netflix-python-eda

October 29, 2025

1 PYTHON Exploratory-Data-Analysis:

[ ]:

```

```

2 IMPORT REQUIRED LIBRARIES:

[32]:

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


import warnings
warnings.filterwarnings('ignore')
```

[33]:

```
df = pd.read_csv("netflix_titles.csv")
print("Initial shape:", df.shape)
df.head()
```

```
Initial shape: (6234, 12)
```

[33]:

```
    show_id     type                                    title  \
0  81145628    Movie  Norm of the North: King Sized Adventure
1  80117401    Movie               Jandino: Whatever it Takes
2  70234439  TV Show                       Transformers Prime
3  80058654  TV Show         Transformers: Robots in Disguise
4  80125979    Movie                             #realityhigh

                   director  \
0  Richard Finn, Tim Maltby
1                       NaN
2                       NaN
3                       NaN
4          Fernando Lebrija
```

```
                                                cast  \
0  Alan Marriott, Andrew Toth, Brian Dobson, Cole…
1                                   Jandino Asporaat
2  Peter Cullen, Sumalee Montano, Frank Welker, J…
3  Will Friedle, Darren Criss, Constance Zimmer, …
4  Nesta Cooper, Kate Walsh, John Michael Higgins…

                                    country         date_added  release_year  \
0  United States, India, South Korea, China  September 9, 2019          2019
1                            United Kingdom  September 9, 2016          2016
2                             United States  September 8, 2018          2013
3                             United States  September 8, 2018          2016
4                             United States  September 8, 2017          2017

     rating  duration                           listed_in  \
0     TV-PG    90 min  Children & Family Movies, Comedies
1     TV-MA    94 min                     Stand-Up Comedy
2  TV-Y7-FV  1 Season                            Kids' TV
3     TV-Y7  1 Season                            Kids' TV
4     TV-14    99 min                            Comedies

                                         description
0  Before planning an awesome wedding for his gra…
1  Jandino Asporaat riffs on the challenges of ra…
2  With the help of three human allies, the Autob…
3  When a prison ship crash unleashes hundreds of…
4  When nerdy high schooler Dani finally attracts…
```

3 DATA CLEANING:

[34]:

```
print("\nMissing values per column:\n")
print(df.isnull().sum())

df['director'].fillna('Unknown', inplace=True)
df['cast'].fillna('Unknown', inplace=True)
df['country'].fillna('Unknown', inplace=True)
df['rating'].fillna('Unknown', inplace=True)
df['duration'].fillna('Unknown', inplace=True)
df['date_added'].fillna('Unknown', inplace=True)

df.drop_duplicates(inplace=True)

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

# Extract Year and Month from date_added
df['year_added'] = df['date_added'].dt.year
```

```python
df['month_added'] = df['date_added'].dt.month_name()

# Extract numeric duration (for Movies only)
def extract_duration(x):
    try:
        if 'min' in x:
            return int(x.split()[0])
        else:
            return np.nan
    except:
        return np.nan

df['duration_num'] = df['duration'].apply(extract_duration)

df['primary_country'] = df['country'].apply(lambda x: x.split(',')[0] if x != 'Unknown' else 'Unknown')
```

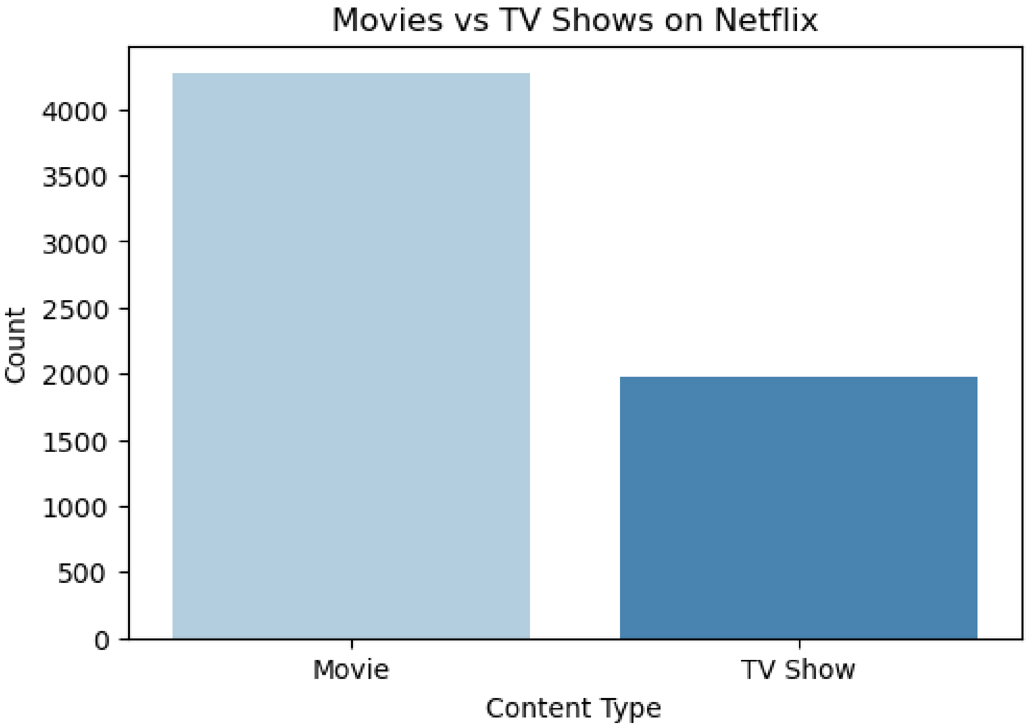
```
Missing values per column:

show_id            0
type               0
title              0
director        1969
cast             570
country          476
date_added        11
release_year       0
rating            10
duration           0
listed_in          0
description        0
dtype: int64
```

4 EDA:

[35]:

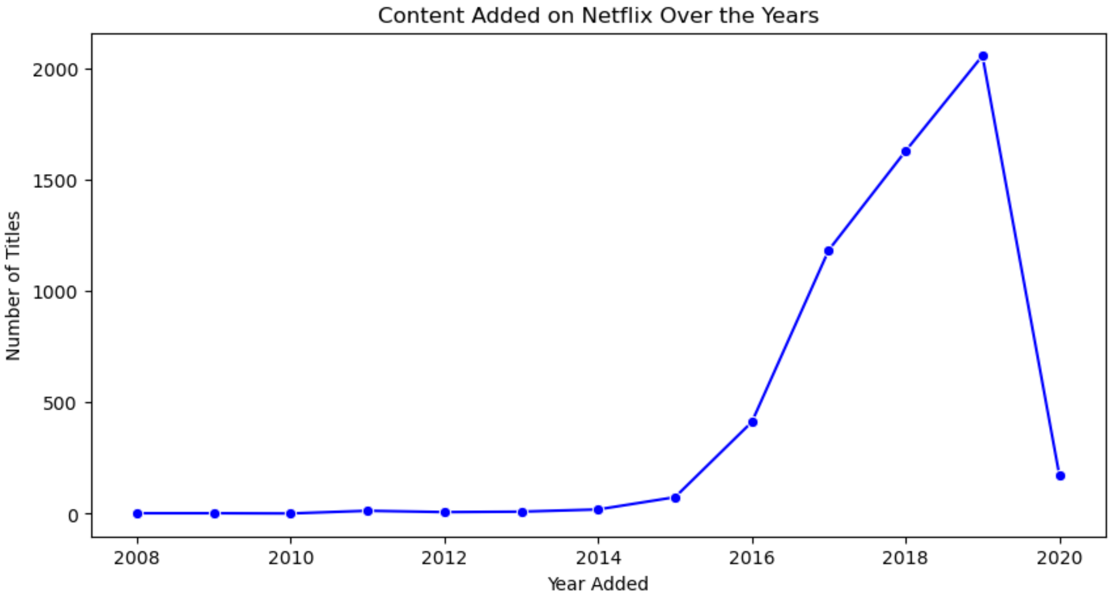
```python
print("\nDataset after cleaning and feature creation:\n")
print(df.info())


# 1. NUMBER OF MOVIES v/s TV Shows:
plt.figure(figsize=(6,4))
sns.countplot(x='type', data=df, palette='Blues')
plt.title('Movies vs TV Shows on Netflix')
plt.xlabel('Content Type')
plt.ylabel('Count')
plt.show()
```

```
Dataset after cleaning and feature creation:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6234 entries, 0 to 6233
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   show_id          6234 non-null   int64
 1   type             6234 non-null   object
 2   title            6234 non-null   object
 3   director         6234 non-null   object
 4   cast             6234 non-null   object
 5   country          6234 non-null   object
 6   date_added       5583 non-null   datetime64[ns]
 7   release_year     6234 non-null   int64
 8   rating           6234 non-null   object
 9   duration         6234 non-null   object
 10  listed_in        6234 non-null   object
 11  description      6234 non-null   object
 12  year_added       5583 non-null   float64
 13  month_added      5583 non-null   object
 14  duration_num     4265 non-null   float64
 15  primary_country  6234 non-null   object
dtypes: datetime64[ns](1), float64(2), int64(2), object(11)
memory usage: 779.4+ KB
None
```



[36]:

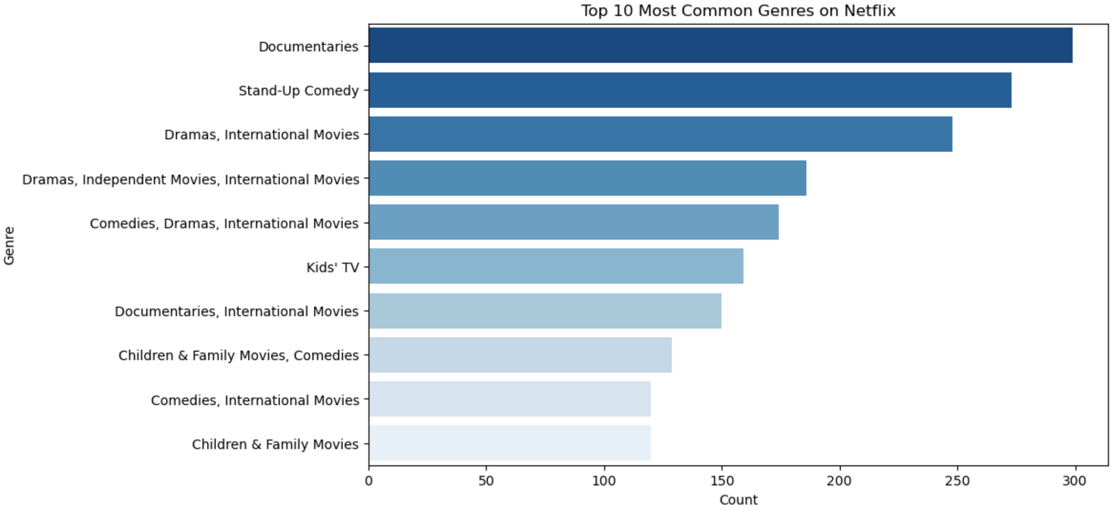
```
# 2. YEARLY TRENDS:
yearly_count = df['year_added'].value_counts().sort_index()
plt.figure(figsize=(10,5))
sns.lineplot(x=yearly_count.index, y=yearly_count.values, marker='o',␣
 ↪color='blue')
plt.title('Content Added on Netflix Over the Years')
plt.xlabel('Year Added')
plt.ylabel('Number of Titles')
plt.show()
```



Content Added on Netflix Over the Years

[37]:

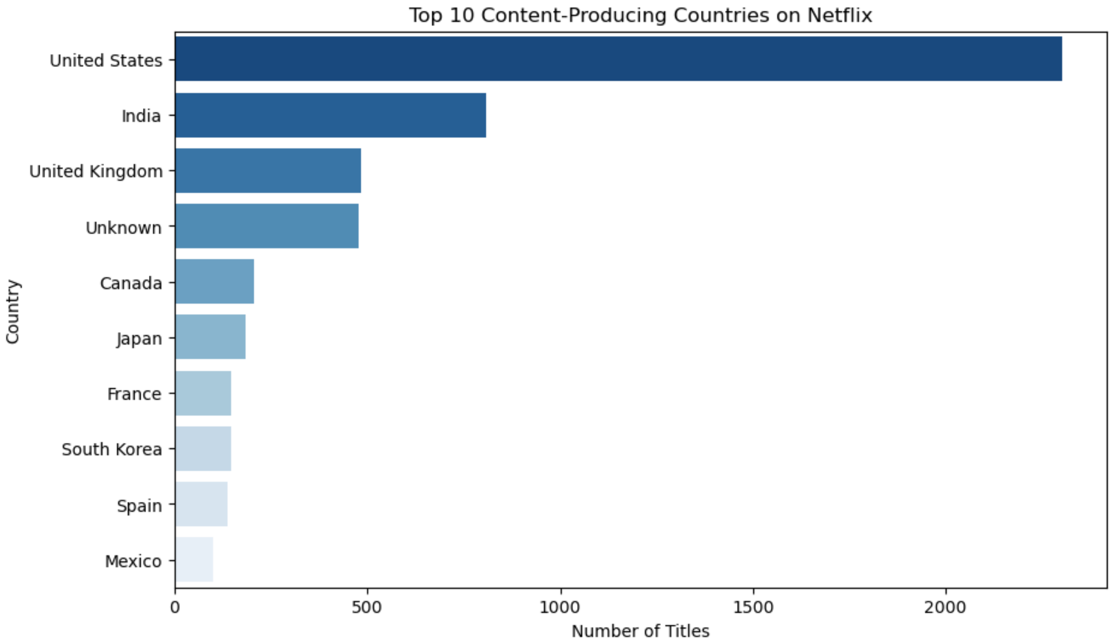
```
# 3. HIGH PERFORMANCE GENRES:
plt.figure(figsize=(10,6))
genre_data = df['listed_in'].value_counts().head(10)
sns.barplot(y=genre_data.index, x=genre_data.values, palette='Blues_r')
plt.title('Top 10 Most Common Genres on Netflix')
plt.xlabel('Count')
plt.ylabel('Genre')
plt.show()
```



Top 10 Most Common Genres on Netflix

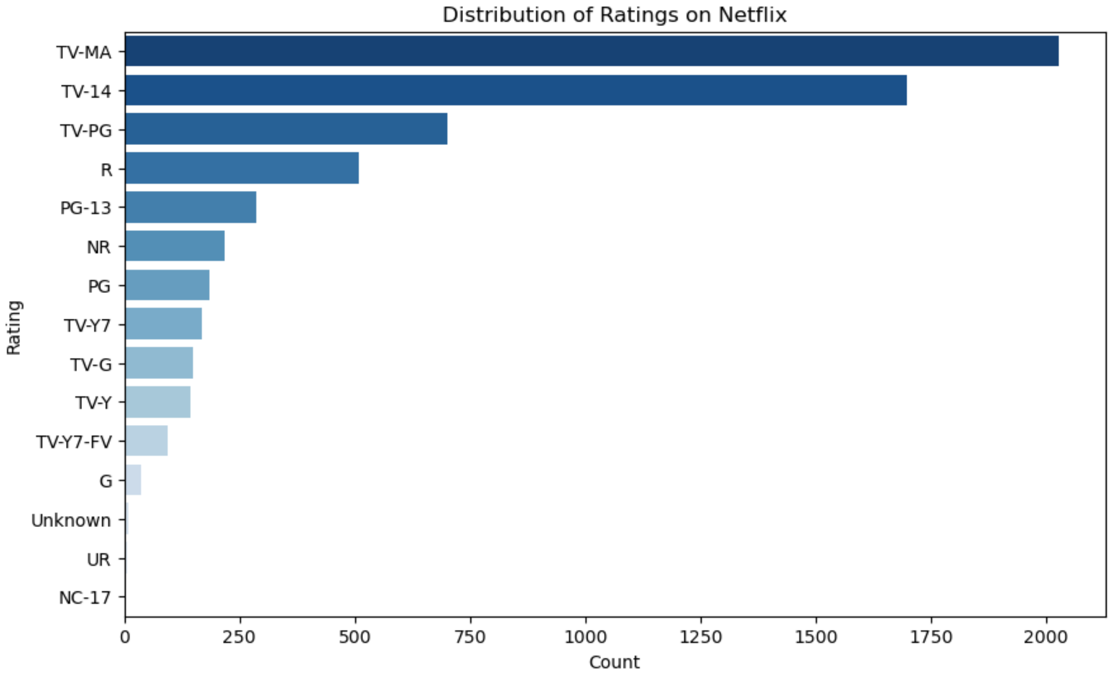
[38]:

```
# 4. TOP-10 CONTRIBUTING COUNTRIES:
country_data = df['primary_country'].value_counts().head(10)
plt.figure(figsize=(10,6))
sns.barplot(y=country_data.index, x=country_data.values, palette='Blues_r')
plt.title('Top 10 Content-Producing Countries on Netflix')
plt.xlabel('Number of Titles')
plt.ylabel('Country')
plt.show()
```



[39]:

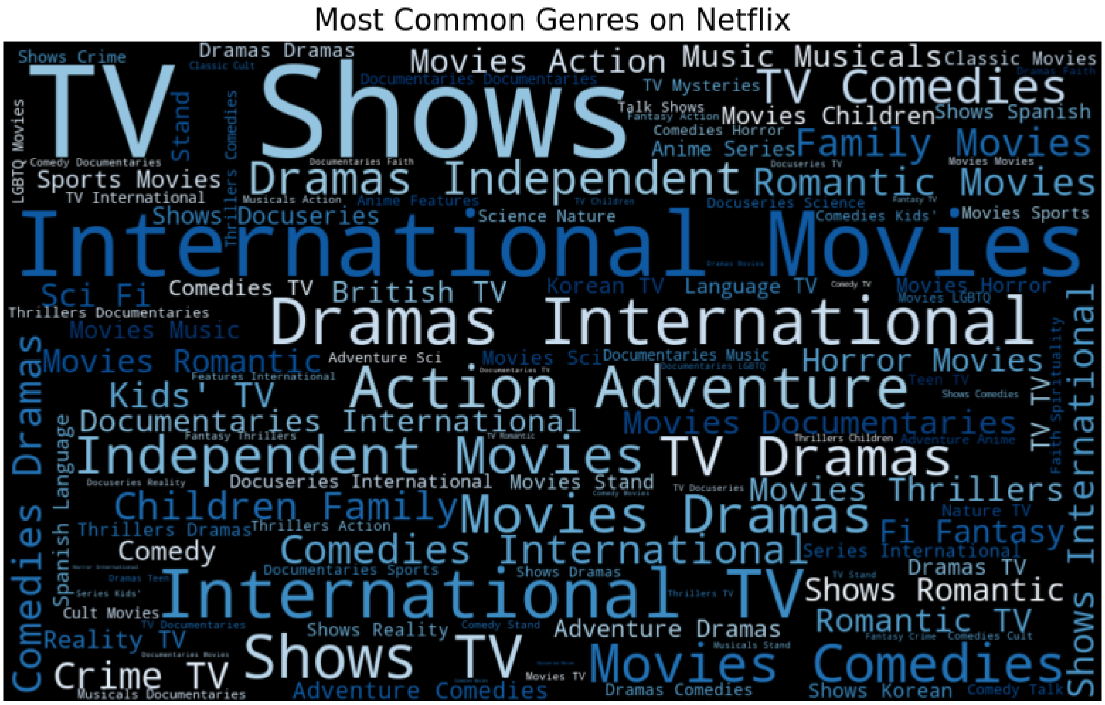
```
# 5.RATINGS DISTRIBUTION:
plt.figure(figsize=(10,6))
sns.countplot(y='rating', data=df, order=df['rating'].value_counts().index,␣
 ↪palette='Blues_r')
plt.title('Distribution of Ratings on Netflix')
plt.xlabel('Count')
plt.ylabel('Rating')
plt.show()
```



[40]:
```
# 6. AVERAGE DURATION OF MOVIES:
avg_duration = df[df['type']=='Movie']['duration_num'].mean()
print(f"\nAverage Movie Duration: {avg_duration:.2f} minutes")
```

```

Average Movie Duration: 99.10 minutes
```

[41]:
```
# 7. GENRES WORDCLOUD:
plt.figure(figsize=(10,7))
text = ' '.join(df['listed_in'].dropna().astype(str))
wordcloud = WordCloud(width=1000, height=600, background_color='black',␣
 ↪colormap='Blues').generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most Common Genres on Netflix', fontsize=15)
plt.show()
```



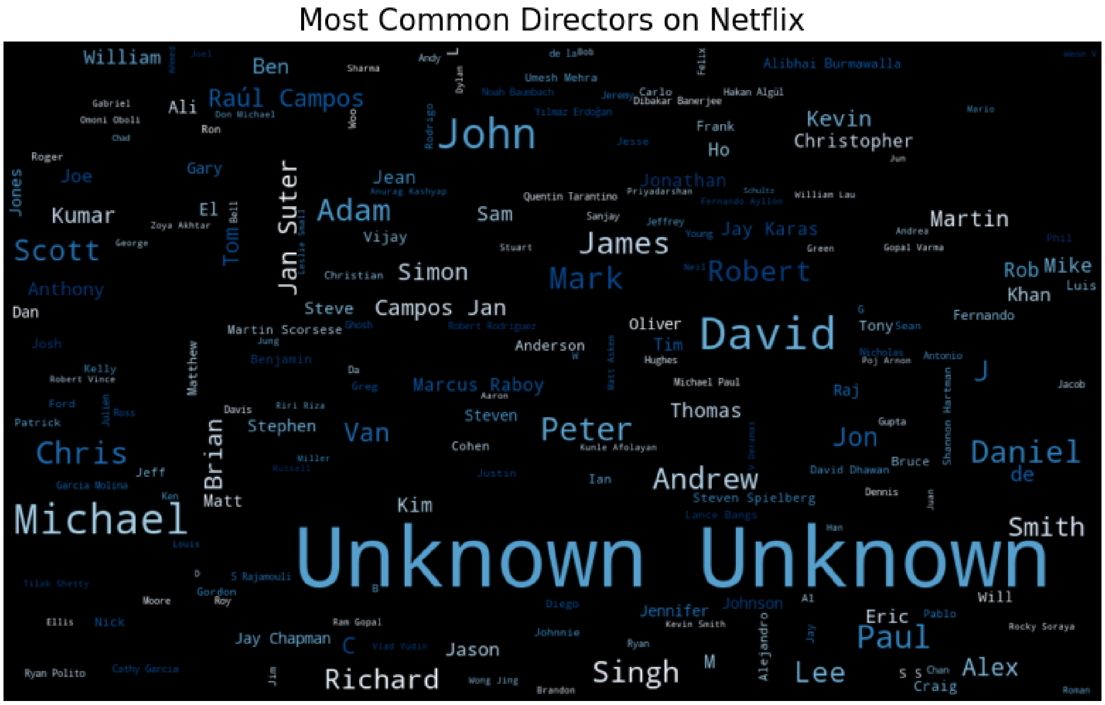
```
[42]: df_directors = df.copy()
      df_directors["director"] = df_directors["director"].str.split(',')
      df_directors=df_directors.explode("director").reset_index(drop=True)
      df_directors.head()
```

```
[42]:     show_id     type                                      title      director  \
      0  81145628    Movie  Norm of the North: King Sized Adventure  Richard Finn
      1  81145628    Movie  Norm of the North: King Sized Adventure    Tim Maltby
      2  80117401    Movie               Jandino: Whatever it Takes       Unknown
      3  70234439  TV Show                       Transformers Prime       Unknown
      4  80058654  TV Show         Transformers: Robots in Disguise       Unknown

                                                 cast  \
      0  Alan Marriott, Andrew Toth, Brian Dobson, Cole…
      1  Alan Marriott, Andrew Toth, Brian Dobson, Cole…
      2                                  Jandino Asporaat
      3  Peter Cullen, Sumalee Montano, Frank Welker, J…
      4  Will Friedle, Darren Criss, Constance Zimmer, …

                                         country date_added  release_year  \
      0  United States, India, South Korea, China 2019-09-09          2019
      1  United States, India, South Korea, China 2019-09-09          2019
      2                             United Kingdom 2016-09-09          2016
      3                              United States 2018-09-08          2013
```

```
4                             United States 2018-09-08          2016

     rating  duration                              listed_in  \
0     TV-PG    90 min  Children & Family Movies, Comedies
1     TV-PG    90 min  Children & Family Movies, Comedies
2     TV-MA    94 min                     Stand-Up Comedy
3  TV-Y7-FV  1 Season                            Kids' TV
4     TV-Y7  1 Season                            Kids' TV

                                         description  year_added month_added  \
0  Before planning an awesome wedding for his gra…      2019.0   September
1  Before planning an awesome wedding for his gra…      2019.0   September
2  Jandino Asporaat riffs on the challenges of ra…      2016.0   September
3  With the help of three human allies, the Autob…      2018.0   September
4  When a prison ship crash unleashes hundreds of…      2018.0   September

   duration_num primary_country
0          90.0   United States
1          90.0   United States
2          94.0  United Kingdom
3           NaN   United States
4           NaN   United States
```

[43]:

```python
# 8. DIRECTORS WORDCLOUD:
plt.figure(figsize=(10,7))
text = ' '.join(df_directors['director'].dropna().astype(str))
wordcloud = WordCloud(width=1000, height=600, background_color='black', colormap='Blues').generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most Common Directors on Netflix', fontsize=15)
plt.show()
```



Most Common Directors on Netflix

[44]:

```
# 9. COUNTRIES-WISE MAP:
try:
    import plotly.express as px
    country_counts = df['primary_country'].value_counts().reset_index()
    country_counts.columns = ['country', 'count']
    fig = px.choropleth(country_counts, locations='country',␣
  ↪locationmode='country names',
                        color='count', title='Netflix Content by Country',
                        color_continuous_scale='Blues')
    fig.show()
except:
    print("Plotly not installed. Install via: pip install plotly")

# SAVE CLEANED DATA:
df.to_csv("netflix_cleaned.csv", index=False)
print("\n Data cleaning complete. File saved as netflix_cleaned.csv")
```

```

 Data cleaning complete. File saved as netflix_cleaned.csv
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: