

```
In [11]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [7]: import pandas as pd

# Read CSV file (skipping metadata rows)
df = pd.read_csv("API_SP.POP.TOTL_DS2_en_csv_v2_87.csv", skiprows=4)

# Display first few rows
print(df.head())

      Country Name Country Code Indicator Name Indicator Code \
0              Aruba          ABW Population, total    SP.POP.TOTL
1 Africa Eastern and Southern AFE Population, total    SP.POP.TOTL
2              Afghanistan AFG Population, total    SP.POP.TOTL
3 Africa Western and Central AFW Population, total    SP.POP.TOTL
4              Angola          AGO Population, total    SP.POP.TOTL

      1960      1961      1962      1963      1964 \
0  54922.0  55578.0  56320.0  57002.0  57619.0
1 130072080.0 133534923.0 137171659.0 140945536.0 144904094.0
2   9035043.0   9214083.0   9404406.0   9604487.0   9814318.0
3  97630925.0  99706674.0 101854756.0 104089175.0 106388440.0
4   5231654.0   5301583.0   5354310.0   5408320.0   5464187.0

      1965      ...      2015      2016      2017      2018 \
0   58190.0      ...      107906.0  108727.0  108735.0  108908.0
1 149033472.0      ...  607123269.0 623369401.0 640058741.0 657801085.0
2  10036008.0      ...   33831764.0  34700612.0  35688935.0  36743039.0
3 108772632.0      ...  418127845.0 429454743.0 440882906.0 452195915.0
4   5521981.0      ...   28157798.0  29183070.0  30234839.0  31297155.0

      2019      2020      2021      2022      2023 \
0  109203.0  108587.0  107700.0  107310.0  107359.0
1 675950189.0 694446100.0 713090928.0 731821393.0 750503764.0
2  37856121.0  39068979.0  40000412.0  40578842.0  41454761.0
3 463365429.0 474569351.0 485920997.0 497387180.0 509398589.0
4  32375632.0  33451132.0  34532429.0  35635029.0  36749906.0

Unnamed: 68
0      NaN
1      NaN
2      NaN
3      NaN
4      NaN

[5 rows x 69 columns]
```

```
In [9]: # Display first few rows
print("First 5 rows of the dataset:")
print(df.head())

First 5 rows of the dataset:
      Country Name Country Code Indicator Name Indicator Code \
0              Aruba          ABW Population, total    SP.POP.TOTL
1 Africa Eastern and Southern AFE Population, total    SP.POP.TOTL
2              Afghanistan AFG Population, total    SP.POP.TOTL
3 Africa Western and Central AFW Population, total    SP.POP.TOTL
4              Angola          AGO Population, total    SP.POP.TOTL

      1960      1961      1962      1963      1964 \
0  54922.0  55578.0  56320.0  57002.0  57619.0
1 130072080.0 133534923.0 137171659.0 140945536.0 144904094.0
2   9035043.0   9214083.0   9404406.0   9604487.0   9814318.0
3  97630925.0  99706674.0 101854756.0 104089175.0 106388440.0
4   5231654.0   5301583.0   5354310.0   5408320.0   5464187.0

      1965      ...      2015      2016      2017      2018 \
0   58190.0      ...      107906.0  108727.0  108735.0  108908.0
1 149033472.0      ...  607123269.0 623369401.0 640058741.0 657801085.0
2  10036008.0      ...   33831764.0  34700612.0  35688935.0  36743039.0
3 108772632.0      ...  418127845.0 429454743.0 440882906.0 452195915.0
4   5521981.0      ...   28157798.0  29183070.0  30234839.0  31297155.0

      2019      2020      2021      2022      2023 \
0  109203.0  108587.0  107700.0  107310.0  107359.0
1 675950189.0 694446100.0 713090928.0 731821393.0 750503764.0
2  37856121.0  39068979.0  40000412.0  40578842.0  41454761.0
3 463365429.0 474569351.0 485920997.0 497387180.0 509398589.0
4  32375632.0  33451132.0  34532429.0  35635029.0  36749906.0

Unnamed: 68
0      NaN
1      NaN
2      NaN
3      NaN
4      NaN

[5 rows x 69 columns]
```

```
In [11]: # Display column names
print("\nColumns in the dataset:")
print(df.columns)

Columns in the dataset:
Index(['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code',
      '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1967', '1968',
      '1969', '1970', '1971', '1972', '1973', '1974', '1975', '1976', '1977',
      '1978', '1979', '1980', '1981', '1982', '1983', '1984', '1985', '1986',
      '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995',
      '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
      '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013',
      '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022',
      '2023', 'Unnamed: 68'],
      dtype='object')
```

```
In [13]: # Choose a numerical variable for histogram (modify as needed)
column_to_plot = "Age" # Replace with a relevant column from your dataset
```

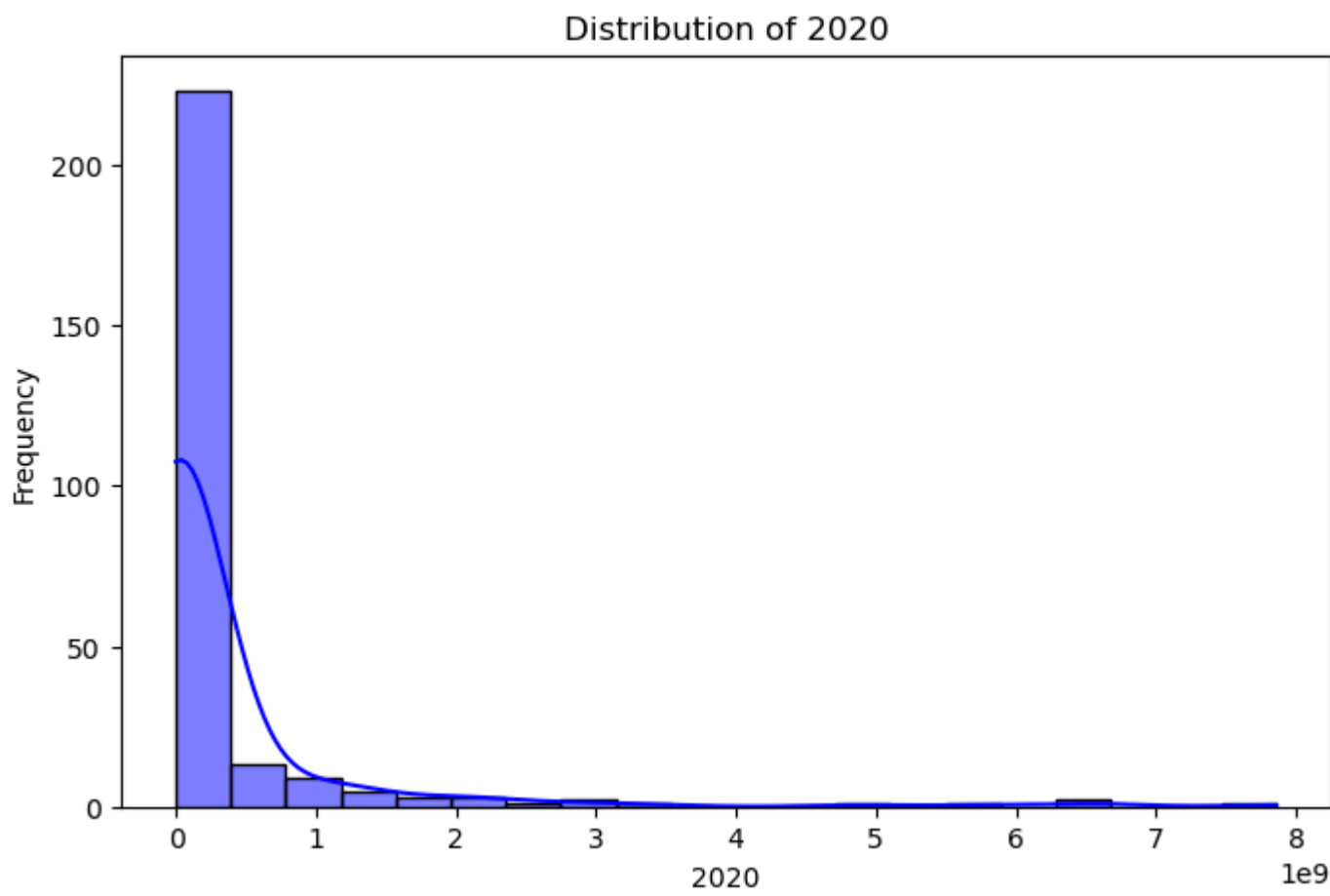
```
In [19]: import matplotlib.pyplot as plt
import seaborn as sns

# Check available columns
print(df.columns)

# Choose a numerical column
column_to_plot = "2020" # Change this to an actual column from your dataset

# Plot histogram
plt.figure(figsize=(8, 5))
sns.histplot(df[column_to_plot], bins=20, kde=True, color="blue")
plt.title(f"Distribution of {column_to_plot}")
plt.xlabel(column_to_plot)
plt.ylabel("Frequency")
plt.show()

Index(['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code',
      '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1967', '1968',
      '1969', '1970', '1971', '1972', '1973', '1974', '1975', '1976', '1977',
      '1978', '1979', '1980', '1981', '1982', '1983', '1984', '1985', '1986',
      '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995',
      '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
      '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013',
      '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022',
      '2023', 'Unnamed: 68'],
      dtype='object')
```



```
In [25]: import matplotlib.pyplot as plt
import seaborn as sns

# Check available categorical columns
print(df.select_dtypes(include=["object"]).columns) # Lists only categorical columns

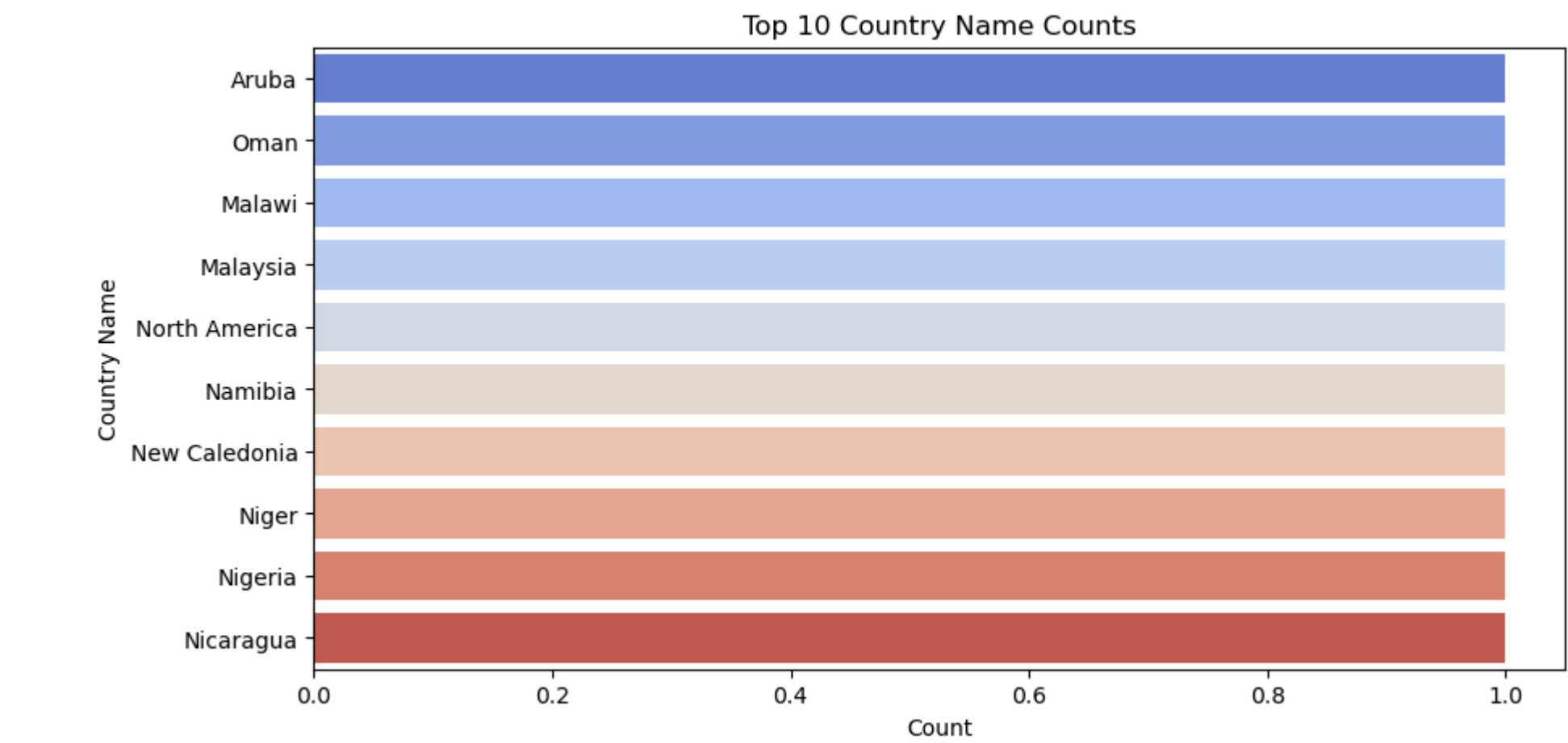
# Replace "Country Name" with a relevant categorical column
categorical_column = "Country Name" # Change this based on your dataset

if categorical_column in df.columns:
    plt.figure(figsize=(10, 5))
    sns.countplot(y=categorical_column, data=df, palette="coolwarm", order=df[categorical_column].value_counts().index[:10])
    plt.title(f"Top 10 {categorical_column} Counts")
    plt.xlabel("Count")
    plt.ylabel(categorical_column)
    plt.show()
else:
    print(f"Column '{categorical_column}' not found in the dataset.")

Index(['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code'], dtype='object')

/var/folders/r3/trn6xwxc7js3_2j79fkcxz880000gn/T/ipykernel_29689/227000036.py:12: FutureWarning:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'y' variable to 'hue' and set 'legend=False' for the same effect.

sns.countplot(y=categorical_column, data=df, palette="coolwarm", order=df[categorical_column].value_counts().index[:10])
```



In []: