In [44]: In [45]:	<pre>import pandas as pd df=pd.read_csv("Netflix.csv") # Create a new DataFrame with only 'title' and 'director' columns (Note using copy to get it from the main data frame) title_director_table = df[['title', 'director']].copy() # Split the 'title' and 'director' columns into lists</pre>
	<pre>title_director_table['title'] = title_director_table['director'].str.split(',') title_director_table['director'] = title_director_table['director'].str.split(',') # Explode the 'title' and 'director' columns to create multiple rows title_director_table = title_director_table.explode('title', ignore_index=True) title_director_table = title_director_table.explode('director', ignore_index=True) # Create a new DataFrame with only 'title' and 'cast' columns title_cast_table = df[['title', 'cast']].copy() # Split the 'title' and 'cast' columns into lists</pre>
In [48]:	<pre>title_cast_table['title'] = title_cast_table['title'].str.split(',') title_cast_table['cast'] = title_cast_table['cast'].str.split(',') # Explode the 'title' and 'cast' columns to create multiple rows title_cast_table = title_cast_table.explode('title', ignore_index=True) title_cast_table = title_cast_table.explode('cast', ignore_index=True) # Create a new DataFrame with only 'title' and 'country' columns title_country_table = df[['title', 'country']].copy() # Split the 'title' and 'country' columns into lists</pre>
In [49]:	<pre>title_country_table['title'] = title_country_table['title'].str.split(',') title_country_table['country'] = title_country_table['country'].str.split(',') # Explode the 'title' and 'country' columns to create multiple rows title_country_table = title_country_table.explode('title', ignore_index=True) title_country_table = title_country_table.explode('country', ignore_index=True) # Create a copy of the DataFrame explicitly title_listed_in_table = df[['title', 'listed_in']].copy() # Split the 'title' and 'listed_in' columns title listed in table['title'] = title listed in table['title'].str.split(',')</pre>
In [50]: Out[50]:	title_listed_in_table['title'] = title_listed_in_table['title'].str.split(',') title_listed_in_table['listed_in'] = title_listed_in_table['listed_in'].str.split(',') # Explode the 'title' and 'listed_in' columns title_listed_in_table = title_listed_in_table.explode('title', ignore_index=True) title_listed_in_table = title_listed_in_table.explode('listed_in', ignore_index=True) title_cast_table.head(1) title_cast_ 0 Dick Johnson Is Dead_NaN
	title_director_table.head(1) title director Dick Johnson Is Dead Kirsten Johnson title_country_table.head(1)
Out[53]:	0 Dick Johnson Is Dead United States title_listed_in_table.head(1) title listed_in 0 Dick Johnson Is Dead Documentaries
In [54]: In [55]: Out[55]:	# Join all the tables together on the 'title' column final_table = title_director_table.merge(title_cast_table, on='title', how='inner') \
In [56]: Out[56]:	Blood & Water NaN Ama Qamata South Africa TV Dramas Blood & Water NaN Ama Qamata South Africa TV Mysteries NaN Khosi Ngema South Africa International TV Shows df . head() ### df is the rw dataset where we started working show_id type title directors South Africa of the country Shows name of the country
In [57]:	<pre>final_table = final_table.merge(df, on='title', how='left')</pre>
In [60]: Out[60]:	Dick Johnson Is Dead Johnson NaN United Dead Johnson NaN United States Documentaries S1 Movie Kirsten Johnson NaN United States September 25, 2021 2020.0 PG-13 90 min Documentaries As her father nears the end of his life, filmm 1 Blood & Water NaN Ama Qamata South Africa South Africa TV Dramas S2 TV NaN Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban South Africa September 24, 2021.0 TV- 2 International TV Shows, TV After crossing paths at a party, a Cape Town t 3 Blood & Water NaN Ama South Africa TV Mysteries S2 TV NaN Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban South Africa September 24, 2021.0 TV- 2 International TV Shows, TV After crossing paths at a party, a Cape Town t
In [61]: Out[61]:	A Blood & Water Nan Qamata South Africa IV Mysteries \$2 Show Nan Mabalane, Thaban South Africa 2021 2021.0 MA Seasons Dramas, TV Mysteries Cape Town t 4 Blood & Water Nan Khosi Ngema South Africa International TV Shows \$2 TV Show Nan Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban South Africa September 24, 2021.0 TV- 2 International TV Shows, TV Mysteries Cape Town t 5 South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t 6 Transport Transport Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t 6 Transport Transport Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t 7 Transport Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t 8 Transport Transport Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas, TV Mysteries Cape Town t South Africa September 24, 2021.0 TV- 2 MA Seasons Dramas,
<pre>In [62]: In [63]: Out[63]:</pre>	final_table = final_table.drop(['director_y', 'cast_y', 'country_y', 'listed_in_y'], axis=1) #dropping unessery col to get better visibity after joining df and the final_table ##note: y col reperesnting the prev colomns therefore dropping them final_table.head(5) title director_x cast_x country_x listed_in_x show_id type date_added release_year rating duration description O Dick Johnson Is Dead Kirsten Johnson NaN United States Documentaries s1 Movie September 25, 2021 2020.0 PG-13 90 min As her father nears the end of his life, filmm 1 Blood & Water NaN Ama Qamata South Africa International TV Shows s2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t
<pre>In [64]: Out[64]:</pre>	#afterdropping unessery colomns now we can get the useful colomns Index(['title', 'director_x', 'cast_x', 'country_x', 'listed_in_x', 'show_id',
<pre>In [65]: In [66]: Out[66]:</pre>	'description'], dtype='object') final_table.rename(columns={"director_x": "director",
In [67]: Out[67]:	O Dick Johnson Is Dead Kirsten Johnson NaN United States Documentaries s1 Movie September 25, 2021 2020.0 PG-13 90 min As her father nears the end of his life, filmm Blood & Water NaN Ama Qamata South Africa International TV Shows s2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Blood & Water NaN Ama Qamata South Africa TV Dramas s2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Blood & Water NaN Ama Qamata South Africa TV Mysteries s2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t final_table.tail(4) title director cast country listed_in show_id type date_added release_year rating duration description
In [30]: In [175	
In [68]: Out[68]:	Dick Johnson Is Dead Kirsten Johnson NaN United States Documentaries s1 Movie September 25, 2021 2020.0 PG-13 90 min As her father nears the end of his life, filmm Blood & Water NaN Ama Qamata South Africa International TV Shows s2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Blood & Water NaN Ama Qamata South Africa TV Dramas s2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Blood & Water NaN Ama Qamata South Africa TV Mysteries s2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t
In [69]:	4 Blood & Water NaN Khosi Ngema South Africa International TV Shows September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t 5 Blood & Water NaN Khosi Ngema South Africa TV Dramas S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t 6 Blood & Water NaN Khosi Ngema South Africa TV Mysteries S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t 7 Blood & Water NaN Gail Mabalane South Africa International TV Shows S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t 8 Blood & Water NaN Gail Mabalane South Africa TV Dramas S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t 9 Blood & Water NaN Gail Mabalane South Africa TV Mysteries S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t 6 Elbod & Water NaN Gail Mabalane South Africa TV Dramas S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t 8 Elbod & Water NaN Gail Mabalane South Africa TV Mysteries S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t 9 Elbod & Water NaN Gail Mabalane South Africa TV Mysteries S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t
Out[69]:	title director cast country listed_in show_id type date_added release_year rating duration Dick Johnson Is Dead Kirsten Johnson NaN United States Documentaries S1 Movie September 25, 2021 2020.0 PG-13 90 min As her father nears the end of his life, filmm Blood & Water NaN Ama Qamata South Africa International TV Shows S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Blood & Water NaN Ama Qamata South Africa TV Dramas S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Blood & Water NaN Khosi Ngema South Africa International TV Shows S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Blood & Water NaN Khosi Ngema South Africa International TV Shows S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Blood & Water NaN Khosi Ngema South Africa International TV Shows S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Blood & Water NaN Khosi Ngema South Africa International TV Shows S2 TV Show September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t Control of the properties of
In [70]:	<pre># Define categorical and continuous variables categorical_columns = ['type', 'title', 'director', 'cast', 'country', 'rating', 'listed_in', 'description'] continuous_columns = ['show_id', 'date_added', 'duration'] # Replace null values in categorical columns with "unknown_column_name" final_table[categorical_columns] = final_table[categorical_columns].fillna("Unknown") # Replace null values in continuous columns with 0 final_table[continuous_columns] = final_table[continuous_columns].fillna(0) ##final_table.head(10)</pre>
In [73]: Out[73]:	<pre>final_table.head()</pre>
In [87]: Out[87]:	<pre>final_table.head()</pre>
_	4 Blood & Water Unknown Khosi Ngema South Africa International TV Shows September 24, 2021 2021.0 TV-MA 2 Seasons After crossing paths at a party, a Cape Town t ####a. Find the number of movies produced in each country and pick the top 10 countries. final_table['country'].nunique() final_table['title'].nunique()
Out[119]:	# Group the data by 'country' and aggregate unique movie titles unique_movies_by_country = final_table.groupby('country')['title'].unique().reset_index() # Count the number of unique movies for each country unique_movies_by_country['unique_movie_count'] = unique_movies_by_country['title'].apply(len) # Sort the DataFrame by 'unique_movie_count' in descending order unique_movies_by_country = unique_movies_by_country.sort_values(by='unique_movie_count', ascending=False)
	192 Unknown 848 190 United Kingdom 632 106 United States 483 122 Canada 276 148 Japan 262 132 France 215 179 South Korea 215
Out[148]:	final_table['type'].value_counts() Movie 200045
In [142	Name: type, dtype: int64 # Filter the DataFrame to include only 'TV Show' entries tv_shows_df = final_table['type'] == 'TV Show'] # Group the data by 'country' and count the number of unique TV show titles tv_show_counts_by_country = tv_shows_df.groupby('country')['title'].nunique().reset_index() # Sort the DataFrame by the count of unique TV show titles in descending order tv_show_counts_by_country = tv_show_countrs_by_country.sort_values(by='title', ascending=False) # Select the top 10 countries
In [151 Out[151]:	top_10_countries_tv_show = tv_show_counts_by_country.head(10) top_10_countries_tv_show country title 100 United States 833 101 Unknown 384 99 United Kingdom 245 69 Japan 173
In 「191	90 South Korea 159 42 United States 90 50 Canada 84 64 India 81 94 Taiwan 70 60 France 64 # Create a bar chart for the top 10 countries for movies
	plt.figure(figsize=(12, 6)) plt.bar(top_10_countries_movies['country'], top_10_countries_movies['unique_movie_count']) plt.xlabel('Country') plt.ylabel('Number of Unique Movies') plt.title('Top 10 Countries with the Most Unique Movies') plt.xticks(rotation=45) plt.tight_layout() plt.show() Top 10 Countries with the Most Unique Movies
	3000 - 2500 - 2000 -
	2000 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 1500 - 15
In [190…	Interest the transfer of the state of the st
	# Create a bar chart for the top 10 TV show-producing countries plt.figure(figsize=(12, 6)) plt.bar(top_10_countries_tv_show['country'], top_10_countries_tv_show['title']) plt.xlabel('Country') plt.ylabel('Number of Unique TV Shows') plt.title('Top 10 Countries with the Most Unique TV Shows') plt.xticks(rotation=45) plt.tight_layout() plt.show() Top 10 Countries with the Most Unique TV Shows Top 10 Countries with the Most Unique TV Shows
	800 - 700 - 88 600 - 89 500 -
	200 - 100 -
	Tinal_table.columns
Out[174]:	<pre>"date_added", 'release_year", 'rating", 'duration', 'description'], dtype='object') # Convert the 'date_added' column to datetime final_table['date_added'] = pd.to_datetime(final_table['date_added']) # Extract the week from the 'date_added' column and create a new column 'added_week' final_table['added_week'] = final_table['date_added'].dt.strftime('%U') # Separate data into TV shows and movies</pre>
	<pre>tv_shows_df = final_table[final_table['type'] == 'TV Show'] movies_df = final_table[final_table['type'] == 'Movie'] # Group the data by the week and count the total number of TV shows and movies added tv_show_counts_by_week = tv_shows_df['added_week'].value_counts().reset_index().rename(columns={'index': 'week', 'added_week': 'tv_show_count'})) movie_counts_by_week = movies_df['added_week'].value_counts().reset_index().rename(columns={'index': 'week', 'added_week': 'movie_count'})) # Find the best week for TV shows best_week_tv_show = tv_show_counts_by_week.loc[tv_show_counts_by_week['tv_show_count'].idxmax()] # Find the best week for movies best_week_movie = movie_counts_by_week.loc[movie_counts_by_week['movie_count'].idxmax()]</pre>
Out[176]:	Name: 0, dtype: object best_week_movie
In [180	# Convert the 'date_added' column to datetime final_table['date_added'] = pd.to_datetime(final_table['date_added']) # Extract the month from the 'date_added' column and create a new column 'added_month' final_table['added_month'] = final_table['date_added'].dt.strftime('%B') # Separate data into TV shows and movies tv_shows_df = final_table[final_table['type'] == 'TV Show'] movies_df = final_table[final_table['type'] == 'Movie'] # Group the data by the month and count the total number of TV shows and movies added
In [196 Out[196]:	tv_show_counts_by_month = tv_shows_df['added_month'].value_counts().reset_index().rename(columns={'index': 'month', 'added_month': 'tv_show_count'}) movie_counts_by_month = movies_df['added_month'].value_counts().reset_index().rename(columns={'index': 'month', 'added_month': 'tv_show_count'}) # Find the best month for TV shows best_month_tv_show = tv_show_counts_by_month.loc[tv_show_counts_by_month['tv_show_count'].idxmax()] # Find the best month for movies best_month_movie = movie_counts_by_month.loc[movie_counts_by_month['movie_count'].idxmax()] best_month_movie.head() month February
In [195 Out[195]: In [193 Out[193]:	Name: 0, dtype: object final_table.head(1)
In [197	O Dick Johnson Is Dead Kirsten Johnson Unknown United States Documentaries s1 Movie 2021-09-25 2020.0 PG-13 90 min As her father nears the end of his life, filmm 38 September # Extract 'director' and 'title' columns director_title_df = final_table[['director', 'title']] # Group by directors and count the unique titles they have directed director_counts = director_title_df.groupby('director')['title'].nunique().reset_index() # Sort directors by the count of unique titles in descending order director_counts = director_counts.sort_values(by='title', ascending=False)
In [198 Out[198]:	# Select the top 10 directors top_10_directors = director_counts.head(10) top_10_directors : director title 4905 Unknown 2672 4020 Rajiv Chilaka 22 4067 Raúl Campos 18
	261 Jan Suter 18 3235 Marcus Raboy 17 4651 Suhas Kadav 16 2450 Jay Karas 15 1382 Cathy Garcia-Molina 15 3306 Martin Scorsese 12 5075 Youssef Chahine 12
In [204	<pre># Extract 'cast' and 'title' columns actor_title_df = final_table[['cast', 'title']] # Group by actors and count the unique titles they have appeared in actor_counts = actor_title_df.groupby('cast')['title'].nunique().reset_index() # Sort actors by the count of unique titles in descending order actor_counts = actor_counts.sort_values(by='title', ascending=False) # Select the top 10 actors top_10_actors = actor_counts.head(10)</pre>
In [206	<pre>import pandas as pd import matplotlib.pyplot as plt # Create a bar chart for the top 10 directors plt.figure(figsize=(12, 6)) plt.bar(top_10_directors['director'], top_10_directors['title']) plt.xlabel('director') plt.ylabel('Name_of_directors') plt.title('Top 10 Directors') plt.xticks(rotation=45)</pre>
	plt.tight_layout() plt.show() Top 10 Directors
	2000 - Sept 1500 -
	Jukroun Rain Children Rain Rain Rain Rain Rain Rain Rain Rai
In [205	<pre>import pandas as pd import matplotlib.pyplot as plt # Create a bar chart for the top 10 directors plt.figure(figsize=(12, 6)) plt.bar(top_10_actors['cast'], top_10_actors['title']) plt.xlabel('cast')</pre>
	plt.ylabel('Name_of_actors') plt.title('Top 10 Actors') plt.xticks(rotation=45) plt.tight_layout() plt.show() Top 10 Actors
	700 - 600 - 800 - 90 - 400 - 400 -
	300 - 200 - 100 -
In [220 Out[220]:	
In [223	# Extract 'listed_in' column genre_df = final_table[['listed_in']] # Group by genres and count the number of unique titles in each genre genre_counts = genre_df['listed_in'].value_counts().reset_index() genre_counts.columns = ['Genre', 'Count'] # Sort genres by the count of unique titles in descending order genre_counts = genre_counts.sort_values(by='Count', ascending=False) # Select the top 10 genres
In [224 Out[224]:	top_10_genres = genre_counts.head(10) top_10_genres Genre Count International Movies 39775 Comedies 25338 Dramas 24354
	3 Romantic Movies 18468 4 Independent Movies 16991 5 Action & Adventure 13004 6 Dramas 10319 7 Children & Family Movies 9381 8 International TV Shows 8398 9 TV Dramas 8018
In [230	# Create a pie chart for the top 10 genres plt.figure(figsize=(8, 8)) plt.pie(top_10_genres['Genre'], labels=top_10_genres['Genre'], autopct='%1.1f%', startangle=140) plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle. plt.title('Genres Distribution') # Show the pie chart plt.show() Genres Distribution Children & Family Movies Dramas
	International TV Shows TV Dramas Action & Adventure 4.8% 5.4% 7.5% Independent Movies
	International Movies 22.9% 10.6% Romantic Movies
	14.6% 14.0% Dramas
In [231	<pre># Convert 'date_added' column to datetime final_table['date_added'] = pd.to_datetime(final_table['date_added'], errors='coerce') # Calculate the time difference in days between 'date_added' and 'release_year' final_table['days_to_netflix'] = (final_table['date_added'] - pd.to_datetime(final_table['release_year'], format='\text{\text{WY'}})).dt.days # Filter out rows with missing or negative time differences (if any) valid_time_diff = final_table[final_table['days_to_netflix'] >= 0] # Calculate the mode (most common) of the time differences</pre>
Out[234]:	<pre>mode_days_to_netflix = valid_time_diff['days_to_netflix'].mode().iloc[0] mode_days_to_netflix</pre>
٦.	