# NETFLIX BUSINESS CASE STUDY

### About Netflix:

Netflix is one of the most popular media and video streaming platforms. They have over 10000 movies or tv shows available on their platform, as of mid-2021, they have over 222M Subscribers globally. This tabular dataset consists of listings of all the movies and tv shows available on Netflix, along with details such as - cast, directors, ratings, release year, duration, etc.

### Business Problem to solve:

To analyze Netflix's content data to uncover actionable insights that inform decisions on the types of shows and movies to produce, and identify opportunities for expanding the platform's presence and engagement across different countries.

The dataset provided consists of a list of all the TV shows/movies available on Netflix:

- **Show_id:** Unique ID for every Movie / Tv Show
- **Type:** Identifier - A Movie or TV Show
- **Title:** Title of the Movie / Tv Show
- **Director:** Director of the Movie
- **Cast:** Actors involved in the movie/show
- **Country:** Country where the movie/show was produced
- **Date_added:** Date it was added on Netflix
- **Release_year:** Actual Release year of the movie/show
- **Rating:** TV Rating of the movie/show
- **Duration:** Total Duration - in minutes or number of seasons
- **Listed_in:** Genre
- **Description:** The summary description

# Important Python Liberaries to Import | Reading Files | Basic Data Exploration

```
#IMPORTING LIBERARIES:
import numpy as numpy
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

#READING THE FILE:
df = pd.read_csv('/content/netflix.csv')
```

```
#EXPLORAING THE DATA FOR THE FIRST TIME:
df.head()
```

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | s1 | Movie | Dick Johnson Is Dead | Kirsten Johnson | NaN | United States | September 25, 2021 | 2020 | PG-13 | 90 min | Documentaries | As her father nears the end of his life, filmm... |
| 1 | s2 | TV Show | Blood & Water | NaN | Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban... | South Africa | September 24, 2021 | 2021 | TV-MA | 2 Seasons | International TV Shows, TV Dramas, TV Mysteries | After crossing paths at a party, a Cape Town t... |
| 2 | s3 | TV | Ganglands | Julien | Sami Bouajila, Tracy Gotoas | NaN | September | 2021 | TV-MA | 1 Season | Crime TV Shows, | To protect his family from a |

Next steps: Generate code with df | View recommended plots | New interactive sheet

### Additional learning for the Viewers:

Various Types of Ratings mentioned in the data above:

- TV-MA = **Mature Audience** (inappropriate for children under 17)
- TV-14 = not suitable for children **aged under 14**
- TV-PG = **Parental Guidance** suggested
- R = Restricted & not allowed for **under 17 age**
- PG-13 = Not Suitable for **children under 13**
- TV-Y7 = Suitable for **age 7+ children**
- TV-Y = Appropriate for children of **all ages**
- PG = **Parental Guidance** Suggested (may not be suitable for younger children)
- TV-G = **General** Audience (all ages)
- NR = **NOT RATED**
- G = **General** Audience
- TV-Y7-FV = suitable for **7+ aged** children (**Fantacy Violence**)
- NC-17 = Not Suitable for children **under 17**
- UR = **UNRATED**
- 74 min
- 84 min
- 66 min

## Basic Observation on the Data

```
#DATA SHAPE:

df.shape
#Our data has 8807 Rows and 12 Columns
```

```
(8807, 12)
```

```
df.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

```
#DATA INFO:

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
#FINDING TOTAL NULL VALUES IN THE DATA:

print('Total Null Values:',df.isna().sum().sum())
```

```
Total Null Values: 4307
```

```
# FINDING STAISTICAL DATA FOR THE GIVEN NETFLIX DATA:

df.describe(include='all')
```

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 8807 | 8807 | 8807 | 6173 | 7982 | 7976 | 8797 | 8807.000000 | 8803 | 8804 | 8807 | 8807 |
| **unique** | 8807 | 2 | 8807 | 4528 | 7692 | 748 | 1767 | NaN | 17 | 220 | 514 | 8775 |
| **top** | s8807 | Movie | Zubaan | Rajiv Chilaka | David Attenborough | United States | January 1, 2020 | NaN | TV-MA | 1 Season | Dramas, International Movies | Paranormal activity at a lush, abandoned prope... |
| **freq** | 1 | 6131 | 1 | 19 | 19 | 2818 | 109 | NaN | 3207 | 1793 | 362 | 4 |
| **mean** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 2014.180198 | NaN | NaN | NaN | NaN |
| **std** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 8.819312 | NaN | NaN | NaN | NaN |
| **min** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 1925.000000 | NaN | NaN | NaN | NaN |
| **25%** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 2013.000000 | NaN | NaN | NaN | NaN |
| **50%** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 2017.000000 | NaN | NaN | NaN | NaN |
| **75%** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 2019.000000 | NaN | NaN | NaN | NaN |

- There is only 1 column with the data type as 'integer' and rest all other columns are Categorical.

```
# FINDINNG THE NUMBER OF UNIQUE VALUES IN ALL THE COLUMNS IN THE DATA:

df.nunique()
```

| | 0 |
|---|---|
| **show_id** | 8807 |
| **type** | 2 |
| **title** | 8807 |
| **director** | 4528 |
| **cast** | 7692 |
| **country** | 748 |
| **date_added** | 1767 |
| **release_year** | 74 |
| **rating** | 17 |
| **duration** | 220 |
| **listed_in** | 514 |
| **description** | 8775 |

**dtype:** int64

## FINDING THE NULL VALUES AND FILLING THEM WITH APPROPRIATE FILLER:

```
# FINDING NULL VALUES

df.isna().sum() #columns-wise
```

| | 0 |
|---|---|
| **show_id** | 0 |
| **type** | 0 |
| **title** | 0 |
| **director** | 2634 |
| **cast** | 825 |
| **country** | 831 |
| **date_added** | 10 |
| **release_year** | 0 |
| **rating** | 4 |
| **duration** | 3 |
| **listed_in** | 0 |
| **description** | 0 |

**dtype:** int64

```
#FILLING THE NULL VALUES:

df['director'].fillna('Unknown', inplace=True)
df['country'].fillna('Unknown', inplace = True)
df['rating'].fillna('Not Available', inplace = True)
df['duration'].fillna('Not Available', inplace = True)
df['date_added'].fillna('Unavailable', inplace = True)

df['cast'].fillna('Not Available', inplace = True)

df.isna().sum()
```

| | 0 |
|---|---|
| **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

**Note:** We need to differentiate the rows of Cast Column as we have Documentaries as well which usually dont have the cast (assumption).

```
# CREATING THE FUNCTION FOR REPLACING THE VALUES FOR THE ROWS OF CAST FOR DOCUMENTARIES:

def cast_filling(x):
  if x['listed_in'] == 'Documentaries':
    x['cast'] = 'No Cast Required'
  return x['cast']
```

```
df['cast'] = df.apply(cast_filling, axis = 1)
```

```
df['cast']
```

| | cast |
|---|---|
| 0 | No Cast Required |
| 1 | Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban... |
| 2 | Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi... |
| 3 | Not Available |
| 4 | Mayur More, Jitendra Kumar, Ranjan Raj, Alam K... |
| ... | ... |
| 8802 | Mark Ruffalo, Jake Gyllenhaal, Robert Downey J... |
| 8803 | Not Available |
| 8804 | Jesse Eisenberg, Woody Harrelson, Emma Stone, ... |
| 8805 | Tim Allen, Courteney Cox, Chevy Chase, Kate Ma... |
| 8806 | Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan... |

8807 rows × 1 columns

**dtype:** object

# TOP 10 RATINGS GIVEN TO THE MOVIES/TV-SHOWS:

```
df['rating'].value_counts().head(10)
```

| rating | count |
|---|---|
| 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 |

**dtype:** int64

- The majority of content on Netflix is rated **TV-MA** (Mature Audience), with **over 3,200 titles**, followed by **TV-14 (suitable for ages 14+)** and **TV-PG (Parental Guidance suggested)**. This indicates that Netflix heavily focuses on content for older teens and adults, catering primarily to a mature audience segment.
- Family-oriented ratings like **TV-Y, TV-Y7, PG, and G have noticeably fewer titles**, suggesting that kids' and general family content is underrepresented. This presents an opportunity for Netflix to diversify its library by investing more in content suitable for younger audiences.

**Converting the 'date_added' Column into a DATETIME datatype:**

```
# CREATING A FUNCTION FOR CONVERTING THE DATA INTO A DATETIME DTYPE:

def datetime(x):
  if x['date_added'] == 'Unavailable':
    return 'Unavailable'
  else:
    return pd.to_datetime(x['date_added'], format = '%B %d, %Y')

```

```
df['date_added'] = df['date_added'].str.strip()
df['date_added'] = df.apply(datetime, axis = 1)
df['date_added'] = pd.to_datetime(df['date_added'], errors  = 'coerce')
```

```
df['month_added'] = df['date_added'].dt.month_name()
df['year_added'] = df['date_added'].dt.year
```

```
df.head(2)
```

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description | month_added |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | s1 | Movie | Dick Johnson Is Dead | Kirsten Johnson | No Cast Required | United States | 2021-09-25 | 2020 | PG-13 | 90 min | Documentaries | As her father nears the end of his life, filmm... | September |
| | | | | | Ama Qamata | | | | | | | After | |

Next steps: Generate code with df | View recommended plots | New interactive sheet

WE NEED TO BIFURCATE TEH DURATION FOR BOTH 'TV SHOWS' AND 'MOVIES':

```
df['movie_min'] = df[df['type']=='Movie']['duration'].apply(lambda x: x.split(' ')[0])
df['season_count'] = df[df['type']=='TV Show']['duration'].apply(lambda x: x.split(' ')[0])
```

```
df.head()
```

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description | month_add |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | s1 | Movie | Dick Johnson Is Dead | Kirsten Johnson | No Cast Required | United States | 2021-09-25 | 2020 | PG-13 | 90 min | Documentaries | As her father nears the end of his life, filmm... | Septemb |
| **1** | s2 | TV Show | Blood & Water | Unknown | Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban... | South Africa | 2021-09-24 | 2021 | TV-MA | 2 Seasons | International TV Shows, TV Dramas, TV Mysteries | After crossing paths at a party, a Cape Town t... | Septemb |
| **2** | s3 | TV Show | Ganglands | Julien Leclercq | Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi... | Unknown | 2021-09-24 | 2021 | TV-MA | 1 Season | Crime TV Shows, International TV Shows, TV Act... | To protect his family from a powerful drug lor... | Septemb |
| **3** | s4 | TV Show | Jailbirds New Orleans | Unknown | Not Available | Unknown | 2021-09-24 | 2021 | TV-MA | 1 Season | Docuseries, Reality TV | Feuds, flirtations and toilet talk go down amo... | Septemb |
| **4** | s5 | TV Show | Kota Factory | Unknown | Mayur More, Jitendra Kumar, Ranjan Raj, Alam K | India | 2021-09-24 | 2021 | TV-MA | 2 Seasons | International TV Shows, Romantic TV Shows, TV ... | In a city of coaching centers known to train I... | Septemb |

Next steps: Generate code with df | View recommended plots | New interactive sheet

SINCE **'DESCRIPTION'** COLUMN IS NOT REQUIRED FOR US TO GET QUANTITATIVE INSIGHTS, WE WILL DROP THAT COLUMN:

```
df.drop('description', axis=1, inplace=True)
```

## NUMBER OF MOVIES/TV-SHOWS MADE BY DIRECTORS:

```
directors = df['director'].str.split(', ').explode()
directors.value_counts()
```

| | count |
|---|---|
| **director** | |
| **Unknown** | 2634 |
| **Rajiv Chilaka** | 22 |
| **Jan Suter** | 21 |
| **Raúl Campos** | 19 |
| **Suhas Kadav** | 16 |
| **...** | ... |
| **Zhang Yimou** | 1 |
| **Phillip Youmans** | 1 |
| **Pawan Kumar** | 1 |
| **Xavier Durringer** | 1 |
| **Scott McAboy** | 1 |

4994 rows × 1 columns

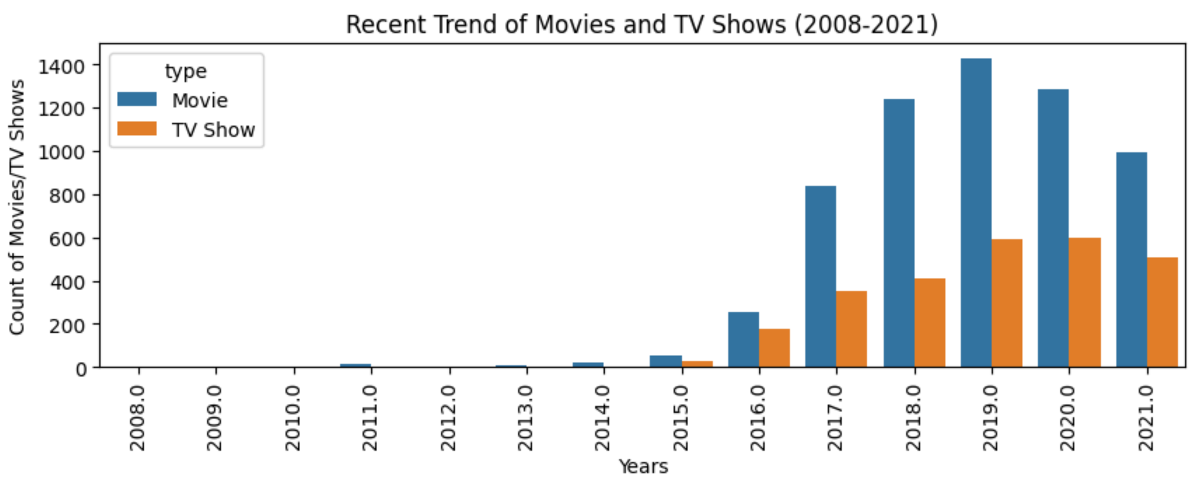
**dtype:** int64

**INSIGHTS:**

- The majority of entries (2,634 out of 4,994) list the director as "Unknown", indicating a large gap in metadata. This is due to missing information or content types like stand-up, reality TV, or documentaries where directors are less emphasized.
- Aside from a few recurring names like Rajiv Chilaka (22), Jan Suter (21), and Raúl Campos (19), most directors appear only once, showing that Netflix content spans a broad range of unique contributors, possibly due to global acquisitions.

## RECENT TREND OF ADDDING OF MOVIE/TV-SHOW ON NETFLIX:

```python
recent_trend = df.groupby('year_added')['type'].value_counts().reset_index()

plt.figure(figsize=(10,3))
sns.barplot(data = recent_trend, x = 'year_added',y='count', hue = 'type')
plt.xticks(rotation=90)
plt.xlabel('Years')
plt.ylabel('Count of Movies/TV Shows')
plt.title('Recent Trend of Movies and TV Shows (2008-2021)')
plt.show()
```

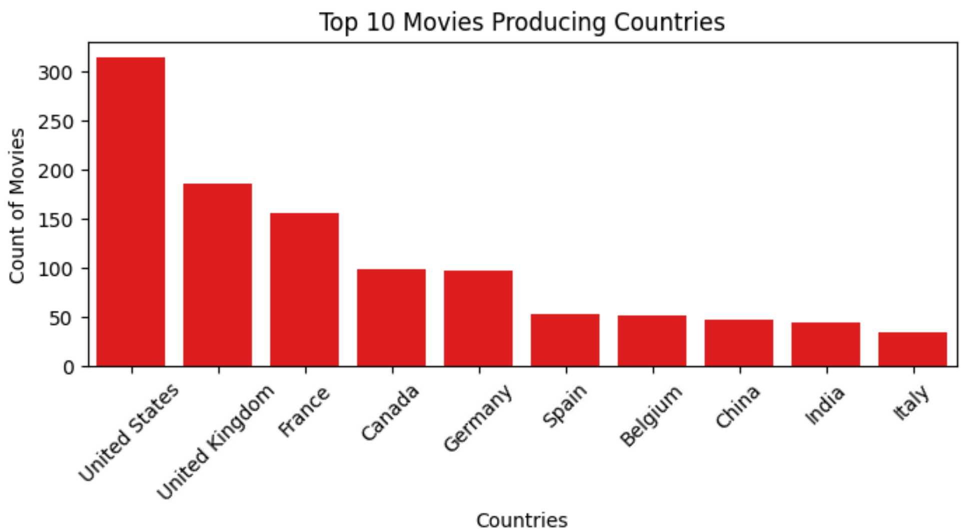



**KEY INSIGHTS:**

- The Platform has showed a massive growth after 2015.
- 2019 and 2020 were the **peak years** in terms of volume:
  - **2019**: 1424 movies and 592 TV shows.
  - **2020**: 1284 movies and 595 TV shows.
- Content volume dropped a bit in 2021, possibly due to pandemic-related production delays catching up.

## TOP 10 MOVIES PRODUCING COUNTRIES ON NETFLIX:

```
country_movie = df.groupby('country')['type'].value_counts()
country_movie = pd.DataFrame(country_movie)
country_movie = country_movie.reset_index()
country_movie.rename(columns = {'type':'Type', 'country':'Country', 'count':'Count'}, inplace = True)
country_movie['Country'] = country_movie['Country'].str.strip()
country_movie['Country'] = country_movie['Country'].str.split(', ')
country_movie = country_movie.explode(column = 'Country')
country_movie.groupby('Country')['Type'].value_counts()

movies_per_country = country_movie.groupby('Type')['Country'].get_group('Movie').value_counts().head(10)

plt.figure(figsize=(8,3))
sns.barplot(data = movies_per_country, color = 'red')
plt.xticks(rotation = 45)
plt.title('Top 10 Movies Producing Countries')
plt.xlabel('Countries')
plt.ylabel('Count of Movies')
plt.show()
```

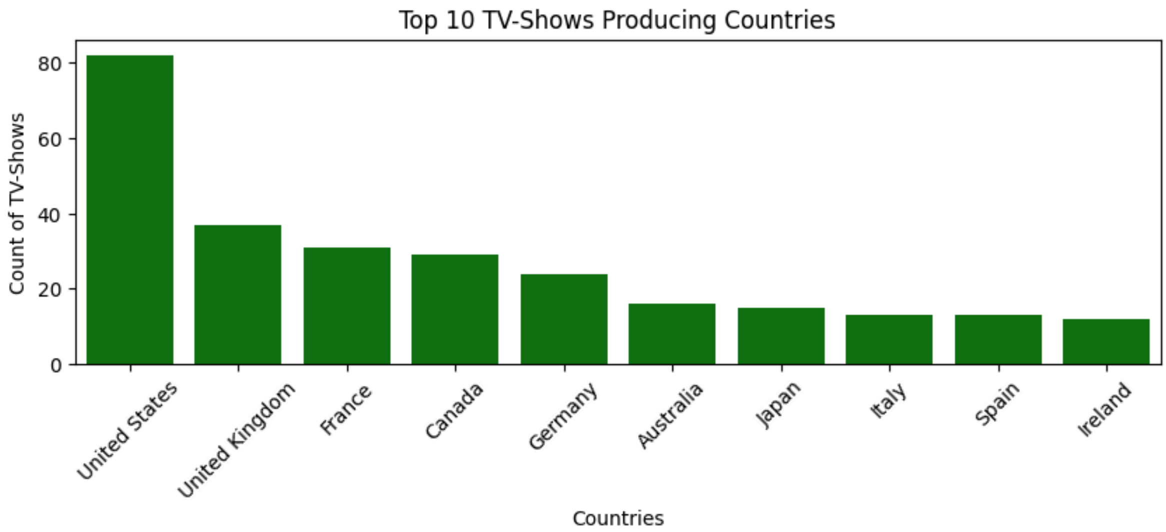



**KEY INSIGHTS:**

- The **United States** alone contributes more than **17%** of all movies on the platform, reflecting Netflix's origin and content production base.
- UK, France, Germany, and Canada together contribute significantly to Netflix's global catalog, indicating strong partnerships or licensing from these regions.
- **6 of the top 10 countries are European**, showing that Europe is a major source of diverse, localized content.

## TOP 10 TV-SHOWS PRODUCING COUNTRIES ON NETFLIX:

```
country_tv = df.groupby('country')['type'].value_counts()
country_tv = pd.DataFrame(country_tv)
country_tv = country_tv.reset_index()
country_tv.rename(columns = {'type':'Type', 'country':'Country', 'count':'Count'}, inplace = True)
country_tv['Country'] = country_tv['Country'].str.strip()
country_tv['Country'] = country_tv['Country'].str.split(', ')
country_tv = country_tv.explode(column = 'Country')
country_tv.groupby('Country')['Type'].value_counts()

shows_per_country = country_tv.groupby('Type')['Country'].get_group('TV Show').value_counts().head(10)

plt.figure(figsize=(10,3))
sns.barplot(data = shows_per_country, color = 'green')
plt.xticks(rotation = 45)
plt.title('Top 10 TV-Shows Producing Countries')
plt.xlabel('Countries')
plt.ylabel('Count of TV-Shows')
plt.show()
```




**KEY INSIGHTS:**

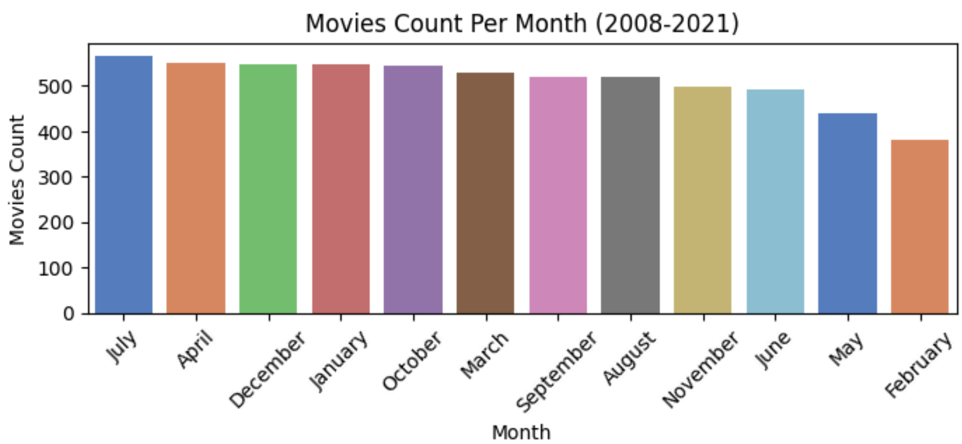
- Again, **United States** alone contributes nearly **one-fifth** of all the TV-Shows on the platform, reflecting Netflix's origin and content production base.
- Compared to movies, Asian countries like India, South Korea, or China are notably absent here—indicating a potential growth area for regional or dubbed/subtitled series.

# WHICH IS THE BEST TIME TO LAUNCH A MOVIE/TV-SHOW?

## BEST MONTH TO LAUNCH A MOVIE:

```
df['day_added'] = df['date_added'].dt.day_name()
df['week_added'] = df['date_added'].dt.isocalendar().week
```

```
movie_data = df.groupby('type').get_group('Movie')
movie_month_data = movie_data.groupby('month_added')['type'].count()
movie_month_data = pd.DataFrame(movie_month_data)
movie_month_data.reset_index(inplace=True)
movie_month_data.rename(columns = {'type':'Movies Count'}, inplace = True)
movie_month_data = movie_month_data.sort_values(by = 'Movies Count',ascending=False)

plt.figure(figsize=(8,2.5))
sns.barplot(data = movie_month_data, x='month_added', y = 'Movies Count', hue = 'month_added', palette = 'muted')
plt.xticks(rotation=45)
plt.xlabel('Month')
plt.ylabel('Movies Count')
plt.title('Movies Count Per Month (2008-2021)')
plt.show()
```




**KEY INSIGHTS:**

- July is the most popular month, with the highest number of movie additions.
- April, December, January, and October closely follow, all with over 530 movies added. These months may align with seasonal trends like summer vacations or holiday periods when viewers are more active.
- February has the lowest movie count, under 400, indicating it's the least utilized month for new releases.

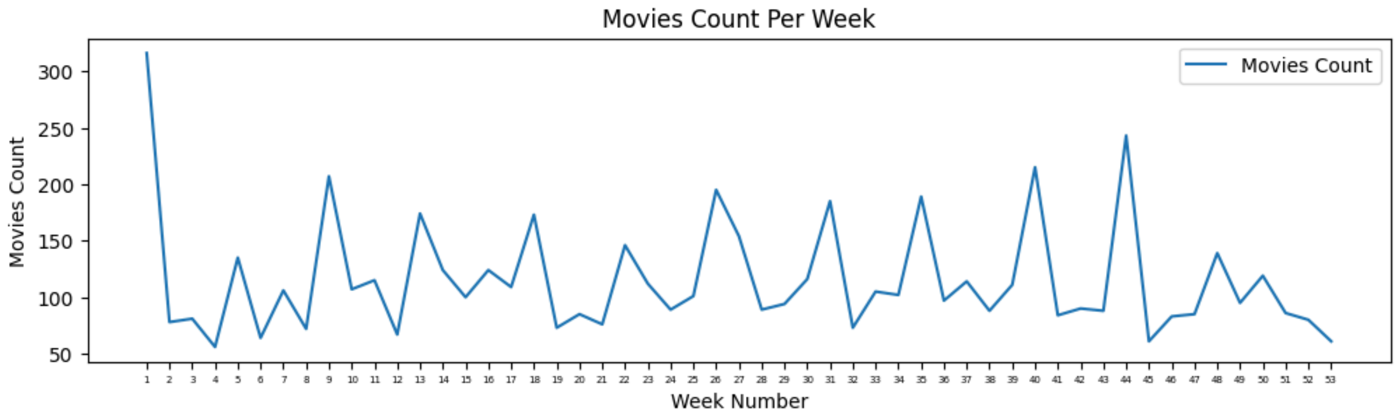
- May and June also show relatively lower activity.

**STRATEGIC IMPLICATIONS:**

- Netflix could analyze why July and the winter months are performing well—perhaps due to holidays, weather, or school breaks—and focus marketing or big releases during these peaks.
- Conversely, February and May might benefit from exclusive or niche content to boost engagement during slower periods.

## BEST WEEK TO LAUNCH A MOVIE:

```
movie_week_data = movie_data.groupby('week_added')['type'].count()
movie_week_data = pd.DataFrame(movie_week_data)
movie_week_data.reset_index()
movie_week_data.rename(columns = {'type':'Movies Count'}, inplace = True)
movie_week_data
plt.figure(figsize=(12,3))
sns.lineplot(data=movie_week_data)
plt.xticks(range(1,54), fontsize=5)
plt.xlabel('Week Number')
plt.ylabel('Movies Count')
plt.title('Movies Count Per Week')
plt.show()
```

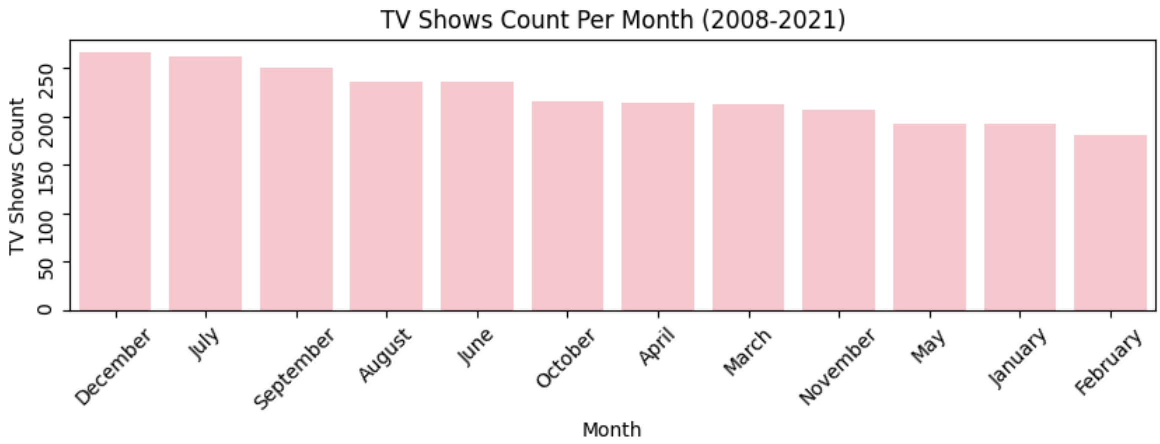



**KEY INSIGHTS**

- **Week 1 (316 movies)** has by far the highest count — likely due to year-end content dumps or New Year strategies.
- **Week 44 (243 movies) and Week 40 (215 movies)** are also strong, indicating **Q4 (Oct–Dec) is a major release window**, possibly targeting holiday and festive seasons.
- Weeks 4 (56), 6 (64), and 2 (78) show lower activity, often early in the year after the initial January push.
- **Week 53 (61 movies) is low**, likely due to it being a partial week in most years.
- There's a spike around Weeks 9, 13, 26, 31, 35, and 44, suggesting:
    - Quarter-end or mid-year surges (Weeks 13, 26)
    - Strategic content drops mid-summer and pre-holiday (Weeks 31–35, 44)

## BEST MONTH TO LAUNCH A TV-SHOW:

```
tv = df.groupby('type').get_group('TV Show')
tv_data = pd.DataFrame(tv.groupby('month_added')['type'].count())
tv_data.reset_index(inplace=True)
tv_data.rename(columns = {'type':'TV Shows Count'}, inplace = True)
tv_data = tv_data.sort_values(by = 'TV Shows Count',ascending=False)
#tv_data
plt.figure(figsize=(10,2.5))
sns.barplot(data = tv_data, x='month_added', y = 'TV Shows Count', color='pink')
plt.xticks(rotation=45)
plt.yticks(rotation=90)
plt.xlabel('Month')
plt.ylabel('TV Shows Count')
plt.title('TV Shows Count Per Month (2008-2021)')
plt.show()
```

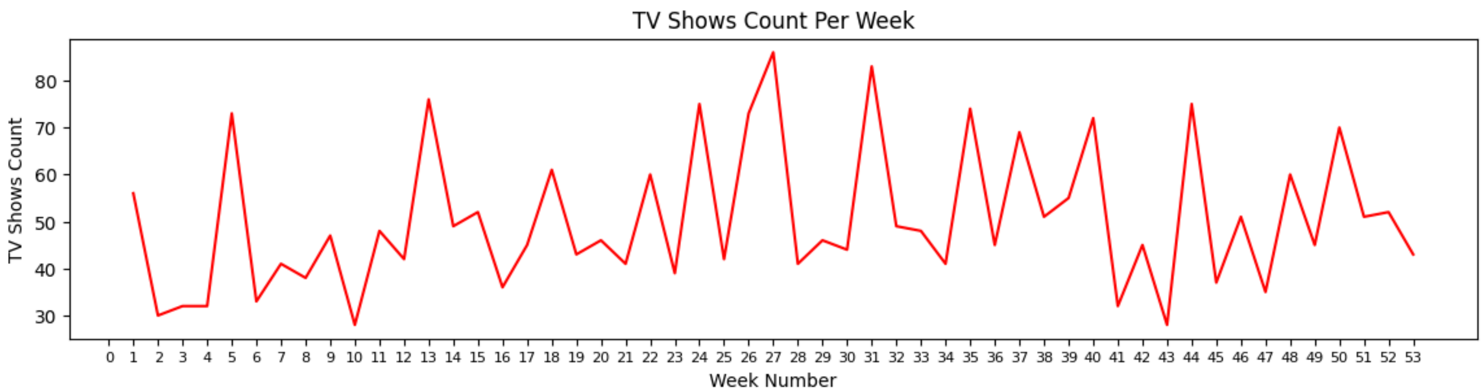



**KEY INSIGHTS:**

- December has the highest number of TV show additions (266), likely due to holiday season releases.
- February has the lowest count (181), possibly because it's the shortest month.
- Overall, Q4 (Oct–Dec) tends to have more TV show releases than other quarters.

## BEST WEEK TO LAUNCH A TV-SHOW:

```
tv_week_data = tv.groupby('week_added')['type'].count()
tv_week_data = pd.DataFrame(tv_week_data)
tv_week_data = tv_week_data.reset_index()
tv_week_data
tv_week_data.rename(columns = {'type':'TV Shows Count'}, inplace = True)
#tv_week_data
plt.figure(figsize=(14,3))
sns.lineplot(data=tv_week_data,x='week_added', y='TV Shows Count', color='red')
plt.xticks(range(0,54), fontsize=8)
plt.xlabel('Week Number')
plt.ylabel('TV Shows Count')
plt.title('TV Shows Count Per Week')
plt.show()
```




**KEY INSIGHTS:**

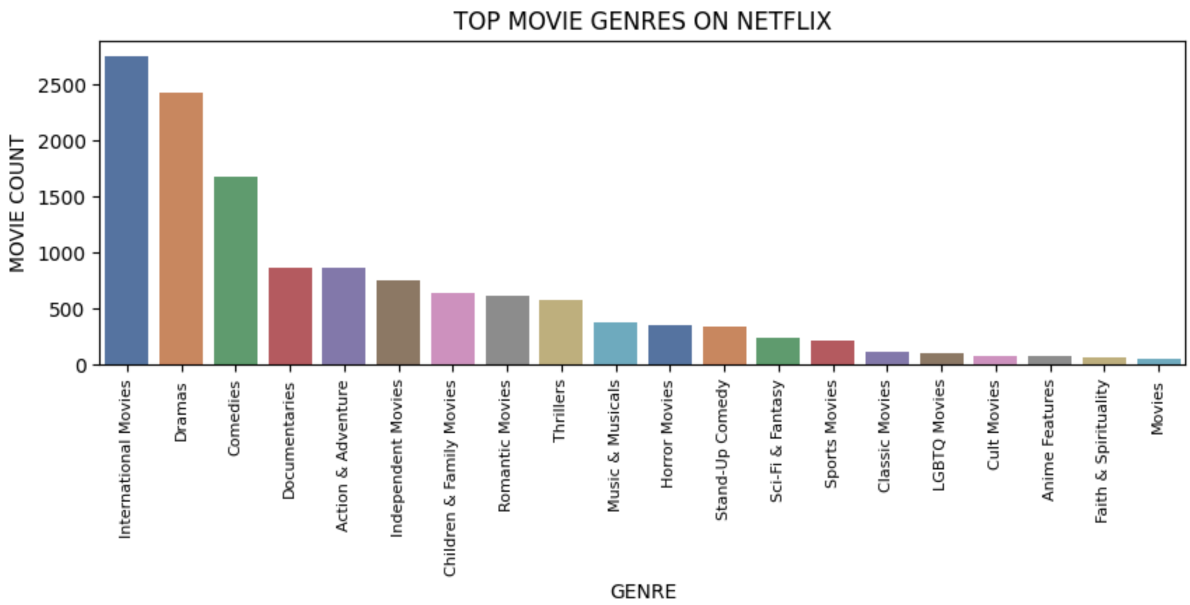
- Weeks 27, 31, and 13 saw the highest number of TV shows added — 86, 83, and 76 respectively — indicating mid-year spikes, likely tied to summer content releases or mid-year platform strategies.
- Weeks 10 and 43 had the lowest counts (28), suggesting possible off-peak periods or slower content rollout in those weeks.
- There's a general increase in content additions from week 24 to week 35, pointing to a trend where streaming services ramp up content in the second quarter to mid-third quarter.

## TOP 'MOVIE/TV-SHOW GENRES' ON THE NETFLIX IN THE LAST 10 YEARS

## TOP MOVIE GENRES:

```
genre_movie = movie_data['listed_in'].apply(lambda x: x.split(', ')).tolist()
genre_movie = pd.DataFrame(genre_movie, index=movie_data['title']).stack()
genre_movie = genre_movie.reset_index()
genre_movie.columns = ['title', 'level_1', 'genre']
```

```
genre_movie.drop('level_1', axis=1, inplace=True)
genre_movie = genre_movie[['title', 'genre']]
#genre_movie.head()

a = genre_movie['genre'].value_counts().sort_values(ascending=False)
a = pd.DataFrame(a)
plt.figure(figsize=(10,3))
sns.barplot(data=a, x='genre', y='count', palette='deep')
plt.xticks(rotation=90, fontsize=8)
plt.xlabel('GENRE')
plt.ylabel('MOVIE COUNT')
plt.title('TOP MOVIE GENRES ON NETFLIX')
plt.show()
```

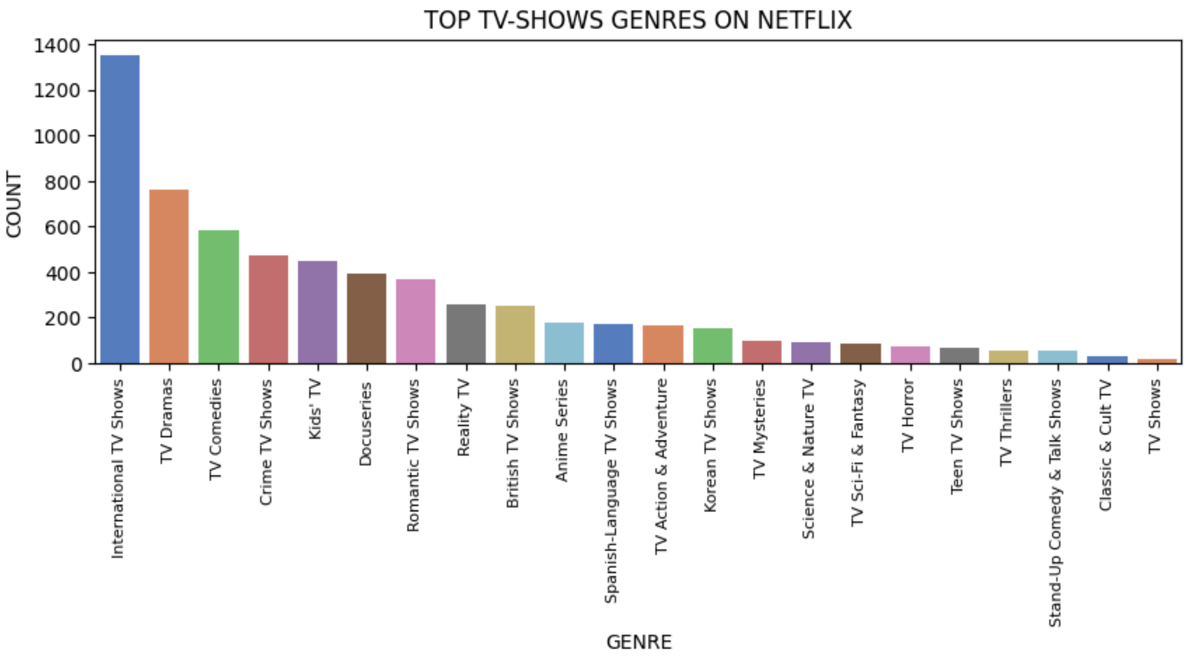



**KEY INSIGHTS:**

- **International Movies** lead with 2752 titles, showing Netflix's strong global content strategy.
- **Dramas and Comedies** are the most popular mainstream genres, highlighting user preference for emotional and lighthearted stories.
- **Documentaries and Action & Adventure** also rank high, indicating audience interest in both real-world topics and thrilling entertainment.

## TOP TV-SHOW GENRES:

```
genre_tv = tv['listed_in'].apply(lambda x: x.split(', ')).tolist()
genre_tv = pd.DataFrame(genre_tv, index=tv['title']).stack()
genre_tv = genre_tv.reset_index()
genre_tv.columns = ['title', 'level_1', 'genre']
genre_tv.drop('level_1', axis=1, inplace=True)
genre_tv = genre_tv[['title', 'genre']]
#genre_tv.head()

a = genre_tv['genre'].value_counts().sort_values(ascending=False)
a = pd.DataFrame(a)
plt.figure(figsize=(10,3))
sns.barplot(data=a, x='genre', y='count', palette='muted')
plt.xticks(rotation=90, fontsize=8)
plt.xlabel('GENRE')
plt.ylabel('COUNT')
plt.title('TOP TV-SHOWS GENRES ON NETFLIX')
plt.show()
```




**KEY INSIGHTS:**

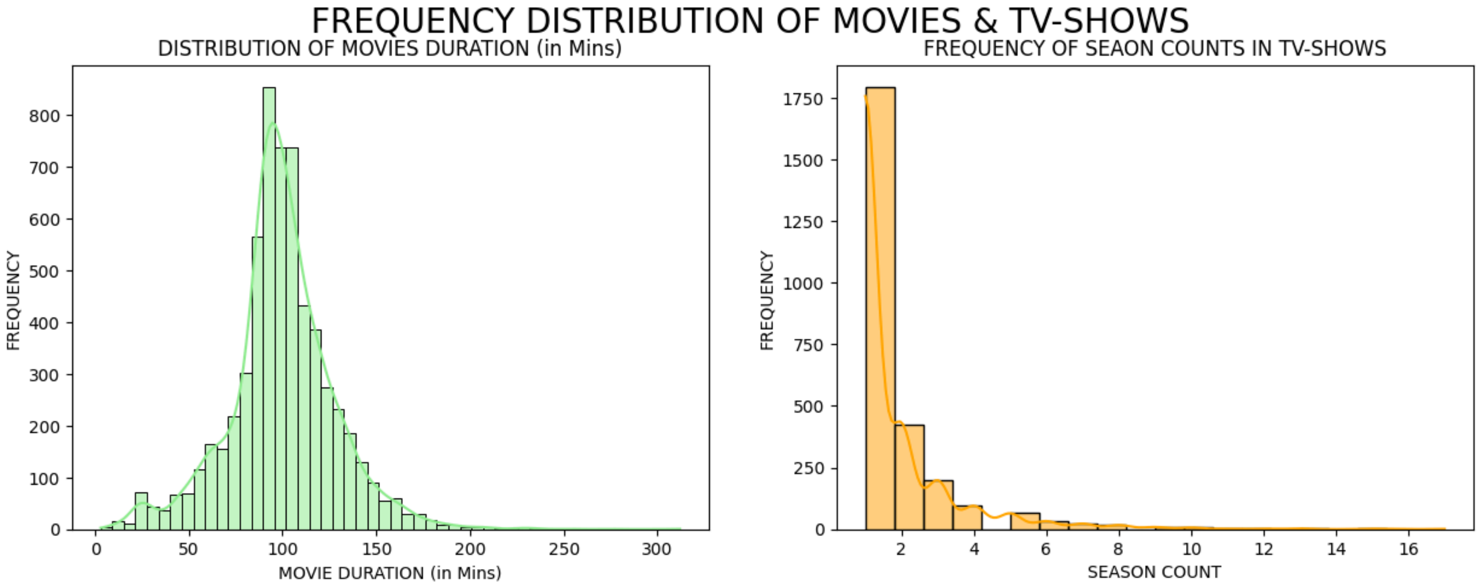
- **International and Drama-based TV shows** dominate Netflix's catalog, indicating a strong global strategy and viewer preference for story-rich content.
- **Kids', Crime, and Anime genres** are heavily featured, catering to both family-friendly and niche, binge-worthy interests.
- Underrepresented genres like **Horror, Sci-Fi, and Classic TV** suggest potential growth areas for diversifying the content portfolio.

# DISTRIBUTION OF TIME IN MOVIES/TV-SHOWS ON NETFLIX:

## TIME DISTRIBUTION OF MOVIES:

```
# Convert 'movie_min' column to numeric, handling errors
movie_data['movie_min'] = pd.to_numeric(movie_data['movie_min'], errors='coerce')
# Convert 'movie_min' column to numeric, handling errors
tv['season_count'] = pd.to_numeric(tv['season_count'], errors='coerce')

# Drop rows with missing values in 'movie_min'
movie_data.dropna(subset=['movie_min'], inplace=True)
# Drop rows with missing values in 'movie_min'
tv.dropna(subset=['season_count'], inplace=True)


fig = plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
sns.histplot(movie_data['movie_min'], bins=50,kde=True, color='lightgreen', edgecolor='black')
plt.xlabel('MOVIE DURATION (in Mins)')
plt.ylabel('FREQUENCY')
plt.title('DISTRIBUTION OF MOVIES DURATION (in Mins)')

#plt.figure(figsize=(10,10))
plt.subplot(1,2,2)
sns.histplot(tv['season_count'], bins=20, kde=True, color='orange', edgecolor='black')
plt.xlabel('SEASON COUNT')
plt.ylabel('FREQUENCY')
plt.title('FREQUENCY OF SEAON COUNTS IN TV-SHOWS')


fig.suptitle('FREQUENCY DISTRIBUTION OF MOVIES & TV-SHOWS', fontsize=20)
```

```
Text(0.5, 0.98, 'FREQUENCY DISTRIBUTION OF MOVIES & TV-SHOWS')
```




**Movie Duration Distribution:**

- Most movies on Netflix have durations centered around 90–110 minutes, forming a near-normal distribution with a slight right skew, indicating the presence of some longer movies (up to ~300 minutes) but relatively fewer in number.

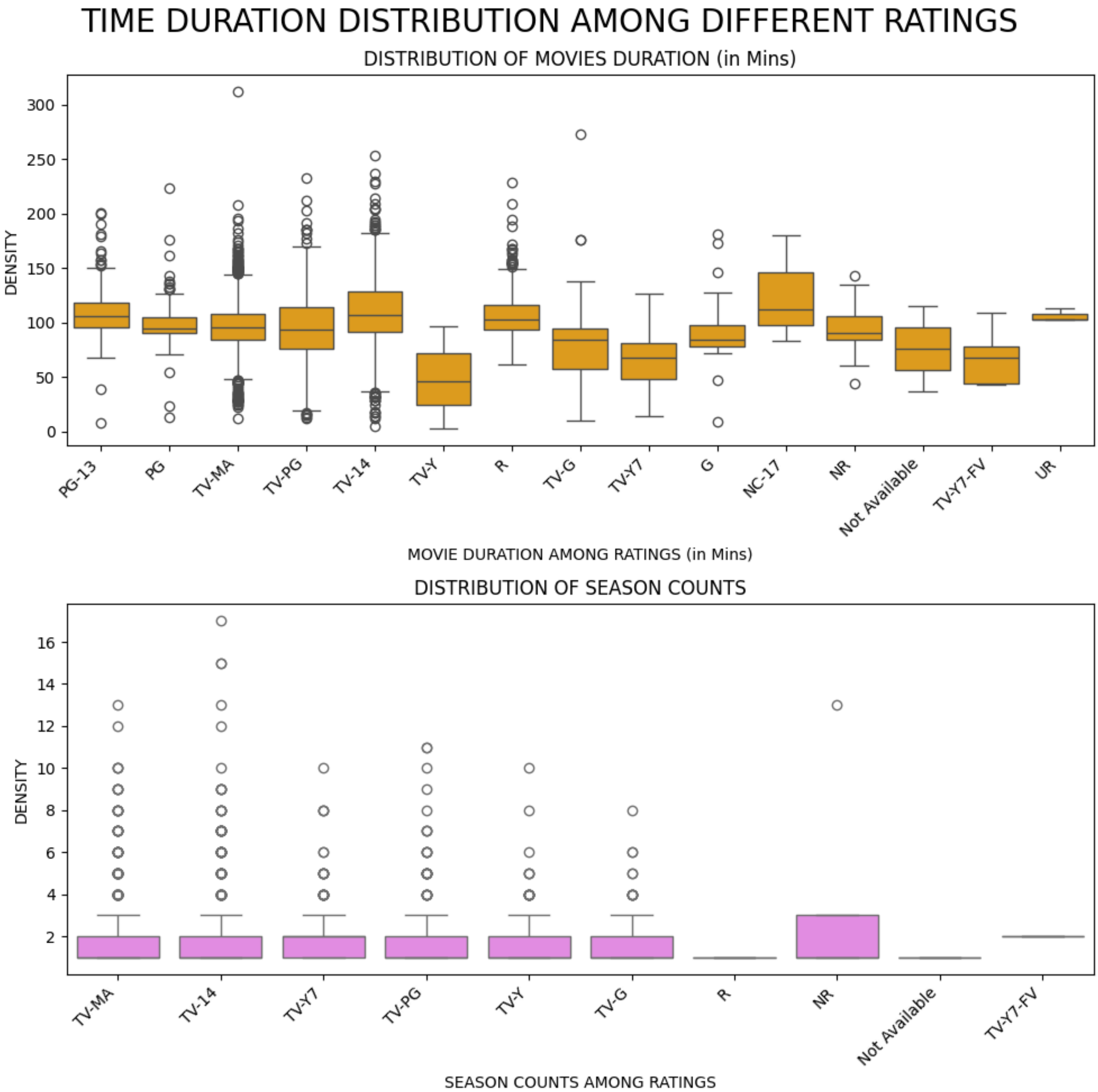
**TV Show Season Count Distribution:**

- A significant majority of TV shows have just 1 or 2 seasons, with frequency dropping sharply beyond 3 seasons.
- The distribution is heavily right-skewed, indicating that long-running shows (more than 5 seasons) are rare on Netflix.

# DISTRIBUTION OF TIME IN MOVIES/TV-SHOWS AMONG DIFFERENT RATINGS:

**FINDING OUT THE OUTLIERS**

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

plt.subplot(2, 1, 1)
molten_movie_data = movie_data.melt(id_vars=['rating'], value_vars=['movie_min'])
sns.boxplot(x='rating', y='value', data=molten_movie_data, color='orange')
plt.xlabel('MOVIE DURATION AMONG RATINGS (in Mins)')
plt.ylabel('DENSITY')
plt.title('DISTRIBUTION OF MOVIES DURATION (in Mins)')
plt.xticks(rotation=45, ha='right')

molten_tv_data = tv.melt(id_vars=['rating'], value_vars=['season_count'])
plt.subplot(2, 1, 2)
sns.boxplot(data = molten_tv_data,x='rating', y='value', color='violet')
plt.xlabel('SEASON COUNTS AMONG RATINGS')
plt.ylabel('DENSITY')
plt.title('DISTRIBUTION OF SEASON COUNTS')
plt.xticks(rotation=45, ha='right')

fig.suptitle('TIME DURATION DISTRIBUTION AMONG DIFFERENT RATINGS', fontsize=20)
plt.tight_layout()
plt.show()
```




**Movies Duration by Rating:**

- 'R', 'TV-MA', and 'TV-14' rated movies tend to have longer durations on average, often exceeding 100 minutes, and also show greater variability in length.
- 'TV-Y', 'TV-Y7', and 'TV-G' rated movies (generally for children) have shorter durations, with medians below 60 minutes, and fewer outliers.
- Most categories show a high number of outliers, indicating some very short or very long movies regardless of the rating.

**Season Counts by Rating:**

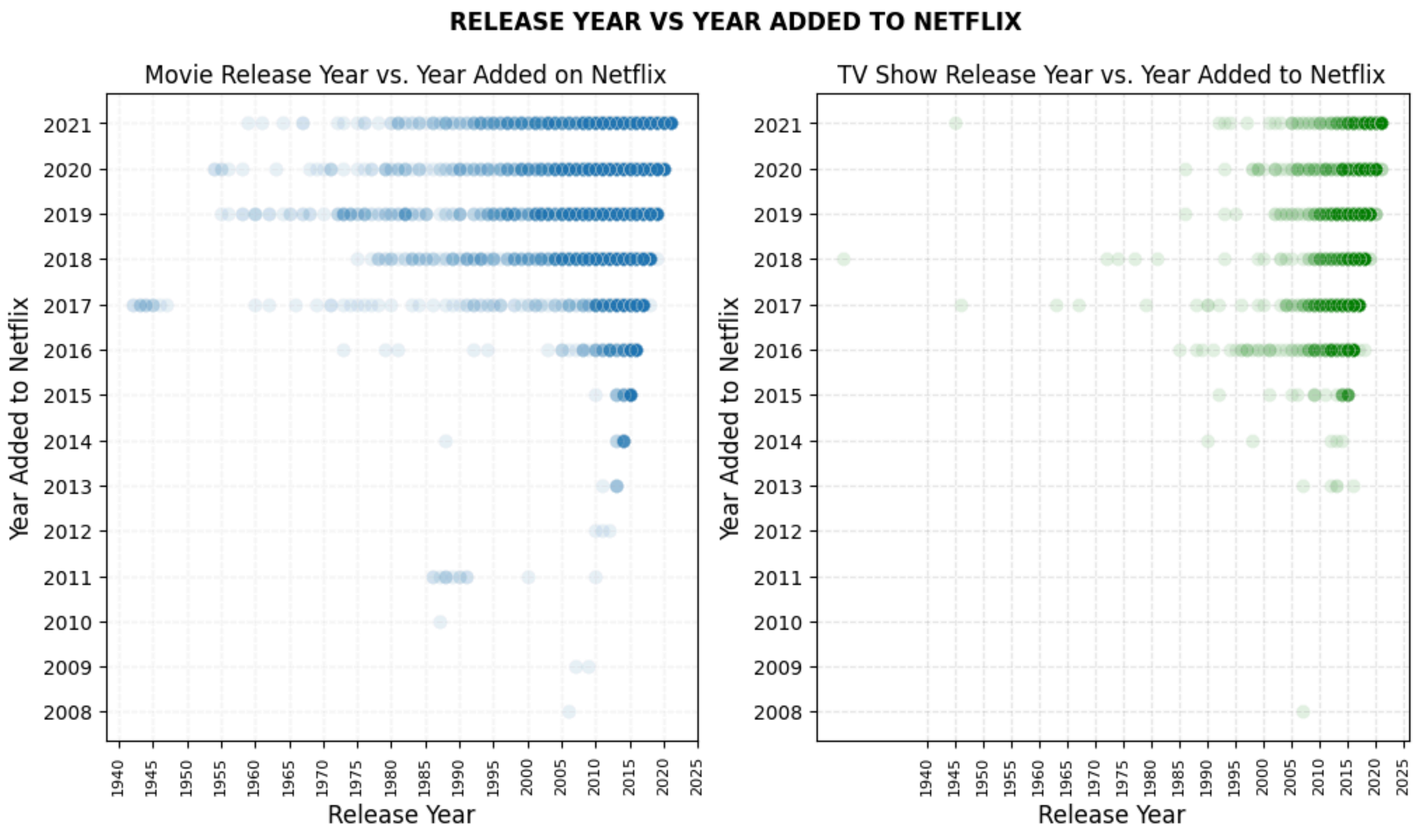
- 'TV-MA' and 'TV-14' have the widest range and most outliers, suggesting these ratings are common for longer-running series.
- 'TV-Y', 'TV-G', and 'TV-PG' rated shows tend to have fewer seasons, indicating they are more likely short series or limited-run kids' shows.
- Ratings like 'R', 'NR', and 'Not Available' are uncommon for TV shows, and their distributions are either narrow or inconsistent.

## RELATIONSHIP BETWEEN RELEASED YEAR AND YEAR ADDED TO THE NETFLIX:

```
fig = plt.figure(figsize=(12, 6))
plt.subplot(1,2,1)
sns.scatterplot(data=movie_data,
                x='release_year',
                y='year_added',
                s=50,
                alpha=0.1, marker='o')
plt.title(' Movie Release Year vs. Year Added on Netflix', fontsize=12)
plt.xticks(ticks=range(1940,2030,5), rotation=90, fontsize=8)
plt.yticks(ticks=range(2008,2022,1))
```

```
plt.xlabel('Release Year', fontsize=12)
plt.ylabel('Year Added to Netflix', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.1)

plt.subplot(1,2,2)
sns.scatterplot(data=tv,
                x='release_year',
                y='year_added',
                s=50,
                alpha=0.1, color='green')
plt.xticks(ticks=range(1940,2030,5), rotation=90, fontsize=8)
plt.title('TV Show Release Year vs. Year Added to Netflix', fontsize=12)
plt.xlabel('Release Year', fontsize=12)
plt.yticks(ticks=range(2008,2022,1))
plt.ylabel('Year Added to Netflix', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.3)

fig.suptitle('RELEASE YEAR VS YEAR ADDED TO NETFLIX',  fontweight='bold')
```

```
Text(0.5, 0.98, 'RELEASE YEAR VS YEAR ADDED TO NETFLIX')
```




Key Insights:

📈 2. Release Year vs. Year Added (Second Image: Separated by Movie & TV Show) Left Plot – Movies: Clustering in recent years (2015–2021) both in release and addition to Netflix.

Titles released even in the 1940s–2000s are being added mostly between 2016–2021.

The vertical lines show many older movies added in bulk during the same years (content licensing drives).

Right Plot – TV Shows: TV Shows are more recent in release years, with few before 2000.

Clear concentration from 2016–2021, with the majority released in the same decade.

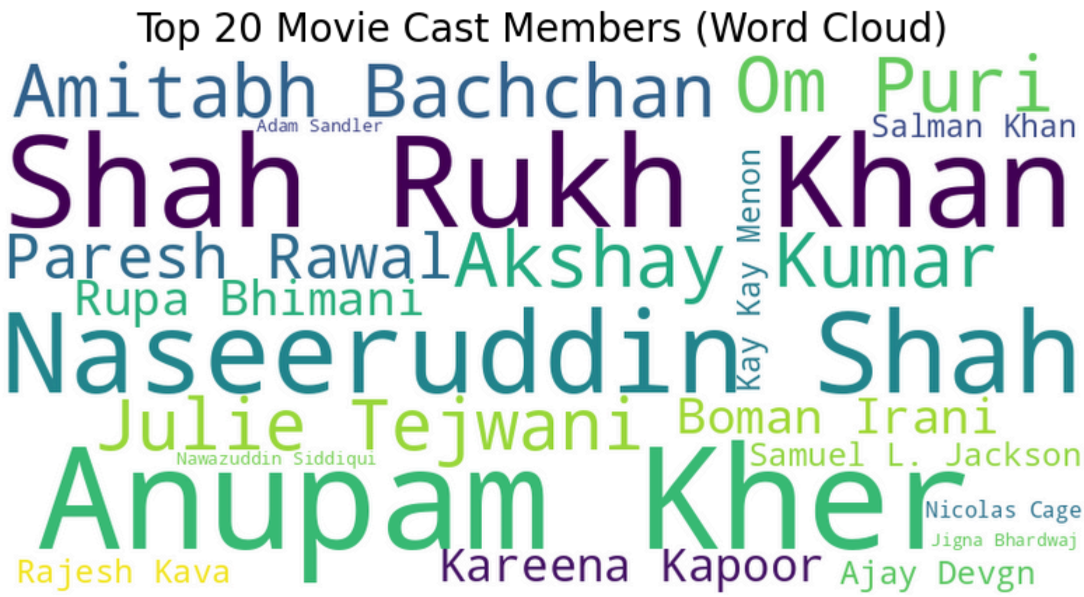
Fewer legacy or classic TV shows are being added compared to movies.

## TOP 20 MOVIE CAST ON NETFLIX

***WORD CLOUD PLOTTING***

```
cast = movie_data['cast'].apply(lambda x: x.split(', ')).tolist()
cast_movie = pd.DataFrame(cast, index=movie_data['title']).stack()
cast_movie = cast_movie.reset_index()
cast_movie.columns = ['title', 'level_1', 'cast']
cast_movie = cast_movie[['title', 'cast']]
#cast_movie
top_20_movie_cast = cast_movie['cast'].value_counts()
top_20_movie_cast.drop(['Not Available', 'No Cast Required'], inplace=True)
top_20_movie_cast = top_20_movie_cast.head(20)
```

```
#top_20_movie_cast

from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Use value_counts result as a dictionary
top_20_movie_cast = top_20_movie_cast.to_dict()

# Create WordCloud from frequencies
wordcloud = WordCloud(width=800,height=400,background_color='white').generate_from_frequencies(top_20_movie_cast)

# Plot the word cloud
plt.figure(figsize=(10,6))
plt.imshow(wordcloud, interpolation='lanczos')
plt.axis('off')
plt.title("Top 20 Movie Cast Members (Word Cloud)", fontsize=20)
plt.show()

```



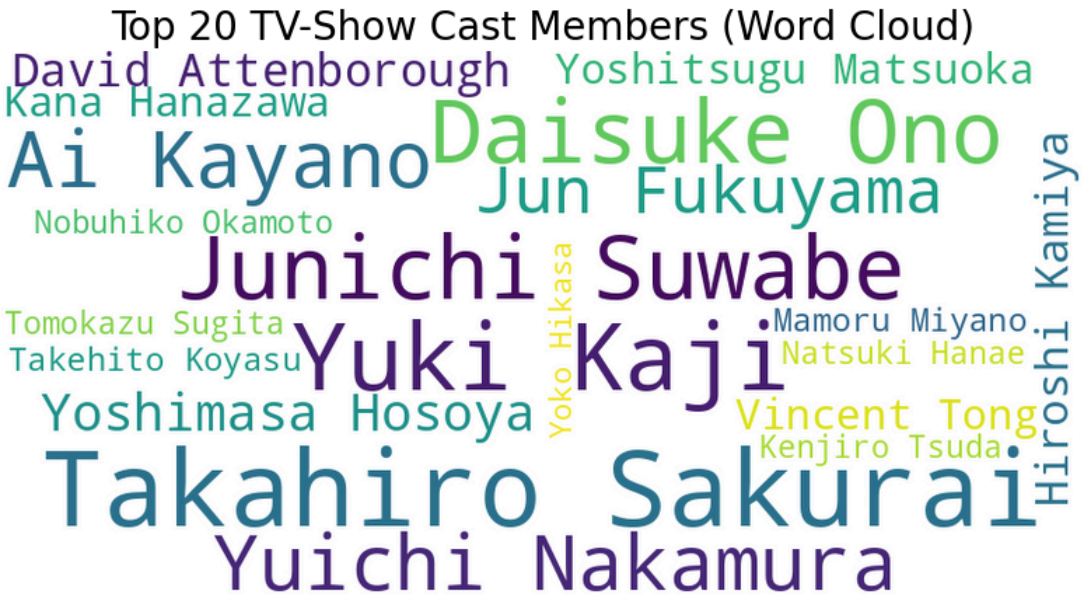

**KEY INSIGHTS:**

- Strong Dominance of Indian Actors
  - The majority of top recurring actors are from Bollywood and Indian cinema (for examlpe, Anupam Kher, Shah Rukh Khan, Naseeruddin Shah).
  - This reflects Netflix's large catalog of Indian films and shows.
- Low Female Representation in Top 20 Cast List
  - Only 3 female actors appear in the top 20: Julie Tejwani (28), Rupa Bhimani (27), and Kareena Kapoor (25).
  - This suggests a gender imbalance, with female actors making up just 15% of the top recurring cast — likely reflecting historical trends in film casting and availability on Netflix.
- Presence of Global Stars
  - A few international actors like Samuel L. Jackson and Nicolas Cage appear, but the list is mostly India-centric.
- Heavy Skew Toward a Few Names
  - The top 5 actors significantly outweigh others in number of appearances — a classic example of a long tail distribution in content casting.

## TOP 20 TV-SHOW CAST ON NETFLIX

```
cast_tv_data = tv['cast'].apply(lambda x: x.split(', ')).tolist()
cast_tv = pd.DataFrame(cast_tv_data, index=tv['title']).stack()
cast_tv = cast_tv.reset_index()
cast_tv.columns = ['title', 'level_1', 'cast']
cast_tv = cast_tv[['title', 'cast']]
#cast_tv
top_20_tv_cast = cast_tv['cast'].value_counts()
top_20_tv_cast.drop(['Not Available'], inplace=True)
top_20_tv_cast = top_20_tv_cast.head(20)
#top_20_tv_cast

from wordcloud import WordCloud
```

```
import matplotlib.pyplot as plt

# Use value_counts result as a dictionary
top_20_tv_cast = top_20_tv_cast.to_dict()

# Create WordCloud from frequencies
wordcloud = WordCloud(width=800,height=400,background_color='white').generate_from_frequencies(top_20_tv_cast)

# Plot the word cloud
plt.figure(figsize=(10,6))
plt.imshow(wordcloud, interpolation='lanczos')
plt.axis('off')
plt.title("Top 20 TV-Show Cast Members (Word Cloud)", fontsize=20)
plt.show()
```



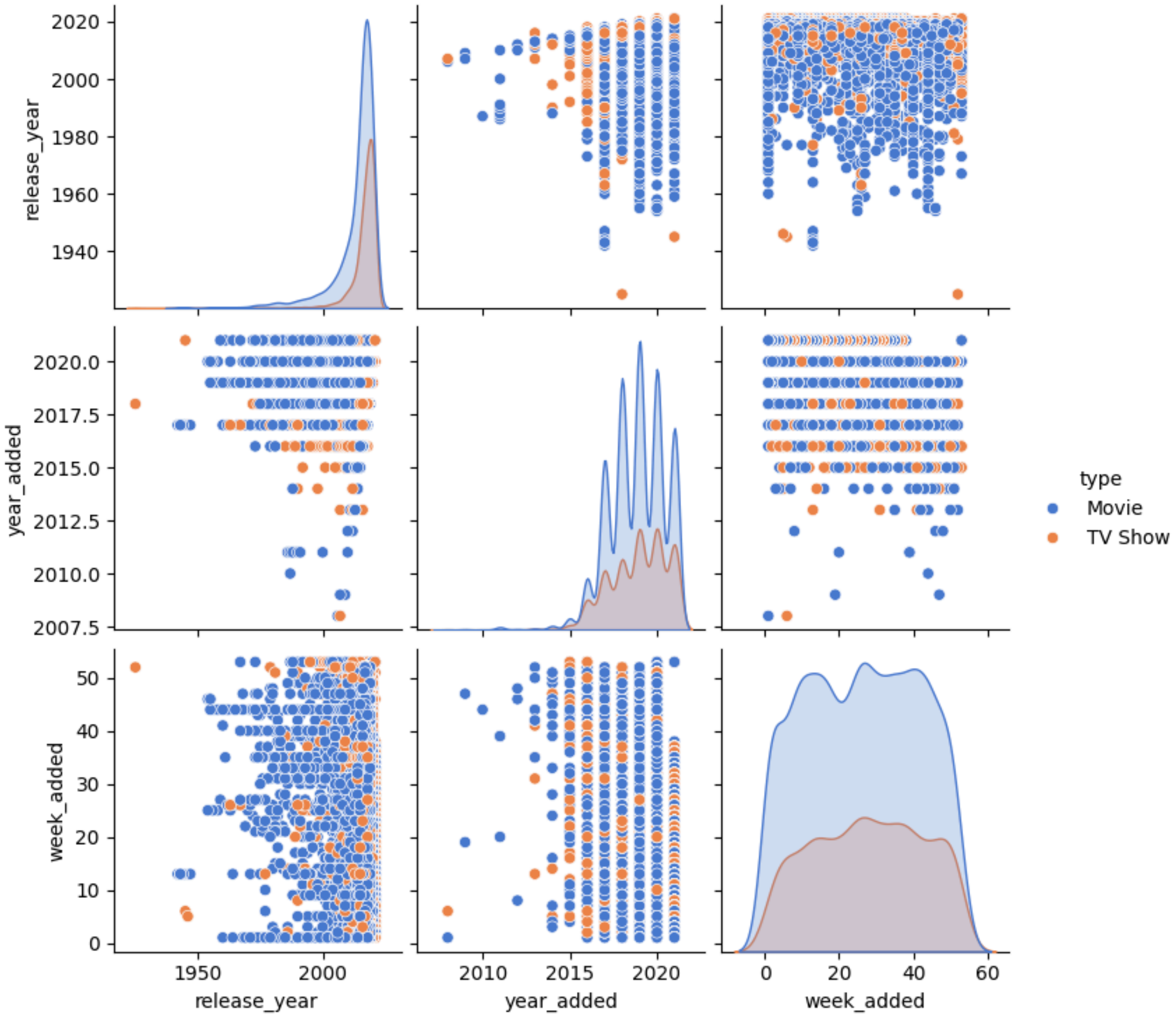

**KEY INSIGHTS:**

- Dominance of Japanese Voice Actors
  - The list is dominated by Japanese voice actors such as Takahiro Sakurai, Yuki Kaji, Daisuke Ono, and others.
  - This reflects the prominent role of anime content in Netflix's catalog, especially given Netflix's strong acquisition of anime shows and films from Japan.
- Presence of International Talent
  - The list includes global talent like David Attenborough (14 appearances), highlighting Netflix's diverse content beyond anime, including documentaries and nature series.
  - This shows Netflix's broad spectrum of content, from anime to global shows.
- Low Female Representation in TV Shows Cast
  - Only 3 female voice actors appear in the top 20: Ai Kayano (17), Kana Hanazawa (13), and Yoko Hikasa (11).
  - This suggests a gender imbalance, with female voice actors making up 15% of the top 20 cast members — reflecting the traditional dominance of male voice actors in anime content.
- Popular and Long-Standing Voice Actors
  - Many of these actors (e.g., Takahiro Sakurai, Yuki Kaji, and Daisuke Ono) are well-established names in the anime industry, suggesting that Netflix's anime catalog relies heavily on veteran voice talent who have been part of long-running or popular series.

## RELATIONSHIP b/w MULTIPLE VARIABLES GIVEN (Release Year, Year Added, Week No. Added)

### PAIRPLOT

```
sns.pairplot(df, hue='type', palette='muted')
plt.show()
```




**INSIGHTS:**

- Release Year Distribution:
  - Movies have a wider historical range, going back to the 1940s.
  - TV Shows are mostly post-2000, with a steep rise in recent years.
- Year Added to Netflix:
  - Both Movies and TV Shows saw a surge from 2016 to 2020, indicating Netflix's aggressive content expansion.
  - More recent additions dominate, with spikes visible in 2018–2020.
- Weekly Additions:
  - Clear seasonality or content addition patterns are seen, especially for Movies, which are added steadily throughout all weeks.
  - TV Shows have fewer additions overall and seem more clustered.
- Correlations:
  - There's a dense diagonal line between release_year and year_added, suggesting many titles are added shortly after release.
  - However, many older titles were also added in recent years (e.g., 2018–2021), especially movies.

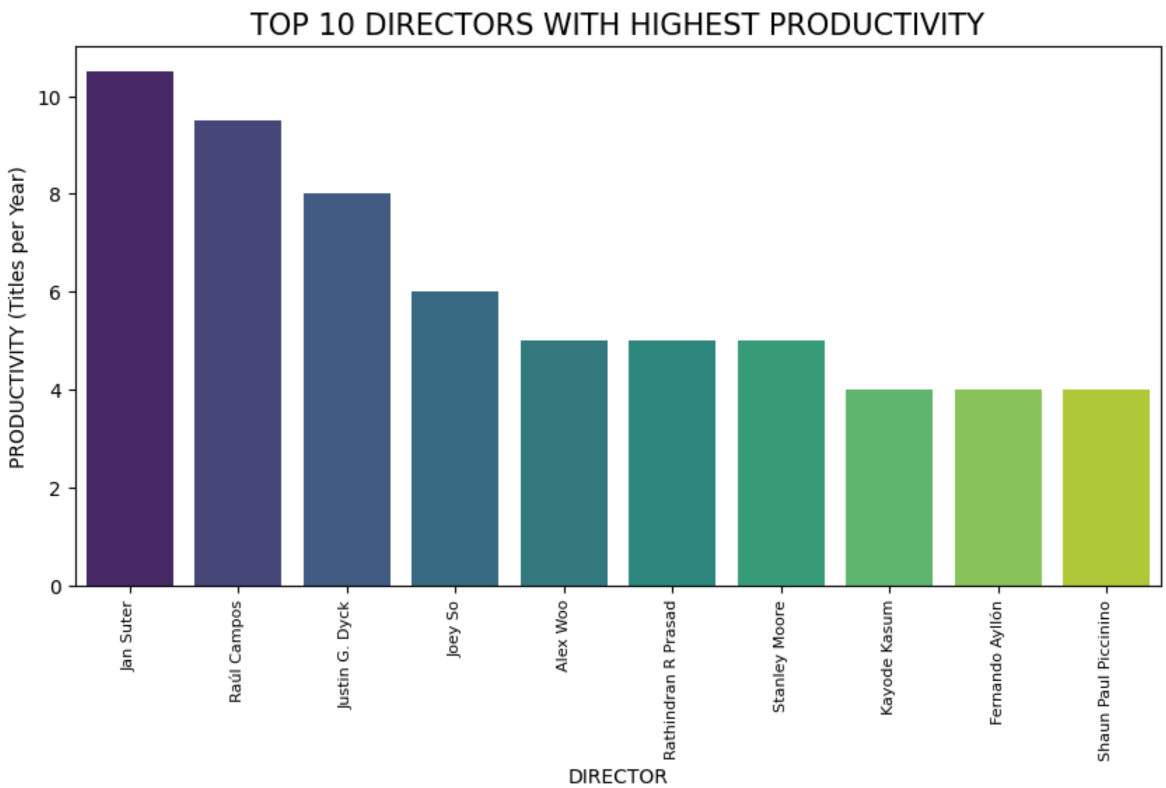
## PRODUCTIVITY OF DIRECTORS DURING THEIR ACTIVE YEARS (TOP 10)

***ASSUMPTIONS:***

- *Since many directors are newcomers to the industry or have worked for a year or less, we are considering their productivity based solely on their title count.*
- *Additionally, due to missing metadata regarding the names of several directors, we are excluding those entries from this specific analysis.*

```
dir_df = df
dir_df['director'] = df['director'].apply(lambda x:x.split(', '))
dir_df = dir_df.explode('director')

dir_df['title_count'] = dir_df.groupby('director')['title'].transform('nunique')
dir_df['min_year'] = dir_df.groupby('director')['release_year'].transform('min')
dir_df['max_year'] = dir_df.groupby('director')['release_year'].transform('max')
active_dir = dir_df[['director','min_year','max_year','title_count']]
```

```python
active_dir.drop_duplicates(inplace=True)

def prod(x):
  if x['max_year'] - x['min_year'] == 0:
    return x['title_count']
  else:
    return x['title_count']/(x['max_year']-x['min_year'])

active_dir['productivity'] = active_dir.apply(prod, axis=1).round(2)
active_dir.drop(1, inplace=True)
active_dir.reset_index(drop=True)
active_dir = active_dir.sort_values(by='productivity', ascending=False)
active_dir = active_dir.reset_index(drop=True)

plt.figure(figsize=(10,5))
sns.barplot(data=active_dir.head(10), x='director', y='productivity', palette='viridis')
plt.xticks(rotation=90, fontsize=8)
plt.xlabel('DIRECTOR')
plt.ylabel('PRODUCTIVITY (Titles per Year)')
plt.title('TOP 10 DIRECTORS WITH HIGHEST PRODUCTIVITY', fontsize=15)
plt.show()
```




**KEY INSIGHTS:**

- Top Directors Show High Short-Term Output:
  - Directors like **Jan Suter and Raúl Campos achieved 10+ titles/year**, indicating they were highly active over **short periods (2016–2018)**.
  - Others like Justin G. Dyck and Joey So also showed very high productivity within 1–2 years, suggesting involvement in series-based or bulk content (e.g., short films or episodes).
- Lower Productivity Over Long Careers:
  - Directors such as Walter Hill or Brian De Palma span decades (e.g., 1980–2016) but have very low productivity scores (~0.05 titles/year), likely because their works were critically spaced out films, not frequent releases.

## Finding out how many directors are having productivity equal to 1:

```python
print('Number of Directors with Productivity equal to 1:',
      int(active_dir.groupby('productivity').get_group(1.0).nunique().loc['director']),
      'out of', int(active_dir.nunique().loc['director']),
      'Directors')
```

```
Number of Directors with Productivity equal to 1: 4107 out of 4993 Directors
```

- 4107 out of 4993 directors (~82%) have a productivity of exactly 1, meaning they contributed only one title or worked in a single year.
- This indicates a high turnover or one-time involvement, possibly due to:

- Debuts or experiments
- Limited licensing from non-recurring creators
- Short-lived or niche projects

# NON-GRAPHICAL AND NON-VISIUAL ANALYSIS:

## MOVIE TIME DURATION ANALYSIS AMONG DIFFERENT RATINGS:

```
df['movie_min'] = pd.to_numeric(df['movie_min'], errors='coerce')
df['season_count'] = pd.to_numeric(df['season_count'], errors='coerce')

a = df.groupby('rating')['movie_min'].aggregate(['mean','min','max']).sort_values(by='mean', ascending=False)
a.dropna(inplace=True)
a
```

| rating | mean | min | max |
|---|---|---|---|
| NC-17 | 125.000000 | 83.0 | 180.0 |
| TV-14 | 110.290820 | 5.0 | 253.0 |
| PG-13 | 108.330612 | 8.0 | 201.0 |
| R | 106.720201 | 62.0 | 229.0 |
| UR | 106.333333 | 103.0 | 113.0 |
| PG | 98.282230 | 13.0 | 224.0 |
| TV-MA | 95.889913 | 12.0 | 312.0 |
| TV-PG | 94.851852 | 12.0 | 233.0 |
| NR | 94.533333 | 44.0 | 143.0 |
| G | 90.268293 | 9.0 | 181.0 |
| TV-G | 79.666667 | 10.0 | 273.0 |
| Not Available | 76.000000 | 37.0 | 115.0 |
| TV-Y7-FV | 68.400000 | 43.0 | 109.0 |
| TV-Y7 | 66.287770 | 14.0 | 127.0 |
| TV-Y | 48.114504 | 3.0 | 97.0 |

Next steps: Generate code with a | View recommended plots | New interactive sheet

**KEY INSIGHTS:**

- Age-Appropriate Trends:
  - As expected, younger audiences (TV-Y, TV-Y7, G) receive shorter content.
  - Teen and adult audiences (PG-13, R, NC-17) receive longer movies, consistent with traditional feature-length films.
- TV Ratings vs. Movie Ratings:
  - TV ratings (TV-MA, TV-14, etc.) often show high variability in duration, suggesting mixed types of content (episodes, specials, possibly films).
  - Movie ratings (G, PG, PG-13, R) have a more predictable average duration.
- Highest Duration Spread:
  - TV-14 and TV-G show some of the widest duration ranges, from extremely short (5–10 mins) to very long (253–273 mins).

## TV-SHOWS SEASON COUNT ANALYSIS AMONG DIFFERENT RATINGS:

```
b = df.groupby('rating')['season_count'].aggregate(['mean','min','max']).sort_values(by='mean', ascending=False)
b.dropna(inplace=True)
b
```

| rating | mean | min | max |
|---|---|---|---|
| NR | 3.800000 | 1.0 | 13.0 |
| TV-Y7 | 2.020513 | 1.0 | 10.0 |
| TV-Y7-FV | 2.000000 | 2.0 | 2.0 |
| TV-Y | 1.852273 | 1.0 | 10.0 |
| TV-G | 1.851064 | 1.0 | 8.0 |
| TV-14 | 1.821282 | 1.0 | 17.0 |
| TV-MA | 1.685590 | 1.0 | 13.0 |
| TV-PG | 1.668731 | 1.0 | 11.0 |
| Not Available | 1.000000 | 1.0 | 1.0 |
| R | 1.000000 | 1.0 | 1.0 |

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

**KEY INSIGHTS:**

- Longer-running series are mostly NR-rated or children's programming (TV-Y7), which makes sense — they are often syndicated or have educational/instructional longevity.
- TV-Y and TV-G shows typically span under 2 seasons, indicating shorter runs, often due to attention span considerations or educational value expiration.
- TV-MA and TV-14 shows, though targeted at older audiences, tend to be shorter-lived on average — reflecting the streaming-era trend of short, high-production-value series.
- "R" and "Not Available" should be flagged as potential misclassifications, especially since they both show only one season per entry. These likely aren't actual series.

## OVERALL MOVIE TIME DURATION ANALYSIS:

```
1 movie_data['movie_min'].aggregate(['mean','min','max'])
```

| | movie_min |
|---|---|
| mean | 99.577187 |
| min | 3.000000 |
| max | 312.000000 |

**dtype:** float64

**KEY INSIGHTS:**

- Most movies fall near the 100-minute mark, which is standard for modern narrative films.
- The 3-minute minimum could be:
  - A true short film (which is fine),
  - A trailer or extra mistakenly classified as a full movie.
- The 312-minute maximum (over 5 hours) is far beyond normal theatrical releases. It may require investigation:
  - Could be a multi-part film, a marathon edition, or an error.

## OVERALL TV-SHOW SEASON COUNT ANALYSIS:

```
1 tv['season_count'].aggregate(['mean','min','max'])
```

| | season_count |
|---|---|
| mean | 1.764948 |
| min | 1.000000 |
| max | 17.000000 |

**dtype:** float64

**KEY INSIGHTS:**

- Average Season Count (1.76): The average TV show in the dataset has approximately 1 to 2 seasons, indicating a prevalence of shows with relatively short durations in terms of seasons.
- Minimum Value (1 season) Some shows have exactly one season, which may include single-season formats, pilot runs, or limited series by design.
- Maximum Value (17 seasons) The longest-running show in the dataset spans 17 seasons, highlighting the inclusion of multi-season series, which suggests a variety of show lifespans are represented.

## TOP 10 DIRECTORS WITH MOST MOVIE TITLES:

```
x = dir_df[['director','title_count', 'type']].groupby('type').get_group('Movie').sort_values(by='title_count', ascending=False)
x.drop_duplicates(inplace=True)
x.reset_index(drop=True, inplace=True)
x.drop(0, inplace=True)
x.drop('type', axis=1, inplace=True)
x.head(10)
```

| | director | title_count |
|---|---|---|
| 1 | Rajiv Chilaka | 22 |
| 2 | Jan Suter | 21 |
| 3 | Raúl Campos | 19 |
| 4 | Marcus Raboy | 16 |
| 5 | Suhas Kadav | 16 |
| 6 | Jay Karas | 15 |
| 7 | Cathy Garcia-Molina | 13 |
| 8 | Jay Chapman | 12 |
| 9 | Martin Scorsese | 12 |
| 10 | Youssef Chahine | 12 |

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

**INSIGHTS:**

- Rajiv Chilaka leads with the highest number of movie titles (22).
- The top 3 directors (Rajiv Chilaka, Jan Suter, Raúl Campos) each have 19 or more titles.
- Marcus Raboy and Suhas Kadav are tied with 16 titles each.
- The bottom 5 directors on the list each have between 12 and 15 titles.
- Three directors (Jay Chapman, Martin Scorsese, Youssef Chahine) are tied at 12 titles.
- The range of movie title counts among the top 10 directors is 12 to 22.

## TOP 10 DIRECTORS WITH MOST TV-SHOW TITLES:

```
y = dir_df[['director','title_count', 'type']].groupby('type').get_group('TV Show').sort_values(by='title_count', ascending=False)
y.drop_duplicates(inplace=True)
y.reset_index(drop=True, inplace=True)
y.drop(0, inplace=True)
y.drop('type', axis=1, inplace=True)
y.head(10)
```

| | director | title_count |
|---|---|---|
| 1 | Marcus Raboy | 16 |
| 2 | Anurag Kashyap | 9 |
| 3 | Ryan Polito | 8 |
| 4 | Quentin Tarantino | 8 |
| 5 | Priyadarshan | 7 |
| 6 | Michael Simon | 6 |
| 7 | Ken Burns | 5 |
| 8 | John Paul Tremblay | 5 |
| 9 | Robb Wells | 5 |
| 10 | Mike Smith | 5 |

Next steps: Generate code with y | View recommended plots | New interactive sheet

**KEY INSIGHTS:**

- Marcus Raboy has the highest number of TV show titles (16), making him the most frequent TV director in the dataset.
- The title counts decrease gradually, from 16 down to 5 across the top 10 directors.
- Three directors (John Paul Tremblay, Robb Wells, and Mike Smith) each have 5 titles, indicating possible collaboration or shared projects.
- Only one director has more than 10 titles; the rest have single-digit counts.
- The range of TV show title counts among the top 10 directors is 5 to 16.

# FINAL BUSINESS INSIGHTS AND OBSERVATIONS:

## *=> Content Composition*

- Movies dominate the Netflix library, forming the majority of the 8807 total titles.
- TV Shows form a smaller portion, but have distinct season-based consumption behavior.
- The average movie duration is approximately 100 minutes, which aligns with industry norms.
- TV shows average around 1.76 seasons, suggesting a prevalence of limited series or recently added shows.

## *=> Content Addition Trends*

- A major surge in both movies and TV shows occurred after 2015, peaking in 2019 and 2020.
- Content additions dropped slightly in 2021, likely reflecting production delays during the pandemic.
- Older titles (from as early as the 1940s) are still added between 2016–2021, showing Netflix's ongoing licensing of classic content.

## *=> Release Timing Trends*

- Movies are most often added in July, followed by December, January, and October.
- TV Shows see the highest additions in December, then in mid-year weeks (27, 31, and 13).
- The first week of the year (Week 1) consistently sees the most movie uploads, possibly as part of a strategic content refresh.

## *=> Geographical Distribution*

- United States is the top producer of both movies and TV shows on Netflix.
- UK, France, Germany, and Canada are also major contributors, especially to movies.
- For TV shows, European and American content dominates, while Asian countries (e.g., India, South Korea) have limited representation in series production.

## *=> Genre Preferences*

- Top movie genres: International, Drama, Comedy, Documentaries, Action & Adventure.
- Top TV genres: Drama, International, Kids, Anime, and Crime.
- There is strong viewer preference for story-driven, emotional, and culturally rich genres.
- Genres like Horror, Sci-Fi, and Classic are underrepresented, indicating untapped segments.

## *=> Ratings and Audience Targeting*

- Majority of content is rated TV-MA (Mature) and TV-14 (Teens).
- Family and kids' ratings (TV-Y, G, TV-Y7) have comparatively fewer titles.
- Mature content also shows a higher average movie duration and more seasons, indicating deeper storytelling.

## *=> Director Productivity & Cast Patterns*

- Most directors (82%) contributed to only one title, reflecting a wide array of creative contributors.
- Top movie directors include Rajiv Chilaka (22 titles) and Jan Suter (21 titles).
- Marcus Raboy leads in TV shows with 16 titles.
- Indian actors dominate the movie cast data, while Japanese voice actors dominate in TV shows, especially due to anime content.
- Female representation is limited in both top 20 movie and TV show cast lists (only ~15%), revealing a potential gender imbalance.

### *=> Duration Distribution*

- Most movies cluster around 90–110 minutes, with a few outliers on either side (as low as 3 mins, as high as 312 mins).
- TV shows are predominantly short-run: 1–2 seasons are most common.
- Movie duration increases with maturity of the content: PG-13, R, NC-17 rated movies are generally longer.
- TV-MA and TV-14 shows have broader season distributions, often linked to mainstream or binge-worthy series.

```
Start coding or generate with AI.
```

## RECOMMENDATIONS:

## WHAT TYPES OF MOVIES/TV-SHOWS TO PRODUCE:

### *=> Produce More Family & Kids Content*

- The majority of titles are rated TV-MA (3207 titles) and TV-14 (2160 titles), indicating a strong emphasis on mature content.
- Ratings like TV-Y, TV-Y7, PG, and TV-G have significantly fewer titles.
- This reveals a gap in content for children and family viewing.

### *=> Focus on High-Interest Genres: Drama, Comedy, Crime*

- For movies, Drama, Comedy, and Documentaries are the most common.
- For TV shows, Drama, Kids', Anime, and Crime dominate.
- This indicates a clear viewer preference for emotional stories and immersive plots.

### *=> Expand Anime & Voice-Driven Content*

- The top 20 TV show cast members are predominantly Japanese voice actors, showing strong anime engagement.
- Names like Takahiro Sakurai, Yuki Kaji, and Daisuke Ono appear frequently.

### *=> Add More Long-Form TV Shows*

- Average TV show seasons = 1.76
- The majority of shows have 1–2 seasons, suggesting short lifespans.

### *=> Explore Underrepresented Genres*

- Genres like Horror, Sci-Fi, and Classic TV are less common.
- This may indicate an opportunity to diversify the catalog with niche or fan-favorite themes.

```
Start coding or generate with AI.
```

## How Netflix Can Grow the Business in Different Countries:

### *=> United States: Keep Leading in Content Supply*

- The United States alone contributes over 17% of movies and nearly 20% of TV shows.
- It is the largest contributor for both formats on the platform.

### *=> India: Increase TV Show Production*

- India appears among the top movie-producing countries but is absent from the top 10 for TV shows.
- This indicates a potential growth area for locally produced series.

### *=> Japan: Strengthen Anime Pipeline*

- High presence of Japanese voice actors in the top 20 cast for TV shows.
- This is due to the large volume of anime content in Netflix's catalog.

### *=> Europe: Continue Strategic Partnerships*

- 6 of the top 10 movie-producing countries are European (UK, France, Germany, Spain, Italy, Turkey).
- Europe is a major source of localized and diverse content.

## *=> Underrepresented Regions: Opportunity to Expand*

- The data shows limited content from African and Middle Eastern countries.
- These regions are not represented in the top 10 lists for movies or TV shows.

## *=> Leverage Strategic Content Launch Windows*

- Movies are most added in July, December, and January.
- TV shows spike in December, and in weeks 27, 31, and 13 (mid-year).
- These seasonal peaks align with holidays and school breaks.

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```