Analyzing Netflix Data with Python

Mary McSweeney #106179736

Introduction

This project analyzes a dataset of Netflix titles to uncover patterns in content type, release trends, and genre diversity. The dataset contains 8,807 rows and key variables include the title, type (Movie or TV Show), duration, release year, genres, cast, director, and country.

The following questions guide this analysis: - Are TV shows trending toward longer or shorter seasons? - How does genre diversity differ between movies and TV shows? - How have content types and genres evolved over time?

Data Cleaning and Preparation

```
import pandas as pd
import numpy as np

netflix = pd.read_csv('Netflix.csv')

netflix.info()
netflix.describe()
netflix.isna().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
```

```
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

The Netflix dataset has 8807 entries and 12 columns. The columns are show_id, type, title, director, cast, country, date_added, release_year, rating, duration, listed_in, and description. Release year is the only numeric column while the other columns are strings. The director, cast, and country columns have many missing values and date_added, rating, and duration are missing a few.

```
# Handle missing values
netflix['director'] = netflix['director'].fillna('Unknown')
netflix['cast'] = netflix['cast'].fillna('Unknown')
netflix['country'] = netflix['country'].fillna('Unknown')
netflix['rating'] = netflix['rating'].fillna('Unknown')
netflix = netflix.dropna(subset=['date_added', 'duration'])
print("Missing values after cleaning:")
print(netflix.isna().sum())
```

```
Missing values after cleaning:
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

After replacing the missing values in the director, cast, country, and rating columns with "Unknown" and dropping the few rows missing date_added and duration, there are no missing values left.

```
# Convert columns
netflix['type'] = netflix['type'].astype('category')
netflix['rating'] = netflix['rating'].astype('category')
```

The type and rating columns are categorical so the type has been converted to category.

```
# Add duration integer and number of genres
netflix['duration_int'] = netflix['duration'].str.extract(r'(\d+)').astype(float)
netflix['num_genres'] = netflix['listed_in'].str.count(',') + 1
netflix['date_added'] = netflix['date_added'].str.strip()
netflix['date_added'] = pd.to_datetime(netflix['date_added'], format="%B %d, %Y")
netflix['year_added'] = netflix['date_added'].dt.year
```

Using the duration column, duration_int has the number of seasons for tv shows and the number of minutes for movies as an integer. The num_genres column counts the number of genres the movie or tv show is listed under.

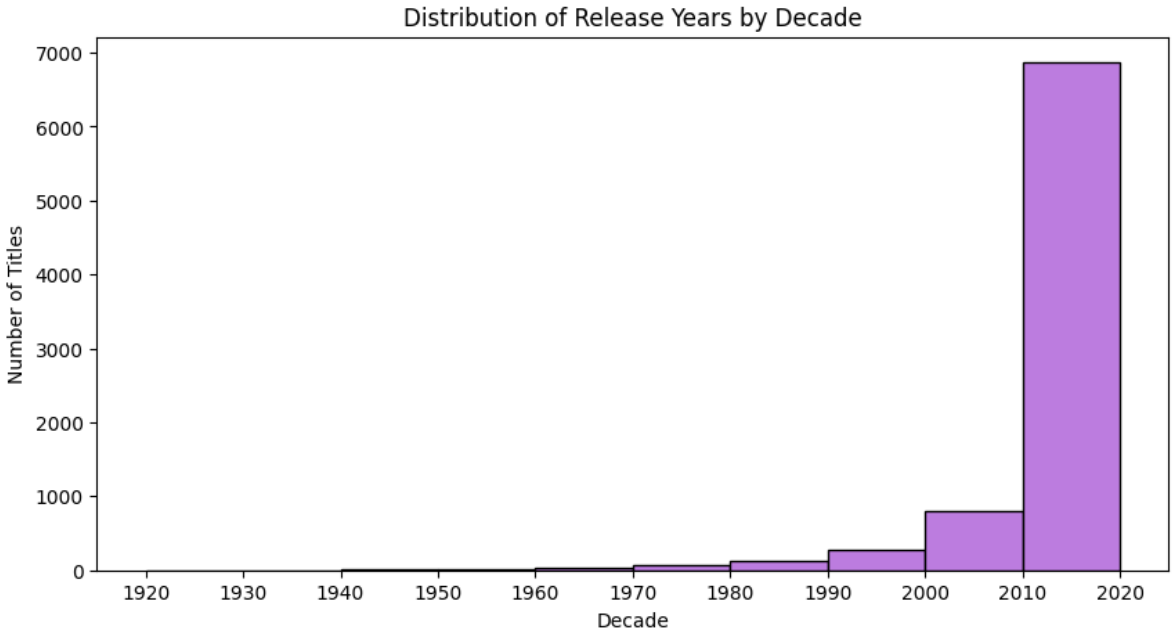
Exploratory Data Analysis

```
# Summary stats for release year
print("Release Year Summary:\n", netflix['release_year'].describe())
# Summary of number of genres per title
print("Number of Genres per Title:\n", netflix['num_genres'].value_counts().sort_index())
```

```
Release Year Summary:
 count    8794.000000
mean     2014.183534
std         8.823527
min      1925.000000
25%      2013.000000
50%      2017.000000
75%      2019.000000
max      2021.000000
Name: release_year, dtype: float64
Number of Genres per Title:
 num_genres
1    2014
2    3054
3    3726
Name: count, dtype: int64
```

The vast majority of titles were released after 2010, with a median of 2017 and a mean of 2014. This indicates that Netflix focuses on newer content. The number of genres listed are between 1 and 3, with most titles having 2 or 3 genres. This suggests that Netflix tends to categorize in multiple genres for more discoverability.

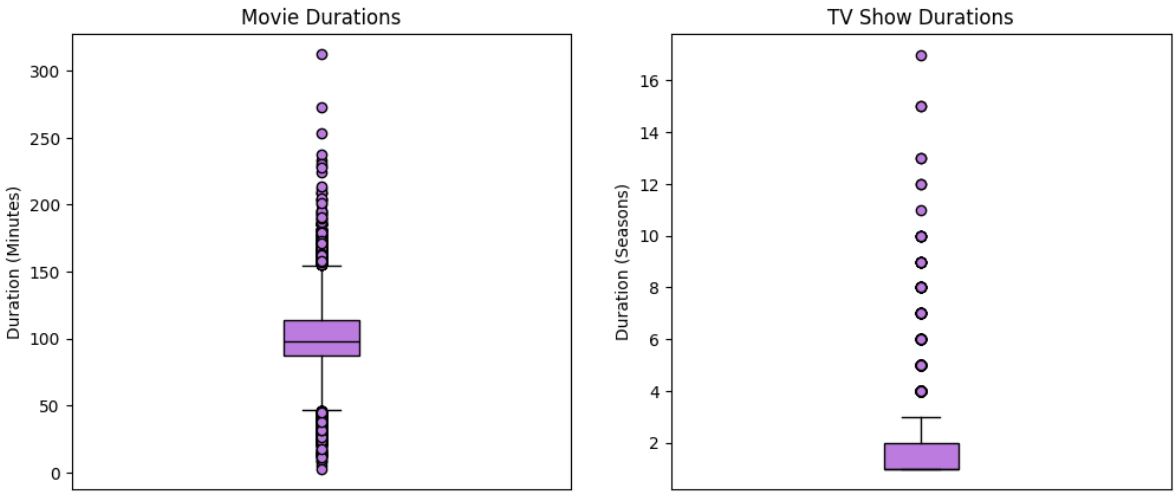
```
# Histogram of release years
import matplotlib.pyplot as plt
light_purple = "#BC7CDF"
plt.figure(figsize=(10, 5))
plt.hist(netflix['release_year'],
         bins=range(1920, 2030, 10),
         edgecolor='black',
         color=light_purple)
plt.title("Distribution of Release Years by Decade")
plt.xlabel("Decade")
plt.ylabel("Number of Titles")
plt.xticks(range(1920, 2030, 10))
plt.show()
```



The histogram above shows the distribution of movies and tv shows by their release year. We can see that Netflix has more movies from recent years, with the vast majority being released after 2010. This suggests that Netflix favors more recent content.

```
# Boxplots of duration
movies = netflix[netflix['type'] == 'Movie']
tv_shows = netflix[netflix['type'] == 'TV Show']
fig, axes = plt.subplots(1, 2, figsize=(12, 5))
axes[0].boxplot(movies['duration_int'], patch_artist=True,
                boxprops={'facecolor': light_purple},
                flierprops={'marker': 'o',
                            'markerfacecolor': light_purple,
                            'markeredgecolor': 'black'},
                medianprops={'color': 'black'})
axes[0].set_title('Movie Durations')
axes[0].set_ylabel('Duration (Minutes)')
axes[0].set_xticks([])
axes[1].boxplot(tv_shows['duration_int'], patch_artist=True,
                boxprops={'facecolor': light_purple},
                flierprops={'marker': 'o',
                            'markerfacecolor': light_purple,
                            'markeredgecolor': 'black'},
                medianprops={'color': 'black'})
axes[1].set_title('TV Show Durations')
```

```
axes[1].set_ylabel('Duration (Seasons)')
axes[1].set_xticks([])
plt.show()
movie_stats = movies['duration_int'].describe()
print("Movie Duration Summary:\n", movie_stats)
tv_stats = tv_shows['duration_int'].describe()
print("\nTV Show Duration Summary:\n", tv_stats)
```

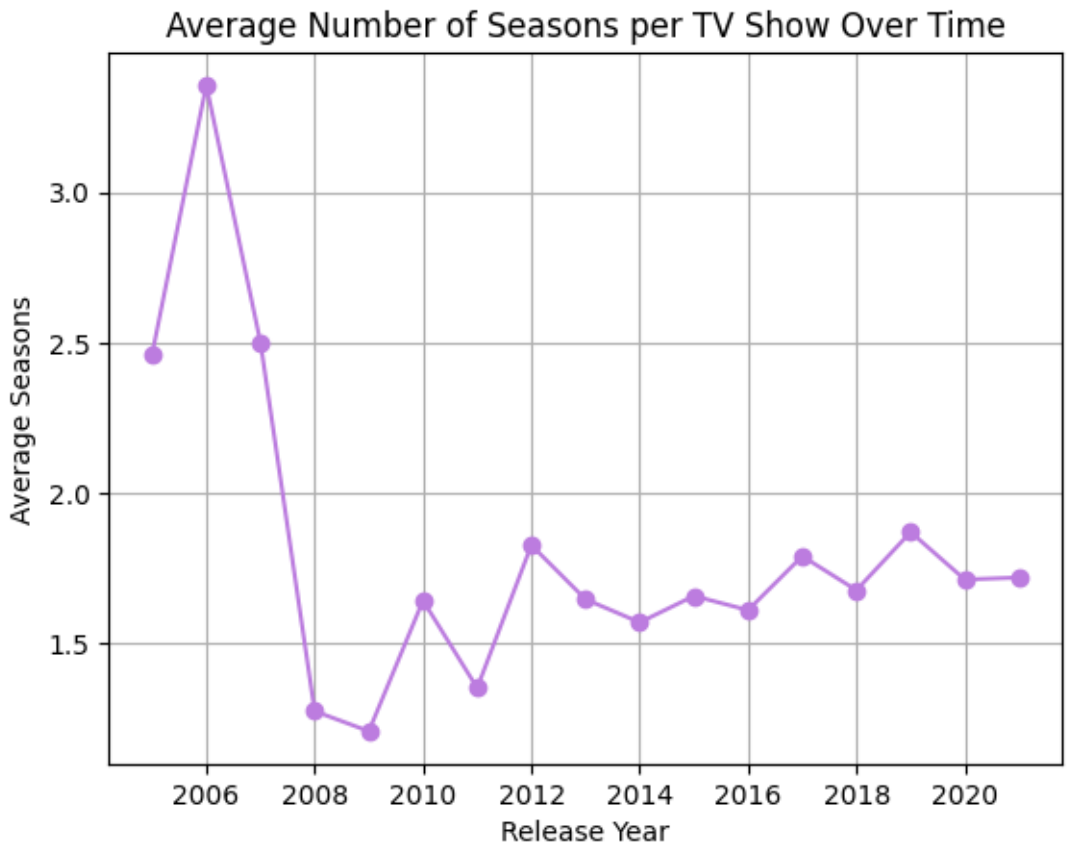



```
Movie Duration Summary:
 count    6128.000000
mean       99.577187
std        28.290593
min         3.000000
25%        87.000000
50%        98.000000
75%       114.000000
max       312.000000
Name: duration_int, dtype: float64

TV Show Duration Summary:
 count    2666.000000
mean        1.751313
std         1.550176
min         1.000000
25%         1.000000
50%         1.000000
75%         2.000000
```

```
max        17.000000
Name: duration_int, dtype: float64
```

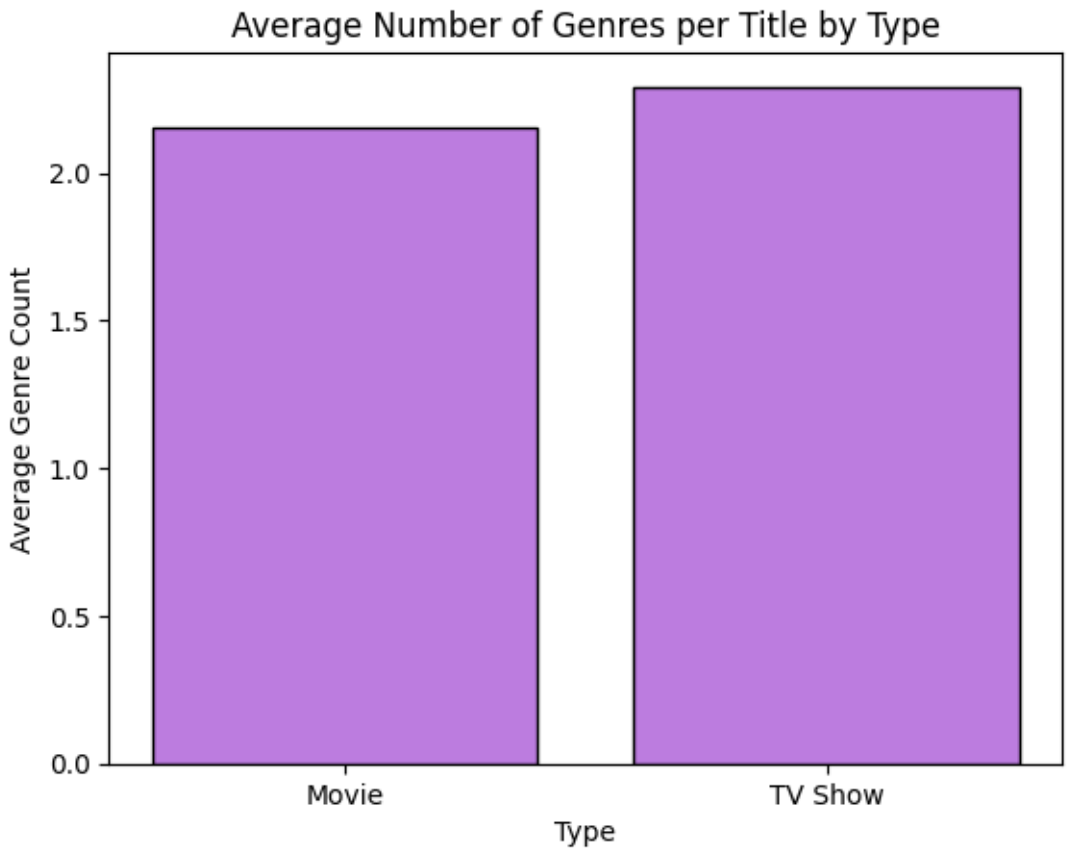
The boxplot and summary statistics show the distribution of duration for movies and tv shows. Movies have a median of 98 minutes and average of 99.6 minutes, while TV shows have a median of 1 season and an average of 1.75 seasons. The boxplots show that movies have a wider range of durations than TV shows. Some movies are as short as 3 minutes and as long as 312 minutes, while TV shows range from 1 to 17 seasons. The majority of movies are between 87 and 114 minutes, while most TV shows have 1 or 2 seasons. This suggests that Netflix movies follow standard feature-length conventions with some variation, the platform strongly favors short-form series when it comes to TV shows.

```python
# Line plot of average number of seasons
year_counts = tv_shows['release_year'].value_counts()
valid_years = year_counts[year_counts >= 10].index
season_trend = tv_shows[tv_shows['release_year'].isin(valid_years)] \
    .groupby('release_year')['duration_int'].mean()

plt.plot(season_trend.index, season_trend.values, marker='o', color=light_purple)
plt.title("Average Number of Seasons per TV Show Over Time")
plt.xlabel("Release Year")
plt.ylabel("Average Seasons")
plt.grid(True)
plt.show()
```



The line plot shows the average number of seasons per TV show from 2005 to 2021, including only years with at least 10 shows to avoid distortion from outliers. There is a peak of around 3.5 seasons in 2006, followed by a sharp decline. From 2009 onward, the trend stabilizes between 1.5 and 2 seasons, suggesting that Netflix tends to favor shorter or limited-run TV series.

```
# Barplot of Average Number of Genres by Type
genre_avg = netflix.groupby('type', observed=True)['num_genres'].mean()
plt.bar(genre_avg.index, genre_avg.values, color=light_purple, edgecolor='black')
plt.title("Average Number of Genres per Title by Type")
plt.ylabel("Average Genre Count")
plt.xlabel("Type")
plt.xticks(rotation=0)
plt.show()
print("Average number of genres listed:")
print(genre_avg)
```



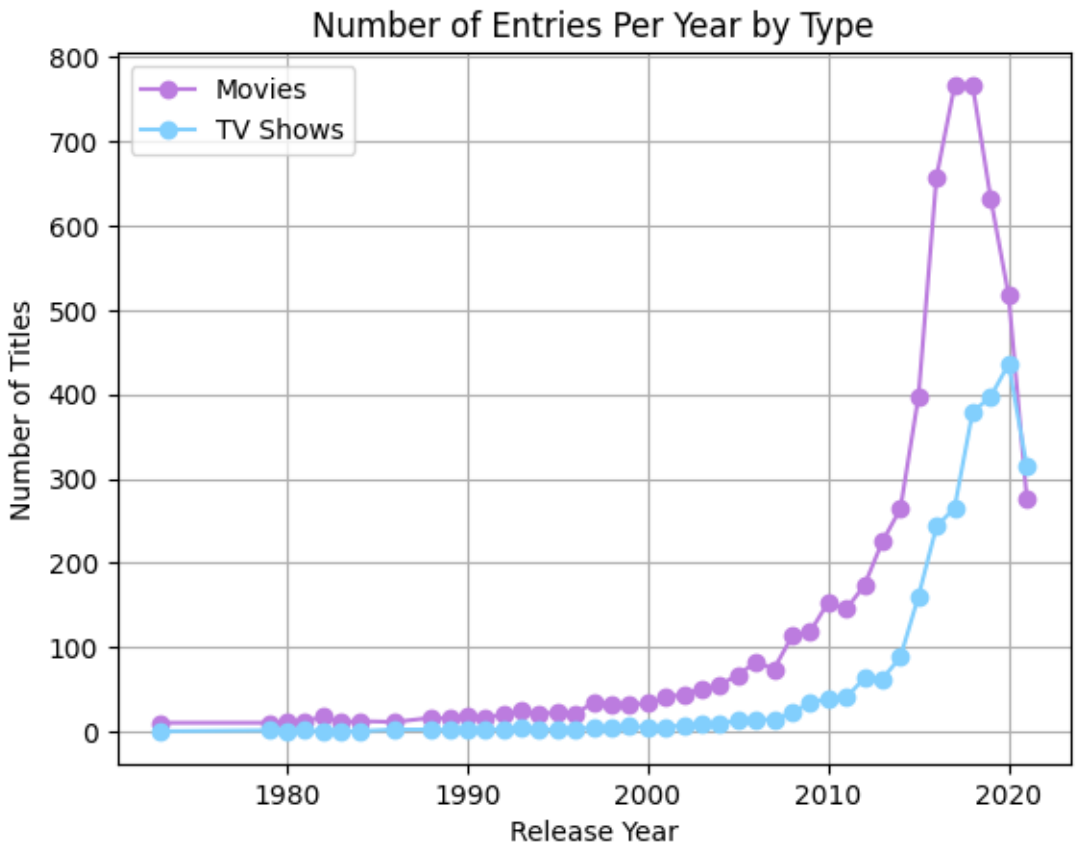

```
Average number of genres listed:
type
Movie      2.151926
TV Show    2.292948
Name: num_genres, dtype: float64
```

This bar plot shows the average number of genre labels per title for movies and TV shows. Both movies and TV shows have an average slightly above 2 genres, with movies averaging 2.15 and TV shows averaging 2.29. This suggests that TV shows may be more likely to be listed under multiple genres.

Further Visualizations

```
# Line plot of number of entries per year by type
year_type_counts = netflix.groupby(['release_year', 'type'],
```

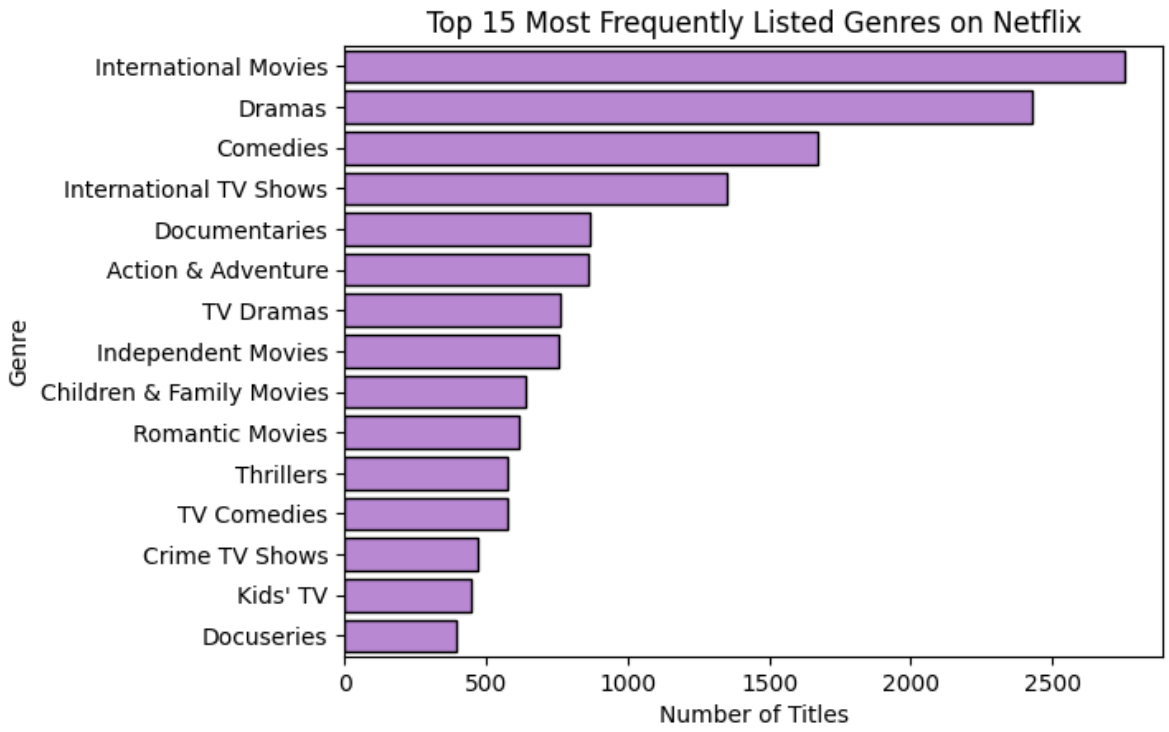
```
                                        observed=True).size().reset_index(name='count')
valid_years = year_type_counts[year_type_counts['count'] >= 10]['release_year'].unique()
filtered_netflix = netflix[netflix['release_year'].isin(valid_years)]
type_trend = filtered_netflix.groupby(['release_year', 'type'],
                                      observed=True).size().unstack(fill_value=0)
light_blue = "#82CFFD"

plt.plot(type_trend.index, type_trend['Movie'],
         label='Movies',
         color=light_purple,
         marker='o')
plt.plot(type_trend.index, type_trend['TV Show'],
         label='TV Shows',
         color=light_blue,
         marker='o')

plt.title("Number of Entries Per Year by Type")
plt.xlabel("Release Year")
plt.ylabel("Number of Titles")
plt.legend()
plt.grid(True)
plt.show()
```



This line plot shows the number of Netflix movies and TV shows released each year, limited to years with at least 10 titles per type. The number of both content types increased dramatically after 2010, with a sharp spike around 2018–2020. Movies have consistently outnumbered TV shows, but they both followed the same trend. This supports the idea that Netflix is greatly skewed towards newer content, with a strong focus on movies.

```
import seaborn as sns
netflix_exploded = netflix.copy()
netflix_exploded['genre_list'] = netflix_exploded['listed_in'].str.split(', ')
netflix_exploded = netflix_exploded.explode('genre_list')
top_genres = netflix_exploded['genre_list'].value_counts().head(15)
sns.barplot(x=top_genres.values,
            y=top_genres.index,
            color=light_purple,
            edgecolor='black')
plt.title("Top 15 Most Frequently Listed Genres on Netflix")
plt.xlabel("Number of Titles")
plt.ylabel("Genre")
```

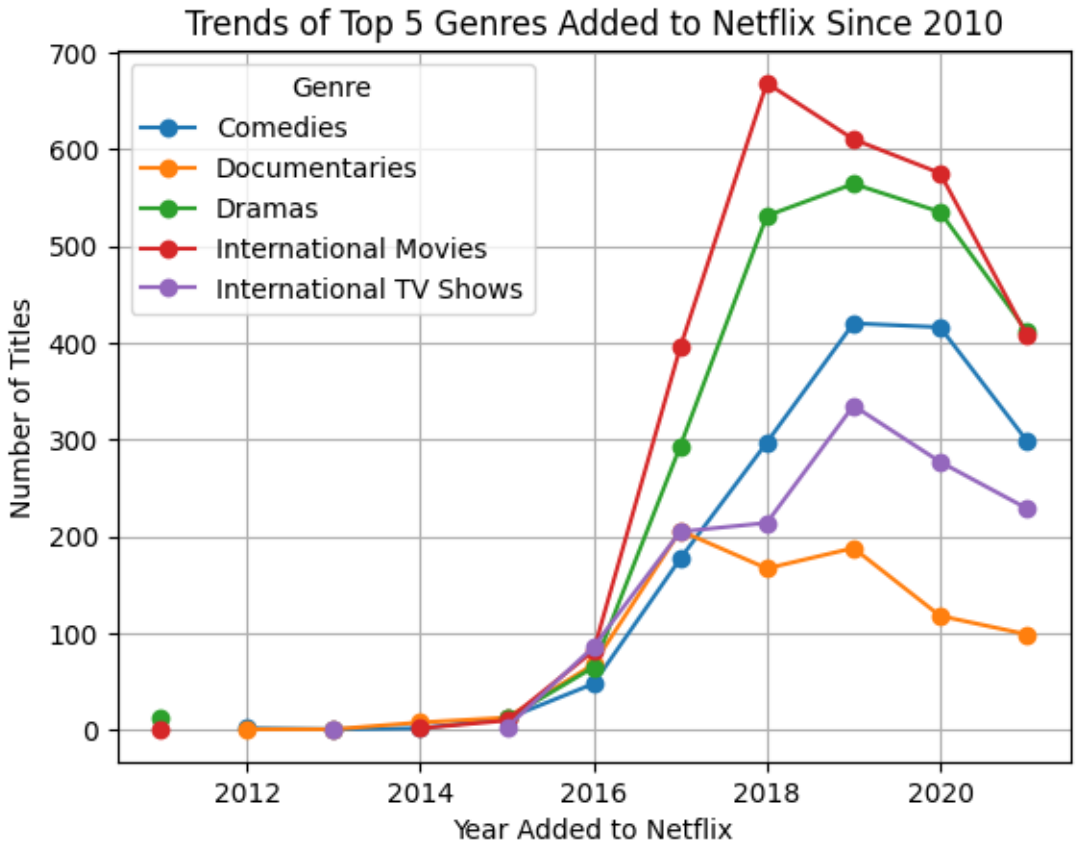
```
plt.show()
```



This horizontal bar chart shows the 15 most listed genres. International Movies, Dramas, and Comedies are the most frequent, with International Movies appearing in nearly 2,800 titles. This highlights Netflix's focus on global content, with a strong emphasis on drama and comedy genres.

```
# Trends of Top 5 Genres Added to Netflix Since 2010
recent = netflix_exploded[netflix_exploded['year_added'] >= 2010]
top_genres = recent['genre_list'].value_counts().nlargest(5).index
recent_top = recent[recent['genre_list'].isin(top_genres)]
genre_year_counts = recent_top.groupby(['year_added',
                                        'genre_list']).size().reset_index(name='count')
pivot_df = genre_year_counts.pivot(index='year_added',
                                   columns='genre_list',
                                   values='count')
for genre in pivot_df.columns:
    plt.plot(pivot_df.index, pivot_df[genre], marker='o', label=genre)
plt.title("Trends of Top 5 Genres Added to Netflix Since 2010")
plt.xlabel("Year Added to Netflix")
```

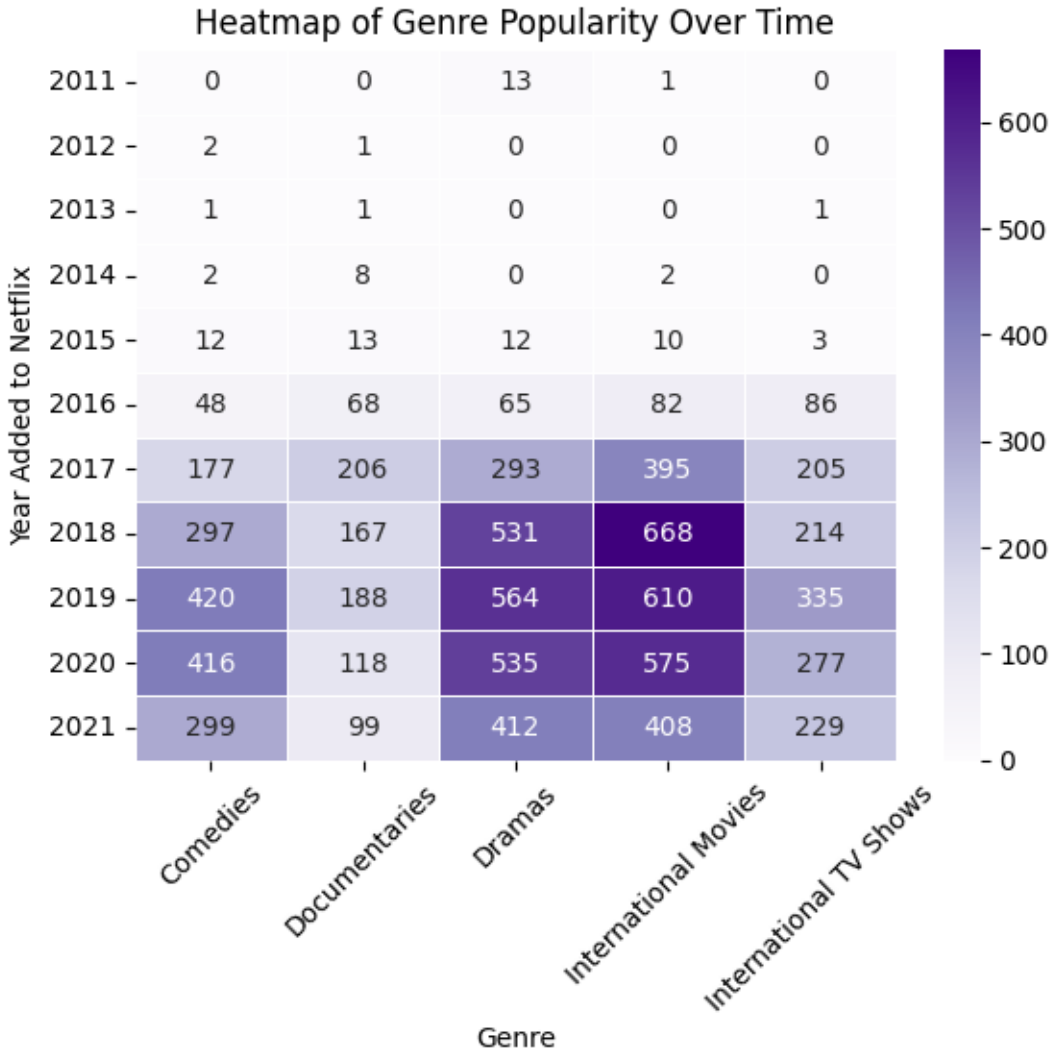
```
plt.ylabel("Number of Titles")
plt.grid(True)
plt.legend(title="Genre")
plt.show()
```



This plot shows how frequently the five most common genres were added to Netflix each year. Beginning around 2016, there was a dramatic increase across all genres, with a peak in around 2018. International Movies and Dramas are added the most frequently, with Comedies and International TV shows following closely behind. This suggests that Netflix is focusing on international content, particularly in the drama and comedy genres.

```
# Heatmap of Genre Popularity Over Time
sns.heatmap(pivot_df, cmap='Purples', annot=True, fmt=".0f", linewidths=.5)
plt.title("Heatmap of Genre Popularity Over Time")
plt.xlabel("Genre")
plt.ylabel("Year Added to Netflix")
plt.xticks(rotation=45)
```

```
plt.yticks(rotation=0)
plt.show()
```



This heatmap shows how often the top 5 genres appeared in titles added to Netflix each year from 2010 to 2021. Darker shades indicate higher counts. Dramas and International Movies consistently have higher counts. The lighter shading in earlier years indicates that genres were more prevalent in recent years. This visualization also shows Netflix's focus on internation content.

Simple Modeling

```
# Logit Model to Predict Movie vs TV Show
import statsmodels.api as sm
netflix['is_movie'] = (netflix['type'] == 'Movie').astype(int)
X = netflix['duration_int']
X = sm.add_constant(X)  # Add constant term for intercept
logit_model = sm.Logit(netflix['is_movie'], X)
result = logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.004912
         Iterations 15
                           Logit Regression Results
==============================================================================
Dep. Variable:               is_movie   No. Observations:                 8794
Model:                          Logit   Df Residuals:                     8792
Method:                           MLE   Df Model:                            1
Date:                Fri, 06 Jun 2025   Pseudo R-squ.:                  0.9920
Time:                        23:29:04   Log-Likelihood:                -43.196
converged:                       True   LL-Null:                       -5395.3
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          -8.5672      0.766    -11.191      0.000     -10.068      -7.067
duration_int    0.6918      0.074      9.375      0.000       0.547       0.836
==============================================================================

Possibly complete quasi-separation: A fraction 0.69 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The goal of this model is the predict whether a title is a movie or TV show based on its duration. The variable is_movie is the outcome with a value of 1 for movies and 0 for TV shows and the predictor is duration_int, the number of seasons for TV shows and the number of minutes for movies.

The coefficient on duration_int is significant and positive so it suggests that higher integers are strongly associated with movies. The model states that each additional unit increase in

duration_int, the log-odds of being a movie increases by about 0.692. The Pseudo R-squared value is 0.992, which is very high, implies that this is an great predictor of content type in this dataset.