```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
from google.colab import files
uploaded = files.upload()

# Assuming the uploaded file is named 'yourfile.csv'
df = pd.read_csv('API_SP.POP.TOTL_DS2_en_csv_v2_900.csv',skiprows=4,header=0)
Metadata_Country = pd.read_csv('Metadata_Country_API_SP.POP.TOTL_DS2_en_csv_v2_900.csv')
Metadata_Indicator = pd.read_csv('Metadata_Indicator_API_SP.POP.TOTL_DS2_en_csv_v2_900.csv')

# Display the first few rows of the DataFrame
df.head()
```

Choose Files 3 files

- **API_SP.POP.TOTL_DS2_en_csv_v2_900.csv**(text/csv) - 191371 bytes, last modified: 1/20/2025 - 100% done
- **Metadata_Country_API_SP.POP.TOTL_DS2_en_csv_v2_900.csv**(text/csv) - 59105 bytes, last modified: 1/20/2025 - 100% done
- **Metadata_Indicator_API_SP.POP.TOTL_DS2_en_csv_v2_900.csv**(text/csv) - 588 bytes, last modified: 1/20/2025 - 100% done

```
Saving API_SP.POP.TOTL_DS2_en_csv_v2_900.csv to API_SP.POP.TOTL_DS2_en_csv_v2_900 (5).csv
Saving Metadata_Country_API_SP.POP.TOTL_DS2_en_csv_v2_900.csv to Metadata_Country_API_SP.POP.TOTL_DS2_en_csv_v2_900 (2).csv
Saving Metadata_Indicator_API_SP.POP.TOTL_DS2_en_csv_v2_900.csv to Metadata_Indicator_API_SP.POP.TOTL_DS2_en_csv_v2_900 (2).csv
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2(
0	Aruba	ABW	Population, total	SP.POP.TOTL	54922.0	55578.0	56320.0	57002.0	57619.0	58190.0	...	10790
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130072080.0	133534923.0	137171659.0	140945536.0	144904094.0	149033472.0	...	60712326
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	9035043.0	9214083.0	9404406.0	9604487.0	9814318.0	10036008.0	...	3383176
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97630925.0	99706674.0	101854756.0	104089175.0	106388440.0	108772632.0	...	41812784
4	Angola	AGO	Population, total	SP.POP.TOTL	5231654.0	5301583.0	5354310.0	5408320.0	5464187.0	5521981.0	...	2815779

5 rows × 69 columns

```
df.info()
```

```
13  1969            264 non-null    float64
14  1970            264 non-null    float64
15  1971            264 non-null    float64
16  1972            264 non-null    float64
17  1973            264 non-null    float64
18  1974            264 non-null    float64
19  1975            264 non-null    float64
20  1976            264 non-null    float64
21  1977            264 non-null    float64
22  1978            264 non-null    float64
```

```
 41  1997           265 non-null    float64
 42  1998           265 non-null    float64
 43  1999           265 non-null    float64
 44  2000           265 non-null    float64
 45  2001           265 non-null    float64
 46  2002           265 non-null    float64
 47  2003           265 non-null    float64
 48  2004           265 non-null    float64
 49  2005           265 non-null    float64
 50  2006           265 non-null    float64
 51  2007           265 non-null    float64
 52  2008           265 non-null    float64
 53  2009           265 non-null    float64
 54  2010           265 non-null    float64
 55  2011           265 non-null    float64
 56  2012           265 non-null    float64
 57  2013           265 non-null    float64
 58  2014           265 non-null    float64
 59  2015           265 non-null    float64
 60  2016           265 non-null    float64
 61  2017           265 non-null    float64
 62  2018           265 non-null    float64
 63  2019           265 non-null    float64
 64  2020           265 non-null    float64
 65  2021           265 non-null    float64
 66  2022           265 non-null    float64
 67  2023           265 non-null    float64
 68  Unnamed: 68    0 non-null      float64
dtypes: float64(65), object(4)
memory usage: 143.5+ KB
```

```
Metadata_Country.head()
```

	Country Code	Region	IncomeGroup	SpecialNotes	TableName	Unnamed: 5
0	ABW	Latin America & Caribbean	High income	NaN	Aruba	NaN
1	AFE	NaN	NaN	26 countries, stretching from the Red Sea in t...	Africa Eastern and Southern	NaN
2	AFG	South Asia	Low income	The reporting period for national accounts dat...	Afghanistan	NaN

Next steps: Generate code with Metadata_Country View recommended plots New interactive sheet

```
Metadata_Indicator.head()
```

	INDICATOR_CODE	INDICATOR_NAME	SOURCE_NOTE	SOURCE_ORGANIZATION	Unnamed: 4
0	SP.POP.TOTL	Population, total	Total population is based on the de facto defi...	(1) United Nations Population Division. World ...	NaN

```
merged_df = pd.merge(df, Metadata_Country, on='Country Code', how='left')
merged_df.head()
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	20
0	Aruba	ABW	Population, total	SP.POP.TOTL	54922.0	55578.0	56320.0	57002.0	57619.0	58190.0	...	10858
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130072080.0	133534923.0	137171659.0	140945536.0	144904094.0	149033472.0	...	69444610
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	9035043.0	9214083.0	9404406.0	9604487.0	9814318.0	10036008.0	...	3906897
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97630925.0	99706674.0	101854756.0	104089175.0	106388440.0	108772632.0	...	47456935
4	Angola	AGO	Population, total	SP.POP.TOTL	5231654.0	5301583.0	5354310.0	5408320.0	5464187.0	5521981.0	...	3345113

5 rows × 74 columns

```
merged_df.info()
```

```
 18  1974            264 non-null    float64
 19  1975            264 non-null    float64
 20  1976            264 non-null    float64
 21  1977            264 non-null    float64
 22  1978            264 non-null    float64
 23  1979            264 non-null    float64
 24  1980            264 non-null    float64
 25  1981            264 non-null    float64
 26  1982            264 non-null    float64
 27  1983            264 non-null    float64
 28  1984            264 non-null    float64
 29  1985            264 non-null    float64
 30  1986            264 non-null    float64
 31  1987            264 non-null    float64
 32  1988            264 non-null    float64
 33  1989            264 non-null    float64
 34  1990            265 non-null    float64
 35  1991            265 non-null    float64
 36  1992            265 non-null    float64
 37  1993            265 non-null    float64
 38  1994            265 non-null    float64
 39  1995            265 non-null    float64
 40  1996            265 non-null    float64
 41  1997            265 non-null    float64
 42  1998            265 non-null    float64
 66  2022            265 non-null    float64
 67  2023            265 non-null    float64
 68  Unnamed: 68     0 non-null      float64
 69  Region          217 non-null    object
 70  IncomeGroup     216 non-null    object
 71  SpecialNotes    127 non-null    object
 72  TableName       265 non-null    object
 73  Unnamed: 5      0 non-null      float64
dtypes: float64(66), object(8)
memory usage: 153.9+ KB
```

```
pd.isnull(merged_df).sum()
```

	0
Country Name	0
Country Code	0
Indicator Name	0
Indicator Code	0
1960	2
...	...
Region	49
IncomeGroup	50
SpecialNotes	139
TableName	1
Unnamed: 5	266

74 rows × 1 columns

dtype: int64

```
merged_df.drop(columns=['Country Code','Indicator Name','Indicator Code','SpecialNotes','Unnamed: 5'],axis=1,inplace=True)
```

```
pd.set_option('display.max_columns', None)
merged_df.head()
```

	Country Name	1960	1961	1962	1963	1964	1965	1966	1967	1968	1
0	Aruba	54922.0	55578.0	56320.0	57002.0	57619.0	58190.0	58694.0	58990.0	59069.0	590
1	Africa Eastern and Southern	130072080.0	133534923.0	137171659.0	140945536.0	144904094.0	149033472.0	153281203.0	157704381.0	162329396.0	1670882
2	Afghanistan	9035043.0	9214083.0	9404406.0	9604487.0	9814318.0	10036008.0	10266395.0	10505959.0	10756922.0	110174
3	Africa Western and Central	97630925.0	99706674.0	101854756.0	104089175.0	106388440.0	108772632.0	111246953.0	113795019.0	116444636.0	1192035
4	Angola	5231654.0	5301583.0	5354310.0	5408320.0	5464187.0	5521981.0	5581386.0	5641807.0	5702699.0	57636

```
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 69 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Country Name  266 non-null    object
 1   1960          264 non-null    float64
 2   1961          264 non-null    float64
 3   1962          264 non-null    float64
 4   1963          264 non-null    float64
 5   1964          264 non-null    float64
 6   1965          264 non-null    float64
 7   1966          264 non-null    float64
 8   1967          264 non-null    float64
 9   1968          264 non-null    float64
 10  1969          264 non-null    float64
 11  1970          264 non-null    float64
 12  1971          264 non-null    float64
 13  1972          264 non-null    float64
 14  1973          264 non-null    float64
 15  1974          264 non-null    float64
 16  1975          264 non-null    float64
 17  1976          264 non-null    float64
 18  1977          264 non-null    float64
 19  1978          264 non-null    float64
 20  1979          264 non-null    float64
 21  1980          264 non-null    float64
 22  1981          264 non-null    float64
 23  1982          264 non-null    float64
 24  1983          264 non-null    float64
 25  1984          264 non-null    float64
 26  1985          264 non-null    float64
 27  1986          264 non-null    float64
 28  1987          264 non-null    float64
 29  1988          264 non-null    float64
 30  1989          264 non-null    float64
 31  1990          265 non-null    float64
 32  1991          265 non-null    float64
 33  1992          265 non-null    float64
 34  1993          265 non-null    float64
 35  1994          265 non-null    float64
 36  1995          265 non-null    float64
 37  1996          265 non-null    float64
 38  1997          265 non-null    float64
 39  1998          265 non-null    float64
 40  1999          265 non-null    float64
 41  2000          265 non-null    float64
 42  2001          265 non-null    float64
 43  2002          265 non-null    float64
 44  2003          265 non-null    float64
 45  2004          265 non-null    float64
 46  2005          265 non-null    float64
 47  2006          265 non-null    float64
 48  2007          265 non-null    float64
 49  2008          265 non-null    float64
 50  2009          265 non-null    float64
 51  2010          265 non-null    float64
 52  2011          265 non-null    float64
```

```
merged_df.describe()
```

	1960	1961	1962	1963	1964	1965	1966	1967	1968	
count	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.64000
mean	1.154482e+08	1.170540e+08	1.192163e+08	1.218881e+08	1.245838e+08	1.273114e+08	1.301584e+08	1.330130e+08	1.359428e+08	1.38970
std	3.626524e+08	3.671661e+08	3.738304e+08	3.824609e+08	3.911398e+08	3.999257e+08	4.091871e+08	4.184362e+08	4.279508e+08	4.37824
min	2.715000e+03	2.970000e+03	3.264000e+03	3.584000e+03	3.922000e+03	4.282000e+03	4.664000e+03	5.071000e+03	5.500000e+03	5.63100
25%	5.152028e+05	5.255230e+05	5.363018e+05	5.475875e+05	5.593638e+05	5.675750e+05	5.711695e+05	5.779525e+05	5.825170e+05	5.86118
50%	3.659633e+06	3.747132e+06	3.831900e+06	3.919710e+06	4.010150e+06	4.102976e+06	4.198738e+06	4.297792e+06	4.396290e+06	4.50342
75%	2.686293e+07	2.761326e+07	2.837302e+07	2.915448e+07	2.995223e+07	3.075921e+07	3.147516e+07	3.203946e+07	3.247057e+07	3.27714
max	3.021529e+09	3.062769e+09	3.117373e+09	3.184063e+09	3.251253e+09	3.318998e+09	3.389087e+09	3.459014e+09	3.530702e+09	3.60481

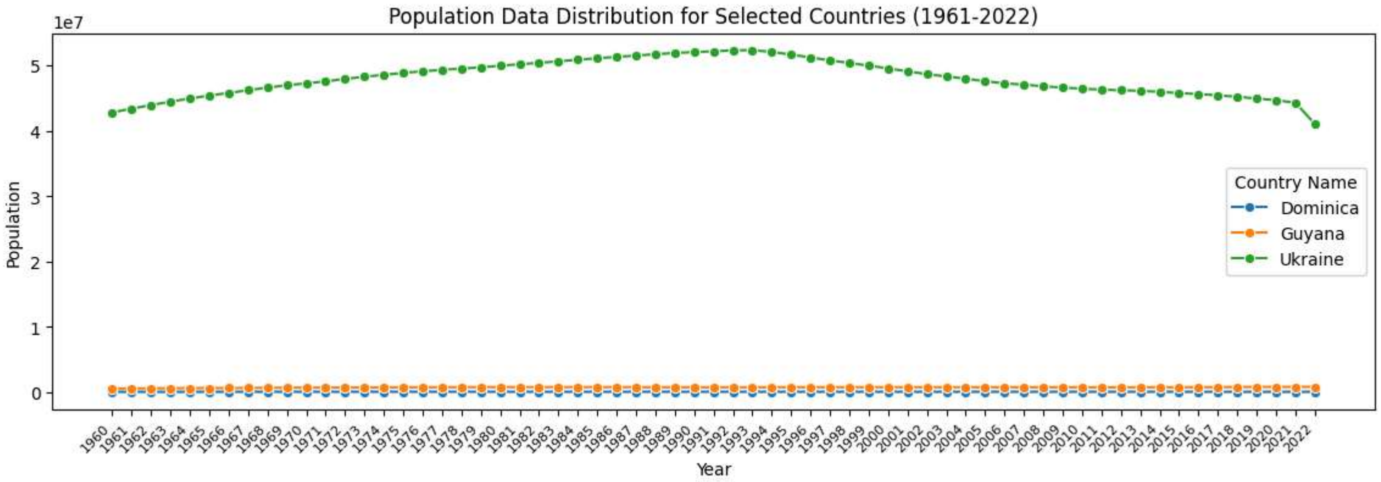
```
merged_df.describe(include='object')
```

	Country Name	Region	IncomeGroup	TableName
count	266	217	216	265
unique	266	7	4	265
top	Aruba	Europe & Central Asia	High income	Aruba
freq	1	58	85	1

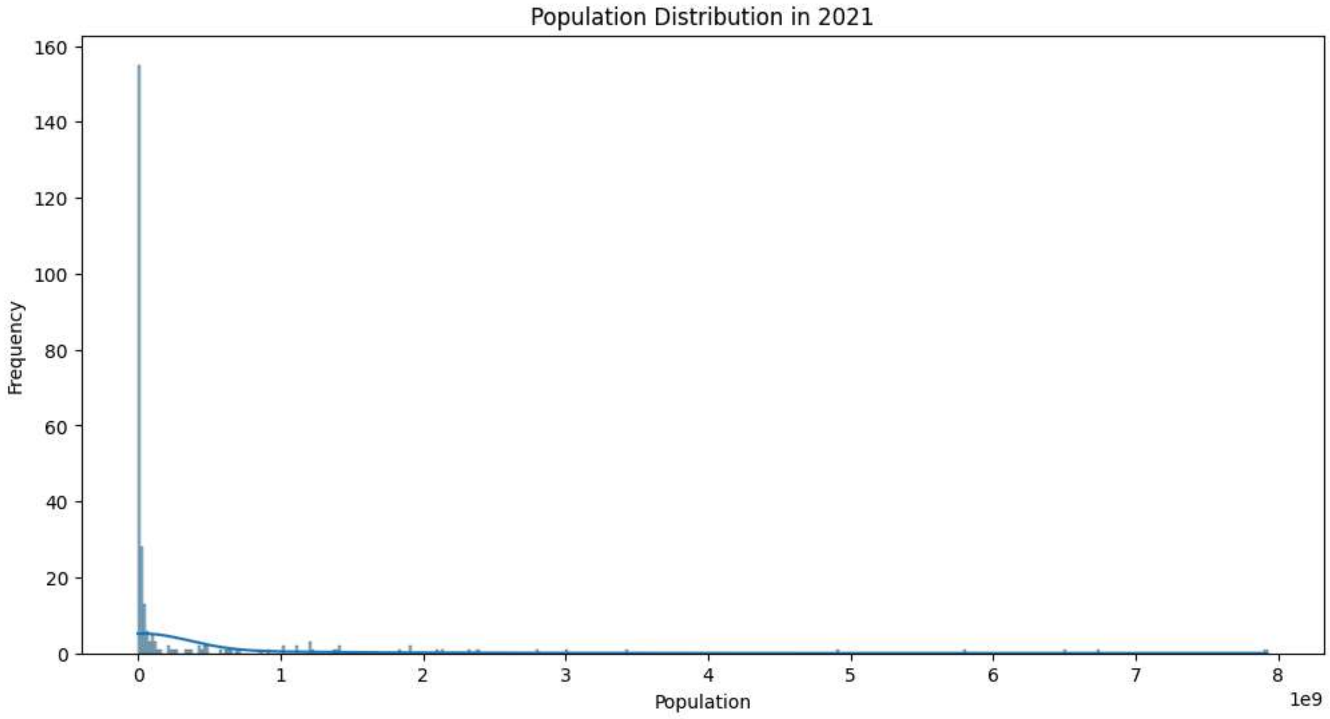
```
from scipy import stats
from scipy.stats import zscore

def detect_outliers_zscore(merged_df, threshold=3):
    numeric_columns = merged_df.select_dtypes(include=[np.number]).columns  # Choosing only numeric columns, fixed syntax error
    numeric_data = merged_df[numeric_columns].T
    z_scores = np.abs(stats.zscore(numeric_data, axis=0))
    outliers = (z_scores > threshold).any(axis=0)
    return outliers, z_scores.T

outliers, z_scores = detect_outliers_zscore(merged_df)    # Calling the function with correct name
# Display rows with outliers
outliers_df = merged_df[outliers]  # Fixed variable name to outliers_df
print("Rows with Outliers:")
print(outliers_df)
```

```
Rows with Outliers:
Empty DataFrame
Columns: [Country Name, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978
Index: []
```

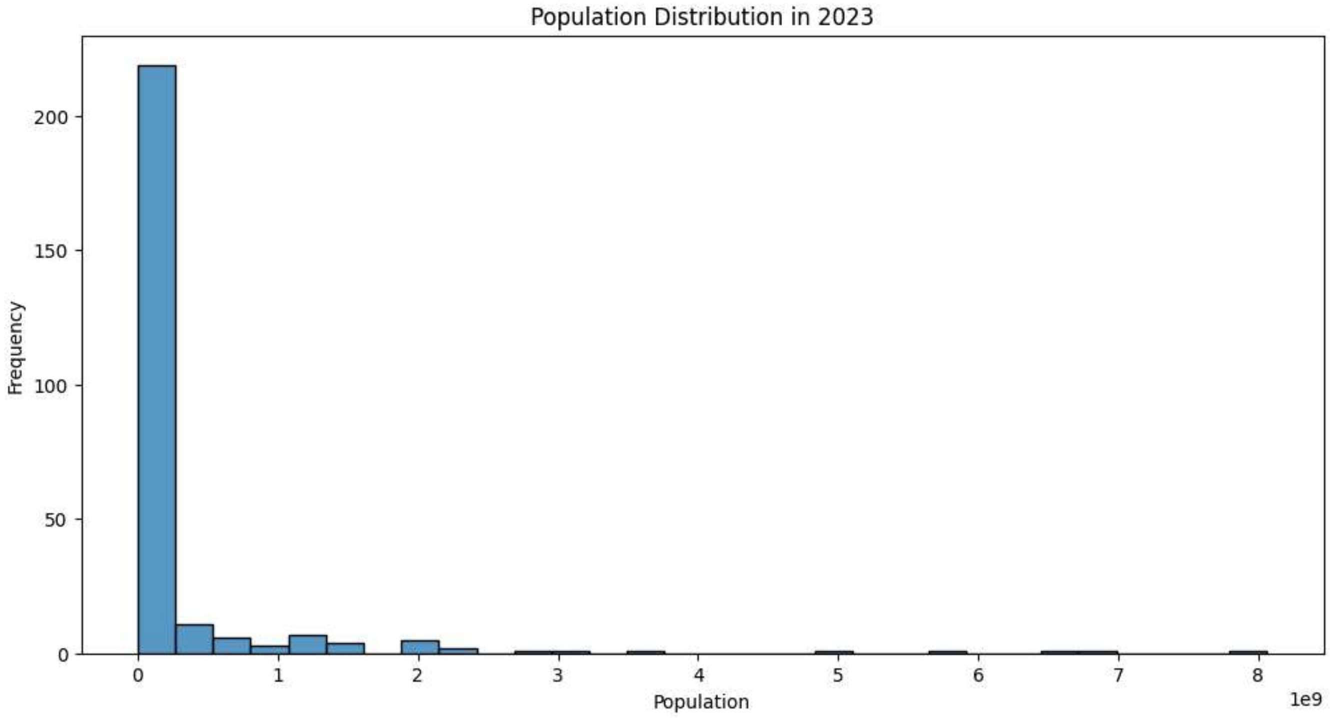
```
columns_of_interest = ['Country Name'] + [str(year) for year in range(1960, 2023)]
data_subset = merged_df[columns_of_interest]
# Choose three countries for visualization
countries_to_plot = ['Dominica', 'Guyana', 'Ukraine']
# Filter data for the selected countries
data_subset_countries = data_subset[data_subset['Country Name'].isin(countries_to_plot)]

melted_data = pd.melt(data_subset_countries, id_vars='Country Name', var_name='Year', value_name='Population')
# Create a line plot to visualize the data distribution for three countries
plt.figure(figsize=(14, 4))
sns.lineplot(x="Year", y='Population', hue='Country Name', data=melted_data, marker="o")
plt.title('Population Data Distribution for Selected Countries (1961-2022)')
plt.xlabel('Year')
plt.ylabel('Population')
plt.xticks(rotation=45, ha="right",fontsize=8)
plt.show()
# print("The graph clearly shows that Ukraine population is on decline, whereas in other two countries no significant positive trend is bein
```



```
year_to_visualize = 2021

# Create the bar chart
plt.figure(figsize=(12, 6))
sns.histplot(merged_df[str(year_to_visualize)], kde=True) # Use histplot for better visualization
plt.title(f'Population Distribution in {year_to_visualize}')
plt.xlabel('Population')
plt.ylabel('Frequency')
plt.show()
```



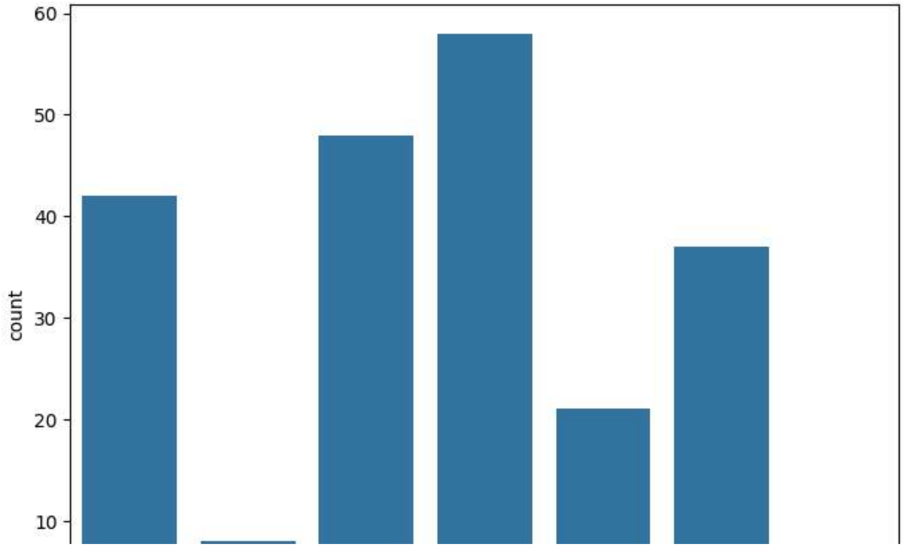
```
# Choose a year to visualize
year_to_visualize = 2023

# Create the bar chart
plt.figure(figsize=(12, 6))
sns.histplot(merged_df[str(year_to_visualize)], kde=False, bins=30)
plt.title(f'Population Distribution in {year_to_visualize}')
plt.xlabel('Population')
plt.ylabel('Frequency')
plt.show()
```



```
#  visualize the distribution of 'Region'
```

```
plt.figure(figsize=(8, 6))
sns.countplot(x='Region', data=merged_df) # Changed
```

```
<Axes: xlabel='Region', ylabel='count'>
```


count