

LIMPIEZA, TRANSFORMACIÓN, ANÁLISIS DE DATOS Y VISUALIZACIÓN DE CONTENIDO DE NETFLIX CON PYTHON

MARÍA FERNANDA RUBÍ EGUEZ
01/08/2025

HERRAMIENTAS Y LIBRERIAS:
MATPLOTLIB, PANDAS, NUMPY, SEABORN, POWER BI, POWER QUERY

INTRODUCCIÓN AL PROBLEMA



OBJETIVO

Realizar limpieza, análisis y visualización de las películas y series en Netflix



PROCEDIMIENTO

- Limpieza de nulos
- Reemplazo de valores
- Eliminar columnas
- Convertir Datos
- Simplificar Columnas
- Analizar Datos
- Obtener graficas

HERRAMIENTAS UTILIZADAS

CLEANING, TRANSFORMATION, ANALYSIS AND VISUALIZATION OF NETFLIX

This notebook shows how to load and analyze a dataset containing information about movies.

Steps:

1. Load data
2. Clean missing values, check for duplicates, remove unnecessary columns
3. Analysis
4. Create visualizations

Note: The dataset is in `netflix_titles.csv`.

```
▶ ▾
  import pandas as pd
  import matplotlib.pyplot as plt
  import seaborn as sns

  #Read the CSV
  df = pd.read_csv("netflix_titles.csv")
  df
[77] ✓ 0.0s
  Activar Windo
  Ver Configuració
```

- **Python (Jupyter Notebook):**
Para análisis, modelado y visualización.
- **Matplotlib:** Para graficar tendencias y predicciones.
- **Pandas:** Para estructurar y manipular datos

EXPLORACIÓN Y ESTRUCTURA DE DATOS

	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 Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Nan	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	Nan	Nan	Nan	September 24, 2021	2021	TV-MA	Activar Windows 1 Season	Docuseries, Reality TV Ve a Configuración para activar Windows.	Feuds, flirtations and toilet talk go down

FUENTE DE DATOS:
DATOS DE NETFLIX_TITLESCSV

LECTURA DE DATOS HISTÓRICOS

Se cargan los datos históricos sobre las películas y series estrenadas en Netflix.

El dataset contiene:

- Show_id
- Type
- Title
- Director
- Cast
- Country
- Date_added
- Release_year
- Rating
- Duration
- Listed_in
- Description

Hay campos vacíos en muchas columnas, por lo cual buscaremos coincidencias para poder llenar valores faltantes.

REVISAMOS VALORES DUPLICADOS Y NULOS

Duplicados: Para los duplicados se agrupa la columna de ID y se realiza un conteo de las veces que se repite un mismo ID, en este caso no existen valores duplicados.

Nulos: Para los valores nulos se utiliza el código `.isnull().sum()` para determinar la cantidad de valores nulos que existen en este caso, por columna.



```
#Check for duplicates ID
duplicates = df.groupby('show_id').size().reset_index(name='count')
duplicates = duplicates[duplicates['count'] > 1]

#Show duplicates
print(duplicates)
```

[78] ✓ 0.0s

... Empty DataFrame
Columns: [show_id, count]
Index: []

```
#Check null values across columns
nulls = {
    'showid_nulls': df['show_id'].isnull().sum(),
    'type_nulls': df['type'].isnull().sum(),
    'title_nulls': df['title'].isnull().sum(),
    'director_nulls': df['director'].isnull().sum(),
    'country_nulls': df['country'].isnull().sum(),
    'date_added_nulls': df['date_added'].isnull().sum(),
    'release_year_nulls': df['release_year'].isnull().sum(),
    'rating_nulls': df['rating'].isnull().sum(),
    'duration_nulls': df['duration'].isnull().sum(),
    'listed_in_nulls': df['listed_in'].isnull().sum(),}
```

[163]

nulls
director_nulls 2634
country_nulls 831
date_added_nulls 10
rating_nulls 4
duration_nulls 3
showid_nulls 0
type_nulls 0
title_nulls 0
release_year_nulls 0
listed_in_nulls 0

Replace null values in the director column by analyzing the directors and cast relationship.

```
#We are left with only valid director + cast combinations
df_director_valid = df[df['director'].notnull() & df['cast'].notnull()]

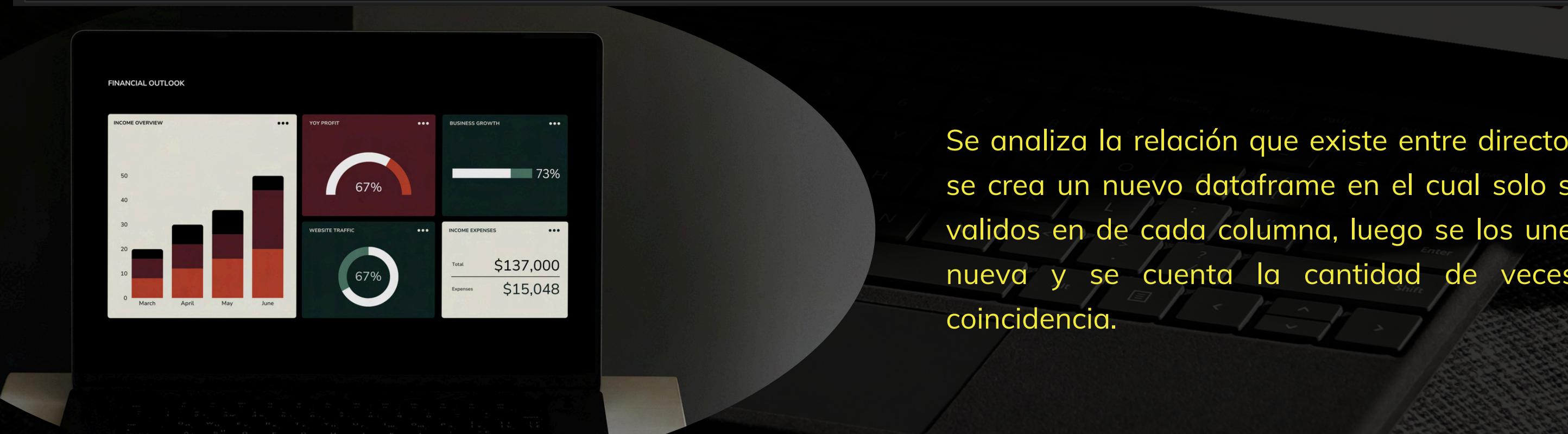
#Create a director + cast combination
df_director_valid['director_cast'] = df_director_valid['director'] + '---' + df_director_valid['cast']

#We count how many times each combination is repeated
counts = df_director_valid['director_cast'].value_counts().reset_index()
counts.columns = ['director_cast', 'count']

df_director_valid.head(5)
```

✓ 0.0s

PowerShell



Se analiza la relación que existe entre directores y el cast, para ello se crea un nuevo dataframe en el cual solo se analizan los valores validos en de cada columna, luego se los une en una sola columna nueva y se cuenta la cantidad de veces que se repite esa coincidencia.

```

#Create a dictionary of the most frequent directors by cast
#Separate director and cast
counts[['director', 'cast']] = counts['director_cast'].str.split('---', expand=True)

#We stick with the most frequent director per cast
directors_per_cast = counts.drop_duplicates(subset=['cast'])[['cast', 'director']]

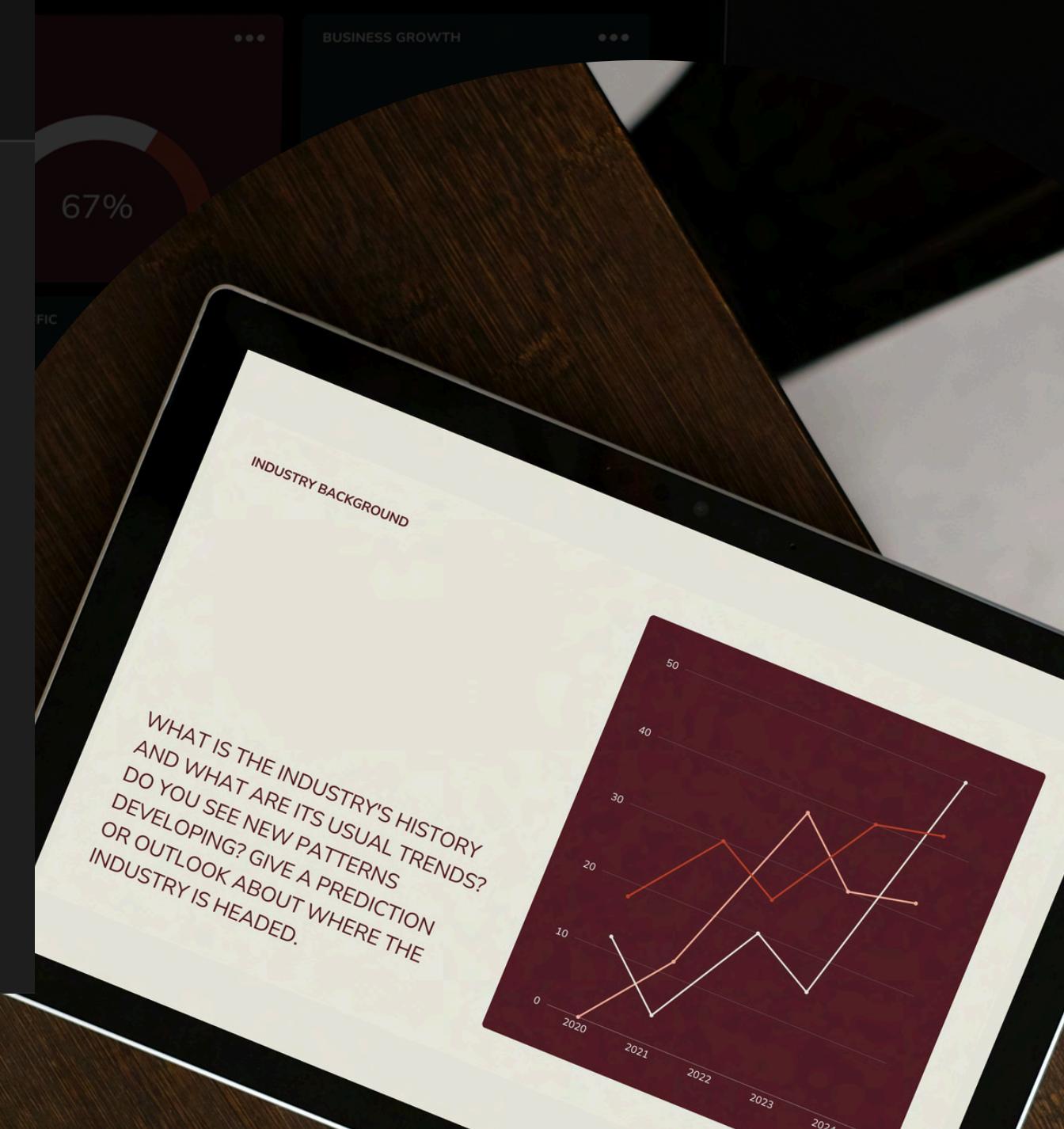
directors_per_cast

```

✓ 0.0s

	cast	director
0	Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jig...	Rajiv Chilaka
1	Aishwarya Rajesh, Vidhu, Surya Ganapathy, Madh...	Rathindran R Prasad
2	Prabhas, Rana Daggubati, Anushka Shetty, Taman...	S.S. Rajamouli
3		Sam Kinison
4		Dave Chappelle
...
5600	Ali Suliman, Saleh Bakri, Yasa, Ali Al-Jabri, ...	Majid Al Ansari
5601	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	David Fincher
5602	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	Ruben Fleischer
5603	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	Peter Hewitt

Se crea un diccionario con los “directores---cast”, para conocer la cantidad de coincidencias, luego esa columna se separa en directores y cast, eliminamos los duplicados y nos quedamos con los mas frecuentes.



Se crea un nuevo df que es llamado "df_merge1" en el cual se guardaran los datos actualizados sobre directores en el cual los campos esten vacios, se reemplazara por el director sugerido y en caso de no haber coincidencias se colocara en ese campo vacio como "Not Given"

```
#We merge with the original DataFrame to bring the suggested director
directors_per_cast.rename(columns={'director': 'director_suggested'}, inplace=True)
df_merge1 = df.merge(directors_per_cast, on='cast', how='left')

#If director is null, we fill it with suggested_director
df_merge1['director'] = df_merge1['director'].fillna(df_merge1['director_suggested'])

#Fill in the remaining text with "Not Given"
df_merge1['director'] = df_merge1['director'].fillna('Not Given')

df_merge1[['director', 'cast']].head(10)
```

✓ 0.0s

	director	cast
0	Kirsten Johnson	NaN
1	Not Given	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...
2	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...
3	Not Given	NaN
4	Not Given	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...
5	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...
6	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...

```

#We are left with only valid country + director combinations
df_country_valid = df[df['country'].notnull() & df['director'].notnull()]

#create a country + director combination
df_country_valid['country_director'] = df_country_valid['country'] + '---' + df_country_valid['director']

#We count how many times each combination is repeated
counts = df_country_valid['country_director'].value_counts().reset_index()
counts.columns = ['country_director', 'count']

#create a dictionary of the most frequent countries by directors
#Separate country and director
counts[['country', 'director']] = counts['country_director'].str.split('---', expand=True)

#We stick with the most frequent country per director
countries_per_director = counts.drop_duplicates(subset=['director'])[['director', 'country']]

#We merge with the original DataFrame to bring the suggested country
countries_per_director.rename(columns={'country': 'country_suggested'}, inplace=True)
df_merge2 = df_merge1.merge(countries_per_director, on='director', how='left')

#if country is null, we fill it with country_suggested
df_merge2['country'] = df_merge2['country'].fillna(df_merge2['country_suggested'])

#Fill in the remaining text with "Not Given"
df_merge2['country'] = df_merge2['country'].fillna('Not Given')

#df_director_valid.head(5)

```

Se analiza la relación entre directores y países para poder obtener coincidencias y así llenar los campos vacíos en la columna de países.

Para ello se realiza el mismo análisis anterior pero en este caso con columnas “country” y “director”.

*Activar Window
Ve a Configuración*

```
#Convert date_added to date
df_merge2['date_added'] = pd.to_datetime(df_merge2['date_added'], errors='coerce')
print(df_merge2.dtypes['date_added'])
```

✓ 0.0s

PC

datetime64[ns]

```
#Create a new column with the difference in years, extract the year from date_added and subtract it from release_year
df_merge2['date_added_year'] = df_merge2['date_added'].dt.year
df_merge2['diff_years'] = df_merge2['date_added_year'] - df['release_year']
```

```
#Calculate the average of that difference (ignoring nulls)
avg_diff = df_merge2['diff_years'].mean()
print(f"The average difference in years is: {avg_diff:.2f}")
```

```
#Round the average difference
```

```
avg_diff_rounded = int(round(avg_diff))
```

```
print(f"The average difference rounded in years is: {avg_diff_rounded:.2f}")
```

✓ 0.0s

PC

Se analiza la diferencia entre “release_year” y “date_added” para reemplazar valores nulos en “date_added”.

Se calcula cuánto tiempo (en años o meses) transcurre desde el lanzamiento de una película o serie (release_year) hasta su publicación en Netflix (date_added).

Luego, a todas las diferencia de tiempo se le saca el promedio para agregarlo a (release_year) y reemplazar los valores nulos.



Se crea una tabla en la cual se toman datos nulos de “date_added” y valores NO nulos de “release_year” para poder realizar la suma con el promedio, previamente obtenido.

Debido a que los formatos de las fechas en ambas columnas son diferentes ya que “release_year” solo muestra el año de estreno, mas no la fecha en DD/MM/YYYY. Y la columna de date_added si muestra la fecha en el formato correspondiente.

Para poder hacer el calculo, se ha tomado en cuenta una fecha media del anual que es el 1ro de Julio.

```
#Filling null values. For values where date_added is empty, i estimate a date using
#release_year + avg_diff (rounded),and assume July 1st as the day (as an average for the year)

#Create a new column with estimated value
df_merge2['date_added_filled'] = df_merge2['date_added']

mask_na = df_merge2['date_added'].isnull() & df_merge2['release_year'].notnull()
df_merge2.loc[mask_na, 'date_added_filled'] = pd.to_datetime(
    df_merge2.loc[mask_na, 'release_year'] + avg_diff_rounded, format='%Y'
) + pd.DateOffset(months=6) #Aproximatively 1st of July

#Replace the original
df_merge2['date_added'] = df_merge2['date_added_filled']
df_merge2.drop(columns='date_added_filled', inplace=True)

#Check for missing nulls
for col in df_merge2.columns:
    nulls = df_merge2[col].isnull().sum()
    print(f'{col}: {nulls}')

✓ 0.0s

show_id: 0
type: 0
title: 0
director: 0
cast: 0
country: 0
date_added: 0
```

Replace null values in the "cast" column with "Not Given"

```
df_merge2['cast'] = df_merge2['cast'].fillna('Not Given')  
✓ 0.0s
```

Removing nulls from rating and duration

```
#Deleting nulls from rating and duration columns  
df_merge2 = df_merge2.dropna(subset=["rating","duration"])  
✓ 0.0s
```

```
#Check for missing nulls  
for col in df_merge2.columns:  
    nulls = df_merge2[col].isnull().sum()  
    print(f"{col}: {nulls}")
```

✓ 0.0s

```
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  
director_suggested: 3042  
country_suggested: 2849  
date_added_year: 98  
diff_years: 98
```

Dentro de la columna “Cast” tiene datos nulos, por lo cual en este caso reemplazaremos los valores nulos de la columna por un “Not Given” ya que no es fácil conocer el Cast de una pelicula por su variabilidad.

Luego eliminamos las filas con valores nulos ya que la columna “Duration” y “Rating” tienen 4 y 3 valores nulos respectivamente, por lo cual si se borran los datos no sera un cambio de mucho impacto ya que tenemos mas de 2500 datoass cargados en este Dataframe.

WHAT IS THE INDUSTRY'S HISTORY
AND WHAT ARE ITS USUAL TRENDS?
DO YOU SEE NEW PATTERNS
DEVELOPING? GIVE A PREDICTION
OR OUTLOOK ABOUT WHERE THE
INDUSTRY IS HEADED.



La columna “listed_in” y “country” contienen diferentes valores separados por comas, se crean columnas nuevas y que mantengan el primer valor, luego se reemplaza esas columnas en sus columnas originales respectivamente.

```
df_merge2['main_genre'] = df_merge2['listed_in'].str.split(',').str[0].str.strip()

#Replace the original
df_merge2['listed_in'] = df_merge2['main_genre']
df_merge2.drop(columns='main_genre', inplace=True)
```

```
df_merge2.head(5)
```

```
✓ 0.0s
```

PowerShell

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	director_suggested
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Not Given	United States	2021-09-25	2020	PG-13	90 min	Documentaries	NaN
1	s2	TV Show	Blood & Water	Not Given	Ama Qamata, Khosi Ngema, Gail Mabalane,	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows	NaN

Activar Windows

```
df_merge2['main_country'] = df_merge2['country'].str.split(',').str[0].str.strip()
```

```
#Replace the original
```

```
df_merge2['country'] = df_merge2['main_country']
df_merge2.drop(columns='main_country', inplace=True)
```

```
df_merge2.head(5)
```

```
✓ 0.0s
```

PowerShell

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	director_suggested
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Not Given	United States	2021-09-25	2020	PG-13	90 min	Documentaries	NaN
1	s2	TV Show	Blood & Water	Not Given	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows	NaN

Activar Windows

Ve a Configuración para activar Windows.

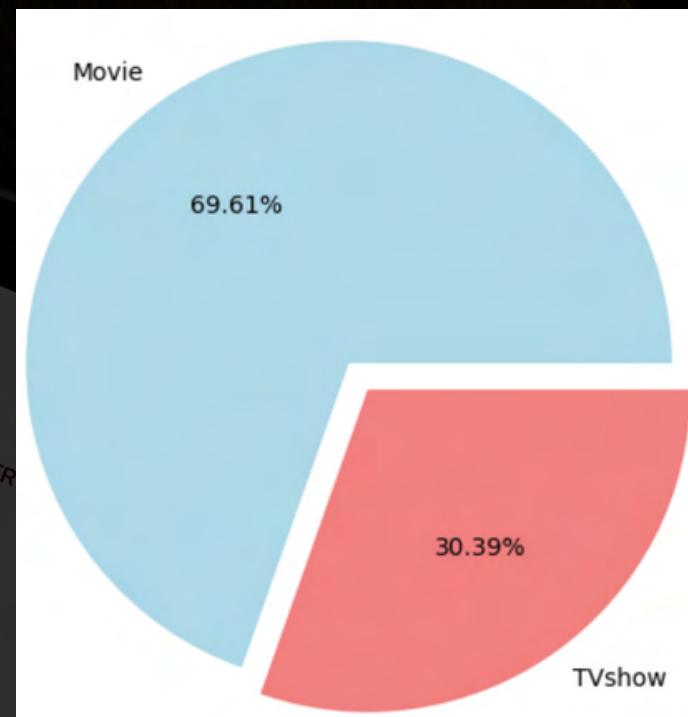


Analysis Vusalization

```
#Total Content on Netflix
types=df_merge2['Type'].value_counts().reset_index()
types
```

✓ 0.0s

Type	count
Movie	6126
TV Show	2674



```
#Create a pie chart to visualize the analysis
plt.figure(figsize=(20,6))
plt.pie(types['count'],labels=['Movie','TVshow'],autopct='%1.2f%%',explode=[0,0.1],colors=['lightblue', 'lightcoral'])
plt.show()
```

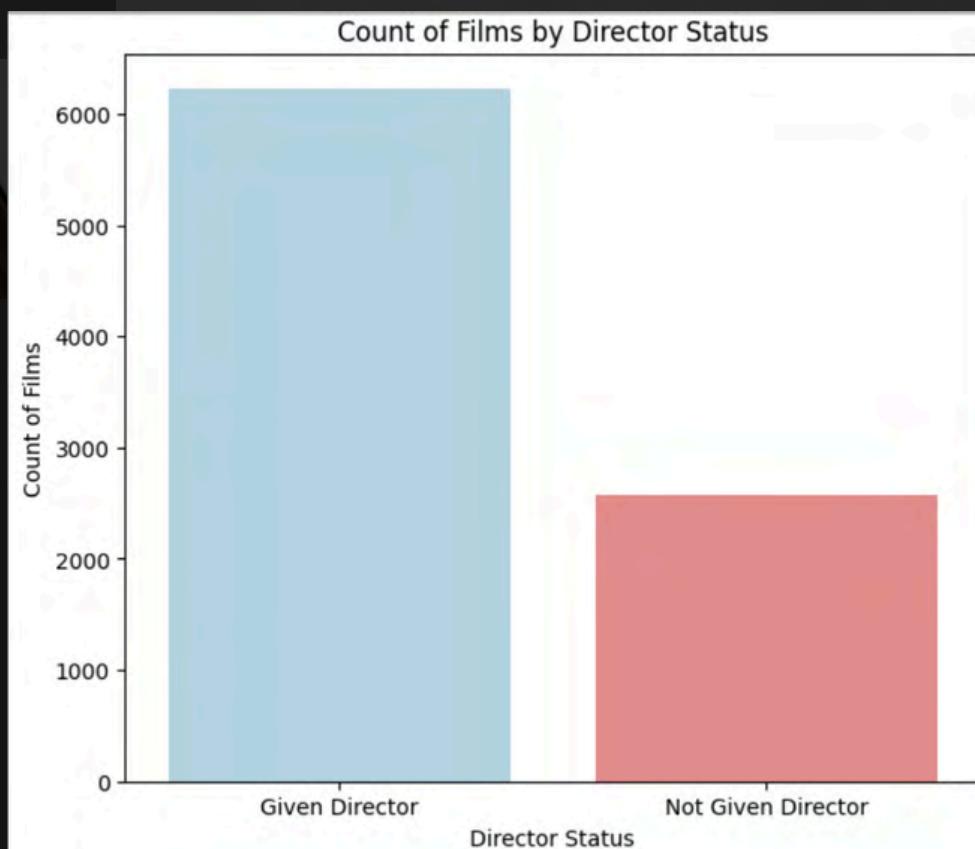
Para poder determinar la cantidad de directores que fueron proporcionados con los que no dentro de la tabla, se realizan dos columnas. Y con eso se hace un conteo de la cantidad de valores, tanto valores dados como valores no dados, y se obtiene la gráfica utilizando “sns.barplot” que es un grafico de barras para poder analizar la diferencia.

El gráfico muestra que mas de 6000 datos en la columna de directores tienen uno asignado y mas de 2000 datos no.

```
#We determine the number of rows with not given values in the directories column
df_merge2['Director'].value_counts().head()
not_given_dir=df_merge2['Director'].value_counts().get('Not Given', 0)
print(not_given_dir)

#We determine the number of rows with true values in the directories column
all_dir = df_merge2['Director'].value_counts().sum()
given_dir=all_dir-not_given_dir
print(given_dir)

#Bar chart showing the number of films in proportion to given and non-given directors
counts = pd.DataFrame({
    'director_status': ['Given Director', 'Not Given Director'],
    'count': [given_dir, not_given_dir]
})
plt.figure(figsize=(7, 6))
sns.barplot(x='director_status', y='count', data=counts, palette=['lightblue', 'lightcoral'])
plt.title('Count of Films by Director Status')
plt.xlabel('Director Status')
plt.ylabel('Count of Films')
plt.xticks(rotation=0)
plt.show()
```



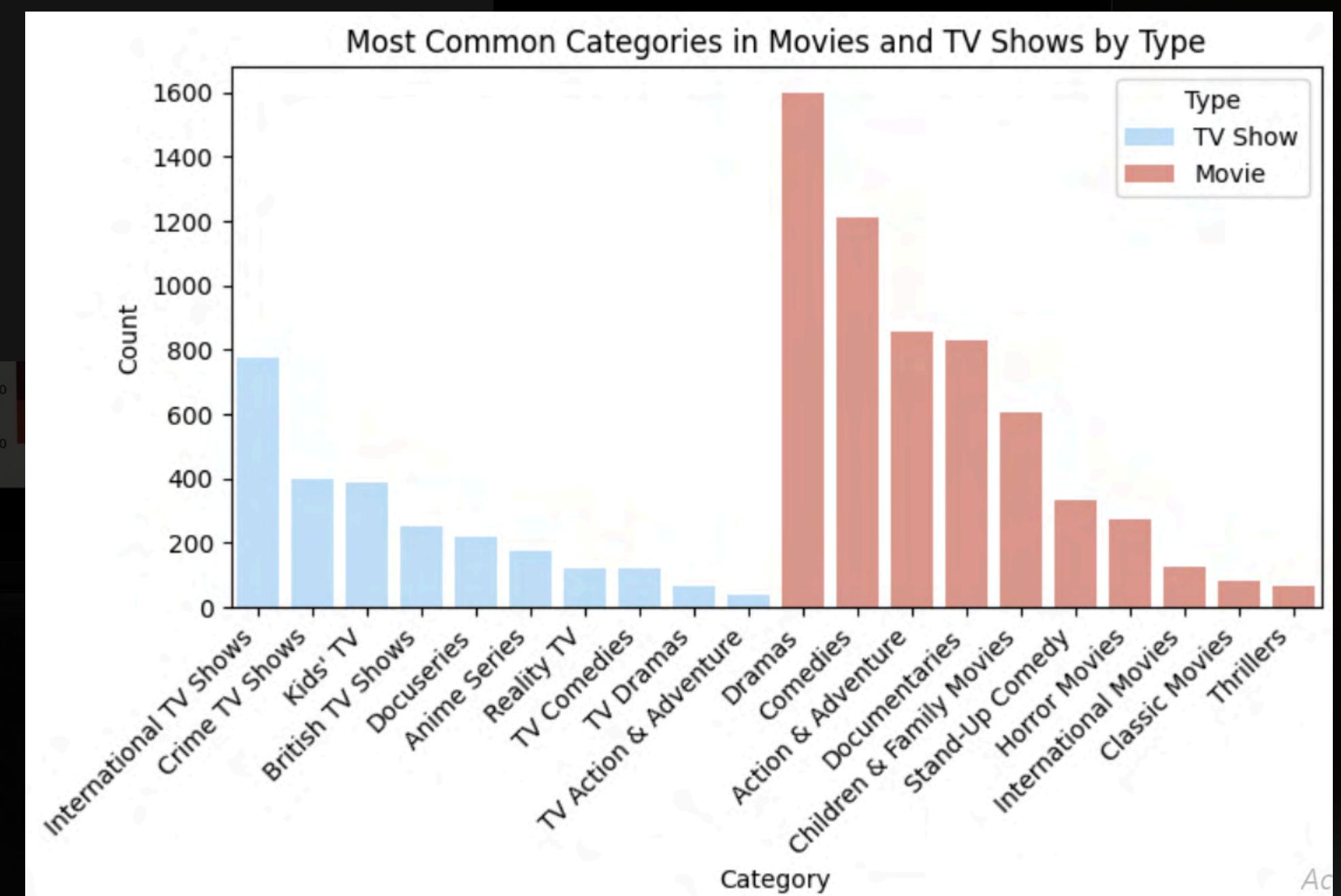
Se realizó un análisis de proporciones mediante un gráfico de torta usando la función “plt.pie” para visualizar la participación de cada tipo de contenido en Netflix, reflejando que un 30,39% son TV Shows y un 69,61% son películas. Esto demuestra que la cantidad de películas en la plataforma superan en un 129,06% a la cantidad de TV Shows.

```

#TOP TV SHOW AND MOVIES
df_split = df_merge2.assign(category=df_merge2['Listed_in'].str.split(', ')).explode('category')
category_count = df_split.groupby(['Type', 'category']).size().reset_index(name='count')
category_count = category_count.sort_values(['Type', 'count'], ascending=False)
category_df = category_count.groupby('Type').head(10).reset_index(drop=True)
plt.figure(figsize=(8, 4))
sns.barplot(
    data=category_df,
    x='category',
    y='count',
    hue='Type',
    palette=[ '#B7E0FF', '#E78F81']
)
plt.title('Most Common Categories in Movies and TV Shows by Type')
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.show()

```

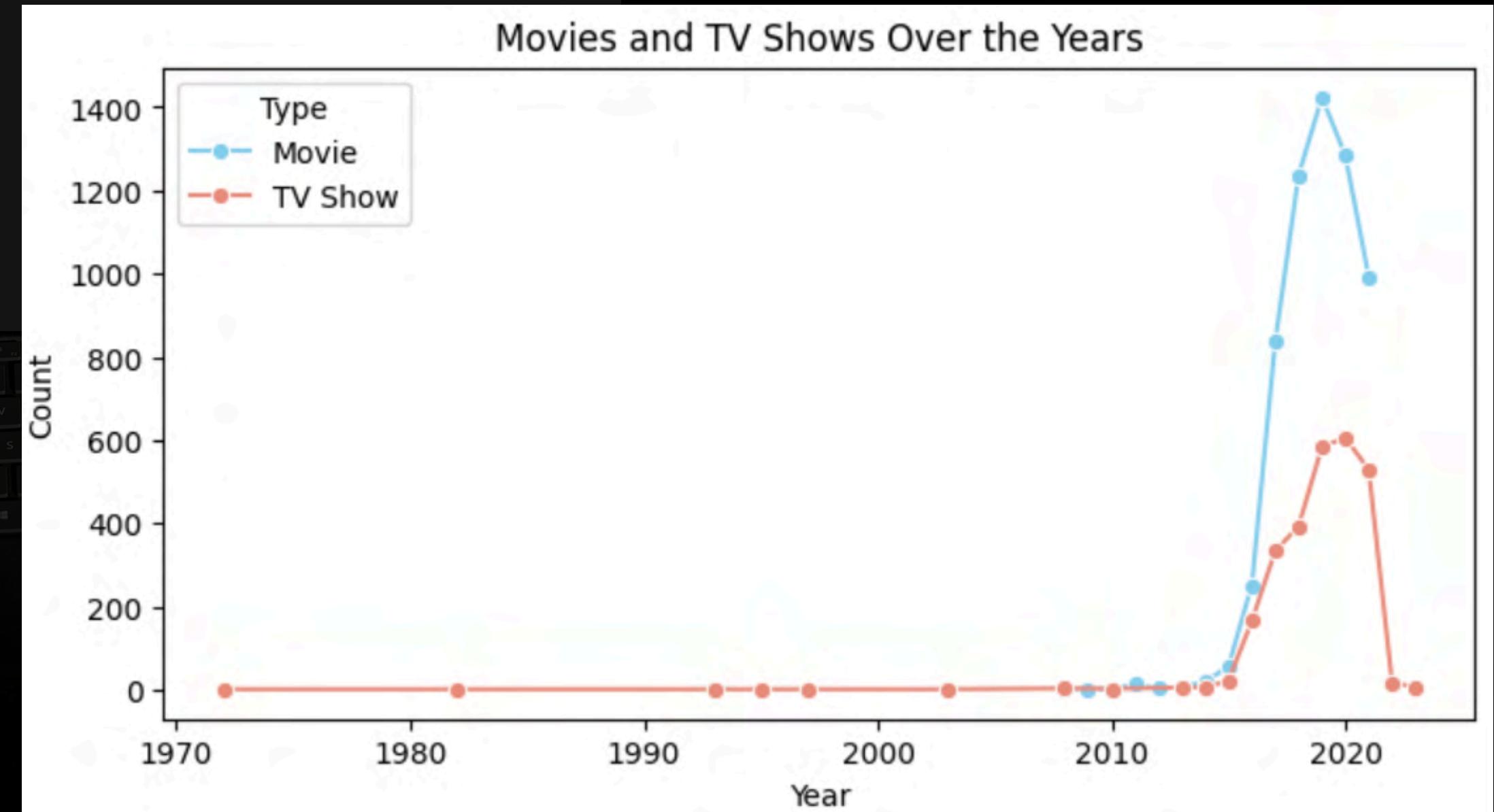
Se realizó un análisis para películas y series sobre cuáles son las categorías más comunes que se estrenan dentro de Netflix. Para ello se utilizó “sns.barplot” que son los gráficos de barra para poder analizar las categorias mas estrenadas en la plataforma.



```

#Yearly Trend of Movie and TV Show Releases
df_merge2['Year'] = df_merge2['Date'].dt.year
cnt_years = df_merge2.groupby(['Type', 'Year']).size().reset_index(name='count')
plt.figure(figsize=(8, 4))
sns.lineplot(
    data=cnt_years,
    x='Year',
    y='count',
    hue='Type',
    marker='o',
    palette=['skyblue', '#E78F81']
)
plt.title('Movies and TV Shows Over the Years')
plt.xlabel('Year')
plt.ylabel('Count')
plt.show()

```

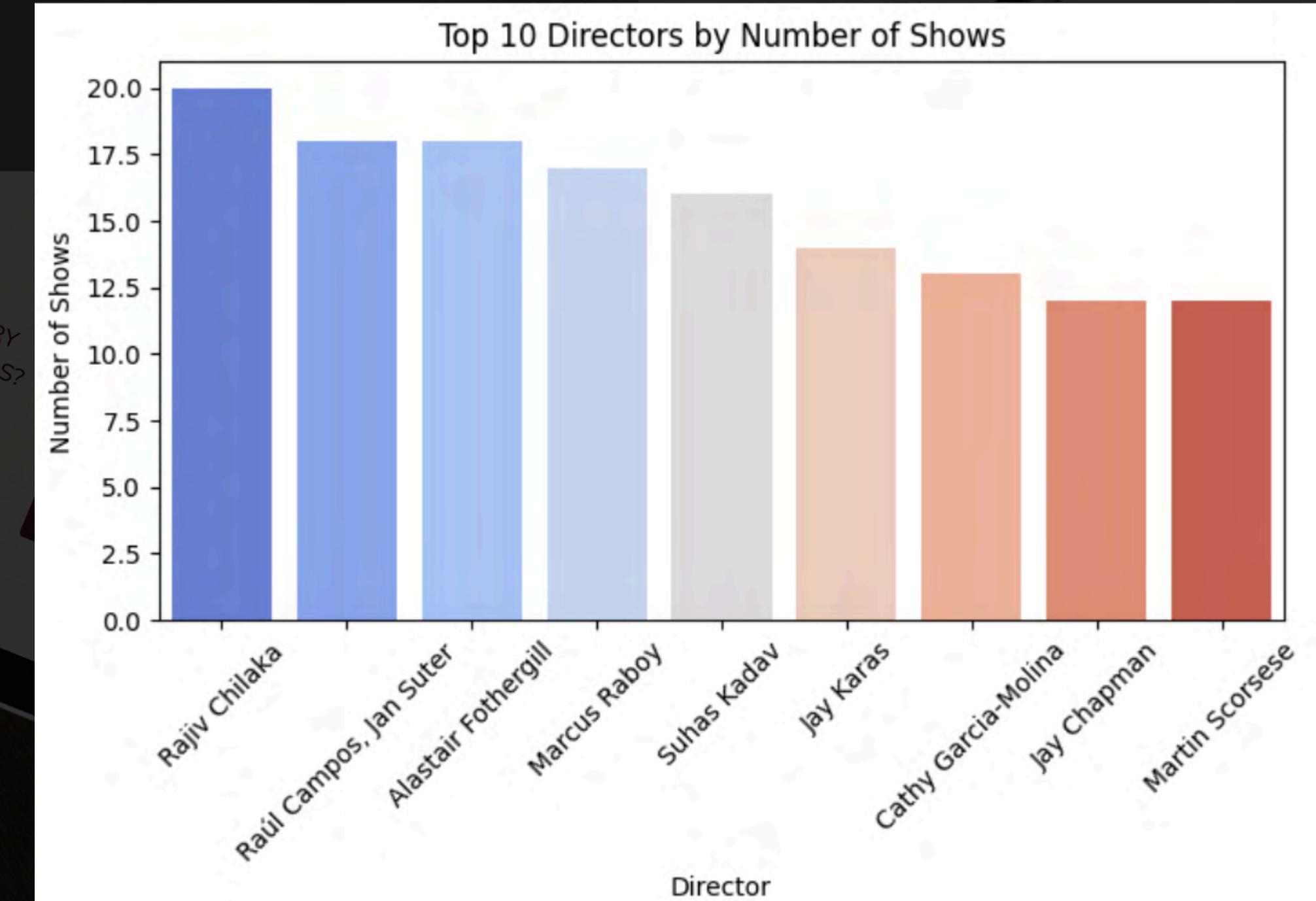


Se realizo un análisis de la cantidad de estrenos tanto en series como en películas a lo largo de los años en Netflix.

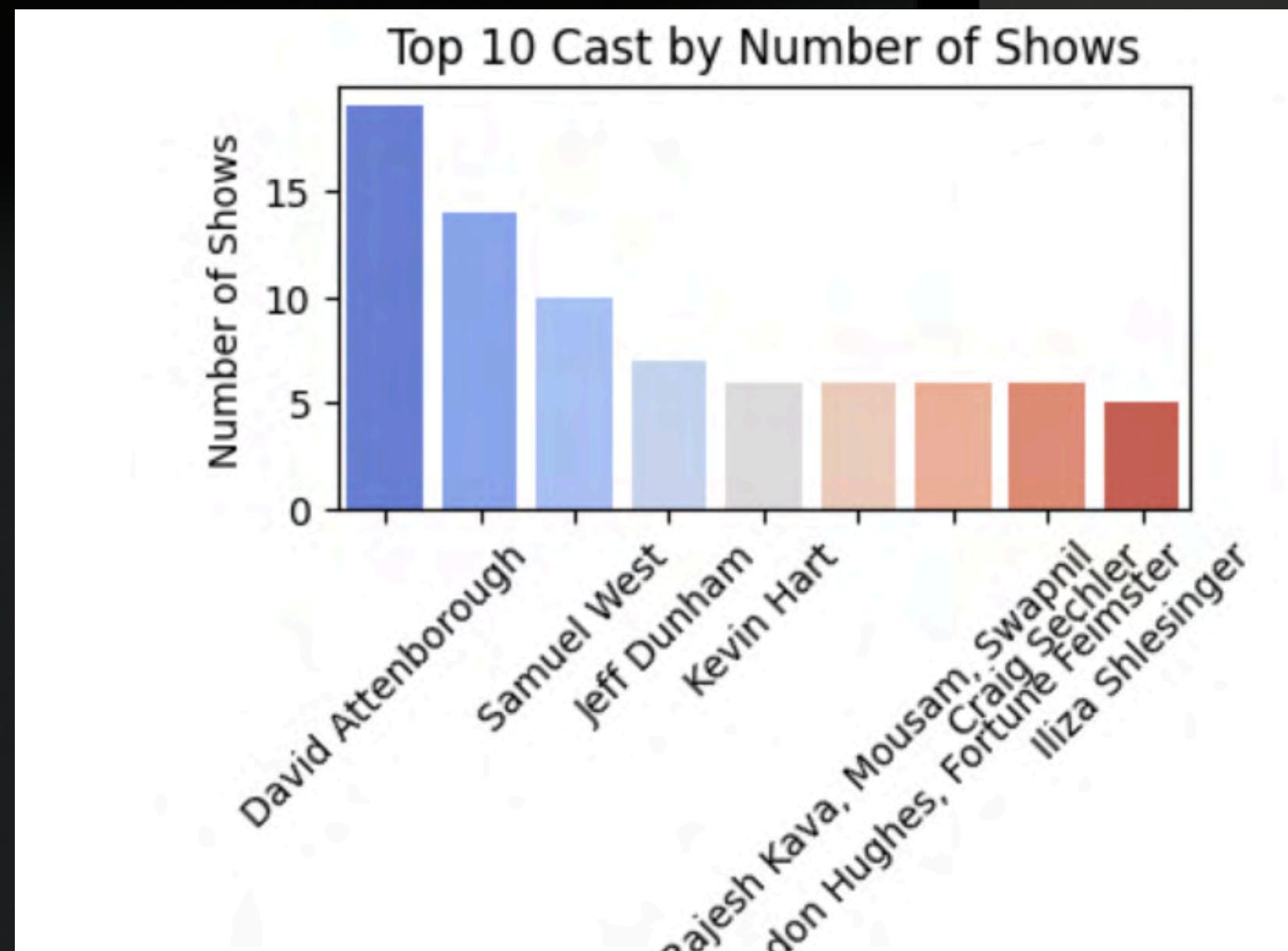
Lo que se puede observar es que a partir de un año en concreto, una creciente significativa de películas en el contenido de Netflix, alcanzando su pico en 2020.

```
#Top 10 Directors in Netflix
top_directors = df_merge2['Director'].value_counts()[1:10]
plt.figure(figsize=(8, 4))
sns.barplot(x=top_directors.index, y=top_directors.values, palette='coolwarm')
plt.title('Top 10 Directors by Number of Shows')
plt.xlabel('Director')
plt.ylabel('Number of Shows')
plt.xticks(rotation=45)
plt.show()
```

Se realizo un gráfico de barras para visualizar la cantidad de contenido que han habido por director y visualizar el top 10.

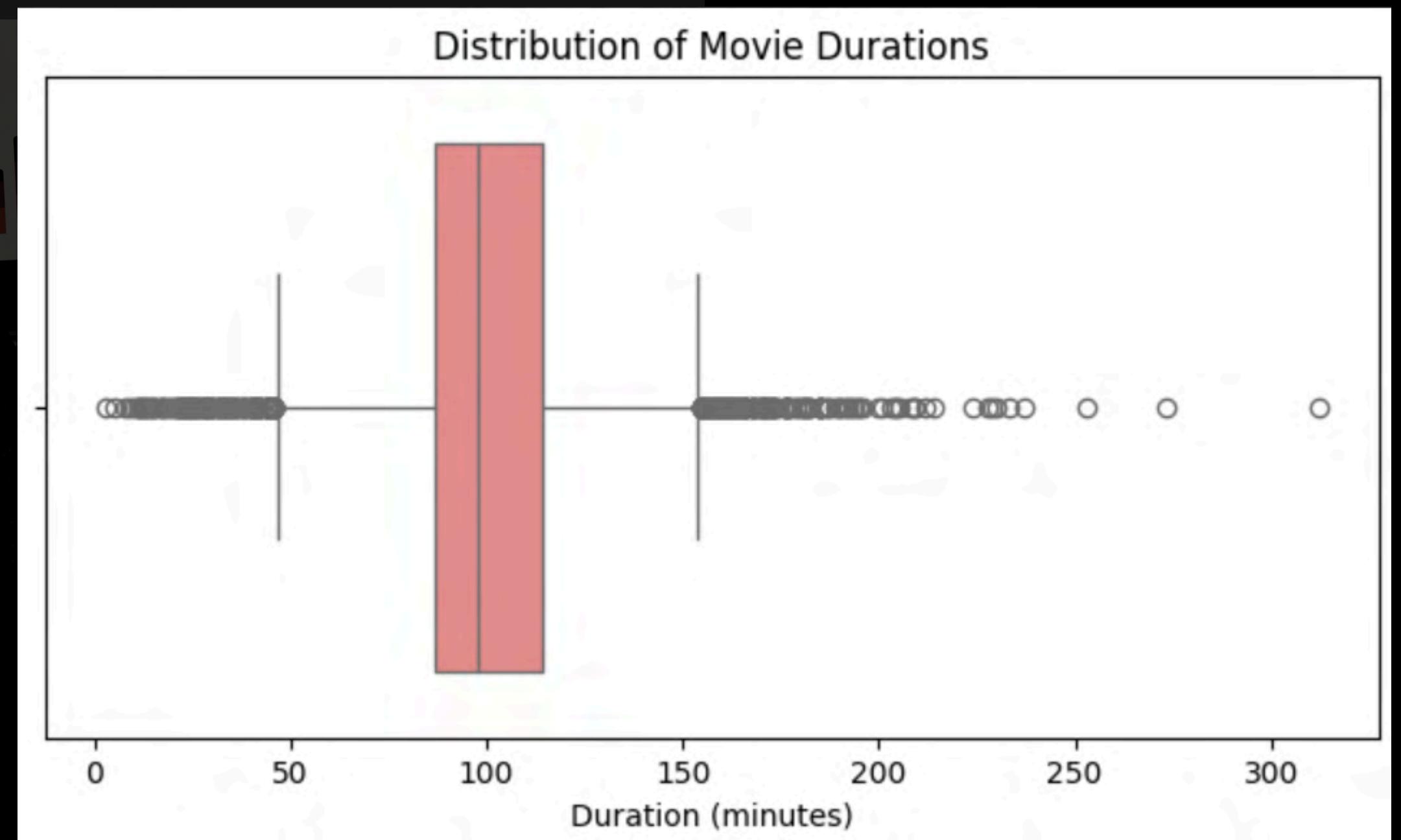


```
#Top 10 Cast in Netflix  
top_directors = df_merge2['Cast'].value_counts()[1:10]  
plt.figure(figsize=(8, 4))  
sns.barplot(x=top_directors.index, y=top_directors.values, palette='coolwarm')  
plt.title('Top 10 Cast by Number of Shows')  
plt.xlabel('Cast')  
plt.ylabel('Number of Shows')  
plt.xticks(rotation=45)  
plt.show()
```



Se realizo el análisis de los cast que han realizado mas contenido dentro de la plataforma, obteniendo un gráfico de barras con los 10 Cast mas vistos.

```
#Movie Durations  
movies_df = df_merge2[df_merge2['Type'] == 'Movie']  
movies_df['Duration'] = movies_df['Duration'].str.replace(' min', '').astype(int)  
plt.figure(figsize=(8, 4))  
sns.boxplot(x=movies_df['Duration'], color= 'lightcoral')  
plt.title('Distribution of Movie Durations')  
plt.xlabel('Duration (minutes)')  
plt.show()
```



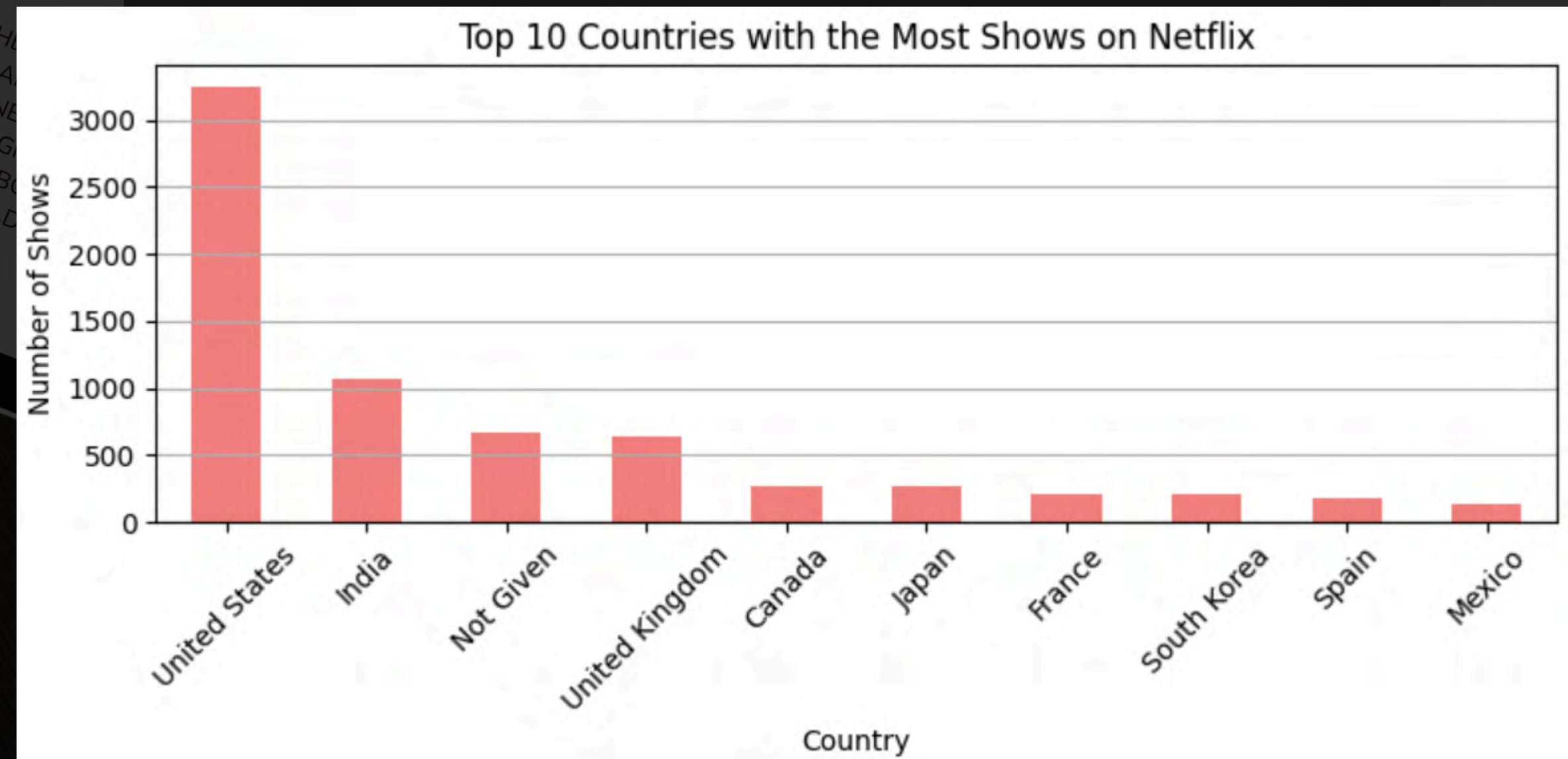
Se analizo la duración que tiene cada pelicula en Netflix, obteniendo una media de 100 min por pelicula.

Por ultimo se realizo un análisis para determinar los paises con mas contenido en Netflix.

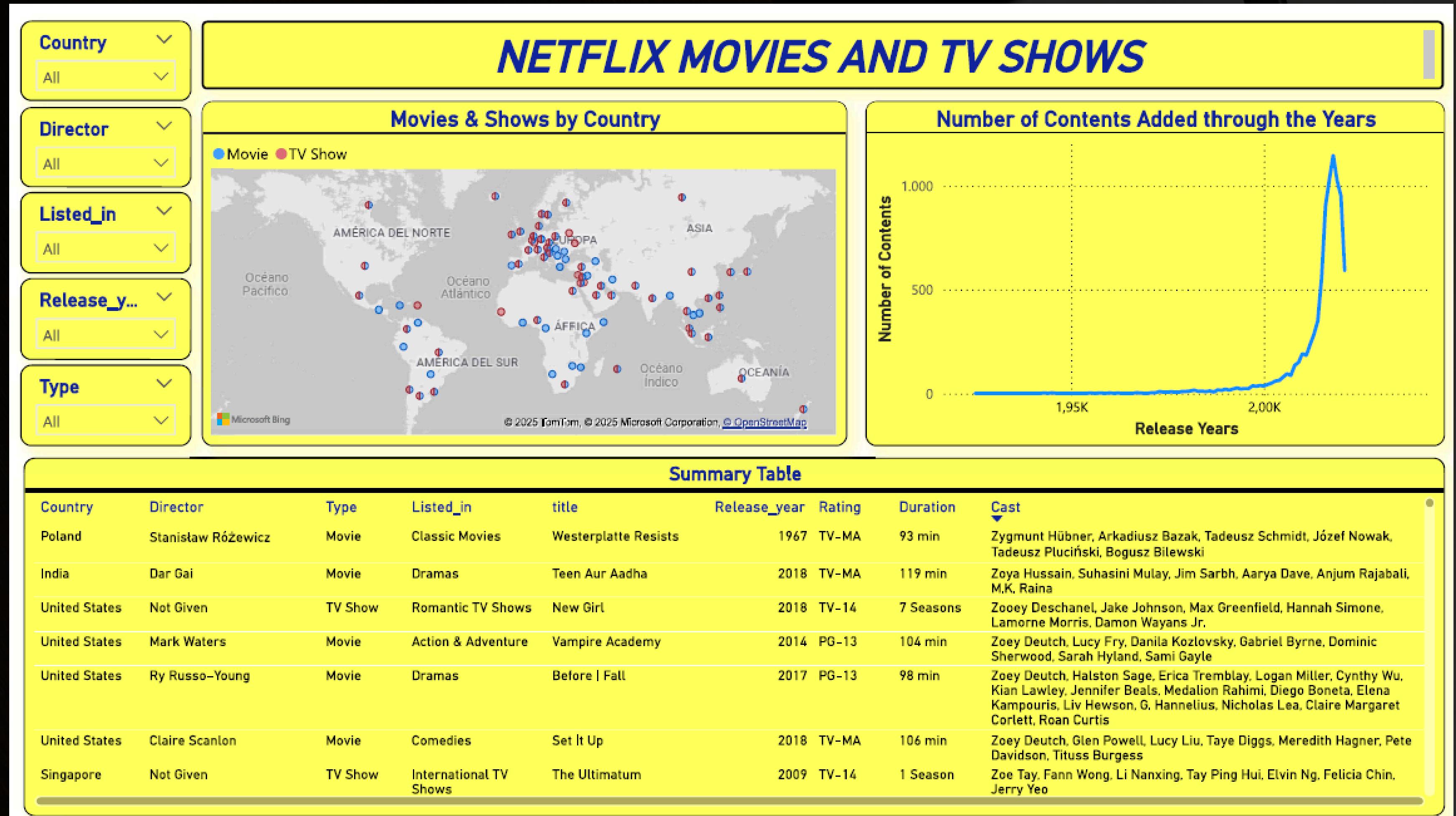
Se obtuvo que estados unidos es el pais con mas contenido anañadido en la plataforma, entre peliculas y series.

De segundo puesta esta la india y en tercer lugar se encuentra el reino unido.

```
#Top 10 countries with Netflix Content
country_counts = df_merge2['Country'].value_counts().head(10)
plt.figure(figsize=(8, 4))
country_counts.plot(kind='bar', color='lightcoral')
plt.title('Top 10 Countries with the Most Shows on Netflix')
plt.xlabel('Country')
plt.ylabel('Number of Shows')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```



DASHBOARD EN POWER BI



DASHBOARD EN POWER BI

NETFLIX MOVIES AND TV SHOWS

Country

All

Summary Table

Country	Director	Type	Listed_in	title	Release_year	Rating	Duration	Cast
Not Given	Not Given	TV Show	TV Shows	Pioneers: First Women Filmmakers*	1925	TV-14	1 Season	Not Given
United States	Frank Capra	Movie	Classic Movies	Prelude to War	1942	TV-14	52 min	Not Given
United States	John Ford	Movie	Classic Movies	The Battle of Midway	1942	TV-14	18 min	Henry Fonda, Jane Darwell
United States	Frank Capra, Anatole Litvak	Movie	Documentaries	Why We Fight: The Battle of Russia	1943	TV-PG	82 min	Not Given
United States	John Ford	Movie	Classic Movies	Undercover: How to Operate Behind Enemy Lines	1943	TV-PG	61 min	Not Given
United States	John Huston	Movie	Documentaries	WWII: Report from the Aleutians	1943	TV-PG	45 min	Not Given
United States	Frank Capra, John Huston, Hugh Stewart, Roy Boulting, Anthony Veiller	Movie	Classic Movies	Tunisian Victory	1944	TV-14	76 min	Burgess Meredith
United States	Stuart Heisler	Movie	Classic Movies	The Negro Soldier	1944	TV-14	40 min	Not Given
United States	Frank Capra, Joris Ivens	Movie	Classic Movies	Know Your Enemy - Japan	1945	TV-14	63 min	Walter Huston, Dana Andrews
United States	George Stevens	Movie	Classic Movies	Nazi Concentration Camps	1945	TV-MA	59 min	Not Given
United States	John Huston	Movie	Classic Movies	San Pietro	1945	TV-14	32 min	Not Given

Director

All

Listed_in

All

Release_y...

All

Type

All

Directors with Most Contents on Netflix

● Movie ● TV Show

Director	Percentage
Not Given	93,54%
Rajiv Chilaka	95,00%
Alastair Fothergill	22,22%
Raúl Campos, Jan ...	100,00%
Marcus Raboy	94,12%
Suhas Kadav	100,00%
Jay Karas	100,00%
Cathy Garcia-Moli...	100,00%
Jay Chapman	100,00%

Movies & Shows

Type ● Movie ● TV Show

Type	Count	Percentage
TV Show	2,67K	(30,39%)
Movie	6,12K	(69,61%)

Number of Contents per Country

Number of Contents

Country	Number of Contents
United States	3,2K
India	1,1K
Not Given	0,7K
United Kingdom	0,6K
Canada	0,3K
Japan	0,3K
France	0,2K
South Korea	0,2K
Spain	0,2K
Mexico	0,1K
Egypt	0,1K
Australia	0,1K
Turkey	0,1K
Nigeria	0,1K
Germany	0,1K
China	0,1K
Brazil	0,1K
Indonesia	0,1K
Taiwan	0,1K

GROUND

THE INDUSTRY'S HISTORY

I WANT TO SAY

THANK YOU

FOR YOUR ATTENTION

