netflix-analyze-1

April 26, 2024

Notebook's Google Colab Link : https://colab.research.google.com/drive/1psXzCcnB-DTu2rpUzXpao0rs72MmKMix?usp=sharing

[84]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from PIL import Image
from collections import Counter
import math
import plotly.graph_objects as go
import plotly.express as px
```

[85]:
```
netflix = pd.read_csv('/content/drive/MyDrive/Dataset/netflix.csv')
netflix.head()
```

[85]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…	NaN	September 24, 2021	2021	TV-MA	1 Season
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…	India	September 24, 2021	2021	TV-MA	2 Seasons

```
                                              listed_in  \
0                                         Documentaries
1    International TV Shows, TV Dramas, TV Mysteries
2  Crime TV Shows, International TV Shows, TV Act…
3                              Docuseries, Reality TV
4  International TV Shows, Romantic TV Shows, TV …

                                           description
0  As her father nears the end of his life, filmm…
1  After crossing paths at a party, a Cape Town t…
2  To protect his family from a powerful drug lor…
3  Feuds, flirtations and toilet talk go down amo…
4  In a city of coaching centers known to train I…
```

[86]:
```
netflix_overall = netflix.copy()
```

[87]:
```
for i in netflix_overall.columns:
    null_rate = netflix_overall[i].isna().sum() / len(netflix_overall) * 100
    if null_rate > 0 :
        print("{} null rate: {}%".format(i,round(null_rate,2)))
```

```
director null rate: 29.91%
cast null rate: 9.37%
country null rate: 9.44%
date_added null rate: 0.11%
rating null rate: 0.05%
duration null rate: 0.03%
```

[88]:
```
# Replacments

netflix_overall['country'] = netflix_overall['country'].
 ↪fillna(netflix_overall['country'].mode()[0])


netflix_overall['cast'].replace(np.nan, 'No Data',inplace  = True)
netflix_overall['director'].replace(np.nan, 'No Data',inplace  = True)

# Drops

netflix_overall.dropna(inplace=True)

# Drop Duplicates

netflix_overall.drop_duplicates(inplace= True)
```

[89]:
```
netflix_overall.isnull().sum()
```

[89]:
```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

[90]:
```
netflix_overall.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 8790 entries, 0 to 8806
Data columns (total 12 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   cast          8790 non-null   object
 5   country       8790 non-null   object
 6   date_added    8790 non-null   object
 7   release_year  8790 non-null   int64 
 8   rating        8790 non-null   object
 9   duration      8790 non-null   object
 10  listed_in     8790 non-null   object
 11  description   8790 non-null   object
dtypes: int64(1), object(11)
memory usage: 892.7+ KB
```

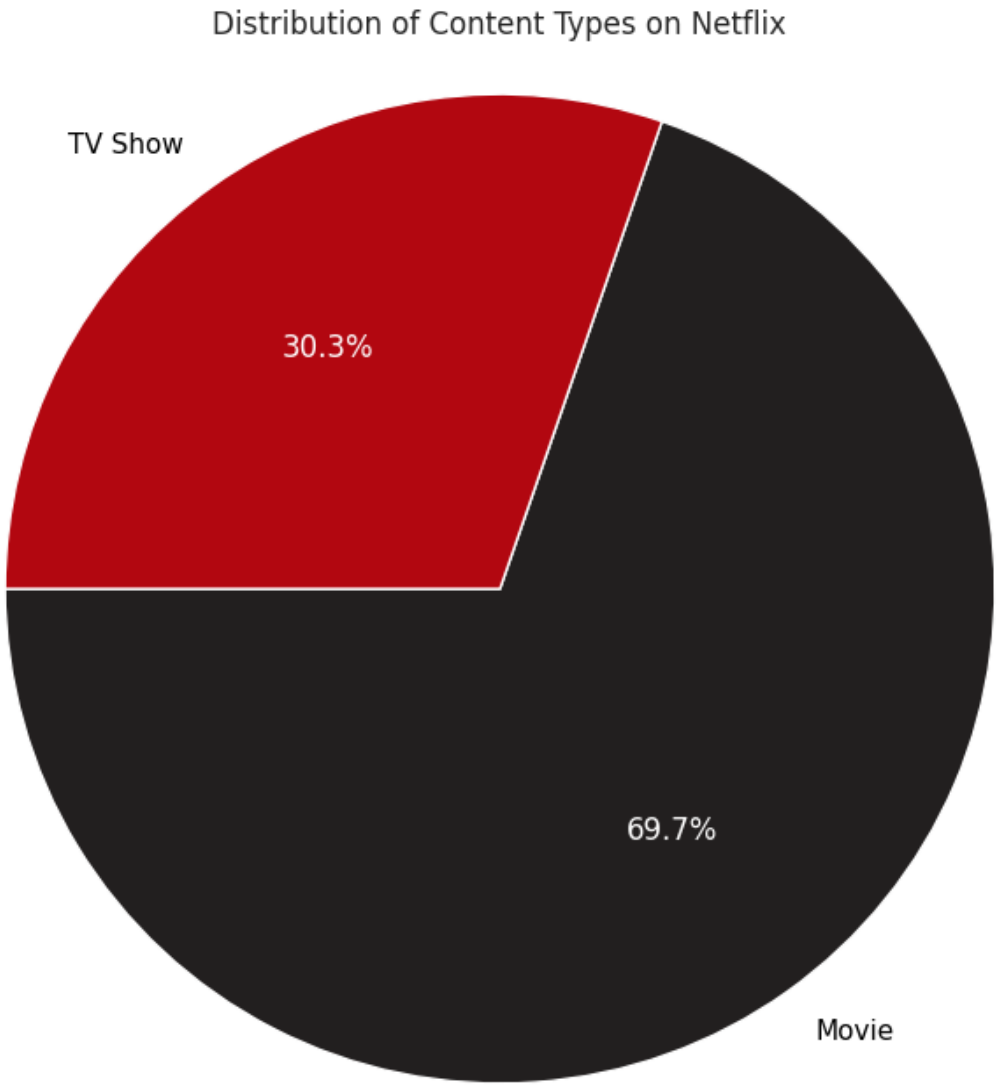
[91]:
```
netflix_shows=netflix_overall[netflix_overall['type']=='TV Show']
netflix_movies=netflix_overall[netflix_overall['type']=='Movie']
```

[92]:
```
custom_palette = ['#221f1f', '#b20710', '#e50914'] #Netflix Palette
```

#Content Distribution Strategy

[93]:
```
sns.set(style="darkgrid")

# Calculate counts for each category
type_counts = netflix_overall['type'].value_counts()
```

```
plt.figure(figsize=(8, 8))
wedges, texts, autotexts = plt.pie(type_counts, labels=type_counts.index,␣
 ↪colors=custom_palette, autopct='%1.1f%%', startangle=180)

for text, autotext in zip(texts, autotexts):
    autotext.set_color('white')  # Text inside pie
    text.set_color('black')      # Text outside pie

plt.title('Distribution of Content Types on Netflix')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



Netflix should continue to focus on producing more movies than TV shows, as the dataset indicates a higher count of movies. However, they should also consider the popularity and demand for TV series, especially in regions like the United States where TV series are highly consumed.

1 Director and Genre Analysis

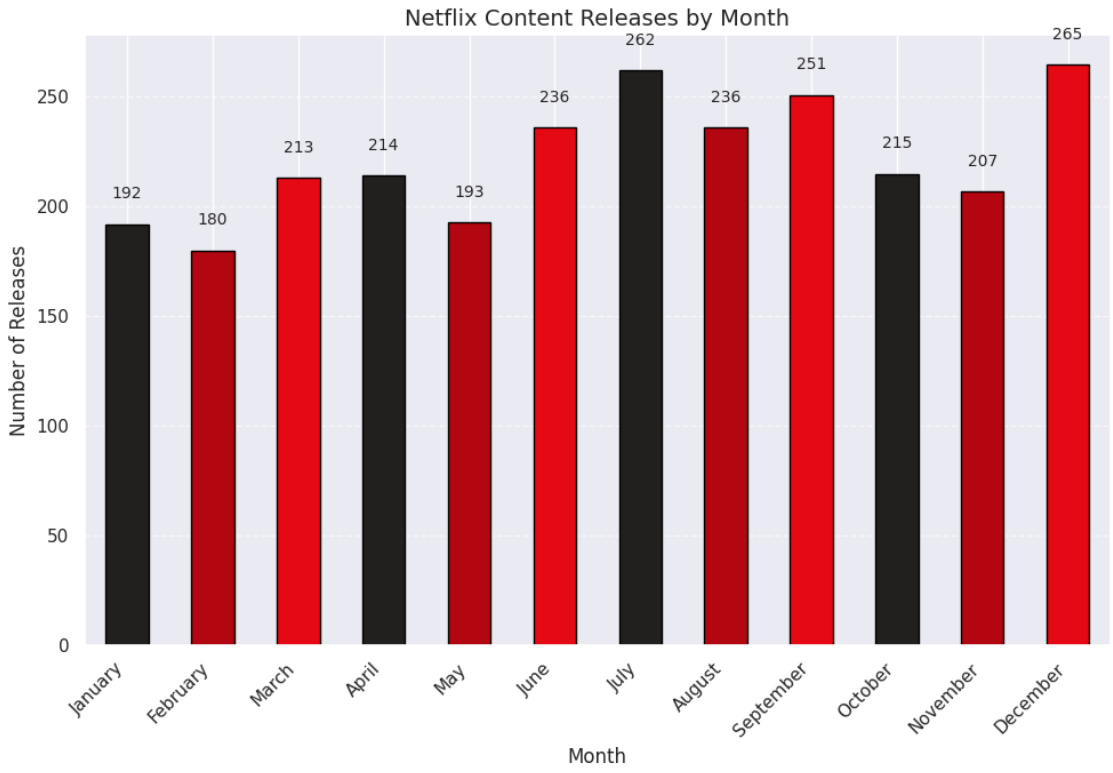
[94]:
```
netflix_fr = netflix_overall.copy()

netflix_fr = netflix_fr.dropna(subset=['director'])

netflix_fr = netflix_fr[netflix_fr['director'] != 'No Data']

# Create the treemap
fig = px.treemap(netflix_fr, path=['director'], color='director',␣
 ↪hover_data=['director', 'title'], color_continuous_scale='Purples')
fig.show()
```

Netflix should consider collaborating with directors like Rajiv Chilaka, known for producing kids cartoons, and Raul Campos, known for Mexican comedy, to cater to specific audience segments. Furthermore, genres like dramas, comedies, and international movies have shown high demand, indicating potential areas for growth.

2 Optimal Release Timing

[95]:
```
netflix_date = netflix_shows[['date_added']].dropna()
netflix_date['year'] = netflix_date['date_added'].apply(lambda x: x.split(',␣
 ↪')[-1])
netflix_date['month'] = netflix_date['date_added'].apply(lambda x: x.lstrip().
 ↪split(' ')[0])

# Defining month order and grouping data
month_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July',␣
 ↪'August', 'September', 'October', 'November', 'December']
df = netflix_date.groupby('month')['year'].count().reindex(month_order)

# Plotting the bar graph with annotations
plt.figure(figsize=(10, 7))
bars = df.plot(kind='bar', color=custom_palette, edgecolor='black')
plt.title('Netflix Content Releases by Month', fontsize=14)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Number of Releases', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Adding annotations
for i, val in enumerate(df):
```

```
    plt.text(i, val + 10, str(val), ha='center', va='bottom', fontsize=10)

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



For producers planning to release content on Netflix, it's advisable to choose months like January or February when the least amount of content is added. This can help new releases gain more visibility and avoid being overshadowed by a flood of new additions.

3 Newest Content On Netflix

[96]:
```
newest_fr_series=netflix_fr.sort_values(by='release_year', ascending=False)[0:
↪20]
newest_fr_series
```

[96]:

	show_id	type	title	\
1370	s1371	Movie	The Dig	
255	s256	Movie	Memories of a Murderer: The Nilsen Tapes	
1189	s1190	Movie	Nate Bargatze: The Greatest Average American	
1188	s1189	Movie	Get the Goat	
1187	s1188	Movie	Deadly Illusions	
363	s364	Movie	The Vault	

1184	s1185	Movie	The Yin Yang Master
683	s684	Movie	A Family
685	s686	Movie	Fatherhood
687	s688	Movie	Jagame Thandhiram
690	s691	Movie	Rurouni Kenshin: The Final
1468	s1469	Movie	What Happened to Mr. Cha?
692	s693	TV Show	The Rational Life
693	s694	Movie	Ali & Ratu Ratu Queens
1173	s1174	TV Show	Men on a Mission
1172	s1173	Movie	Loyiso Gola: Unlearning
1171	s1172	Movie	Seaspiracy
1170	s1171	Movie	Secret Magic Control Agency
707	s708	TV Show	Let's Eat
1167	s1168	Movie	Caught by a Wave

	director	cast
1370	Simon Stone	Carey Mulligan, Ralph Fiennes, Lily James, Joh…
255	Michael Harte	No Data
1189	Troy Miller	Nate Bargatze
1188	Vitor Brandt	Matheus Nachtergaele, Edmilson Filho, Letícia …
1187	Anna Elizabeth James	Kristin Davis, Dermot Mulroney, Greer Grammer,…
363	Jaume Balagueró	Freddie Highmore, Astrid Bergès-Frisbey, Sam R…
1184	Li Weiran	Chen Kun, Zhou Xun, Qu Chuxiao, Shen Yue, Will…
683	Michihito Fujii	Go Ayano, Hiroshi Tachi, Machiko Ono, Yukiya K…
685	Paul Weitz	Kevin Hart, Alfre Woodard, Lil Rel Howery, DeW…
687	Karthik Subbaraj	Dhanush, Aishwarya Lekshmi, James Cosmo, Joju …
690	Keishi Otomo	Takeru Sato, Emi Takei, Mackenyu, Munetaka Aok…
1468	Kim Dong-kyu	Cha In-pyo, Cho Dal-hwan, Song Jae-ryong
692	Hsu Fu-chun	Qin Lan, Dylan Wang, Li Zonghan, Bao Wenjing, …
693	Lucky Kuswandi	Iqbaal Ramadhan, Nirina Zubir, Asri Welas, Tik…
1173	Jung-ah Im	Ho-dong Kang, Soo-geun Lee, Sang-min Lee, Youn…
1172	Kagiso Lediga	Loyiso Gola
1171	Ali Tabrizi	No Data
1170	Aleksey Tsitsilin	Nicholas Corda, Sylvana Joyce, Alyson Leigh Ro…
707	Sarawut Wichiensarn	Apinya Sakuljaroensuk, Mawin Taweephol, Nat Sa…
1167	Massimiliano Camaiti	Elvira Camarrone, Christian Roberto, Donatella…

	country	date_added	release_year	rating	duration
1370	United Kingdom	January 29, 2021	2021	PG-13	113 min
255	United States	August 18, 2021	2021	TV-MA	86 min
1189	United States	March 18, 2021	2021	TV-G	60 min
1188	Brazil	March 18, 2021	2021	TV-14	98 min
1187	United States	March 18, 2021	2021	R	115 min
363	United States	July 31, 2021	2021	R	118 min
1184	China	March 19, 2021	2021	TV-14	120 min
683	United States	June 18, 2021	2021	TV-MA	135 min
685	United States	June 18, 2021	2021	PG-13	111 min

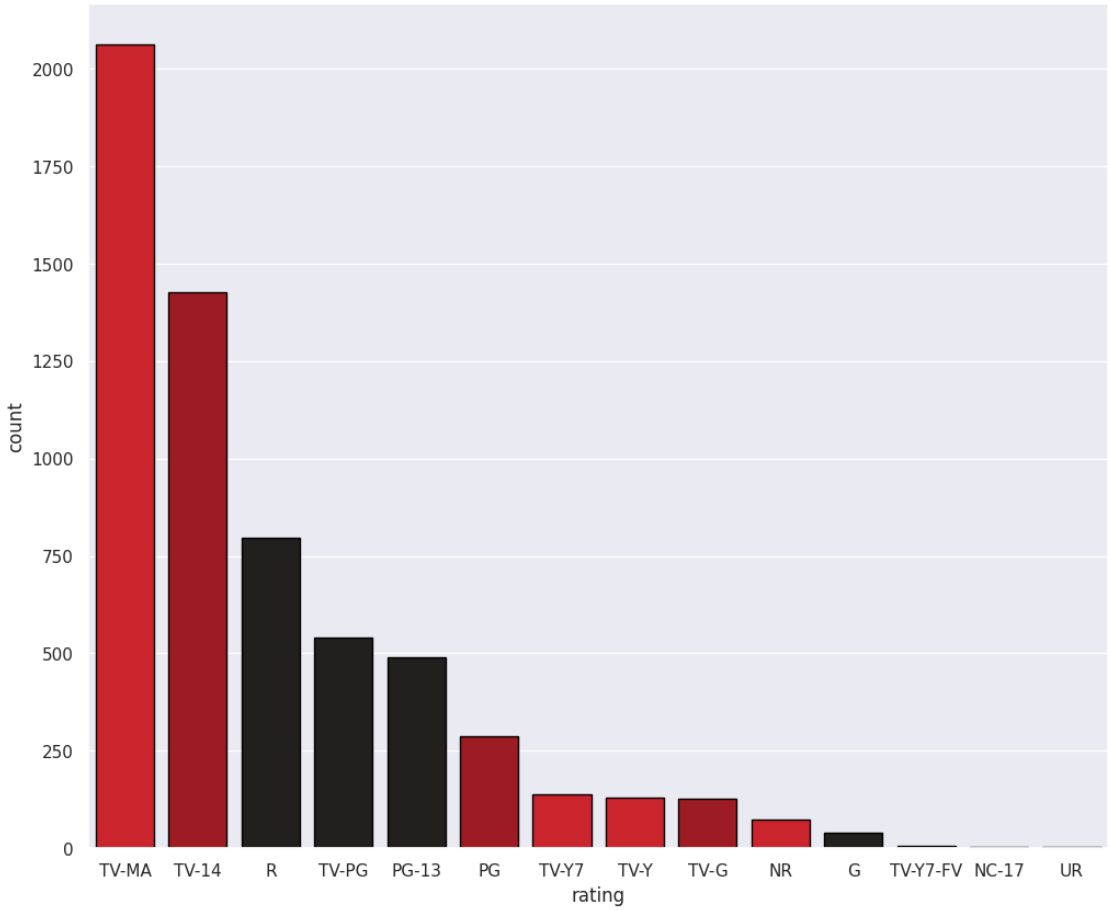
```
687           United States     June 18, 2021          2021  TV-MA    159 min
690           United States     June 18, 2021          2021  TV-14    137 min
1468            South Korea   January 1, 2021          2021  TV-MA    102 min
692           United States     June 18, 2021          2021  TV-14   1 Season
693           United States     June 17, 2021          2021  TV-14    101 min
1173            South Korea    March 23, 2021          2021  TV-14  6 Seasons
1172           South Africa    March 23, 2021          2021  TV-MA     60 min
1171          United States    March 24, 2021          2021  TV-14     90 min
1170  Russia, United States    March 25, 2021          2021  TV-Y7    105 min
707           United States     June 16, 2021          2021  TV-14   1 Season
1167                  Italy    March 25, 2021          2021  TV-14    100 min

                                                listed_in  \
1370                      Dramas, International Movies
255                Documentaries, International Movies
1189                                  Stand-Up Comedy
1188  Action & Adventure, Comedies, International Mo…
1187                                        Thrillers
363   Action & Adventure, International Movies, Thri…
1184          Action & Adventure, International Movies
683                       Dramas, International Movies
685                                             Dramas
687    Action & Adventure, Dramas, International Movies
690           Action & Adventure, International Movies
1468                    Comedies, International Movies
692   International TV Shows, Romantic TV Shows, TV …
693              Comedies, Dramas, International Movies
1173  International TV Shows, Korean TV Shows, Stand…
1172                                  Stand-Up Comedy
1171                                    Documentaries
1170                 Children & Family Movies, Comedies
707   International TV Shows, Romantic TV Shows, TV …
1167      Dramas, International Movies, Romantic Movies

                                               description
1370  On the eve of World War II, a British widow hi…
255   Serial killer Dennis Nilsen narrates his life …
1189  Nate Bargatze touches on Zoom comedy shows, lo…
1188  Two hapless cops find themselves in over their…
1187  After a bestselling novelist suffering from wr…
363   A genius engineer and his crew of amateur thie…
1184  A feud erupts between realms, and Yin Yang mas…
683   Taken in by the yakuza at a young age, Kenji s…
685   A widowed new dad copes with doubts, fears, he…
687   When a clever, carefree gangster is recruited …
690   In 1879, Kenshin and his allies face their str…
1468  With the peak of his career long behind him, a…
```

```
692    A career-driven 30-something must contend with…
693    After his father's passing, a teenager sets ou…
1173   Male celebs play make-believe as high schooler…
1172   South African comedian Loyiso Gola serves up f…
1171   Passionate about ocean life, a filmmaker sets …
1170   Hansel and Gretel of fairy tale fame - now act…
707    A food-blogging insurance agent encounters a f…
1167   After falling in love at a sailing summer camp…
```

4 Rating and Audience Preferences

[97]:
```
netflix_movies=netflix_overall[netflix_overall['type']=='Movie']
plt.figure(figsize=(12,10))
sns.set(style="darkgrid")
ax = sns.countplot(x="rating", data=netflix_movies, palette=custom_palette,␣
↪order=netflix_movies['rating'].value_counts().index[0:15], hue = 'rating',␣
↪legend= False,edgecolor='black')
```

```
<ipython-input-97-19cea8b07ba4>:4: UserWarning:


The palette list has fewer values (3) than needed (14) and will cycle, which may
produce an uninterpretable plot.
```

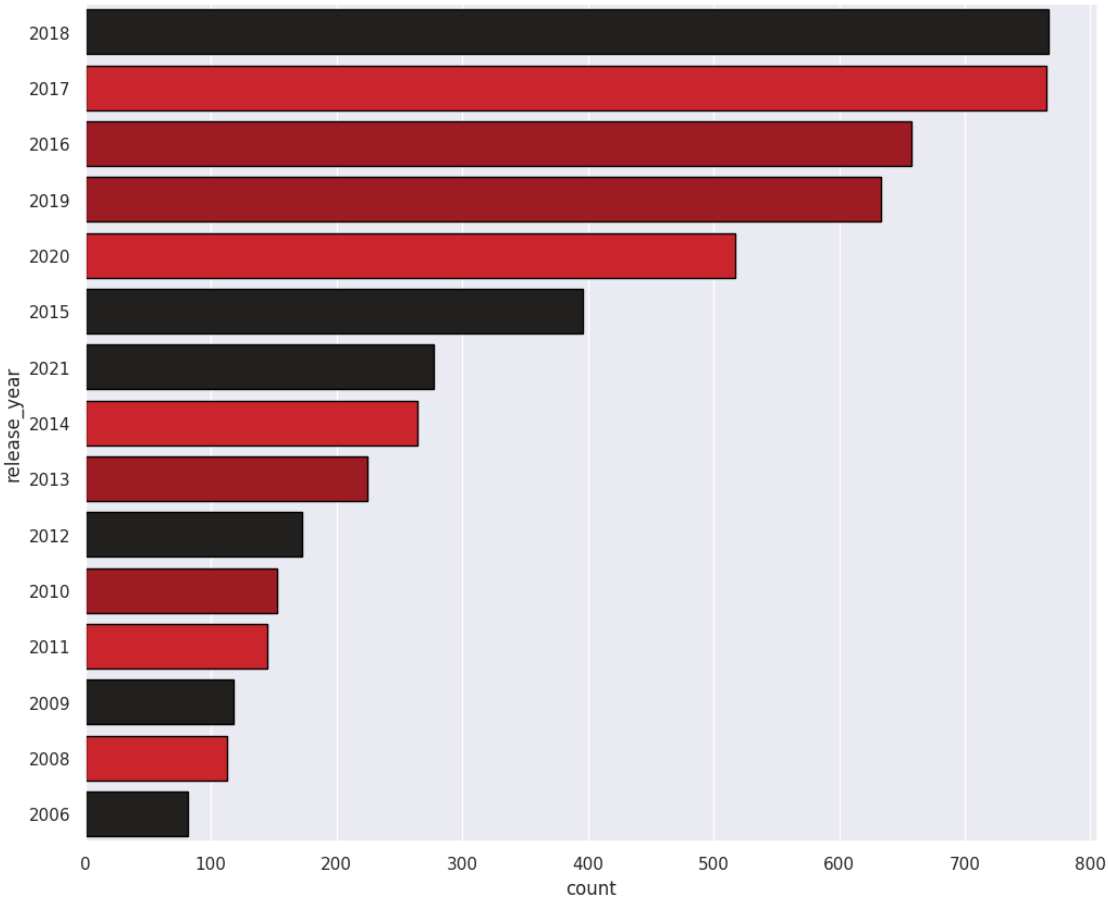


The majority of content on Netflix is rated TV-MA, TV-14, and R, indicating a preference for mature and diverse content. However, there is also a significant audience for family-friendly content, as seen in the popularity of genres like family movies and children's shows.

5 Yearly Content Strategy

[98]:
```
plt.figure(figsize=(12,10))
sns.set(style="darkgrid")
ax = sns.countplot(y="release_year", data=netflix_movies,␣
↪palette=custom_palette, order=netflix_movies['release_year'].value_counts().
↪index[0:15],hue="release_year", legend=False,edgecolor='black')
```

```
<ipython-input-98-b55351296fb9>:3: UserWarning:


The palette list has fewer values (3) than needed (73) and will cycle, which may
produce an uninterpretable plot.
```



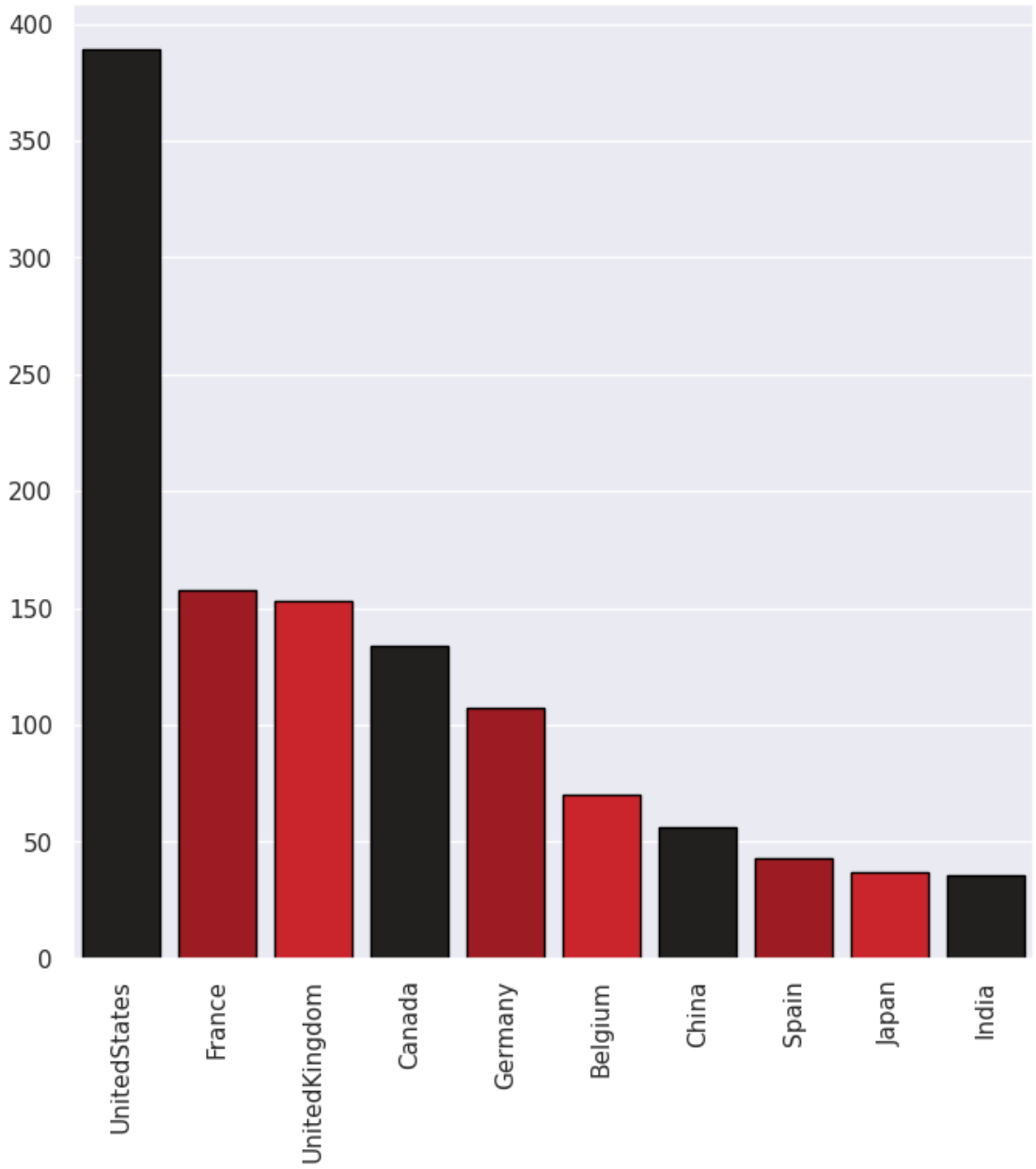
Netflix should continue to focus on producing a diverse range of content each year, with a particular emphasis on releasing more movies in years like 2018, when audience engagement and demand were higher.

6 International Expansion

[99]:
```
countries = {}
netflix_movies['country'] = netflix_movies['country'].fillna('Unknown')
cou = list(netflix_movies['country'])
for i in cou:
    i = list(i.split(','))
    if len(i) == 1:
        if i in list(countries.keys()):
            countries[i] += 1
        else:
            countries[i[0]] = 1
    else:
        for j in i:
```

```
            if j in list(countries.keys()):
                countries[j] += 1
            else:
                countries[j] = 1

countries_fin = {}
for country, no in countries.items():
    country = country.replace(' ', '')
    if country in list(countries_fin.keys()):
        countries_fin[country] += no
    else:
        countries_fin[country] = no

countries_fin = {k: v for k, v in sorted(countries_fin.items(), key=lambda item:
 ↪ item[1], reverse=True)}

plt.figure(figsize=(8, 8))
ax = sns.barplot(x=list(countries_fin.keys())[0:10], y=list(countries_fin.
 ↪values())[0:10], palette=custom_palette, hue = list(countries_fin.keys())[0:
 ↪10], legend = False,edgecolor='black')
ax.set_xticklabels(list(countries_fin.keys())[0:10], rotation=90)
plt.show()
```

```
<ipython-input-99-e80e1670add8>:2: 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/user_guide/indexing.html#returning-a-view-versus-a-copy

<ipython-input-99-e80e1670add8>:29: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-99-e80e1670add8>:30: UserWarning:

FixedFormatter should only be used together with FixedLocator
```



Considering the top countries for content creation on Netflix are USA, France, UK, Canada, and others, Netflix should prioritize producing content that resonates with the cultural preferences of these regions. This could involve investing in local productions, language-specific content, and collaborations with regional talents.

7 Duration Preferences

[100]:
```
netflix_movies['duration']=netflix_movies['duration'].str.replace(' min','')
netflix_movies['duration']=netflix_movies['duration'].astype(str).astype(int)
netflix_movies['duration']
```

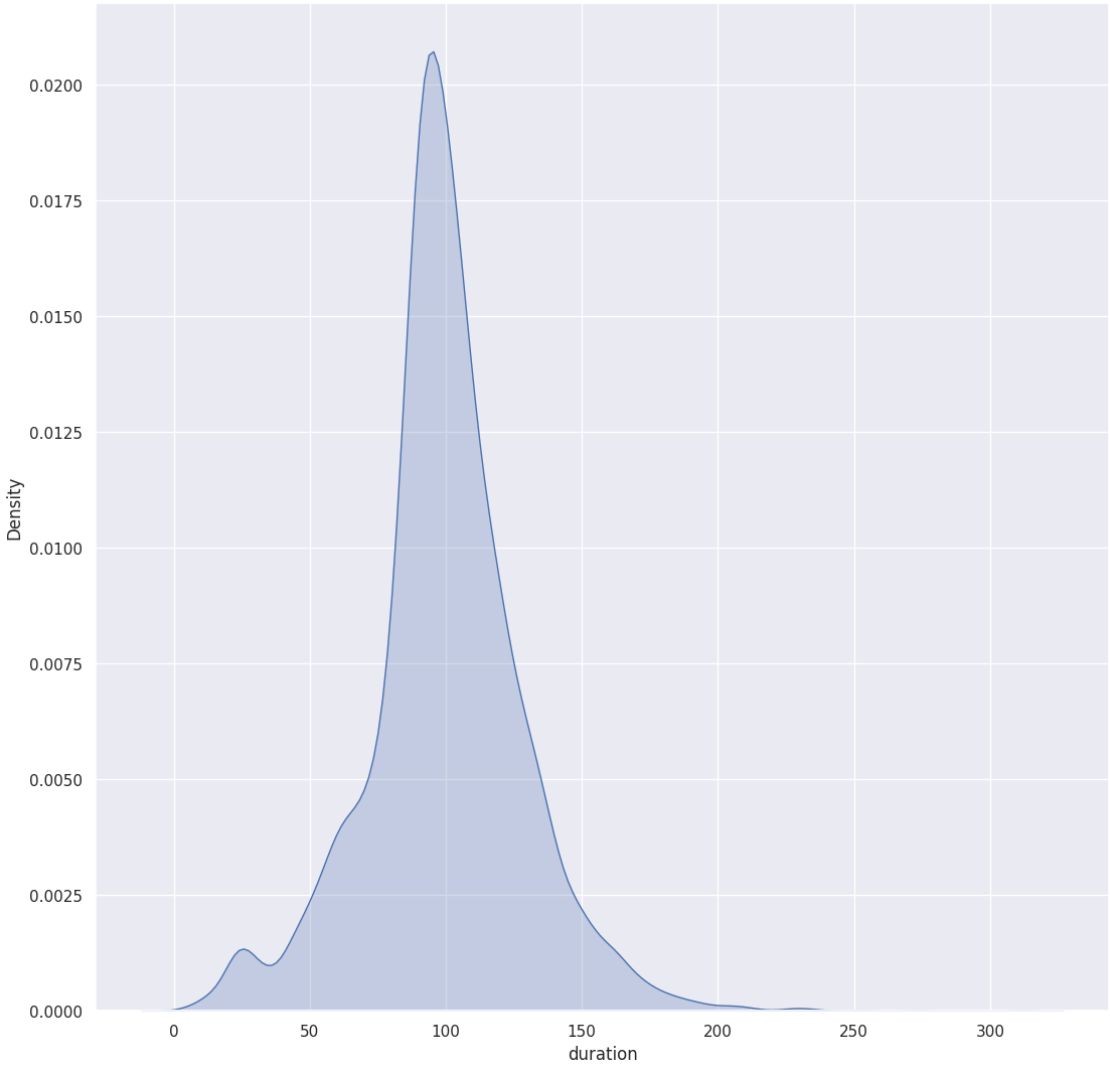
```
<ipython-input-100-738641204206>:1: SettingWithCopyWarning:


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/user_guide/indexing.html#returning-a-view-versus-a-copy

<ipython-input-100-738641204206>:2: 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/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[100]:
```
0        90
6        91
7       125
9       104
12      127
       ...
8801     96
8802    158
8804     88
8805     88
8806    111
Name: duration, Length: 6126, dtype: int64
```

[101]:
```
sns.set(style="darkgrid")
sns.kdeplot(data=netflix_movies['duration'], shade=True)
```

```
<ipython-input-101-7ac85c43151c>:2: FutureWarning:


`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.
```

[101]: <Axes: xlabel='duration', ylabel='Density'>



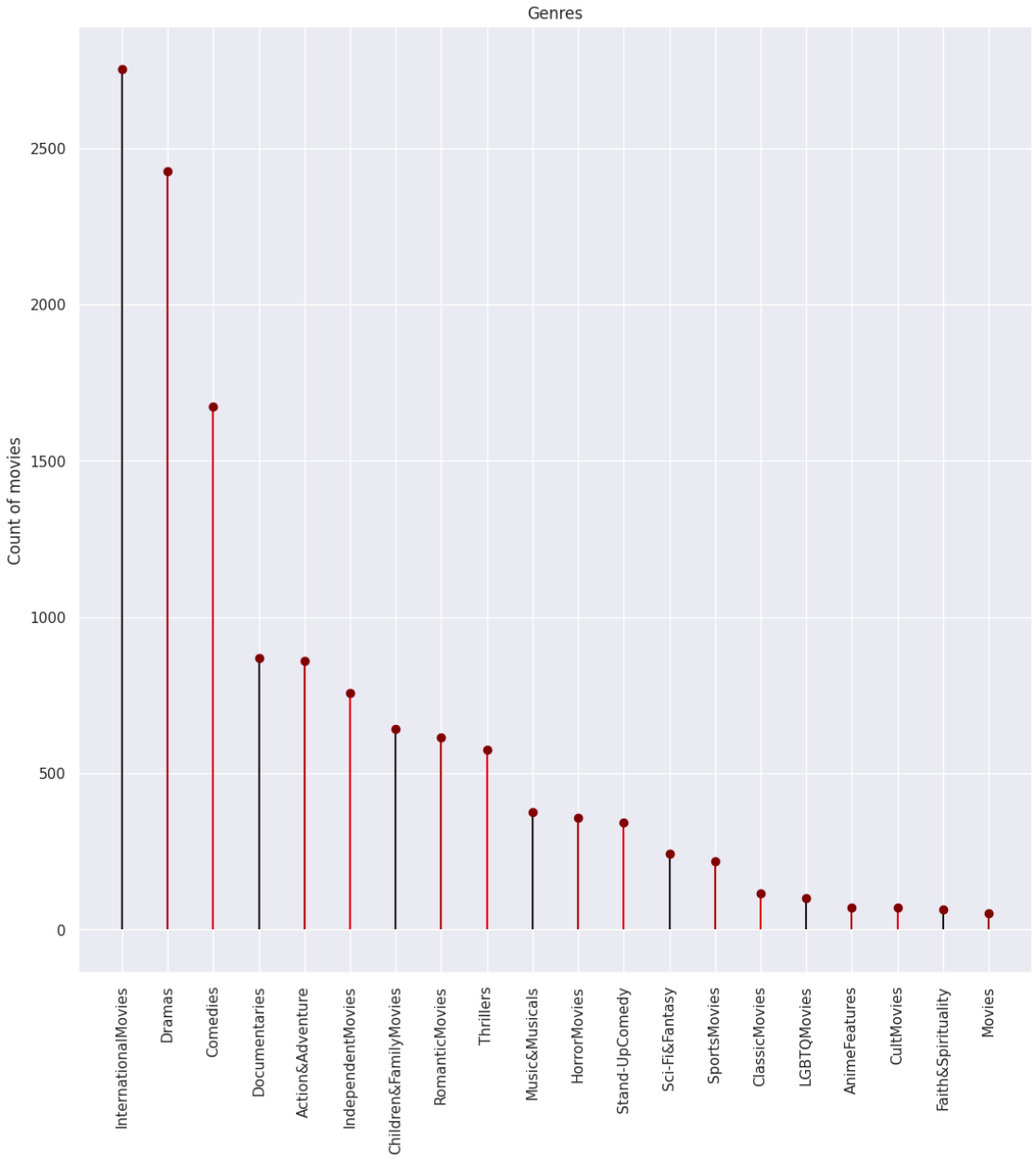
Audience preference for movie duration falls within the range of 75-120 minutes, indicating a preference for content that can be consumed in one sitting.

8 Genres Preferences

[102]:
```
genres=list(netflix_movies['listed_in'])
gen=[]

for i in genres:
    i=list(i.split(','))
```

```
    for j in i:
        gen.append(j.replace(' ',""))
g=Counter(gen)

g={k: v for k, v in sorted(g.items(), key=lambda item: item[1], reverse= True)}


fig, ax = plt.subplots()

fig = plt.figure(figsize = (0.2, 0.2))
x=list(g.keys())
y=list(g.values())
ax.vlines(x, ymin=0, ymax=y, color=custom_palette)
ax.plot(x,y, "o", color='maroon')
ax.set_xticklabels(x, rotation = 90)
ax.set_ylabel("Count of movies")
# set a title
ax.set_title("Genres");
```

```
<ipython-input-102-3896da2a22d9>:20: UserWarning:

FixedFormatter should only be used together with FixedLocator
```




```
<Figure size 20x20 with 0 Axes>
```

Therefore, it is clear that international movies, dramas and comedies are the top three genres that have the highest amount of content on Netflix.

9 WordCloud for Movie Genres.

[103]:
```
genres=list(netflix_movies['listed_in'])
gen=[]

for i in genres:
    i=list(i.split(','))
    for j in i:
        gen.append(j.replace(' ',""))
g=Counter(gen)
```

[104]:
```
text = list(set(gen))
plt.rcParams['figure.figsize'] = (13, 13)

# Assigning shape to the word cloud
mask = np.array(Image.open('/content/drive/MyDrive/Dataset/comment.png'))

wordcloud = WordCloud(max_words=1000000, background_color="white", mask=mask,␣
 ↪colormap='rocket').generate(str(text))

plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

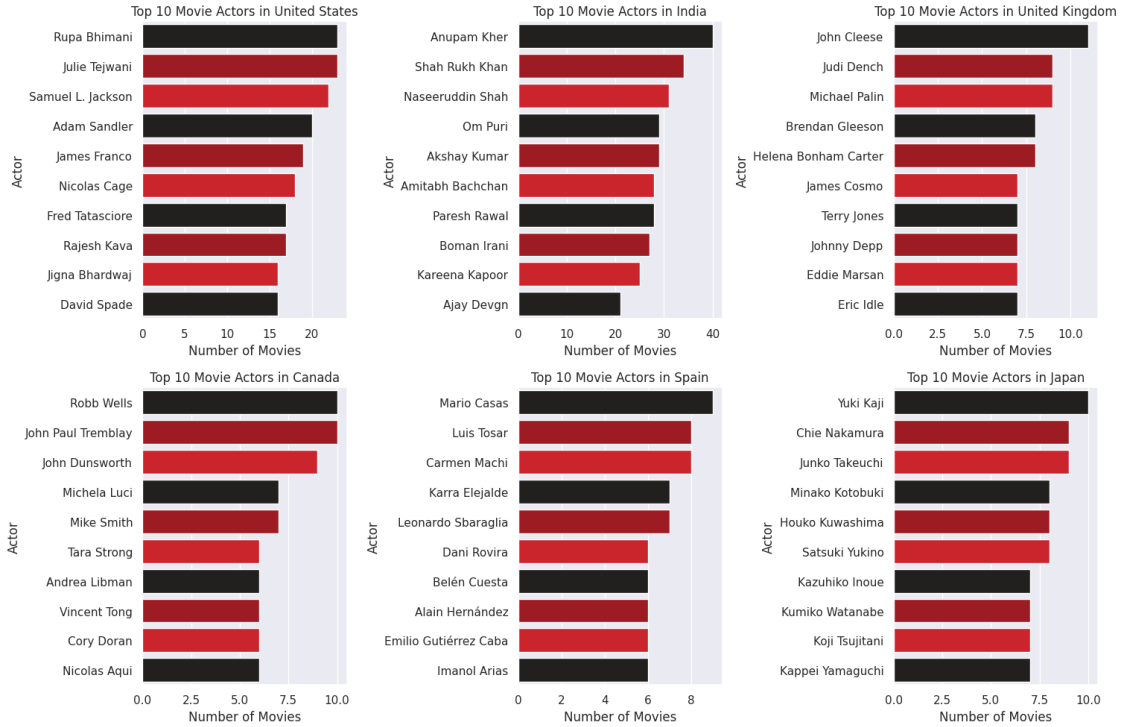
10 Top Movie Actors

[105]:
```
# Define the countries
countries = ["United States", "India", "United Kingdom", "Canada", "Spain",␣
 ↪"Japan"]

# Function to plot top actors for a given country
def plot_top_actors(country, ax):
    country_data = netflix_overall[(netflix_overall['country'].str.
 ↪contains(country, na=False)) & (netflix_overall['type'] == 'Movie') &
                                   (netflix_overall['cast'] != 'No Data')]
```

```
    # Extract and split the cast column to get individual actors
    actors = country_data['cast'].str.split(', ')
    actors = actors.explode()

    # Count the occurrences of each actor
    top_actors = actors.value_counts().head(10)

    sns.barplot(x=top_actors.values, y=top_actors.index,␣
 ↪palette=custom_palette, ax=ax, hue = top_actors.index,legend=False)
    ax.set_title(f'Top 10 Movie Actors in {country}')
    ax.set_xlabel('Number of Movies')
    ax.set_ylabel('Actor')


num_countries = len(countries)
num_cols = 3
num_rows = math.ceil(num_countries / num_cols)


fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 10))


if num_rows == 1:
    axes = axes.reshape(1, -1)


for i, country in enumerate(countries):
    plot_top_actors(country, axes[i // num_cols, i % num_cols])

for i in range(num_countries, num_rows * num_cols):
    axes.flatten()[i].axis('off')

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

```
<ipython-input-105-e323a8530485>:16: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-105-e323a8530485>:16: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.
```

```
<ipython-input-105-e323a8530485>:16: UserWarning:

The palette list has fewer values (3) than needed (10) and will cycle, which may produce an uninterpretable plot.

<ipython-input-105-e323a8530485>:16: UserWarning:

The palette list has fewer values (3) than needed (10) and will cycle, which may produce an uninterpretable plot.

<ipython-input-105-e323a8530485>:16: UserWarning:

The palette list has fewer values (3) than needed (10) and will cycle, which may produce an uninterpretable plot.

<ipython-input-105-e323a8530485>:16: UserWarning:

The palette list has fewer values (3) than needed (10) and will cycle, which may produce an uninterpretable plot.
```



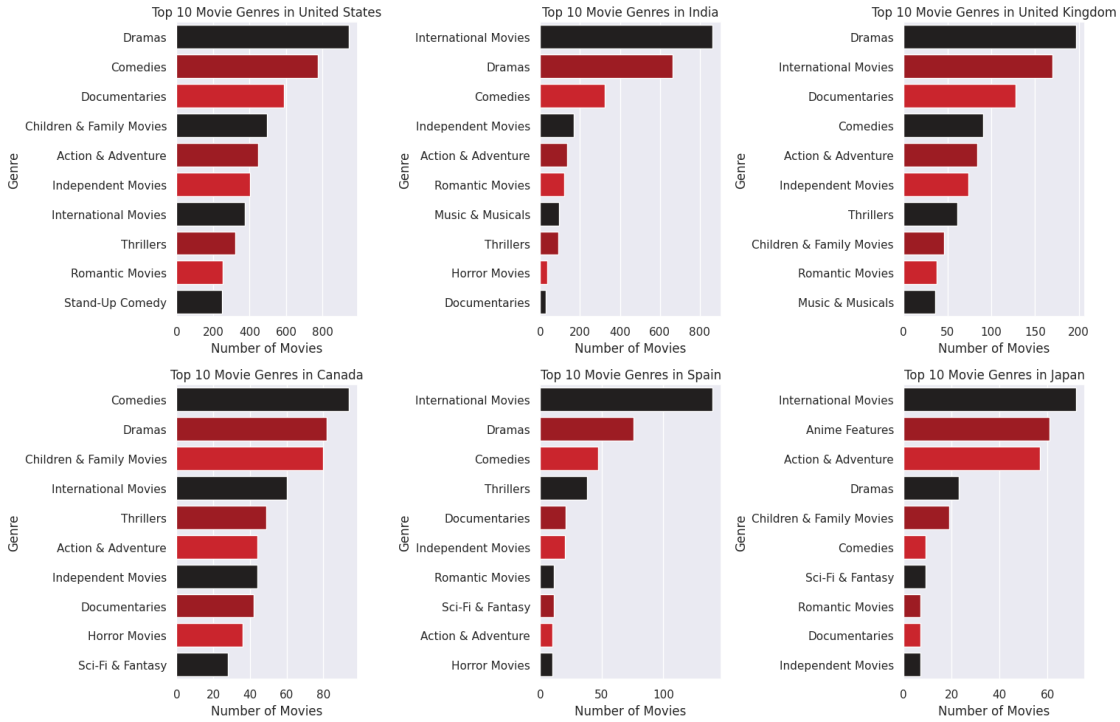
Netflix can leverage the popularity of actors like Rupa Bhimani, Anupam Kher, John Cleese, and others to attract audiences from different regions. Collaborations with renowned directors and actors can enhance the quality and appeal of Netflix originals.

11 Top Genre in Movies

[106]:
```
countries = ["United States", "India", "United Kingdom", "Canada", "Spain",␣
 ↪"Japan"]


def plot_top_movie_genres(country, ax):
    country_data = netflix_overall[(netflix_overall['country'].str.
 ↪contains(country, na=False)) &
                                   (netflix_overall['type'] == 'Movie')]


    genres = country_data['listed_in'].str.split(', ')
    genres = genres.explode()


    top_genres = genres.value_counts().head(10)


    sns.barplot(x=top_genres.values, y=top_genres.index, ax=ax,␣
 ↪palette=custom_palette, hue = top_genres.index,legend=False)
    ax.set_title(f'Top 10 Movie Genres in {country}')
    ax.set_xlabel('Number of Movies')
    ax.set_ylabel('Genre')


num_countries = len(countries)
num_cols = 3
num_rows = math.ceil(num_countries / num_cols)


fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 10))


if num_rows == 1:
    axes = axes.reshape(1, -1)


for i, country in enumerate(countries):
    plot_top_movie_genres(country, axes[i // num_cols, i % num_cols])

for i in range(num_countries, num_rows * num_cols):
```

```
    axes.flatten()[i].axis('off')

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

```
<ipython-input-106-b298ed19bdaa>:16: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-106-b298ed19bdaa>:16: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-106-b298ed19bdaa>:16: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-106-b298ed19bdaa>:16: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-106-b298ed19bdaa>:16: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-106-b298ed19bdaa>:16: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.
```



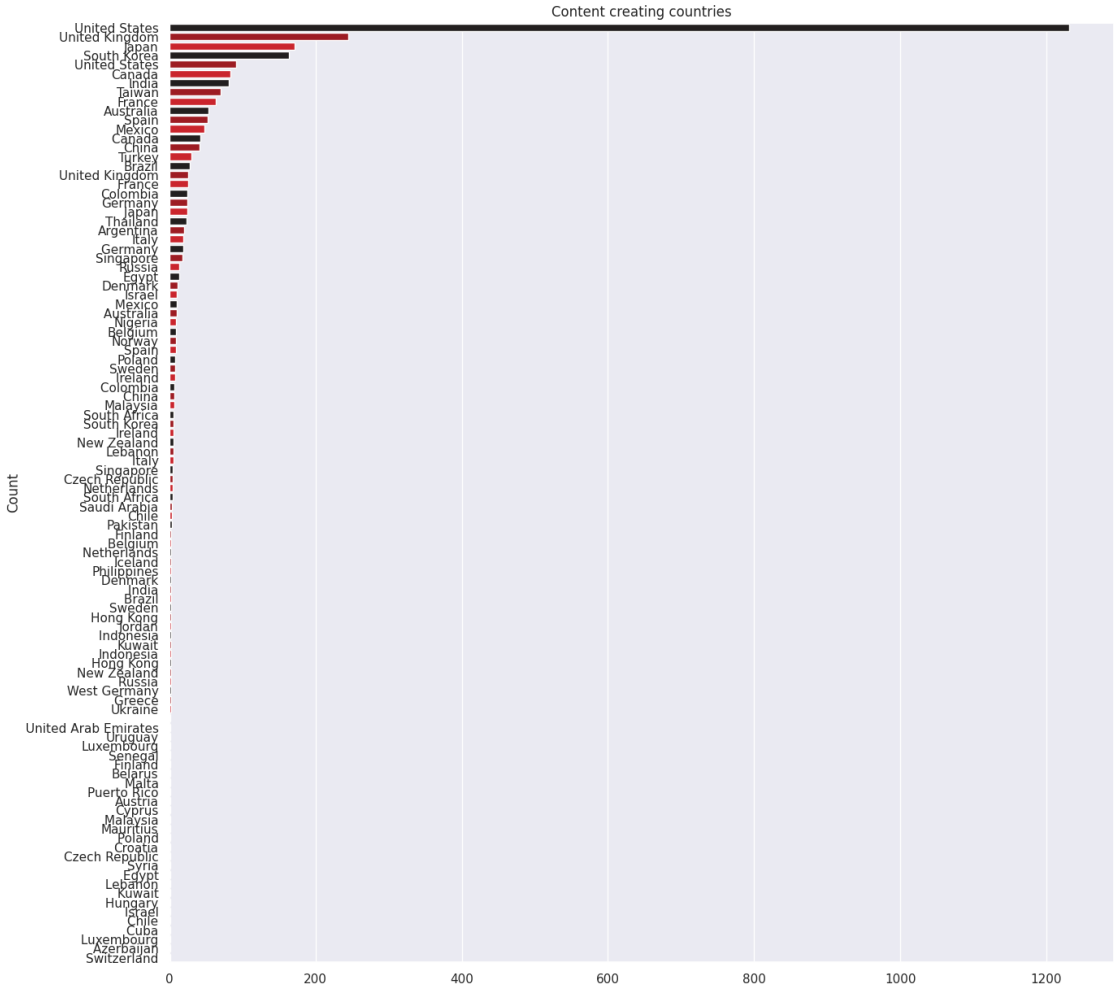
Preferences vary by country, with dramas dominating in the USA, international movies in India, and comedies in Canada. Understanding regional preferences can inform content production strategies.

12 Analysis of TV SERIES on Netflix

[107]:
```
countries1 = {}
netflix_shows.loc[:, 'country'] = netflix_shows['country'].fillna('Unknown')
cou1 = list(netflix_shows['country'])
for i in cou1:
    i = list(i.split(','))
    if len(i) == 1:
        if i[0] in countries1:
            countries1[i[0]] += 1
        else:
            countries1[i[0]] = 1
    else:
        for j in i:
            if j in countries1:
                countries1[j] += 1
            else:
                countries1[j] = 1
```

[108]:
```
plt.figure(figsize=(15, 15))

plt.title("Content creating countries")

sorted_countries = dict(sorted(countries1.items(), key=lambda item: item[1],␣
↪reverse=True))

sns.barplot(y=list(sorted_countries.keys()), x=list(sorted_countries.values()),␣
↪palette=custom_palette,hue =list(sorted_countries.keys()),legend=False )

plt.ylabel("Count")

plt.show()
```

```
<ipython-input-108-029b50e319a0>:7: UserWarning:


The palette list has fewer values (3) than needed (102) and will cycle, which
may produce an uninterpretable plot.
```



The USA leads in producing TV series content for Netflix, indicating a strong market presence and potential for further growth in this category.

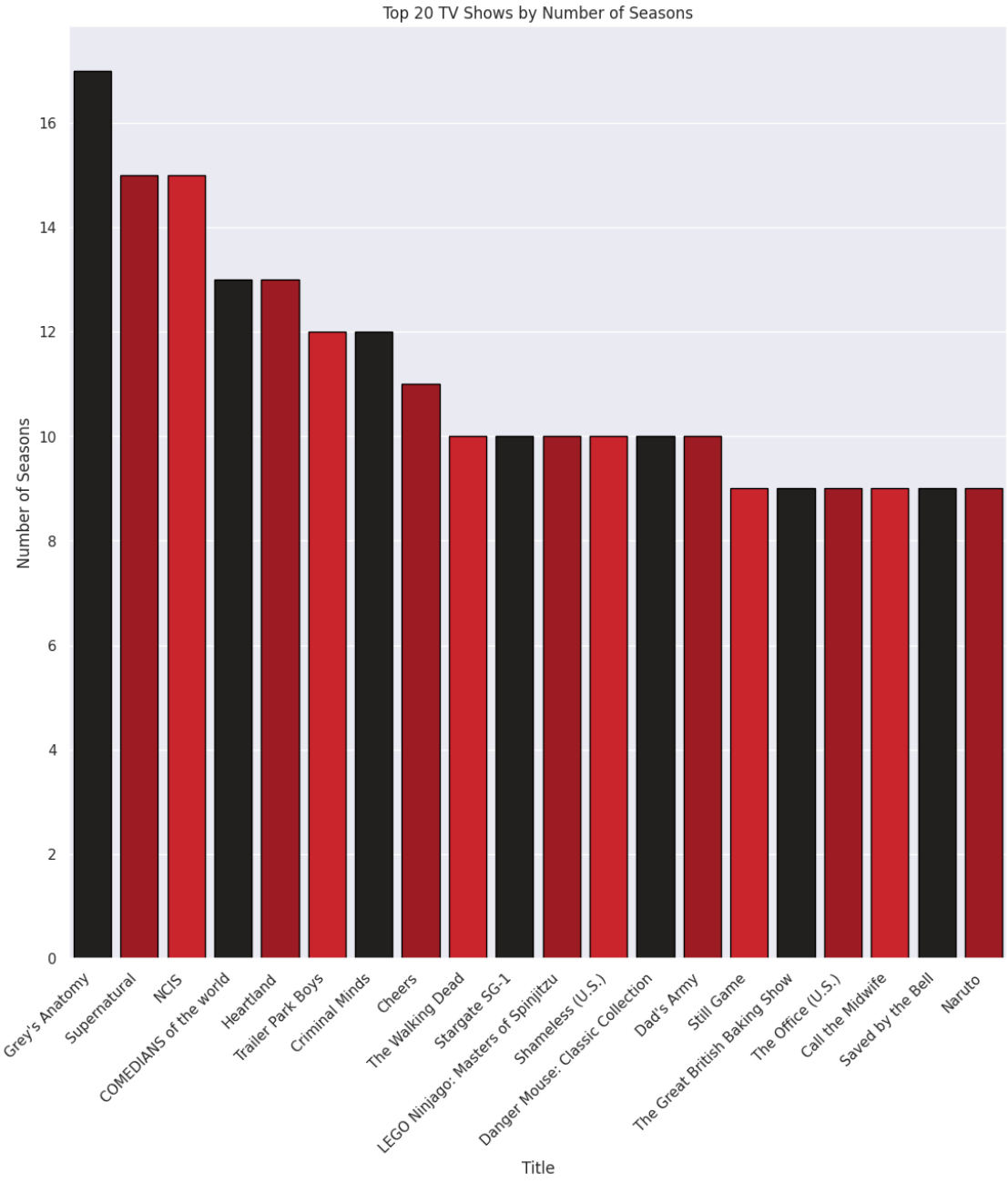
13 TV shows with highest number of seasons

[109]:
```
netflix_shows = netflix_overall[netflix_overall['type'] == 'TV Show']
features = ['title', 'duration']
durations = netflix_shows[features].copy()

durations['no_of_seasons'] = durations['duration'].str.replace(' Season', '')
durations['no_of_seasons'] = durations['no_of_seasons'].str.replace('s', '')

durations.loc[:, 'no_of_seasons'] = durations['no_of_seasons'].astype(int)
t=['title','no_of_seasons']
top=durations[t]
```

```
top=top.sort_values(by='no_of_seasons', ascending=False)
top20=top[0:20]
```

[110]:
```
sns.barplot(data=top20, x='title', y='no_of_seasons',␣
 ↪palette=custom_palette,hue ='title',legend=False,edgecolor='black')

plt.xticks(rotation=45, ha='right')
plt.xlabel('Title')
plt.ylabel('Number of Seasons')
plt.title('Top 20 TV Shows by Number of Seasons')
plt.show()
```

```
<ipython-input-110-767c27c9eb62>:1: UserWarning:


The palette list has fewer values (3) than needed (20) and will cycle, which may
produce an uninterpretable plot.
```



TV series like "NCIS," "Grey's Anatomy," and "Supernatural" with multiple seasons are popular, while shorter series offer binge-worthy content. This insight helps in understanding audience preferences for series length.

14 Lowest number of seasons

[111]:
```
bottom=top.sort_values(by='no_of_seasons')
bottom=bottom[20:50]

fig = go.Figure(data=[go.Table(header=dict(values=['Title', 'No of seasons']),
                 ␣
 ↪cells=dict(values=[bottom['title'],bottom['no_of_seasons']],fill_color='pink'))
                     ])
fig.show()
```

These are some binge-worthy shows that are short and have only one season.

15 Top Duration Shows

[112]:
```
topdirs=pd.value_counts(netflix_overall['duration'])
fig = go.Figure([go.Bar(x=topdirs.index, y=topdirs.values , text=topdirs.
 ↪values,marker_color=custom_palette)])
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')
fig.show()
```

Single-season TV shows are preferred, suggesting that concise storytelling resonates well with Netflix audiences.

16 Word Cloud for TV Shows Genres

[113]:
```
genres=list(netflix_shows['listed_in'])
gen=[]

for i in genres:
    i=list(i.split(','))
    for j in i:
        gen.append(j.replace(' ',""))
g=Counter(gen)
```

[114]:
```
text = list(set(gen))

mask=np.array(Image.open('/content/drive/MyDrive/Dataset/comment.png'))
wordcloud =␣
 ↪WordCloud(max_words=1000000,background_color="white",mask=mask,colormap='rocket').
 ↪generate(str(text))
plt.rcParams['figure.figsize'] = (13, 13)
plt.imshow(wordcloud,interpolation="bilinear")
plt.axis("off")
plt.show()
```

Keywords like “TV Sci,” “comedy,” “romantic TV shows,” and “talk shows” indicate popular genres in TV series, guiding content development efforts.

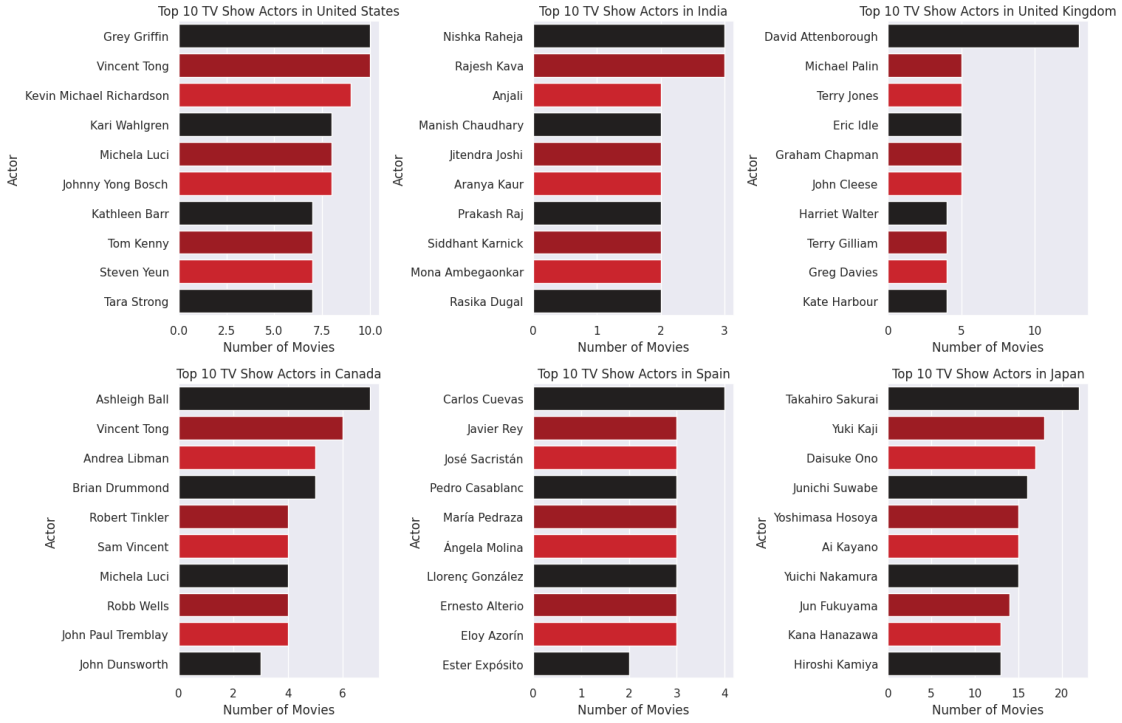
17 Top TV Shows Actors

[115]:
```
countries = ["United States", "India", "United Kingdom", "Canada", "Spain",␣
 ↪"Japan"]


def plot_top_actors(country, ax):
    country_data = netflix_overall[(netflix_overall['country'].str.
 ↪contains(country, na=False)) & (netflix_overall['type'] == 'TV Show') &
```

```
                                    (netflix_overall['cast'] != 'No Data')]


    actors = country_data['cast'].str.split(', ')
    actors = actors.explode()
    top_actors = actors.value_counts().head(10)
    sns.barplot(x=top_actors.values, y=top_actors.index,␣
 ↪palette=custom_palette, ax=ax, hue = top_actors.index,legend=False)
    ax.set_title(f'Top 10 TV Show Actors in {country}')
    ax.set_xlabel('Number of Movies')
    ax.set_ylabel('Actor')


num_countries = len(countries)
num_cols = 3
num_rows = math.ceil(num_countries / num_cols)


fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 10))


if num_rows == 1:
    axes = axes.reshape(1, -1)
for i, country in enumerate(countries):
    plot_top_actors(country, axes[i // num_cols, i % num_cols])
for i in range(num_countries, num_rows * num_cols):
    axes.flatten()[i].axis('off')

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

```
<ipython-input-115-69c55e03ce91>:12: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-115-69c55e03ce91>:12: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-115-69c55e03ce91>:12: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
```

```
produce an uninterpretable plot.

<ipython-input-115-69c55e03ce91>:12: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-115-69c55e03ce91>:12: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-115-69c55e03ce91>:12: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.
```



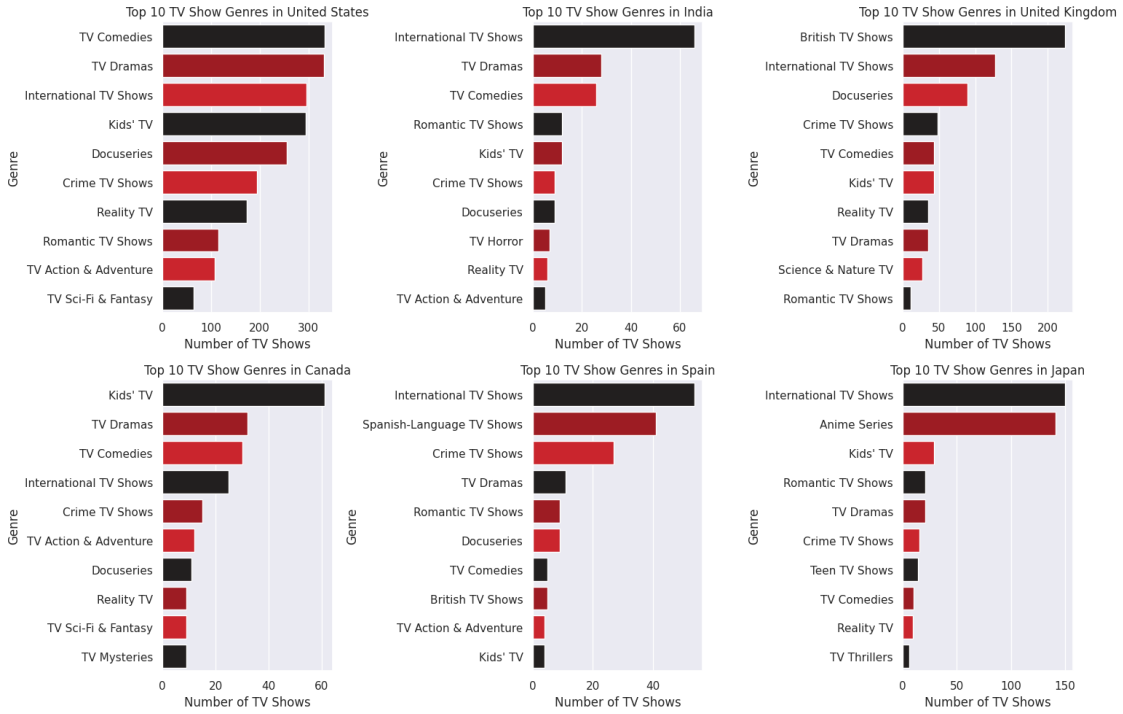
Recognizable talents like Grey Griffin in the USA and David Attenborough in the UK contribute to Netflix's TV series content, enhancing its appeal to audiences.

18 Top TV Shows Genres

[116]:
```
countries = ["United States", "India", "United Kingdom", "Canada", "Spain",␣
 ↪"Japan"]


def plot_top_genres(country, ax):
    country_data = netflix_overall[(netflix_overall['country'].str.
 ↪contains(country, na=False)) &
                                   (netflix_overall['type'] == 'TV Show')]

    genres = country_data['listed_in'].str.split(', ')
    genres = genres.explode()


    top_genres = genres.value_counts().head(10)
    sns.barplot(x=top_genres.values, y=top_genres.index, ax=ax,␣
 ↪palette=custom_palette, hue = top_genres.index,legend=False)
    ax.set_title(f'Top 10 TV Show Genres in {country}')
    ax.set_xlabel('Number of TV Shows')
    ax.set_ylabel('Genre')


num_countries = len(countries)
num_cols = 3
num_rows = math.ceil(num_countries / num_cols)


fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 10))


if num_rows == 1:
    axes = axes.reshape(1, -1)


for i, country in enumerate(countries):
    plot_top_genres(country, axes[i // num_cols, i % num_cols])

for i in range(num_countries, num_rows * num_cols):
    axes.flatten()[i].axis('off')

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

```
<ipython-input-116-c6ff2eadfc6f>:13: UserWarning:
```

```
The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-116-c6ff2eadfc6f>:13: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-116-c6ff2eadfc6f>:13: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-116-c6ff2eadfc6f>:13: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-116-c6ff2eadfc6f>:13: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.

<ipython-input-116-c6ff2eadfc6f>:13: UserWarning:


The palette list has fewer values (3) than needed (10) and will cycle, which may
produce an uninterpretable plot.
```



Preferences for TV show genres vary by country, with TV comedies in the USA, international TV shows in India and Spain, and British TV shows in the UK. Tailoring content to regional tastes can maximize viewership.

19 WordCloud for Description

[117]:
```
genres=list(netflix_movies['description'])
gen=[]

for i in genres:
    i=list(i.split(','))
    for j in i:
        gen.append(j.replace(' ',""))
g=Counter(gen)

text = " ".join(gen)

mask = np.array(Image.open('/content/drive/MyDrive/Dataset/comment.png'))

wordcloud = WordCloud(max_words=1000000, background_color="white", mask=mask,␣
 ↪colormap = 'rocket').generate(text)

plt.figure(figsize=(10, 10))
plt.imshow(wordcloud, interpolation="bilinear")
```

```
plt.axis("off")
plt.show()
```

Keywords like “year,” “special,” “based on true events,” and “inspired by true events” reflect popular themes in Netflix descriptions, providing insights into viewer interests and potential content themes.

20 Conclusion

In conclusion, the analysis of Netflix’s content dataset provides valuable insights into audience preferences, content trends, and potential growth opportunities for the streaming platform. By synthesizing the findings, we can offer strategic recommendations to guide Netflix in its content production and business expansion efforts.

Firstly, it's evident that Netflix should continue to prioritize a diverse content library that includes a mix of movies, TV shows, documentaries, and other genres. While movies dominate the platform, TV series also play a significant role, particularly in regions like the United States. Therefore, Netflix should maintain a balanced approach to content production, investing in both long-form storytelling and shorter, one-season shows.

Genre analysis highlights the popularity of dramas, comedies, international movies, and TV comedies across different countries. Netflix should capitalize on these trends by producing content that resonates with diverse audience preferences while also exploring emerging genres and niche markets.

Collaborations with renowned directors, actors, and production studios can enhance the quality and appeal of Netflix originals. By leveraging regional talent and partnerships, Netflix can create authentic, culturally relevant content for specific markets, driving engagement and subscriber growth.

Furthermore, data-driven decision-making should underpin Netflix's content strategy, with a focus on personalized recommendations, targeted marketing, and continuous innovation. By analyzing viewer behavior, engagement metrics, and content performance, Netflix can optimize its content offerings and enhance viewer satisfaction.

In terms of content production, Netflix should prioritize quality over quantity and focus on creating unique, compelling stories that resonate with audiences worldwide. Investing in original content, while also acquiring and licensing localized content, can help Netflix penetrate new markets and strengthen its position as a global streaming platform.

Overall, by incorporating these insights and recommendations into its content strategy, Netflix can effectively meet the diverse needs and preferences of its audience while driving business growth and success in different countries and markets.