

Netflix data analysis in python

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

This project explores Netflix's content library using Python and pandas. It analyzes trends across movies and TV shows, including genre, release patterns, actors, and directors. Visualizations such as bar charts, linecharts etc help uncover insights into Netflix's streaming catalog and viewer preferences.

```
#To download the data
df=pd.read_csv('/content/scaler netflix.csv',index_col=0)
```

```
#To remove the duplicates
df=df.drop_duplicates()
```

```
#This gives us top 5 movies/tvshows
df.head()
```


	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	week_number	year_added
0	Movie	Dick Johnson Is Dead	Kirsten Johnson	Not known	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	38	2021.0
1	TV Show	Blood & Water	Not known director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...	38	2021.0
2	TV Show	Blood & Water	Not known director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...	38	2021.0
3	TV Show	Blood & Water	Not known director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	TV Mysteries	After crossing paths at a party, a Cape Town t...	38	2021.0
4	TV Show	Blood & Water	Not known director	Khosi Ngema	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...	38	2021.0



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 8807 entries, s1 to s8807
Data columns (total 11 columns):
#   Column          Non-Null Count  Dtype
#   ...
```


```
---  -----  -----  ----
0   type      8807 non-null  object
1   title     8807 non-null  object
2   director  6173 non-null  object
3   cast      7982 non-null  object
4   country   7976 non-null  object
5   date_added 8797 non-null  object
6   release_year 8807 non-null  int64
7   rating    8803 non-null  object
8   duration  8804 non-null  object
9   listed_in 8807 non-null  object
10  description 8807 non-null  object
dtypes: int64(1), object(10)
memory usage: 1.1+ MB
```

```
df.describe()
```




	release_year	
count	8807.000000	
mean	2014.180198	
std	8.819312	
min	1925.000000	
25%	2013.000000	
50%	2017.000000	
75%	2019.000000	
max	2021.000000	

```
df.shape
```



```
(202065, 13)
```

```
type_val=df['type'].value_counts()
type_val
```



	count
type	
Movie	6131
TV Show	2676

```
dtype: int64
```

```
df['cast']=df['cast'].fillna('Not known')
df['cast']=df['cast'].apply(lambda x:x.split(','))
df=df.explode('cast').reset_index(drop=True)
df['cast']=df['cast'].str.strip()

df['director']=df['director'].fillna('Not known director')
df['director']=df['director'].apply(lambda x:x.split(','))
df=df.explode('director').reset_index(drop=True)
df['director']=df['director'].str.strip()

df['listed_in']=df['listed_in'].fillna('genre unavailable')
df['listed_in']=df['listed_in'].apply(lambda x:x.split(','))
df=df.explode('listed_in').reset_index(drop=True)
df['listed_in']=df['listed_in'].str.strip()

df['country']=df['country'].fillna('country unavailable')
df['country']=df['country'].apply(lambda x:x.split(','))
df=df.explode('country').reset_index(drop=True)
df['country']=df['country'].str.strip()
```

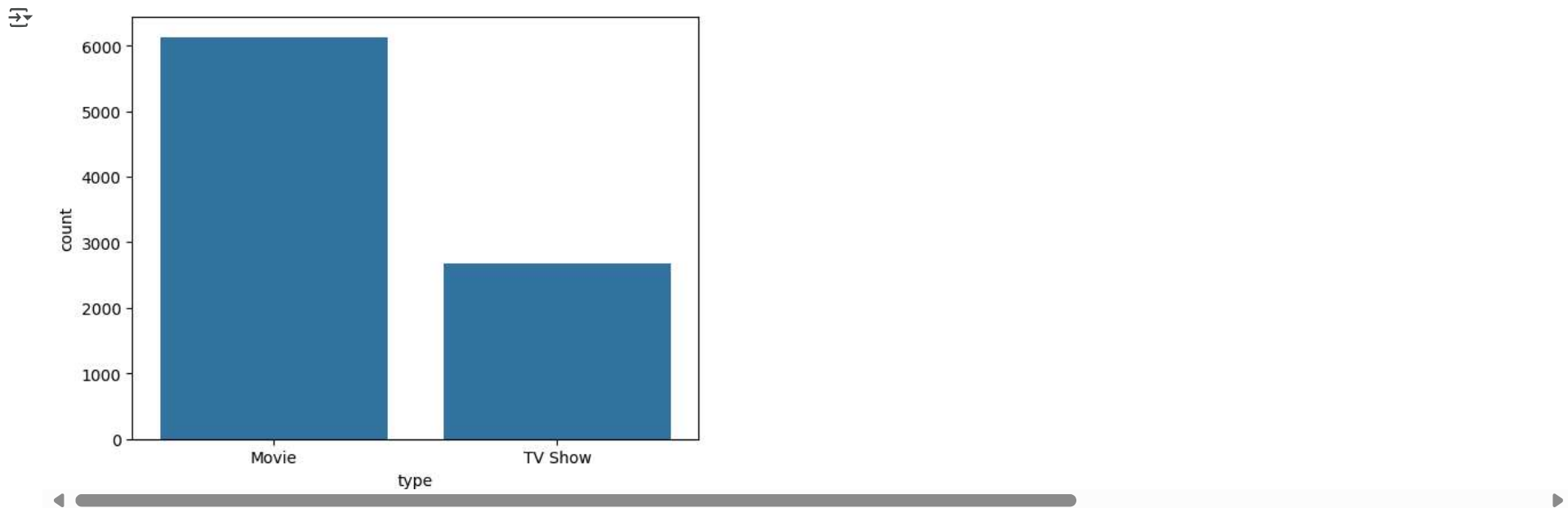
Data conversion and week extraction

```
df['date_added']=pd.to_datetime(df['date_added'],format='mixed')
df['week_number']=df['date_added'].dt.isocalendar().week
df.head(2)
```

</

This chart below gives us the comparison of movies vs tvshows

```
cp=sns.countplot(data=df,x='type',order=type_val.index)
plt.show()
```



clearly from the above chart we can see number of movies released on netflix are far more than tv shows. so we can conclude from the above example that Netflix releases more number of movies than the tvshows

```
movies_df = df[df['type'] == 'Movie']
```

Change in number of movies released over last 30 years

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

```
movies_df['release_year'] = pd.to_numeric(movies_df['release_year'], errors='coerce')
```

```
movies_df = movies_df.dropna(subset=['release_year'])
movies_df['release_year'] = movies_df['release_year'].astype(int)
```

```
current_year = pd.Timestamp.now().year
start_year = current_year - 30
```


```
movies_30yrs = movies_df[(movies_df['release_year'] >= start_year) & (movies_df['release_year'] <= current_year)]
```

```
movies_per_year = movies_30yrs['release_year'].value_counts().sort_index()
```

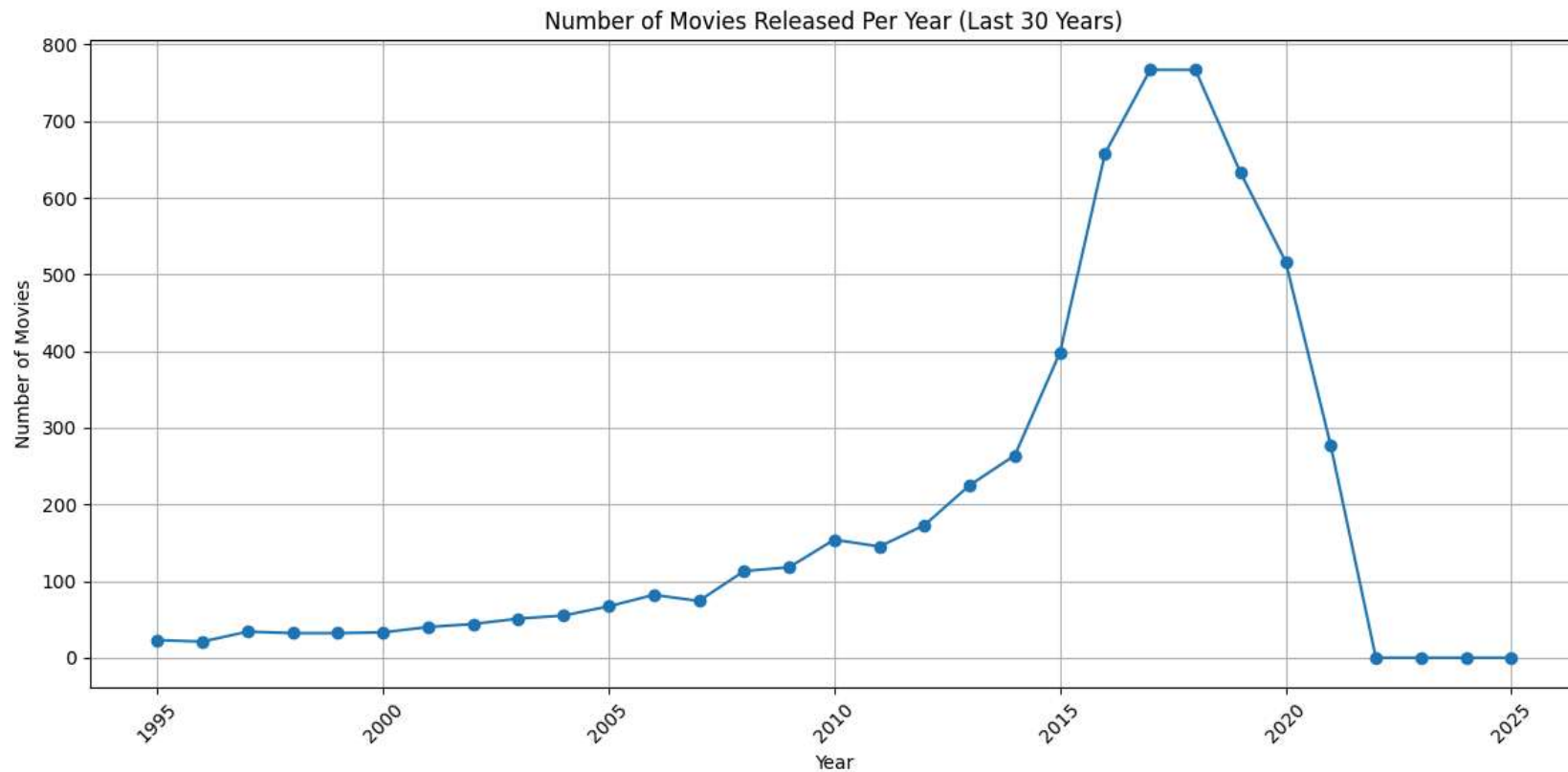
```
full_year_range = pd.Series(0, index=range(start_year, current_year + 1))
```

```
movies_per_year = full_year_range.add(movies_per_year, fill_value=0).astype(int)
```

```
plt.figure(figsize=(12, 6))
plt.plot(movies_per_year.index, movies_per_year.values, marker='o', linestyle='-')
plt.title('Number of Movies Released Per Year (Last 30 Years)')
plt.xlabel('Year')
plt.ylabel('Number of Movies')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

 <ipython-input-87-b065eeba2cf9>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
 movies_df['release_year'] = pd.to_numeric(movies_df['release_year'], errors='coerce')



The line chart above represents the change in number of movies released over last 30 years. we can see by above line chart that the most number of movies are released in 2015 to 2020. And almost same number of movies were released from 1995 to 2010

Best time to launch tvshow

```
tv_shows = df[df['type'] == 'TV Show'].copy()

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

tv_shows['month'] = tv_shows['date_added'].dt.month
tv_shows['weekday'] = tv_shows['date_added'].dt.day_name()

monthly_counts = tv_shows['month'].value_counts().sort_index()

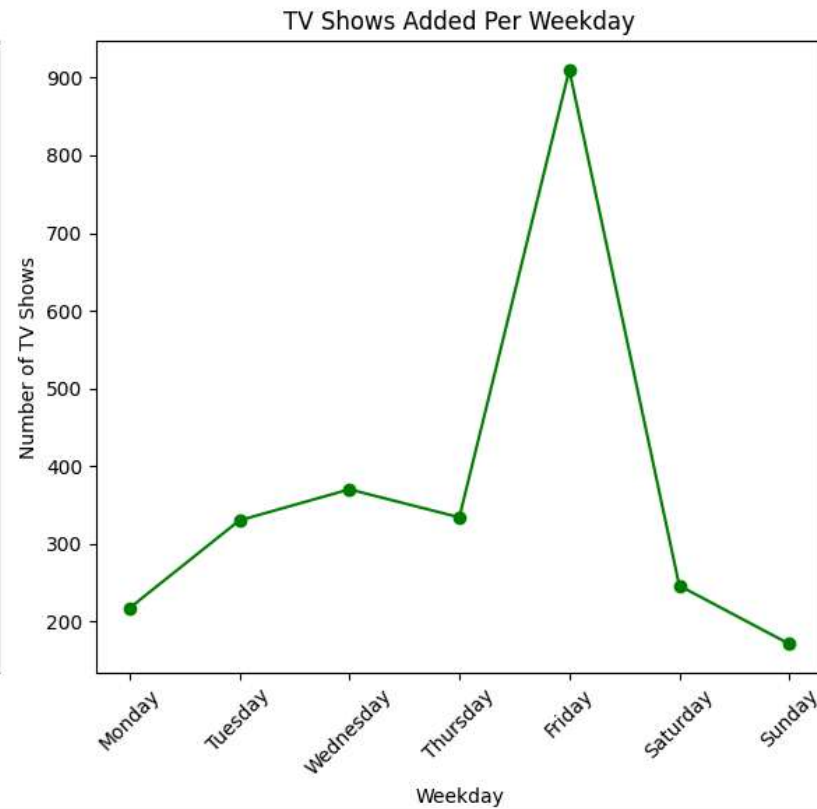
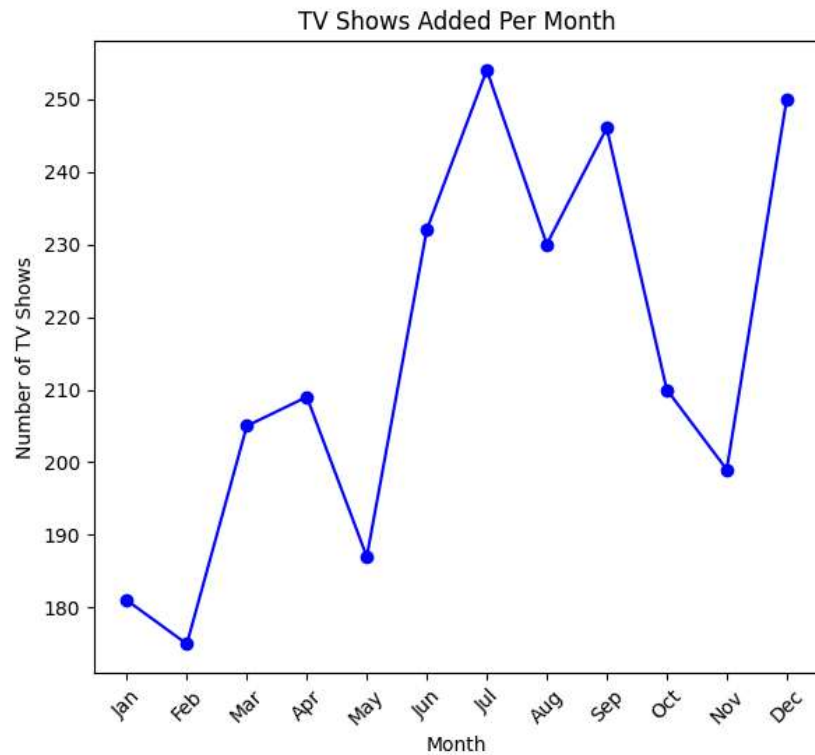
weekday_counts = tv_shows['weekday'].value_counts().reindex(
    ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'], fill_value=0)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.plot(monthly_counts.index, monthly_counts.values, marker='o', linestyle='-', color='blue')
plt.title('TV Shows Added Per Month')
plt.xlabel('Month')
plt.ylabel('Number of TV Shows')
plt.xticks(monthly_counts.index, labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], rotation=45)

plt.subplot(1, 2, 2)
plt.plot(weekday_counts.index, weekday_counts.values, marker='o', linestyle='-', color='green')
plt.title('TV Shows Added Per Weekday')
plt.xlabel('Weekday')
plt.ylabel('Number of TV Shows')
plt.xticks(rotation=45)

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



As we go monthwise netflix releases its shows in the month of JULY and DECEMBER most of the times. And they choose FRIDAY to release the show, so that people could watch it on Saturday and Sunday.

Top 10 most frequent actors on netflix

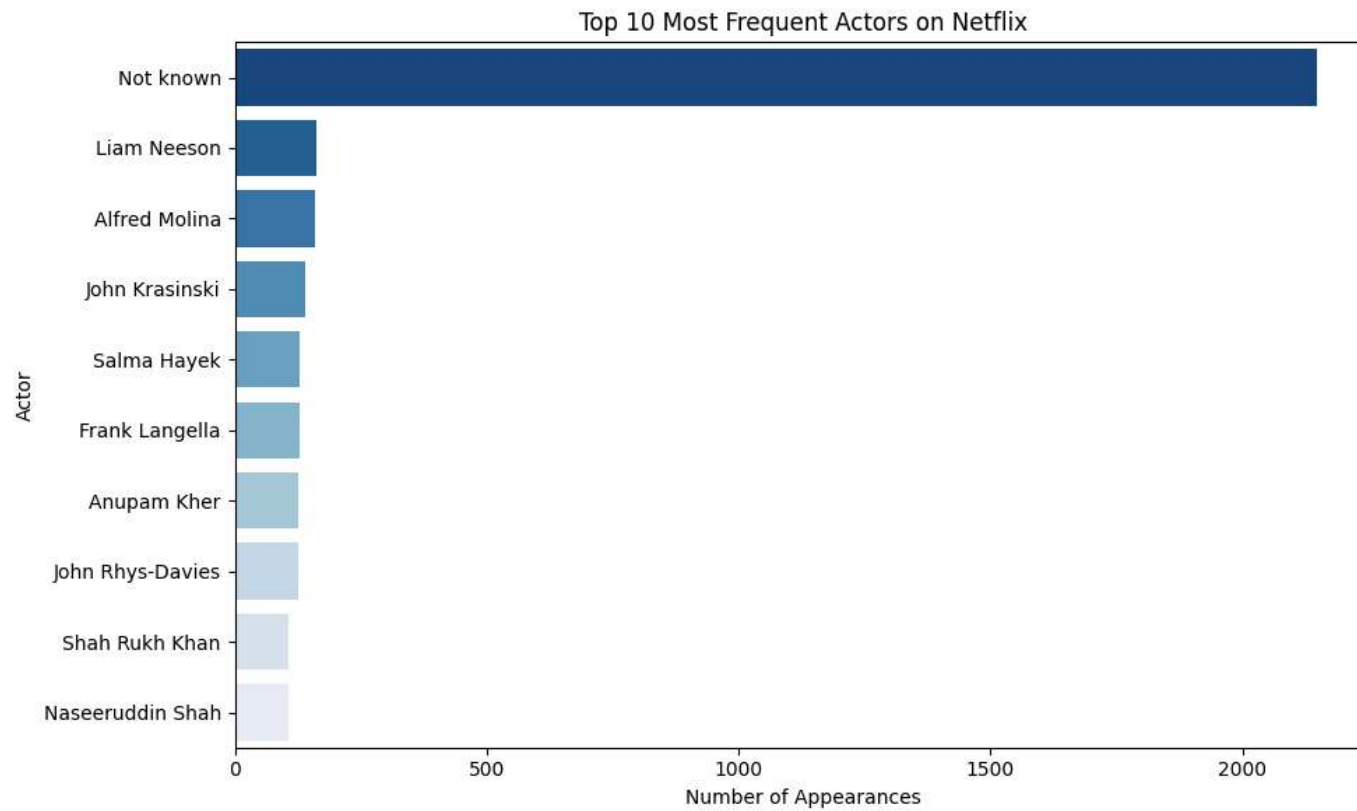
```
top_actors = df['cast'].value_counts().head(10)
```

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df[df['cast'].isin(top_actors.index)], y='cast', order=top_actors.index, palette='Blues_r')
plt.title('Top 10 Most Frequent Actors on Netflix')
plt.xlabel('Number of Appearances')
plt.ylabel('Actor')
plt.tight_layout()
plt.show()
```

<ipython-input-172-ae7f741e3ea2>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df[df['cast'].isin(top_actors.index)], y='cast', order=top_actors.index, palette='Blues_r')
```



In the bar chart above the most frequent actors(top 10) are given.

1) Liam Neeson, Alfred Molina and John Krasinski are in top 5.

2) Also Indian actors like Anupam Kher, Shahrukh Khan and Naseeruddin Shah are in top 10 most frequent actors.


Top 10 most frequent Directors on netflix

```
top_directors = df['director'].value_counts().head(10)
```

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df[df['director'].isin(top_directors.index)], y='director', order=top_directors.index, palette='Oranges_r')
plt.title('Top 10 Most Frequent Directors on Netflix')
plt.xlabel('Number of Titles Directed')
plt.ylabel('Director')
```

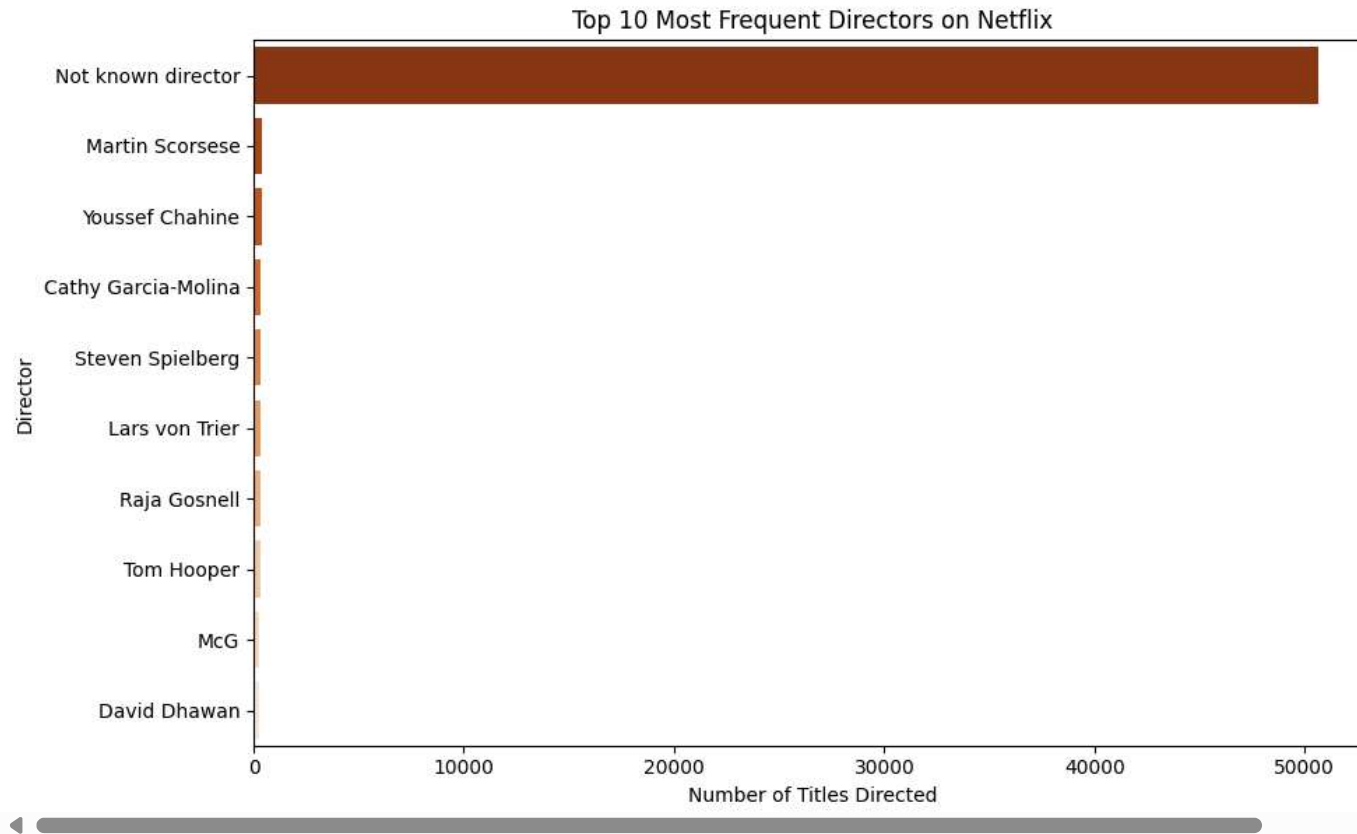


```
plt.tight_layout()
plt.show()
```

 <ipython-input-158-87aad5799ff3>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df[df['director'].isin(top_directors.index)], y='director', order=top_directors.index, palette='Oranges_r')
```




- 1) Directors like Martin Scorsese, Youssef Chahine, Cathy Garcia-Molina are in top 5 most frequent directors.
- 2) Indian Director David Dhawan is at 10th position.

Top 10 Genres on Netflix

```
top_genres = df['listed_in'].value_counts().head(10)
```

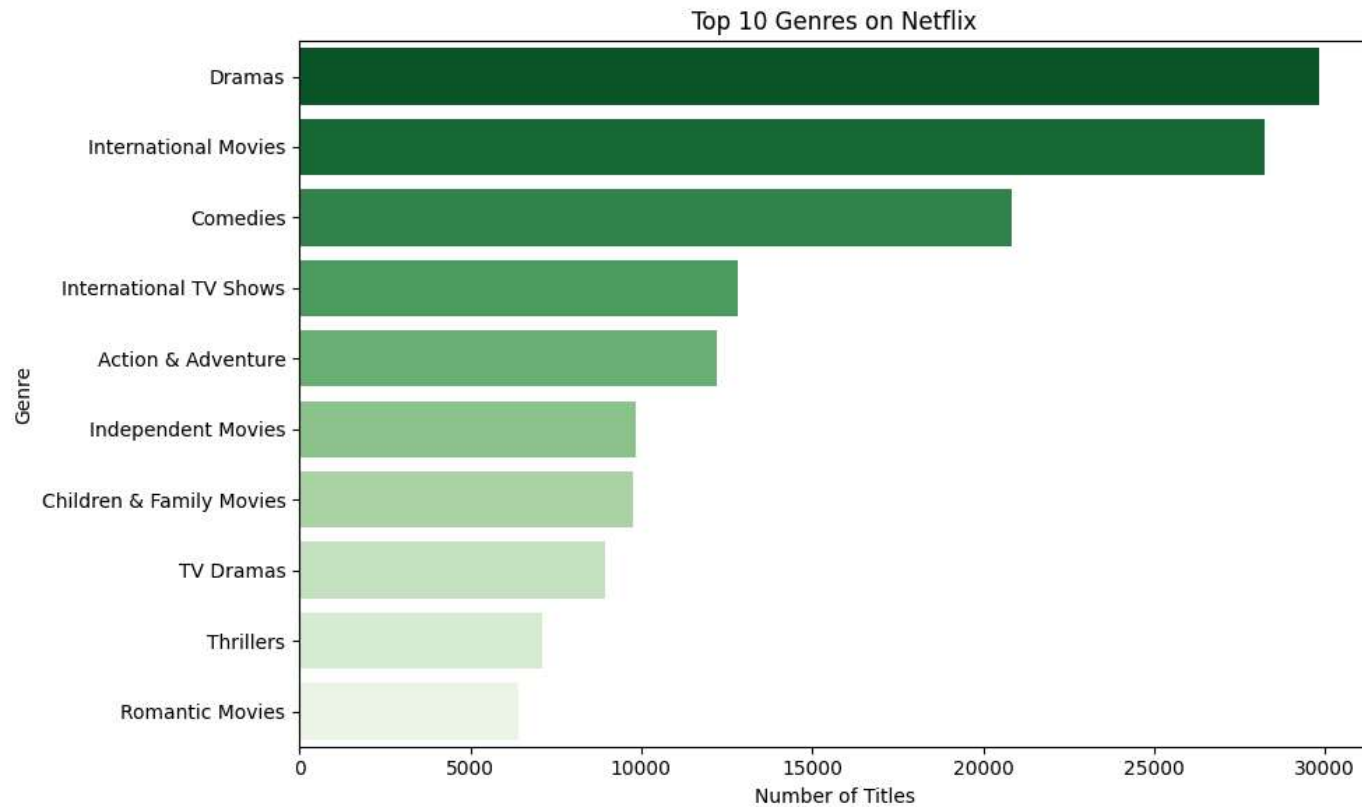
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df[df['listed_in'].isin(top_genres.index)], y='listed_in', order=top_genres.index, palette='Greens_r')
plt.title('Top 10 Genres on Netflix')
plt.xlabel('Number of Titles')
```

```
plt.ylabel('Genre')
plt.tight_layout()
plt.show()
```

 <ipython-input-159-2cec0cc65c04>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df[df['listed_in'].isin(top_genres.index)], y='listed_in', order=top_genres.index, palette='Greens_r')
```



- 1) Dramas, International Movies, and Comedies are the top 3 genres released the most frequently.
- 2) Romantic Movies are released least in numbers

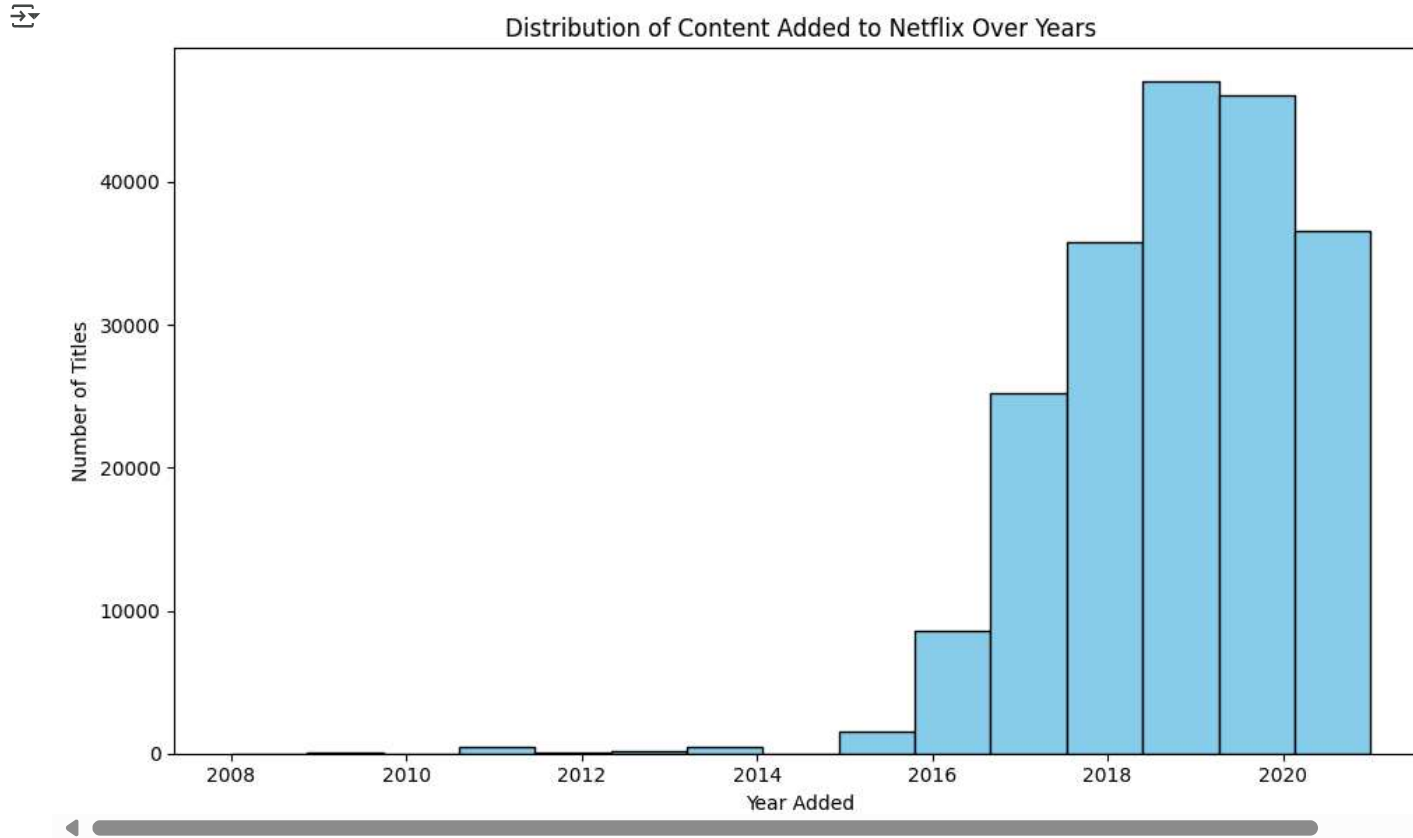
Distribution of content added to the netflix over years

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

```
df['year_added'] = df['date_added'].dt.year
```

```
plt.figure(figsize=(10, 6))
```

```
plt.hist(df['year_added'].dropna(), bins=15, color='skyblue', edgecolor='black')
plt.title('Distribution of Content Added to Netflix Over Years')
plt.xlabel('Year Added')
plt.ylabel('Number of Titles')
plt.tight_layout()
plt.show()
```



- 1)The number of titles added to Netflix increased significantly from 2016 to 2019, peaking during this period.
- 2)A slight drop is observed in 2020, likely due to production delays during the COVID-19 pandemic.
- 3)Overall, the trend shows Netflix rapidly expanding its library in recent years, especially for original content.

Variation of duration of movies according to genre

```
df_movies = df[df['type'] == 'Movie'].copy()
df_movies['duration'] = df_movies['duration'].str.extract('(\d+)').astype(float)
```

```
df_movies = df_movies.dropna(subset=['duration', 'listed_in'])

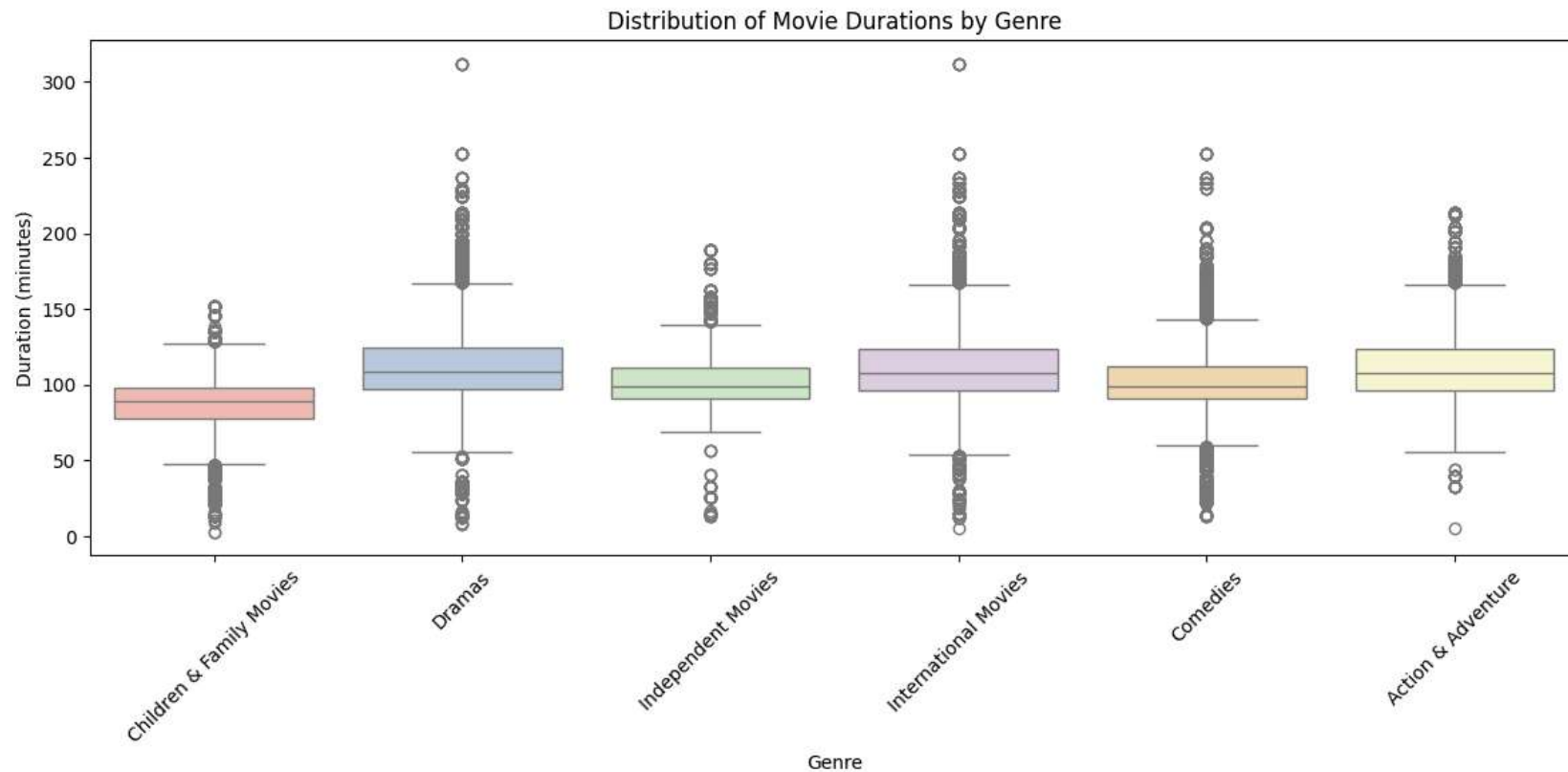
top_genres = df_movies['listed_in'].value_counts().head(6).index
df_movies = df_movies[df_movies['listed_in'].isin(top_genres)]

plt.figure(figsize=(12, 6))
sns.boxplot(data=df_movies, x='listed_in', y='duration', palette='Pastell1')
plt.title('Distribution of Movie Durations by Genre')
plt.xlabel('Genre')
plt.ylabel('Duration (minutes)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

↔ <ipython-input-165-42c700674fc0>:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(data=df_movies, x='listed_in', y='duration', palette='Pastell1')
```



1)Most genres have movie durations clustered between 80 to 120 minutes, indicating a standard feature-length format across genres.

2)Action and Drama tends to have slightly longer average

Top 10 actors collaboration with top 10 directors

```
top_actors = df['cast'].value_counts().head(10).index
top_directors = df['director'].value_counts().head(10).index

filtered_df = df[df['cast'].isin(top_actors) & df['director'].isin(top_directors)]

collab_matrix = pd.crosstab(filtered_df['cast'], filtered_df['director'])

plt.figure(figsize=(10, 8))
sns.heatmap(collab_matrix, annot=True, fmt='d', cmap='YlGnBu', linewidths=0.5)
plt.title('Actor-Director Collaboration Heatmap (Top 10)')
plt.xlabel('Director')
plt.ylabel('Actor')
plt.tight_layout()
plt.show()
```



- 1) Most actor-director collaborations are limited, with only a few pairs appearing together more than once.
- 2) A small number of directors consistently work with specific actors, suggesting preferred creative partnerships.

Tv shows vs movies added to netflix over last 10 years

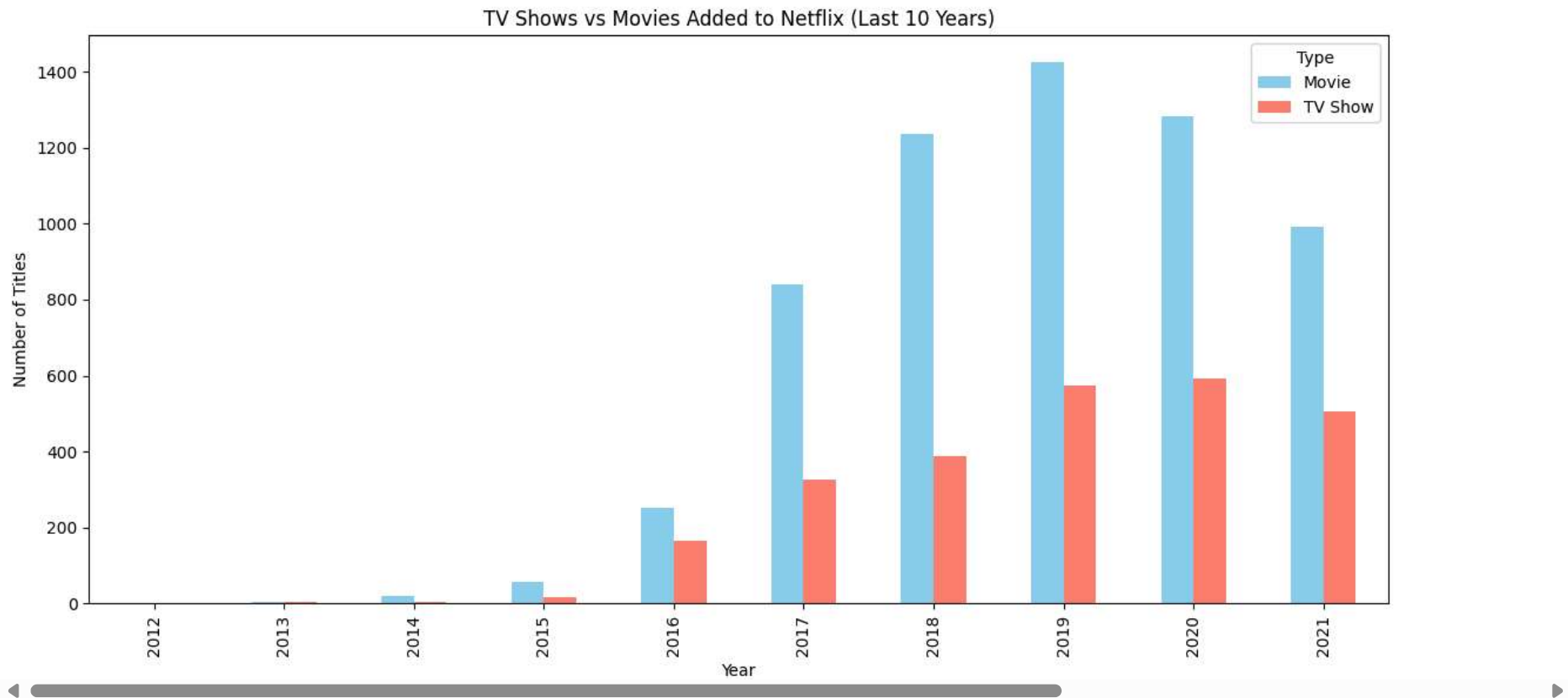
```
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
```

```
df = df.dropna(subset=['date_added'])
```

```
df['year_added'] = df['date_added'].dt.year
last_10_years = df[df['year_added'] >= (df['year_added'].max() - 9)]

year_type_counts = last_10_years.groupby(['year_added', 'type']).size().unstack(fill_value=0)

year_type_counts.plot(kind='bar', figsize=(12, 6), color=['skyblue', 'salmon'])
plt.title('TV Shows vs Movies Added to Netflix (Last 10 Years)')
plt.xlabel('Year')
plt.ylabel('Number of Titles')
plt.legend(title='Type')
plt.tight_layout()
plt.show()
```



- 1) Movies have consistently outnumbered TV shows on Netflix, especially between 2016 and 2019.
- 2) TV show additions have grown steadily, reflecting Netflix's investment in long-form and original series content

Content available in top 10 countries

```

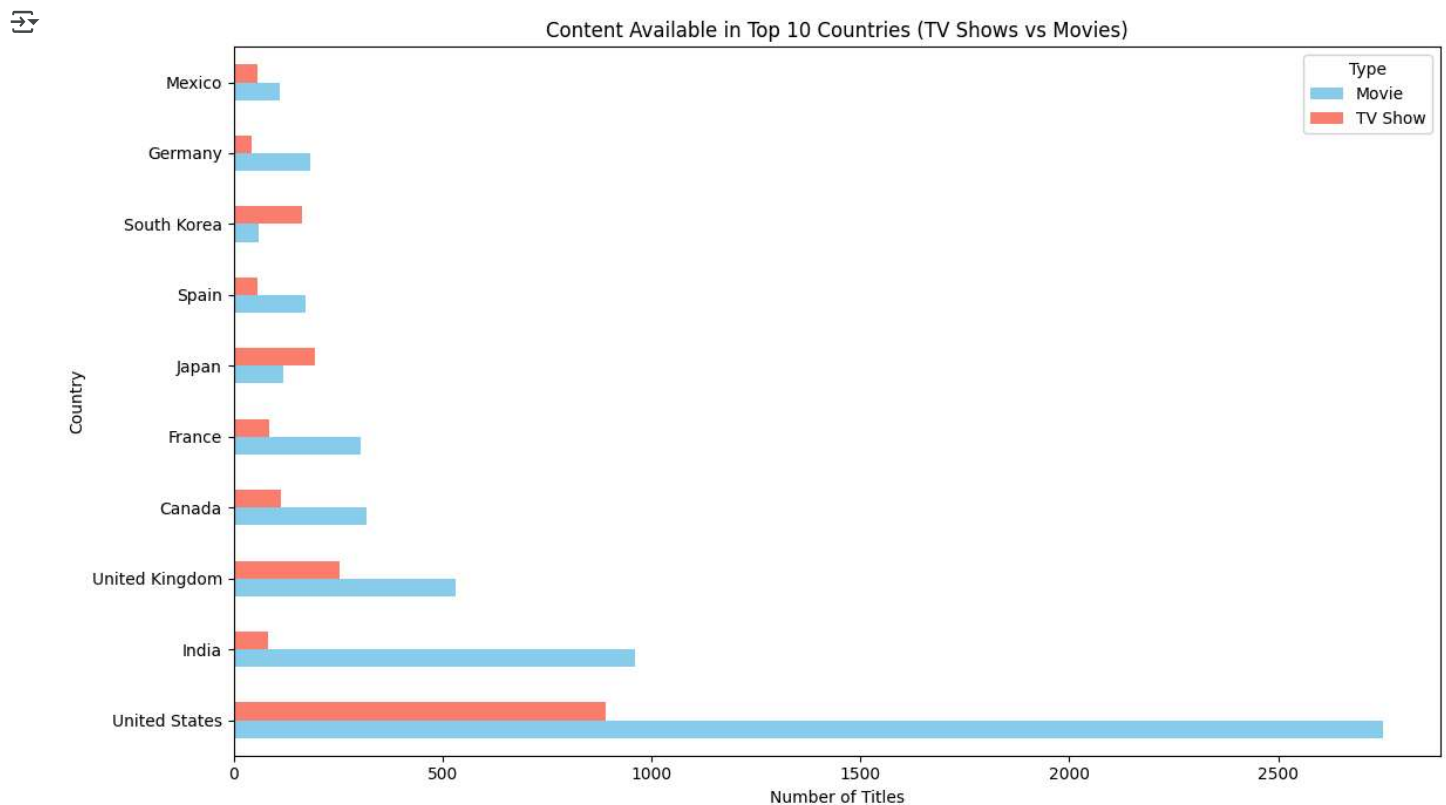
df['country'] = df['country'].str.split(', ')
df_exploded = df.explode('country')

country_content = df_exploded.groupby(['country', 'type']).size().unstack(fill_value=0)

top_countries = country_content.sum(axis=1).sort_values(ascending=False).head(10)
country_content_top10 = country_content.loc[top_countries.index]

country_content_top10.plot(kind='barh', figsize=(12, 7), color=['skyblue', 'salmon'])
plt.title('Content Available in Top 10 Countries (TV Shows vs Movies)')
plt.xlabel('Number of Titles')
plt.ylabel('Country')
plt.legend(title='Type')
plt.tight_layout()
plt.show()

```



1)The United States leads with the highest volume of content, followed by India, the United Kingdom, and Canada. 2)Japan and South Korea are the only two countries which have more number of tvshows than movies.