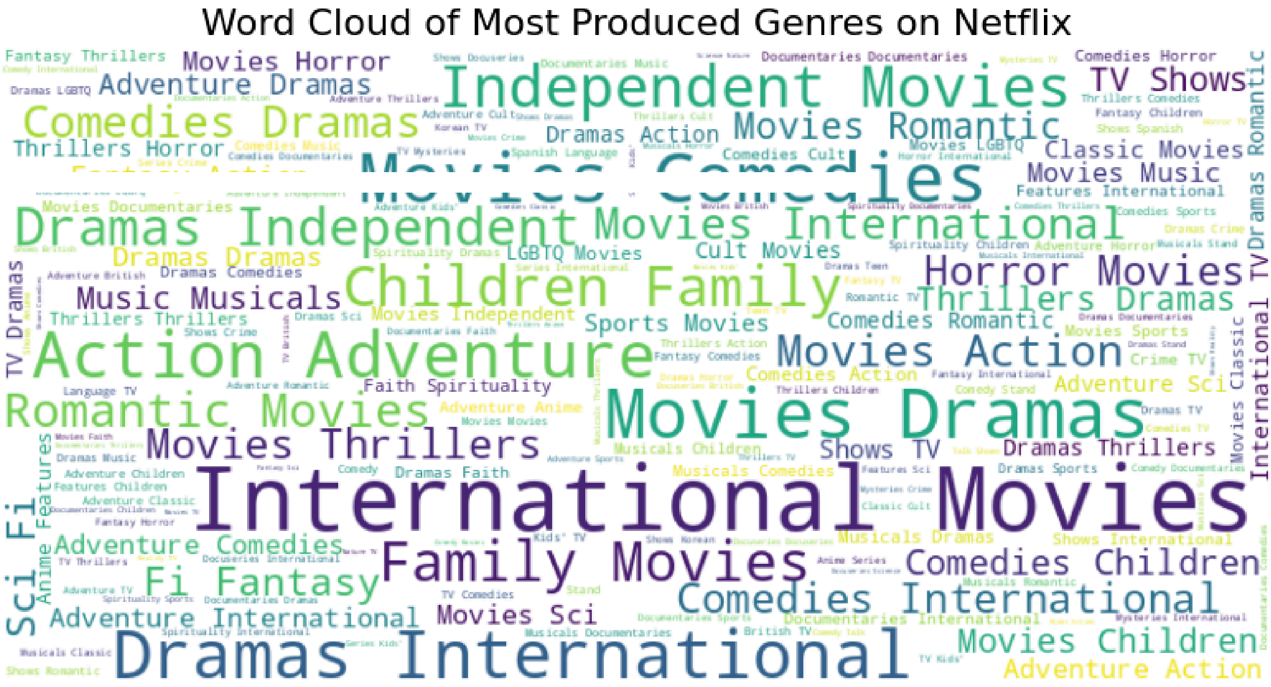


```
                 Director  Number of Titles
1650            Jan Suter                21
3283          Raúl Campos                19
1690            Jay Karas                15
2477         Marcus Raboy                15
656   Cathy Garcia-Molina                13
4263      Youssef Chahine                12
2539      Martin Scorsese                12
1687          Jay Chapman                12
3821     Steven Spielberg                11
1054     Don Michael Paul                10
```

5. Which genre movies are more popular or produced more

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

netflix_data['listed_in'] = netflix_data['listed_in'].fillna('')
```

```
all_genres = ' '.join(netflix_data['listed_in'])

all_genres = all_genres.replace(', ', ' ').replace('  ', ' ')

wordcloud = WordCloud(width=800, height=400, background_color='white', colormap='viridis'

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Most Produced Genres on Netflix', fontsize=16)
plt.show()
```



6. Find after how many days the movie will be added to Netflix after the release of the movie

```
netflix_data['date_added'] = pd.to_datetime(netflix_data['date_added'], errors='coerce')
netflix_data['release_year'] = pd.to_datetime(netflix_data['release_year'], format='%Y',

netflix_data = netflix_data.dropna(subset=['date_added', 'release_year'])
netflix_data['days_to_add'] = (netflix_data['date_added'] - netflix_data['release_year'])

mode_days_to_add = netflix_data['days_to_add'].mode()[0]

print(f"The mode of days after which movies are added to Netflix after their release is:
```

```
The mode of days after which movies are added to Netflix after their release is: 1369
<ipython-input-25-300740c54113>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
    See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
      netflix_data['days_to_add'] = (netflix_data['date_added'] - netflix_data['release_y
```

Identify the top 10 actors who have appeared in most movies or TV shows.

```
netflix_data['cast'] = netflix_data['cast'].fillna('')
```