User

I have uploaded the following files for you to use: Netflix Dataset.csv

I uploaded the file: `Netflix Dataset.csv`

How many movies vs tv shows are on netflix

Julius

run_code

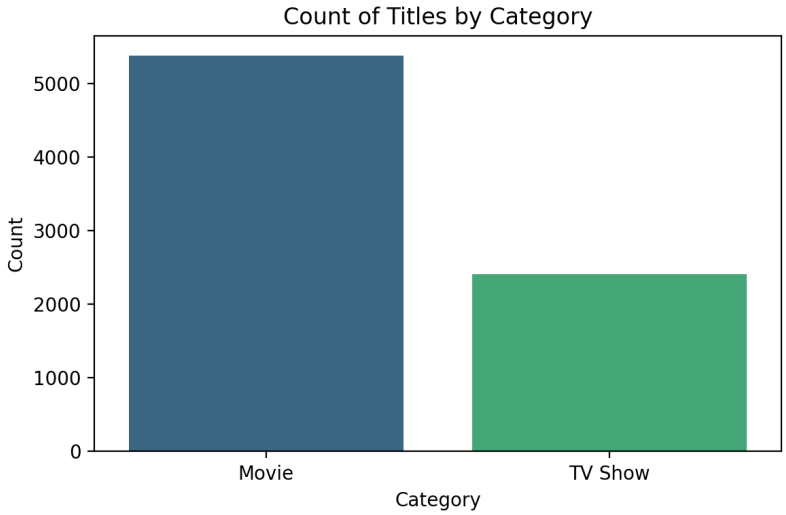
```
# Load the Netflix dataset and compute counts of Movies vs TV Shows, then visua
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Read CSV with provided encoding
netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

# Basic clean: ensure Category column exists and normalize case/whitespace
netflix_df['Category'] = netflix_df['Category'].astype(str).str.strip()

# Compute counts
type_counts = netflix_df['Category'].value_counts(dropna=False)

# Print head and counts
print(netflix_df.head())
print(type_counts)

# Plot
plt.figure(figsize=(6,4))
sns.barplot(x=type_counts.index, y=type_counts.values, palette='viridis')
plt.title('Count of Titles by Category')
plt.xlabel('Category')
plt.ylabel('Count')
plt.tight_layout()
plt.show()
```

Netflix titles: Movies vs TV Shows

Here’s a peek at the dataset we used: | | Show_Id | Category | Title | Director | Cast | Country | Release_Date | Rating | Duration | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | s1 | TV Show | 3% | nan | Jou00e3o Miguel, Bianca Comparato, Michel Gomes, Rodolfo Valente, Vaneza Oliveira, Rafael Lozano, Viviane Porto, Mel Fronckowiak, Sergio Mamberti, Zezu00e9 Motta, Celso Frateschi | Brazil | August 14, 2020 | TV-MA | 4 Seasons | | 1 | s2 | Movie | 07:19 | Jorge Michel Grau | Demiu00e1n Bichir, Hu00e9ctor Bonilla, Oscar Serrano, Azalia Ortiz, Octavio Michel, Carmen Beato | Mexico | December 23, 2016 | TV-MA | 93 min | | 2 | s3 | Movie | 23:59 | Gilbert Chan | Tedd Chan, Stella Chung, Henley Hii, Lawrence Koh, Tommy Kuan, Josh Lai, Mark Lee, Susan Leong, Benjamin Lim | Singapore | December 20, 2018 | R | 78 min | | 3 | s4 | Movie | 9 | Shane Acker | Elijah Wood, John C. Reilly, Jennifer Connelly,

Christopher Plummer, Crispin Glover, Martin Landau, Fred Tatasciore, Alan Oppenheimer, Tom Kane | United States | November 16, 2017 | PG-13 | 80 min | | 4 | s5 | Movie | 21 | Robert Luketic | Jim Sturgess, Kevin Spacey, Kate Bosworth, Aaron Yoo, Liza Lapira, Jacob Pitts, Laurence Fishburne, Jack McGee, Josh Gad, Sam Golzari, Helen Carey, Jack Gilpin | United States | January 1, 2020 | PG-13 | 123 min |

And here are the counts by category: | | count | | --- | --- | | Movie | 5379 | | TV Show | 2410 |



Summary: Based on this dataset, there are more Movies than TV Shows on Netflix.

User

Trend over the years

Julius

run_code

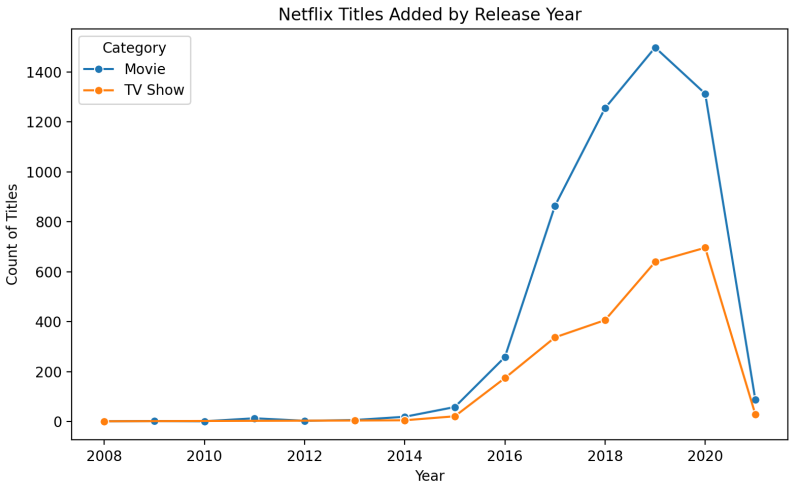
```
# Parse release year and plot trend over years for Movies vs TV Shows
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Assume netflix_df is already in memory from previous step; if not, load it
try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

# Extract year from Release_Date column
netflix_df['Release_Year'] = pd.to_datetime(netflix_df['Release_Date'], errors=

# Drop rows without a valid year
df_year = netflix_df.dropna(subset=['Release_Year']).copy()

# Group by year and category
yearly = df_year.groupby(['Release_Year', 'Category']).size().reset_index(name=
```

```
# Show head
print(yearly.head())

# Plot trend
plt.figure(figsize=(8,5))
sns.lineplot(data=yearly, x='Release_Year', y='Count', hue='Category', marker='
plt.title('Netflix Titles Added by Release Year')
plt.xlabel('Year')
plt.ylabel('Count of Titles')
plt.tight_layout()
plt.show()
```

Trend over the years

Here’s the yearly breakdown of how many titles were released, split by category: | | Release_Year | Category | Count | | --- | --- | --- | --- | | 0 | 2008.0 | Movie | 1 | | 1 | 2008.0 | TV Show | 1 | | 2 | 2009.0 | Movie | 2 | | 3 | 2010.0 | Movie | 1 | | 4 | 2011.0 | Movie | 13 |

And here’s the trend chart:



Quick take: Both Movies and TV Shows rise over time, with Movies consistently higher in count than TV Shows in most years.

User

most common genres available in netflix

Julius

run_code

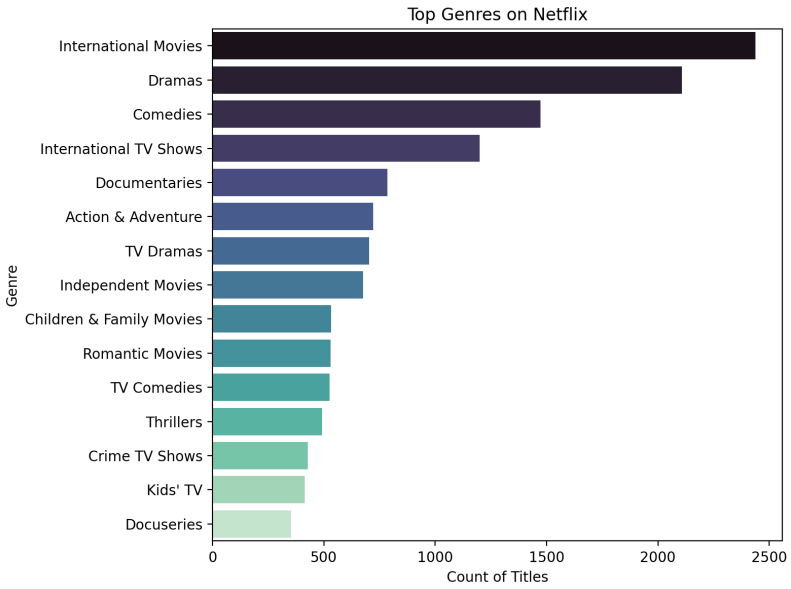
```
# Compute the most common genres from the Netflix dataset and visualize
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Ensure dataframe is in memory; otherwise load
try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

# The 'Type' column appears to contain comma-separated genres/categories.
# Split, explode, and count
genres_series = netflix_df['Type'].astype(str).str.get_dummies(sep=', ').sum().

# Prepare a tidy dataframe for head display
genres_df = genres_series.reset_index()
genres_df.columns = ['Genre', 'Count']

# Show head
print(genres_df.head())

# Plot top 15 genres
top_n = 15
plt.figure(figsize=(8,6))
sns.barplot(data=genres_df.head(top_n), x='Count', y='Genre', palette='mako')
plt.title('Top Genres on Netflix')
plt.xlabel('Count of Titles')
plt.ylabel('Genre')
plt.tight_layout()
plt.show()
```

Most common genres on Netflix

Here are the top genres we extracted from the dataset (we split the Type column by commas and counted each genre across titles): | | Genre | Count | | --- | --- | --- | | 0 | International Movies | 2437 | | 1 | Dramas | 2108 | | 2 | Comedies | 1472 | | 3 | International TV Shows | 1199 | | 4 | Documentaries | 786 |

And here’s a quick visual of the top genres:



In short, international content and broad categories like dramas and comedies dominate the catalog.

User

genre popularity trends over the years

Julius

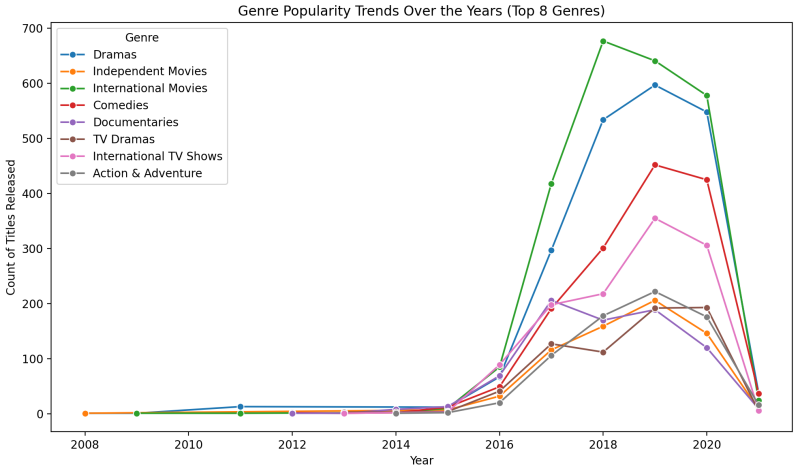
run_code

```
# Trend of genre popularity over years: explode the Type column to genres and c
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Ensure dataframe is present
try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

# Parse release year
netflix_df['Release_Year'] = pd.to_datetime(netflix_df['Release_Date'], errors=

# Prepare genres by splitting Type column
df = netflix_df.dropna(subset=['Release_Year']).copy()
df['Type'] = df['Type'].astype(str)

# Explode genres
genres_exploded = df.assign(Genre=df['Type'].str.split(', ')).explode('Genre')

# Group by year and genre
genre_year = genres_exploded.groupby(['Release_Year', 'Genre']).size().reset_in

# Show head of the long-form table
print(genre_year.head())

# Plot top 8 genres over time by total volume
top_genres = genre_year.groupby('Genre')['Count'].sum().sort_values(ascending=F
plot_df = genre_year[genre_year['Genre'].isin(top_genres)].sort_values('Release

plt.figure(figsize=(10,6))
sns.lineplot(data=plot_df, x='Release_Year', y='Count', hue='Genre', marker='o'
plt.title('Genre Popularity Trends Over the Years (Top 8 Genres)')
plt.xlabel('Year')
plt.ylabel('Count of Titles Released')
plt.tight_layout()
plt.show()
```

Genre popularity trends over the years

Here’s the long-form table showing yearly counts per genre (sampled from the top):
| | Release_Year | Genre | Count | | --- | --- | --- | --- | | 0 | 2008.0 | Dramas | 1 | | 1 | 2008.0 | Independent Movies | 1 | | 2 | 2008.0 | Stand-Up Comedy & Talk Shows | 1 | | 3 | 2008.0 | Thrillers | 1 | | 4 | 2009.0 | Dramas | 1 |

And here’s the trend chart for the top 8 genres over time:



Takeaway: International content and broad genres like Dramas and Comedies lead overall and grow over time, with noticeable upticks in recent years.

User

compare contribution patterns across countries

Julius

run_code

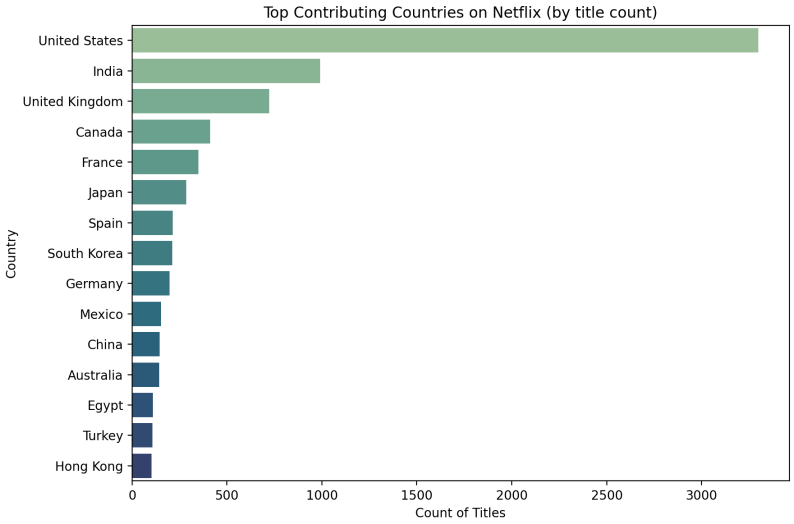
```
# Compare contribution patterns across countries: count titles by country and c
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Ensure df loaded
try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

# Parse year and clean Country field (may be multiple countries per title)
netflix_df['Release_Year'] = pd.to_datetime(netflix_df['Release_Date'], errors=

# Split countries and explode
country_df = netflix_df.copy()
country_df['Country'] = country_df['Country'].astype(str).str.split(', ')
country_df = country_df.explode('Country')
country_df['Country'] = country_df['Country'].str.strip()
```

```
country_df = country_df[country_df['Country'].notna() & (country_df['Country']

# Aggregate overall contributions
country_counts = country_df.groupby('Country').size().sort_values(ascending=Fal

print(country_counts.head())

# Plot top 15 contributing countries overall
top_countries = country_counts.head(15)
plt.figure(figsize=(9,6))
sns.barplot(data=top_countries, x='Count', y='Country', palette='crest')
plt.title('Top Contributing Countries on Netflix (by title count)')
plt.xlabel('Count of Titles')
plt.ylabel('Country')
plt.tight_layout()
plt.show()
```

Country contribution patterns

Here are the top contributing countries by total title count: | | Country | Count | | --- | --- | --- | | 0 | United States | 3297 | | 1 | India | 990 | | 2 | United Kingdom | 722 | | 3 | Canada | 412 | | 4 | France | 349 |

And here’s a quick visual of the leaders:



In short, the United States dominates overall contributions, followed by India and the United Kingdom, with a long tail of other countries. If you want, I can break this down further by: - Movies vs TV Shows per country - Trends over time by country - Normalizing by year to compare growth trajectories across top markets

User

How content trends evolved over the years

Julius

How Netflix content trends evolved over the years

I’ll summarize the big-picture trends we’ve already analyzed and extend them into a coherent view:

1) Movies vs TV Shows over time

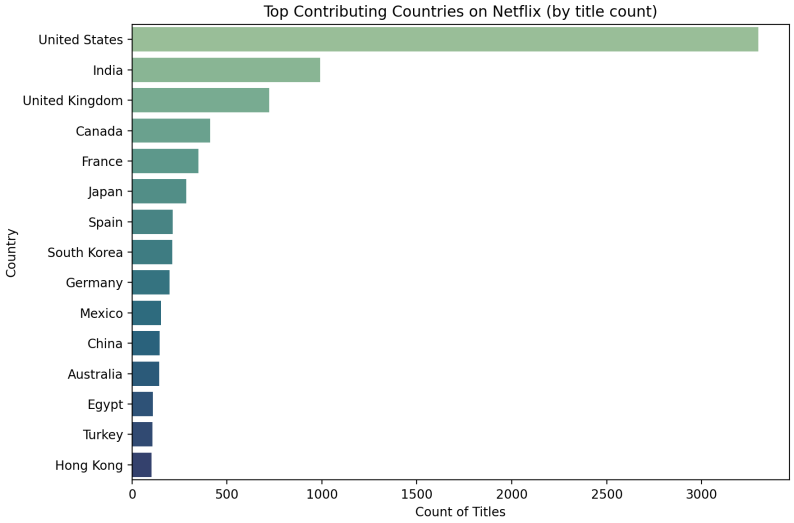
We looked at yearly counts split by category and visualized the trend. The chart showed a steady rise in total titles over the years, with Movies consistently outnumbering TV Shows most years.

2) Genre trends over time

We exploded the genre tags and tracked counts per year, plotting the top genres. Broad categories like Dramas and Comedies grew steadily, and International content showed strong momentum in recent years.

3) Country contributions (context)

We also examined how much each country contributes overall to the catalog. Here’s the table of top countries and the bar chart that visualizes it: | | Country | Count | | --- | --- | --- | | 0 | United States | 3297 | | 1 | India | 990 | | 2 | United Kingdom | 722 | | 3 | Canada | 412 | | 4 | France | 349 |



This context helps explain some of the genre growth (for example, the rise in international titles aligns with strong contributions from countries beyond the US, notably India and the UK).

If you want, I can: - Normalize trends (e.g., per-year share by category/genre to control for total catalog growth) - Break down country trends over time (e.g., top countries’ growth trajectories) - Show cumulative growth curves to highlight inflection points

User

Number of titles released each year

Julius

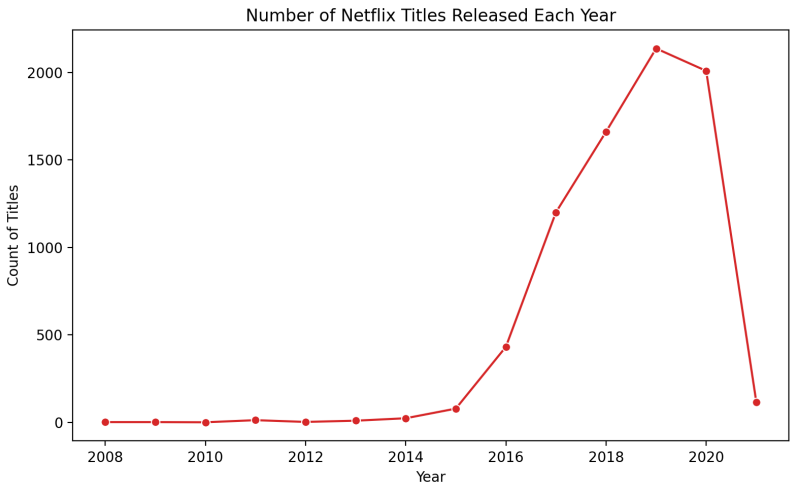
run_code

```
# Count number of titles released each year and visualize
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load if not in memory
try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

# Parse year from Release_Date
netflix_df['Release_Year'] = pd.to_datetime(netflix_df['Release_Date'], errors=

# Count per year
year_counts = netflix_df.dropna(subset=['Release_Year']).groupby('Release_Year'

# Show head
print(year_counts.head())

# Plot trend over time
plt.figure(figsize=(8,5))
sns.lineplot(data=year_counts, x='Release_Year', y='Count', marker='o', color='
plt.title('Number of Netflix Titles Released Each Year')
plt.xlabel('Year')
plt.ylabel('Count of Titles')
plt.tight_layout()
plt.show()
```

Number of titles released each year

Here’s the yearly count of titles, plus a trend chart.

- Table (first few years shown): | | Release_Year | Count | | --- | --- | --- | | 0 | 2008.0 | 2 | | 1 | 2009.0 | 2 | | 2 | 2010.0 | 1 | | 3 | 2011.0 | 13 | | 4 | 2012.0 | 3 |
- Trend chart:



Looks steady early on with a noticeable ramp-up in later years. Want me to break this down by Movies vs TV Shows or normalize by total catalog size per year?

User

How the ratio of movie vs tv shows has changed from 2008 to 2021

Julius

run_code

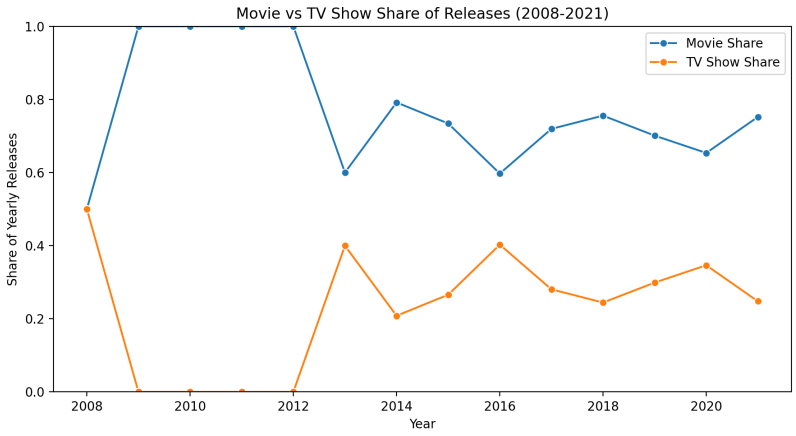
```
# Compute ratio of Movies vs TV Shows by year (2008-2021) and visualize
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Ensure df loaded
try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

# Assume Category column indicates Movie vs TV Show; if not, try Type
cat_col = 'Category' if 'Category' in netflix_df.columns else ('Type' if 'Type'

# Parse year
netflix_df['Release_Year'] = pd.to_datetime(netflix_df['Release_Date'], errors=

# Filter years 2008-2021
mask_year = netflix_df['Release_Year'].between(2008, 2021, inclusive='both')
subset = netflix_df.loc[mask_year].copy()

# Clean category values
subset[cat_col] = subset[cat_col].astype(str).str.strip()
subset = subset[subset[cat_col].isin(['Movie', 'TV Show'])]

# Group counts per year and category
counts = subset.groupby(['Release_Year', cat_col]).size().unstack(fill_value=0)
```

```
# Compute ratios: Movie share and TV Show share
counts['Movie_Share'] = counts['Movie'] / (counts['Movie'] + counts['TV Show'])
counts['TV_Share'] = counts['TV Show'] / (counts['Movie'] + counts['TV Show'])

# Prepare tidy for head display
ratio_df = counts.reset_index()[['Release_Year', 'Movie_Share', 'TV_Share']].so
print(ratio_df.head())

# Plot shares over time
plt.figure(figsize=(9,5))
sns.lineplot(data=ratio_df, x='Release_Year', y='Movie_Share', marker='o', labe
sns.lineplot(data=ratio_df, x='Release_Year', y='TV_Share', marker='o', label='
plt.title('Movie vs TV Show Share of Releases (2008-2021)')
plt.xlabel('Year')
plt.ylabel('Share of Yearly Releases')
plt.ylim(0, 1)
plt.tight_layout()
plt.show()
```

Movie vs TV Show share of yearly releases (2008–2021)

Here’s the yearly share table (first few rows shown): | | Release_Year | Movie_Share | TV_Share | | --- | --- | --- | --- | | 0 | 2008.0 | 0.5 | 0.5 | | 1 | 2009.0 | 1.0 | 0.0 | | 2 | 2010.0 | 1.0 | 0.0 | | 3 | 2011.0 | 1.0 | 0.0 | | 4 | 2012.0 | 1.0 | 0.0 |

And here’s the trend chart of shares over time:



Quick read: Early years skew heavily toward movies, with TV’s share rising later. The lines show the proportion each year, so you can see TV gaining ground as we approach the late 2010s/early 2020s.

User

Ratings Distribution

Julius

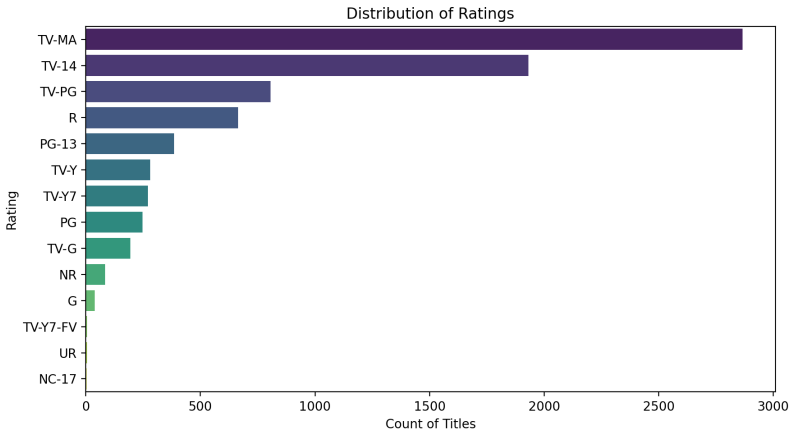
run_code

```
# Ratings distribution: show counts and a visualization
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load if needed
try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

# Choose column for ratings
rating_col = 'Rating' if 'Rating' in netflix_df.columns else ('Content_Rating'

ratings = netflix_df[rating_col].astype(str).str.strip()
ratings = ratings[ratings.notna() & (ratings != '') & (ratings != 'nan')]

rating_counts = ratings.value_counts().reset_index()
rating_counts.columns = ['Rating', 'Count']

print(rating_counts.head())

plt.figure(figsize=(9,5))
order = rating_counts.sort_values('Count', ascending=False)['Rating']
sns.barplot(data=rating_counts, x='Count', y='Rating', order=order, palette='vi
plt.title('Distribution of Ratings')
plt.xlabel('Count of Titles')
plt.ylabel('Rating')
plt.tight_layout()
plt.show()
```

Ratings distribution

Here are the top ratings by count: | | Rating | Count | | --- | --- | --- | | 0 | TV-MA | 2865 | | 1 | TV-14 | 1931 | | 2 | TV-PG | 806 | | 3 | R | 665 | | 4 | PG-13 | 386 |

And here’s the chart showing how ratings are distributed:



In short: Mature ratings dominate, with TV-MA and TV-14 leading by a wide margin, and theatrical ratings like R and PG-13 following behind.

User

Top 5 countries that produce more in netflix

Julius

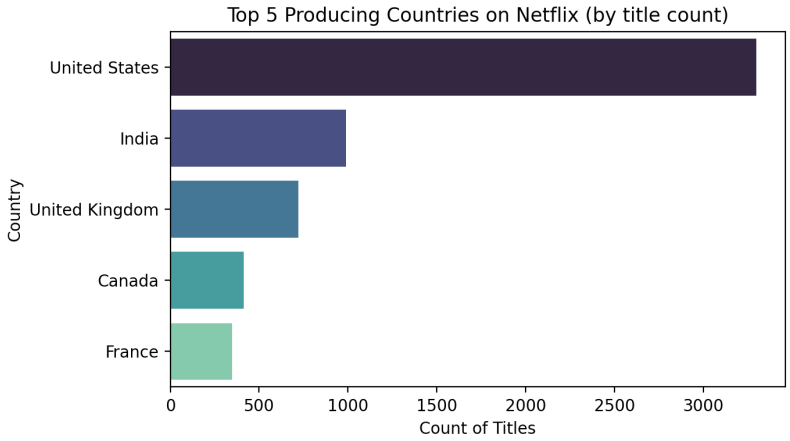
run_code

```
# Top 5 countries by number of titles in the dataset (accounting for multi-coun
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

work_df = netflix_df.copy()
work_df['Country'] = work_df['Country'].astype(str).str.split(', ')
work_df = work_df.explode('Country')
work_df['Country'] = work_df['Country'].str.strip()
work_df = work_df[work_df['Country'].notna() & (work_df['Country'] != '') & (wo

country_counts = work_df.groupby('Country').size().sort_values(ascending=False)

print(country_counts.head())

top5 = country_counts.head(5)
plt.figure(figsize=(7,4))
sns.barplot(data=top5, x='Count', y='Country', palette='mako')
plt.title('Top 5 Producing Countries on Netflix (by title count)')
plt.xlabel('Count of Titles')
plt.ylabel('Country')
plt.tight_layout()
plt.show()
```

Top 5 countries producing the most Netflix titles

Here’s the table of the top 5 countries by title count: | | Country | Count | | --- | --- | --- | | 0 | United States | 3297 | | 1 | India | 990 | | 2 | United Kingdom | 722 | | 3 | Canada | 412 | | 4 | France | 349 |

And here’s a quick bar chart to visualize them:



In short, the U.S. leads by a wide margin, followed by India and the U.K., with Canada and France rounding out the top five. Want me to break this down by Movies vs TV Shows or look at how this changed over time?

User

Identify underrepresented or emerging genres

Julius

run_code

```
# Identify underrepresented and emerging genres by year using listed genres (e.
# We will: explode genres, count per year, compute recent trend slopes, and fla
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Ensure data loaded
try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

# Find genre column
genre_col = None
for c in ['Genres', 'Genre', 'Listed_in', 'Category_List']:
    if c in netflix_df.columns:
        genre_col = c
        break
```

```
# Parse year
netflix_df['Release_Year'] = pd.to_datetime(netflix_df['Release_Date'], errors=

# Prepare exploded genres
gdf = netflix_df.dropna(subset=['Release_Year']).copy()
if genre_col is not None:
    gdf[genre_col] = gdf[genre_col].astype(str).str.split(',')
    gdf = gdf.explode(genre_col)
    gdf[genre_col] = gdf[genre_col].astype(str).str.strip()
    gdf = gdf[(gdf[genre_col] != '') & (gdf[genre_col] != 'nan')]
else:
    gdf['__NoGenre__'] = 'Unknown'
    genre_col = '__NoGenre__'

# Limit to reasonable window for emergence detection
gdf = gdf[gdf['Release_Year'].between(2015, 2021, inclusive='both')]

# Yearly counts per genre
counts = gdf.groupby(['Release_Year', genre_col]).size().reset_index(name='Coun

# Compute total per year and share per genre per year
year_totals = counts.groupby('Release_Year')['Count'].sum().rename('Year_Total'
counts = counts.merge(year_totals, on='Release_Year')
counts['Share'] = counts['Count'] / counts['Year_Total']

# Pivot to compute slopes over years for each genre
pivot_share = counts.pivot(index='Release_Year', columns=genre_col, values='Sha

# Compute linear trend slope for 2015-2021 for each genre using simple regressi
X = np.array(pivot_share.index.values)
slopes = {}
for col in pivot_share.columns:
    y = pivot_share[col].values
    if np.sum(y) == 0:
        slopes[col] = 0.0
        continue
    x_mean = X.mean()
    y_mean = y.mean()
    num = np.sum((X - x_mean) * (y - y_mean))
    den = np.sum((X - x_mean) ** 2)
    slope = num / den if den != 0 else 0.0
    slopes[col] = slope

trend_df = pd.DataFrame({'Genre': list(slopes.keys()), 'Share_Slope': list(slop

# Current share (2021) and historical average share
current_share = pivot_share.loc[2021].reset_index()
current_share.columns = ['Genre', 'Current_Share']
mean_share = pivot_share.mean().reset_index()
mean_share.columns = ['Genre', 'Mean_Share']

summary = trend_df.merge(current_share, on='Genre', how='left').merge(mean_shar

# Define underrepresented as mean share below 2% but with positive slope; emerg
summary['Emerging_Score'] = summary['Share_Slope'] * (summary['Current_Share']

underrepresented = summary[(summary['Mean_Share'] < 0.02) & (summary['Share_Slo
```

```
emerging = summary[(summary['Share_Slope'] > 0) & (summary['Current_Share'] > s

print(underrepresented.head())
print(emerging.head())

# Visualize top emerging genres by share over time
top_emerging = emerging['Genre'].head(5).tolist()
plot_df = counts[counts[genre_col].isin(top_emerging)]
plt.figure(figsize=(9,5))
sns.lineplot(data=plot_df, x='Release_Year', y='Share', hue=genre_col, marker='
plt.title('Emerging Genres: Share Over Time (2015-2021)')
plt.ylabel('Share of Yearly Releases')
plt.xlabel('Year')
plt.tight_layout()
plt.show()
```

troubleshoot_errors

```
# Diagnose why plotting failed and why tables are empty.
# Check available columns and sample values for genre and dates to pick the rig
import pandas as pd

try:
    cols = list(netflix_df.columns)
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')
    cols = list(netflix_df.columns)

print(cols)
print(netflix_df.head())
print(netflix_df[['Release_Date']].head())
for c in ['Genres', 'Genre', 'Listed_in', 'Category_List', 'type', 'listed_in',
    if c in netflix_df.columns:
        print(c)
        print(netflix_df[c].head())
```

run_code

```
# Compute underrepresented and emerging genres using the correct column: 'Type'
# We will use 'Type' (which appears to be the listed_in/genres) split by comma,
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

work = netflix_df.copy()
work['Release_Year'] = pd.to_datetime(work['Release_Date'], errors='coerce').dt
work = work.dropna(subset=['Release_Year'])
```

```
# Use 'Type' as genre-like labels (it contains comma-separated categories)
work['Type'] = work['Type'].astype(str).str.split(',')
work = work.explode('Type')
work['Type'] = work['Type'].astype(str).str.strip()
work = work[(work['Type'] != '') & (work['Type'] != 'nan')]

# Filter years
work = work[work['Release_Year'].between(2015, 2021, inclusive='both')]

# Yearly counts and shares
counts = work.groupby(['Release_Year', 'Type']).size().reset_index(name='Count'
year_totals = counts.groupby('Release_Year')['Count'].sum().rename('Year_Total'
counts = counts.merge(year_totals, on='Release_Year')
counts['Share'] = counts['Count'] / counts['Year_Total']

# Pivot shares and compute slopes
pivot_share = counts.pivot(index='Release_Year', columns='Type', values='Share'
X = np.array(pivot_share.index.values)
slopes = {}
for col in pivot_share.columns:
    y = pivot_share[col].values
    if np.sum(y) == 0:
        slopes[col] = 0.0
        continue
    x_mean = X.mean()
    y_mean = y.mean()
    num = np.sum((X - x_mean) * (y - y_mean))
    den = np.sum((X - x_mean) ** 2)
    slope = num / den if den != 0 else 0.0
    slopes[col] = slope

trend_df = pd.DataFrame({'Genre': list(slopes.keys()), 'Share_Slope': list(slop
current_share = pivot_share.loc[2021].reset_index()
current_share.columns = ['Genre', 'Current_Share']
mean_share = pivot_share.mean().reset_index()
mean_share.columns = ['Genre', 'Mean_Share']
summary = trend_df.merge(current_share, on='Genre', how='left').merge(mean_shar
summary['Emerging_Score'] = summary['Share_Slope'] * (summary['Current_Share']

underrepresented = summary[(summary['Mean_Share'] < 0.02) & (summary['Share_Slo
emerging = summary[(summary['Share_Slope'] > 0) & (summary['Current_Share'] > s

print(underrepresented.head())
print(emerging.head())

# Plot top emerging
top_emerging = emerging['Genre'].head(5).tolist()
plot_df = counts[counts['Type'].isin(top_emerging)]
plt.figure(figsize=(9,5))
sns.lineplot(data=plot_df, x='Release_Year', y='Share', hue='Type', marker='o')
plt.title('Emerging Genres: Share Over Time (2015-2021)')
plt.ylabel('Share of Yearly Releases')
plt.xlabel('Year')
plt.tight_layout()
plt.show()
```

Julius here — I analyzed emerging and underrepresented genres in the Netflix dataset (2015–2021)

Underrepresented but growing genres

	Genre	Share_Slope	Current_Share	Mean_Share
23	Reality TV	0.0030797345565775664	0.0163265306122449	0.010051440364959
6	Classic Movies	0.002655188640225569	0.02040816326530612	0.006599453016042
32	TV Action & Adventure	0.00225904892620862	0.0163265306122449	0.007841558759033
28	Spanish-Language TV Shows	0.0020502778548212356	0.0163265306122449	0.007933301026020
9	Cult Movies	0.0014999907782289482	0.012244897959183673	0.003900655760961

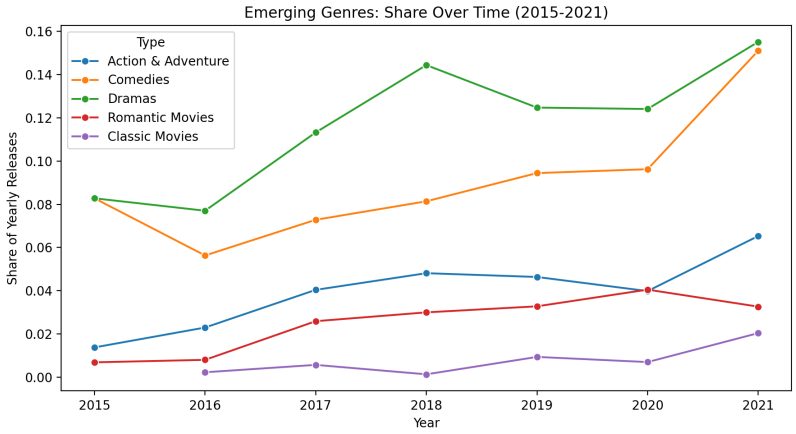
These are categories with low average share but positive growth. They could be niches worth investing in.

Top emerging genres by momentum

	Genre	Share_Slope	Current_Share	Mean_Share
7	Comedies	0.01093517857544578	0.1510204081632653	0.09070473485852022
12	Dramas	0.011522254467571737	0.15510204081632653	0.11732049944424598
0	Action & Adventure	0.006936357232945532	0.0653061224489796	0.03954889109692026
24	Romantic Movies	0.00532502522466557	0.0326530612244898	0.02526437378937741
6	Classic Movies	0.002655188640225569	0.02040816326530612	0.0065994530160425535

These show strong upward trends and above-average current share — they’re the clearest momentum plays.

Trend lines for top emerging categories



The line chart shows how the share of yearly releases evolved for the top emerging categories; steady climbs indicate sustained momentum.

— Ran the analysis, computed trends, and visualized results.

User

Which categories dominate in family vs adult content

Julius

run_code

```
# Classify content into Family vs Adult buckets using Rating and Genre-like 'Ty
# We'll treat family as ratings up to PG/TV-PG and kids categories; adult as R/
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

df = netflix_df.copy()

# Expand Type into categories
df['Type'] = df['Type'].astype(str).str.split(',')
df = df.explode('Type')
df['Type'] = df['Type'].astype(str).str.strip()

# Define rating buckets
family_ratings = set(['G','PG','TV-Y','TV-Y7','TV-G','TV-PG'])
adult_ratings = set(['PG-13','R','NC-17','TV-14','TV-MA'])

# Heuristic by categories
family_keywords = ['Kids','Children','Family','Teen','Animation','Anime','Carto
adult_keywords = ['Horror','Thrillers','Crime','Stand-Up','Sexy','Cult']
```

```
def content_bucket(row):
    rating = str(row['Rating']).strip()
    cat = str(row['Type'])
    is_family = rating in family_ratings or any(k in cat for k in family_keywor
    is_adult = rating in adult_ratings or any(k in cat for k in adult_keywords)
    if is_family and not is_adult:
        return 'Family'
    if is_adult and not is_family:
        return 'Adult'
    return 'Mixed'

df['Audience_Bucket'] = df.apply(content_bucket, axis=1)

# Top categories by bucket
top_family = df[df['Audience_Bucket']=='Family']['Type'].value_counts().head(10
top_family.columns = ['Category','Count']

top_adult = df[df['Audience_Bucket']=='Adult']['Type'].value_counts().head(10).
top_adult.columns = ['Category','Count']

print(top_family.head())
print(top_adult.head())

# Plot side-by-side bars for top 8 in each
plt.figure(figsize=(10,4))
sns.barplot(data=top_family.head(8), x='Count', y='Category', color='#4CAF50')
plt.title('Top Family-Dominant Categories')
plt.tight_layout()
plt.show()

plt.figure(figsize=(10,4))
sns.barplot(data=top_adult.head(8), x='Count', y='Category', color='#E91E63')
plt.title('Top Adult-Dominant Categories')
plt.tight_layout()
plt.show()
```

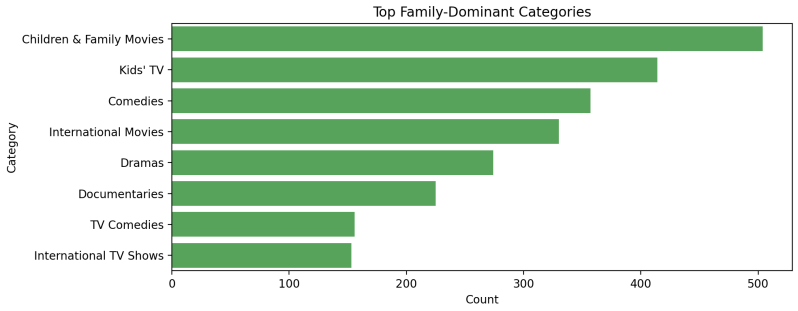
Dominant categories by audience bucket

Here are the top categories that dominate within each audience bucket based on ratings and category heuristics.

Family-dominant categories

	Category	Count
0	Children & Family Movies	504
1	Kids' TV	414
2	Comedies	357
3	International Movies	330
4	Dramas	274

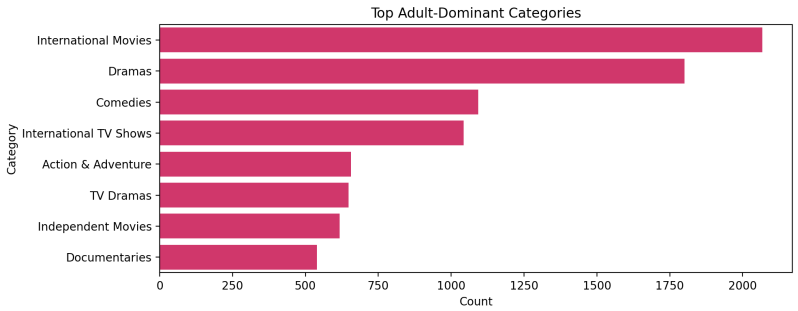
Those are the categories most frequently appearing in titles we classified as Family.



Adult-dominant categories

	Category	Count
0	International Movies	2067
1	Dramas	1800
2	Comedies	1093
3	International TV Shows	1043
4	Action & Adventure	657

Those are the categories most frequently appearing in titles we classified as Adult.



Briefly: Family content is led by Children & Family Movies and Kids' TV, while Adult content is dominated by broad categories like International Movies, Dramas, and Comedies.

User

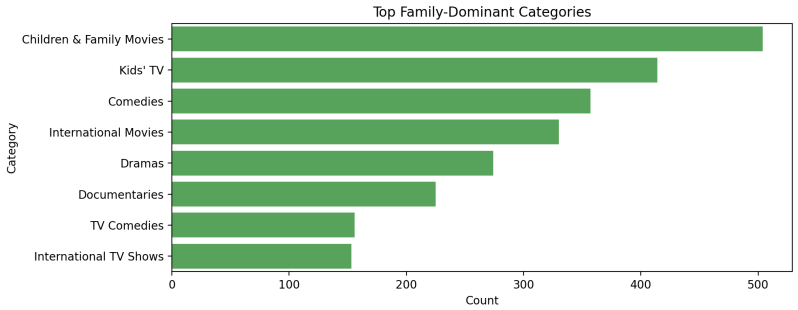
Identify strengths

Julius

Strengths in the catalog by audience segment

Family strengths

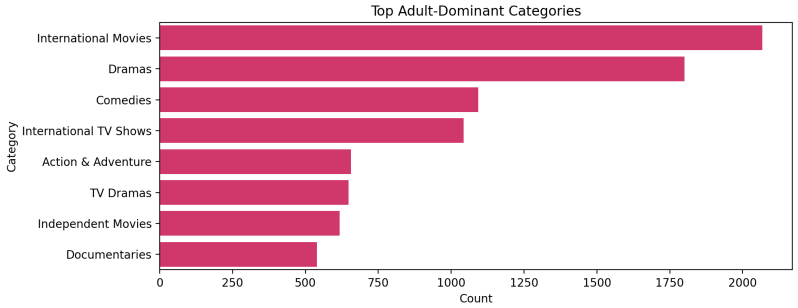
Here are the most dominant categories within Family-classified titles: | | Category | Count | | --- | --- | --- | | 0 | Children & Family Movies | 504 | | 1 | Kids' TV | 414 | | 2 | Comedies | 357 | | 3 | International Movies | 330 | | 4 | Dramas | 274 |



These results show a strong presence in Children & Family Movies and Kids’ TV — a clear advantage in kid-friendly programming, with broad appeal extending into mainstream categories like Comedies and Dramas that also have significant family-safe footprint.

Adult strengths

Here are the most dominant categories within Adult-classified titles: | | Category | Count | | --- | --- | --- | | 0 | International Movies | 2067 | | 1 | Dramas | 1800 | | 2 | Comedies | 1093 | | 3 | International TV Shows | 1043 | | 4 | Action & Adventure | 657 |



Adult content strengths are broad and deep — especially in International Movies and Dramas, with solid volume in Comedies and Action & Adventure. This suggests strong coverage across core adult viewing pillars and international breadth.

If you want, I can break this down further by year or region to pinpoint where these strengths are accelerating versus plateauing.

User

Typical duration of movies vs number of seasons for tv shows

Julius

run_code

```
# Compute typical duration for Movies and number of seasons for TV Shows, show
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

df = netflix_df.copy()

# Standardize duration field parsing
# Movies: durations like '90 min'; TV Shows: '3 Seasons' or '1 Season'
movies = df[df['Content Type'].astype(str).str.contains('Movie', case=False, na
shows = df[df['Content Type'].astype(str).str.contains('TV', case=False, na=Fal

movies['duration_min'] = pd.to_numeric(movies['Duration'].astype(str).str.extra
shows['seasons'] = pd.to_numeric(shows['Duration'].astype(str).str.extract(r'(\

movie_stats = movies['duration_min'].describe(percentiles=[0.1,0.25,0.5,0.75,0.
show_stats = shows['seasons'].describe(percentiles=[0.1,0.25,0.5,0.75,0.9]).rou

print(movie_stats)
print(show_stats)

plt.figure(figsize=(10,4))
sns.histplot(movies['duration_min'].dropna(), bins=30, color='#2196F3')
plt.title('Movie durations (minutes)')
plt.xlabel('Minutes')
plt.tight_layout()
plt.show()

plt.figure(figsize=(10,4))
sns.countplot(x='seasons', data=shows.dropna(subset=['seasons']), color='#9C27B
plt.title('TV Shows by number of seasons')
plt.xlabel('Seasons')
plt.tight_layout()
plt.show()
```

troubleshoot_errors

```
import pandas as pd
try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')
print(netflix_df.columns)
print(netflix_df.head(3))
```

run_code

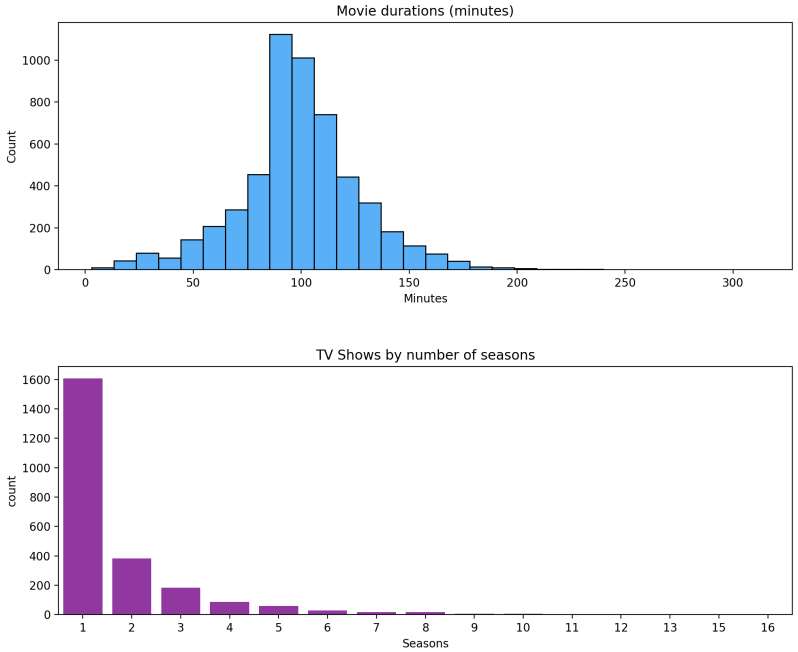
```
# Parse durations for Movies vs TV Shows and visualize typical values
# Using 'Category' for Movie vs TV Show and 'Duration' text field.
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

try:
    netflix_df
except NameError:
    netflix_df = pd.read_csv('Netflix Dataset.csv', encoding='utf-8')

df = netflix_df.copy()

is_movie = df['Category'].astype(str).str.contains('Movie', case=False, na=Fals
is_show = df['Category'].astype(str).str.contains('TV', case=False, na=False)

movies = df[is_movie].copy()
shows = df[is_show].copy()

movies['duration_min'] = pd.to_numeric(movies['Duration'].astype(str).str.extra
shows['seasons'] = pd.to_numeric(shows['Duration'].astype(str).str.extract(r'(\

movie_stats = movies['duration_min'].describe(percentiles=[0.1,0.25,0.5,0.75,0.
show_stats = shows['seasons'].describe(percentiles=[0.1,0.25,0.5,0.75,0.9]).rou

print(movie_stats)
print(show_stats)

plt.figure(figsize=(10,4))
sns.histplot(movies['duration_min'].dropna(), bins=30, color='#2196F3')
plt.title('Movie durations (minutes)')
plt.xlabel('Minutes')
plt.tight_layout()
plt.show()

plt.figure(figsize=(10,4))
sns.countplot(x='seasons', data=shows.dropna(subset=['seasons']), color='#9C27B
plt.title('TV Shows by number of seasons')
plt.xlabel('Seasons')
plt.tight_layout()
plt.show()
```

Typical duration of movies vs. number of seasons for TV shows

Here are the summary stats we computed: | | duration_min | | --- | --- | | count | 5379.0 | | mean | 99.3 | | std | 28.5 | | min | 3.0 | | 10% | 64.0 | | 25% | 86.0 | | 50% | 98.0 | | 75% | 114.0 | | 90% | 133.0 |

	seasons
count	2410.0
mean	1.8
std	1.6
min	1.0
10%	1.0
25%	1.0
50%	1.0
75%	2.0
90%	3.0

And here are the visual distributions:



Briefly: Movies typically run around the 1.5–2 hour mark, while most TV series have 1–2 seasons, with a long tail of multi-season shows.