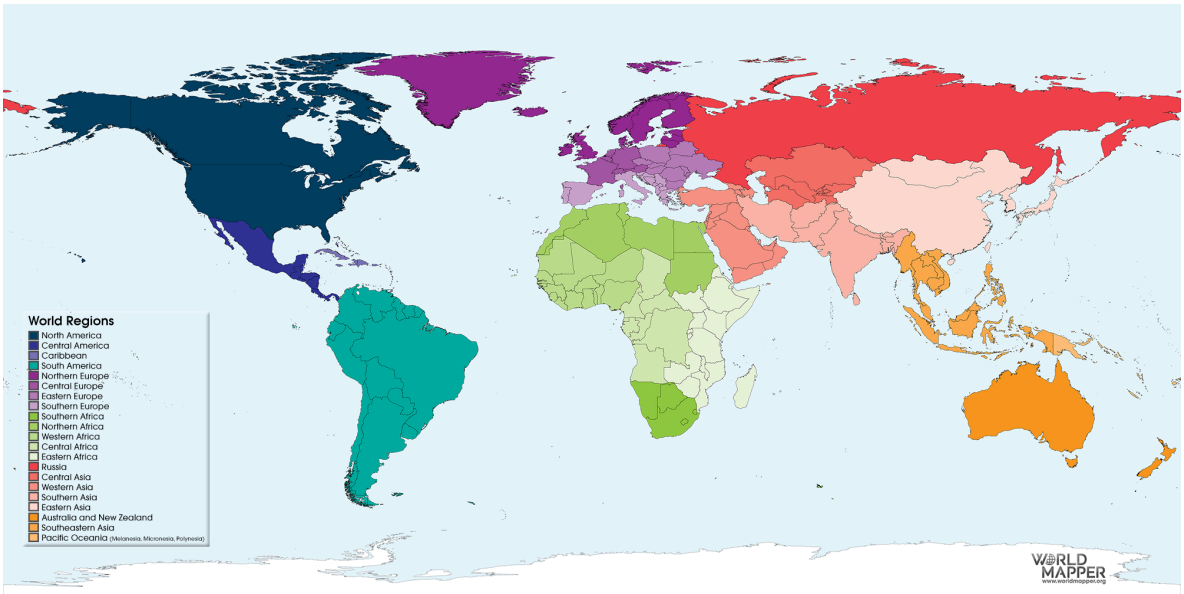
Data Visualization - World Population Data



About the project

This project is a comprehensive Data Visualization and Exploratory Data Analysis (EDA) effort focused on the World Population Dataset. The primary goal is to use graphical techniques to uncover trends, patterns, and insights related to global and regional population data.

Key Objectives

Trend Analysis: Visualize and analyze how the total global population has changed over the years.

Geographical Comparison: Create visualizations to compare population metrics across different countries and continents.

Ranking and Distribution: Identify and visualize the top most populated countries and examine the distribution of population by rank and density.

Methodology and Tools

The project utilizes powerful Python libraries to generate compelling and interactive visualizations:

Data Handling: Pandas and NumPy for loading and manipulating the population data.

Static Visualization: Matplotlib and Seaborn for creating standard statistical plots (e.g., bar plots, histograms, heatmaps).

Interactive Visualization: Plotly Express and Plotly Graph Objects for creating dynamic and interactive charts, including choropleth maps to display population data directly on a world map.

This project serves as a showcase for effective data storytelling and leveraging modern visualization tools to derive clear, meaningful insights from large-scale demographic data.

Importing Essential Libaries

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

Loading Data

In [ ]:
```
df=pd.read_csv('/content/world_population.csv')
df
```

Out[ ]:

	Rank	CCA3	Country/ Territory	Capital	Continent	2022 Population	2020 Population	Popul
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	3375
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	288
2	34	DZA	Algeria	Algiers	Africa	44903225	43451666	3954
3	213	ASM	American Samoa	Pago Pago	Oceania	44273	46189	5
4	203	AND	Andorra	Andorra la Vella	Europe	79824	77700	7
...	...	...	...	...	...	...	...	
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572	11655	1
230	172	ESH	Western Sahara	El Aaiún	Africa	575986	556048	49
231	46	YEM	Yemen	Sanaa	Asia	33696614	32284046	2851
232	63	ZMB	Zambia	Lusaka	Africa	20017675	18927715	1624
233	74	ZWE	Zimbabwe	Harare	Africa	16320537	15669666	1415

234 rows × 17 columns

First five values

In [ ]:
```
df.head()
```

Out[ ]:

	Rank	CCA3	Country/ Territory	Capital	Continent	2022 Population	2020 Population	20 Populati
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	337534
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	28824
2	34	DZA	Algeria	Algiers	Africa	44903225	43451666	395431
3	213	ASM	American Samoa	Pago Pago	Oceania	44273	46189	513
4	203	AND	Andorra	Andorra la Vella	Europe	79824	77700	717

Last five values

In [ ]:
```
df.tail()
```

Out[ ]:

	Rank	CCA3	Country/ Territory	Capital	Continent	2022 Population	2020 Population	2 Popula
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572	11655	12
230	172	ESH	Western Sahara	El Aaiún	Africa	575986	556048	491
231	46	YEM	Yemen	Sanaa	Asia	33696614	32284046	28516
232	63	ZMB	Zambia	Lusaka	Africa	20017675	18927715	16248
233	74	ZWE	Zimbabwe	Harare	Africa	16320537	15669666	14154

Data Information

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 234 entries, 0 to 233
Data columns (total 17 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Rank                          234 non-null    int64  
 1   CCA3                          234 non-null    object 
 2   Country/Territory            234 non-null    object 
 3   Capital                       234 non-null    object 
 4   Continent                     234 non-null    object 
 5   2022 Population               234 non-null    int64  
 6   2020 Population               234 non-null    int64  
 7   2015 Population               234 non-null    int64  
 8   2010 Population               234 non-null    int64  
 9   2000 Population               234 non-null    int64  
 10  1990 Population               234 non-null    int64  
 11  1980 Population               234 non-null    int64  
 12  1970 Population               234 non-null    int64  
 13  Area (km²)                    234 non-null    int64  
 14  Density (per km²)             234 non-null    float64
 15  Growth Rate                   234 non-null    float64
 16  World Population Percentage  234 non-null    float64
dtypes: float64(3), int64(10), object(4)
memory usage: 31.2+ KB
```

Columns of dataset

In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['Rank', 'CCA3', 'Country/Territory', 'Capital', 'Continent',
       '2022 Population', '2020 Population', '2015 Population',
       '2010 Population', '2000 Population', '1990 Population',
       '1980 Population', '1970 Population', 'Area (km²)', 'Density (per km
²)',
       'Growth Rate', 'World Population Percentage'],
      dtype='object')
```

Datatype of each columns

In [ ]:
```
df.dtypes
```

Out[ ]:
```
Rank                             int64
CCA3                            object
Country/Territory               object
Capital                         object
Continent                       object
2022 Population                  int64
2020 Population                  int64
2015 Population                  int64
2010 Population                  int64
2000 Population                  int64
1990 Population                  int64
1980 Population                  int64
1970 Population                  int64
Area (km²)                       int64
Density (per km²)              float64
Growth Rate                    float64
World Population Percentage    float64
dtype: object
```

Shape of dataset

In [ ]:
```
df.shape
```

Out[ ]:
```
(234, 17)
```

In [ ]:
```
df.describe().T.sort_values("50%", ascending = False).style.background_gradier
    .bar(subset = ["mean"], color = "red").bar(subset = ["max"], color = "gree
```

Out[ ]:

	count	mean	std	min	
2022 Population	234.000000	34074414.709402	136766424.804763	510.000000	419738.
2020 Population	234.000000	33501070.952991	135589876.924439	520.000000	415284.
2015 Population	234.000000	31729956.243590	130404992.751760	564.000000	404676.
2010 Population	234.000000	29845235.034188	124218487.632998	596.000000	393149.
2000 Population	234.000000	26269468.816239	111698206.719070	651.000000	327242.
1990 Population	234.000000	22710220.790598	97832173.346751	700.000000	264115.
1980 Population	234.000000	18984616.970085	81785186.084201	733.000000	229614.
1970 Population	234.000000	15786908.807692	67795091.643236	752.000000	155997.
Area (km²)	234.000000	581449.384615	1761840.864063	1.000000	2650.
Rank	234.000000	117.500000	67.694165	1.000000	59.
Density (per km²)	234.000000	452.127044	2066.121904	0.026100	38.
Growth Rate	234.000000	1.009577	0.013385	0.912000	1.
World Population Percentage	234.000000	0.427051	1.714977	0.000000	0.

Checking for any duplicate values

In [ ]:
```
df.duplicated().sum()
```

Out[ ]:
```
0
```

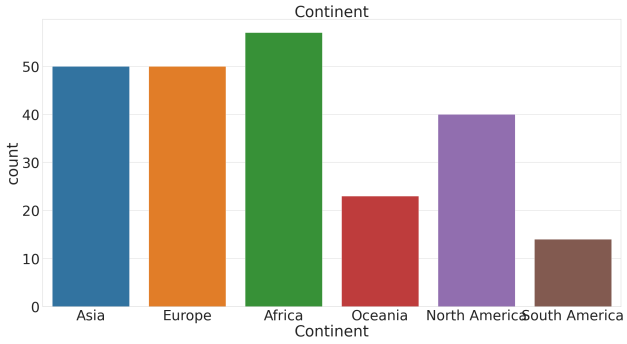
In [ ]:
```
df['Continent'].value_counts()
```

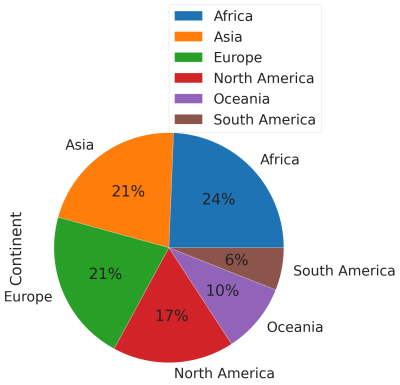
Out[ ]:
```
Africa           57
Asia             50
Europe           50
North America    40
Oceania          23
South America    14
Name: Continent, dtype: int64
```

Data Visualization

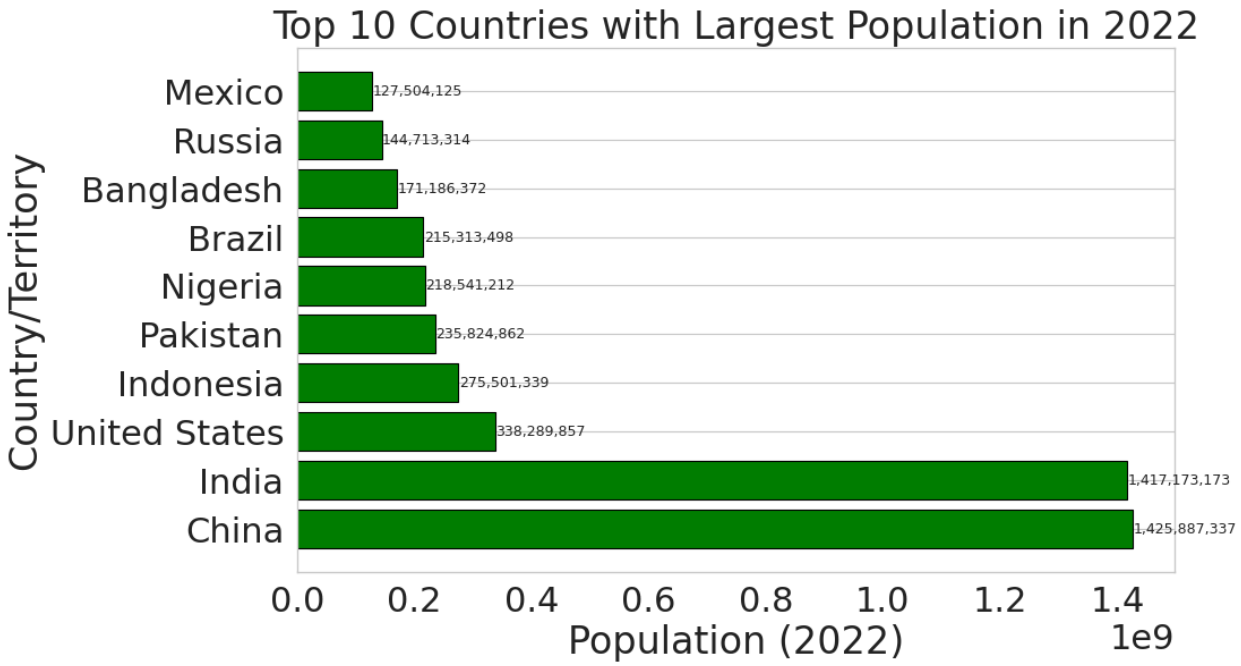
In [ ]:
```
sns.set_style('whitegrid')
plt.figure(figsize=(100,50))
sns.set_context('paper', font_scale=4.5)

plt.subplot(3,3,1)
sns.countplot(x='Continent', data = df).set_title('Continent')

plt.subplot(3,3,2)
df["Continent"].value_counts().plot.pie(autopct='%1.0f%%')
plt.legend(loc=(0.5,0.9))
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7963f1279d80>
```



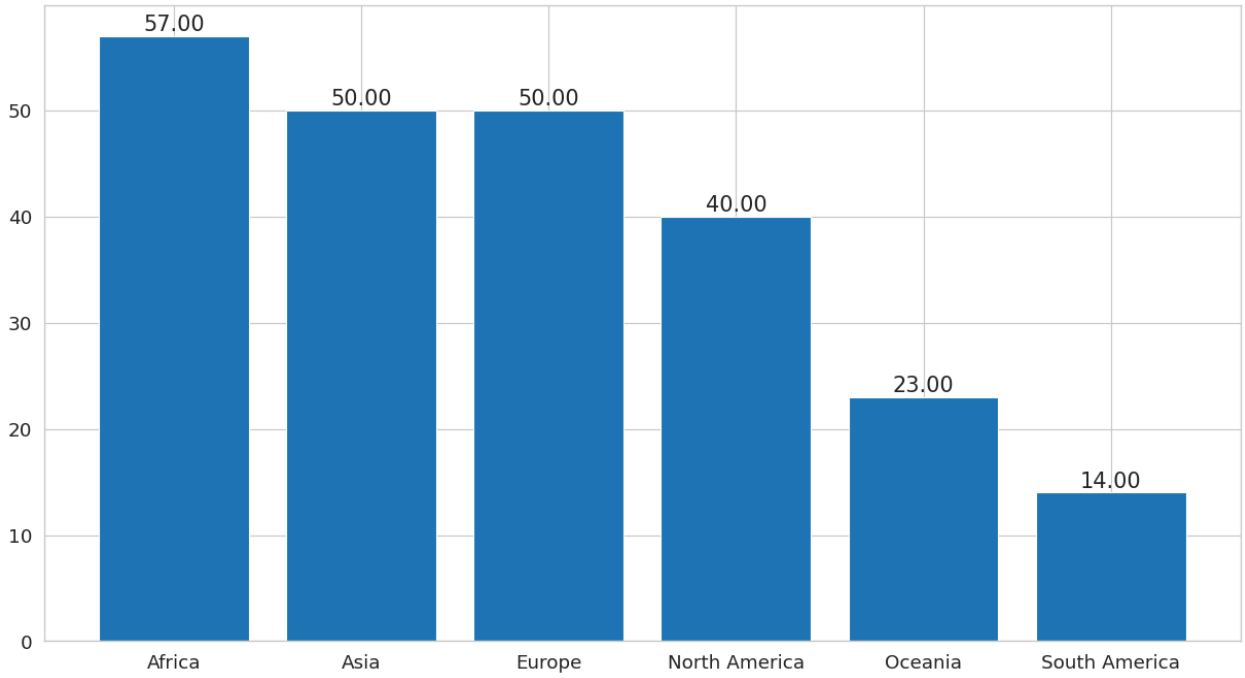


In [ ]:
```
# Plotting a horizontal bar chart to show the top 5 countries with the largest
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=2.5)


top_5_population = df.nlargest(10, '2022 Population')

plt.figure(figsize=(10, 6))
bars = plt.barh(top_5_population['Country/Territory'], top_5_population['2022

plt.xlabel('Population (2022)')
plt.ylabel('Country/Territory')
plt.title('Top 10 Countries with Largest Population in 2022')
plt.grid(axis='x')

# Adding data labels on the bars
for bar in bars:
    width = bar.get_width()
    plt.text(width + 1000000, bar.get_y() + bar.get_height() / 2, f'{width:,}'

plt.show()
```



In [ ]:
```
sns.set_style('whitegrid')
plt.figure(figsize=(100,50))
sns.set_context('paper', font_scale=1.5)


country_count=df["Continent"].value_counts()
x=country_count.index
y=country_count.values

fig, ax = plt.subplots(figsize =(15,8))
ax.bar(x, y)
for i in ax.patches:
    ax.annotate(format(i.get_height(), '.2f'),
                   (i.get_x() + i.get_width() / 2,
                    i.get_height()), ha='center', va='center',
                   size=15, xytext=(0,8),
                   textcoords='offset points')
plt.show()
```

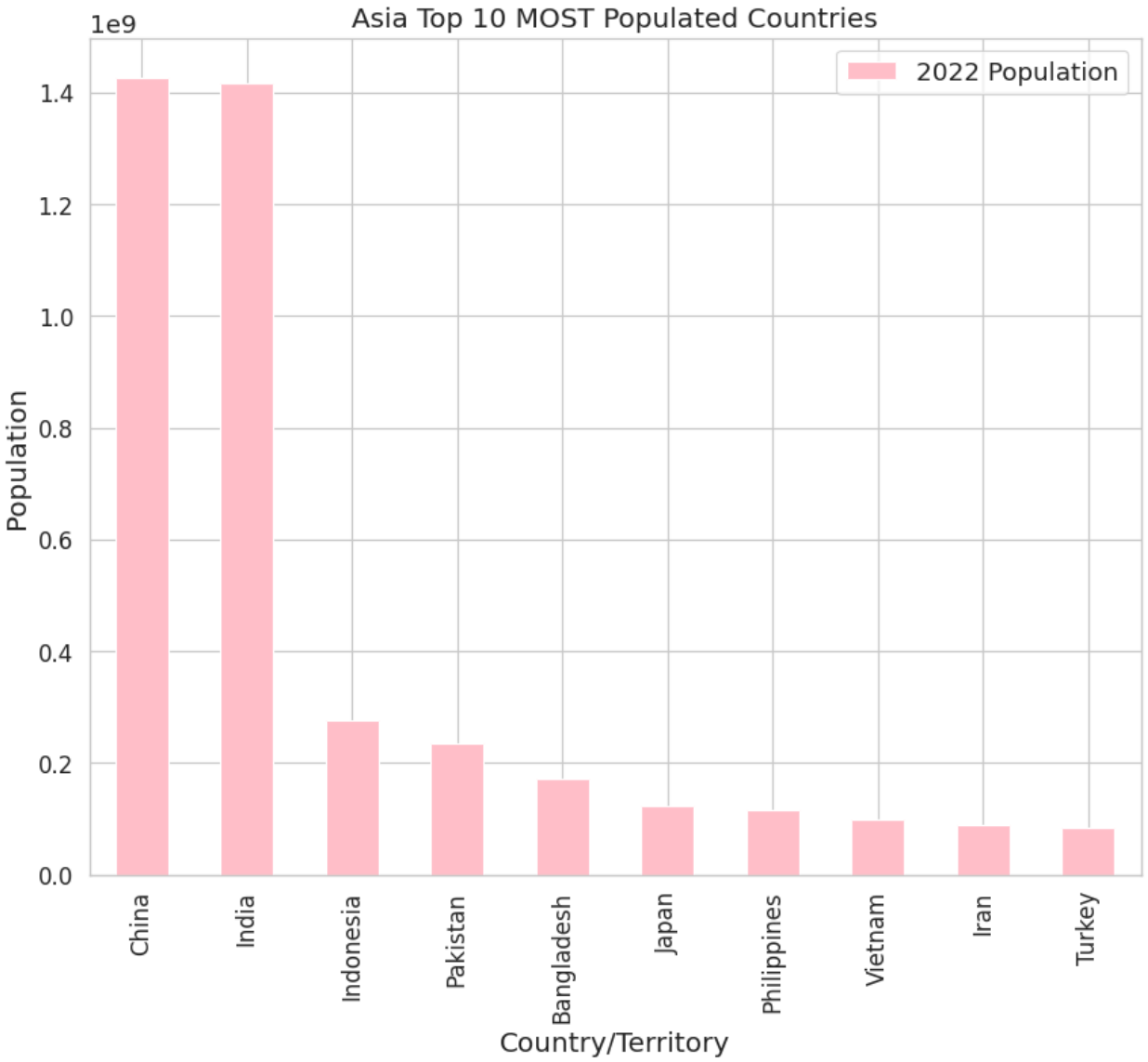
```
<Figure size 10000x5000 with 0 Axes>
```



plotting top 10 most populated countries by continent

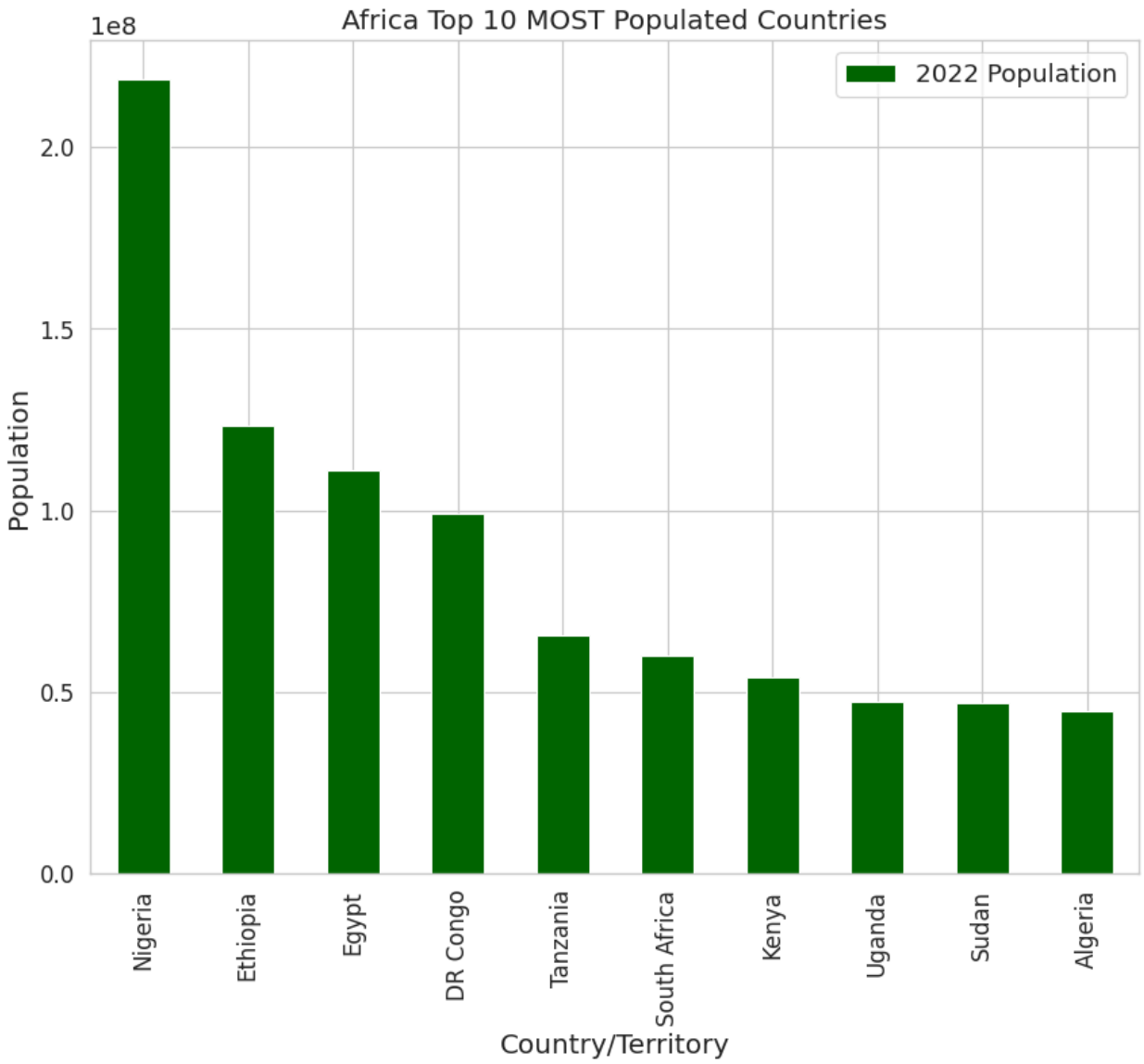
In [ ]:
```
# Asia
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)
asian_countries = df.loc[df["Continent"]=="Asia"].sort_values(by=["2022 Popula
asian_countries[["Country/Territory", "2022 Population"]].sort_values(by="2022
```

Out[ ]:
```
<Axes: title={'center': 'Asia Top 10 MOST Populated Countries'}, xlabel='Coun
try/Territory', ylabel='Population'>
```



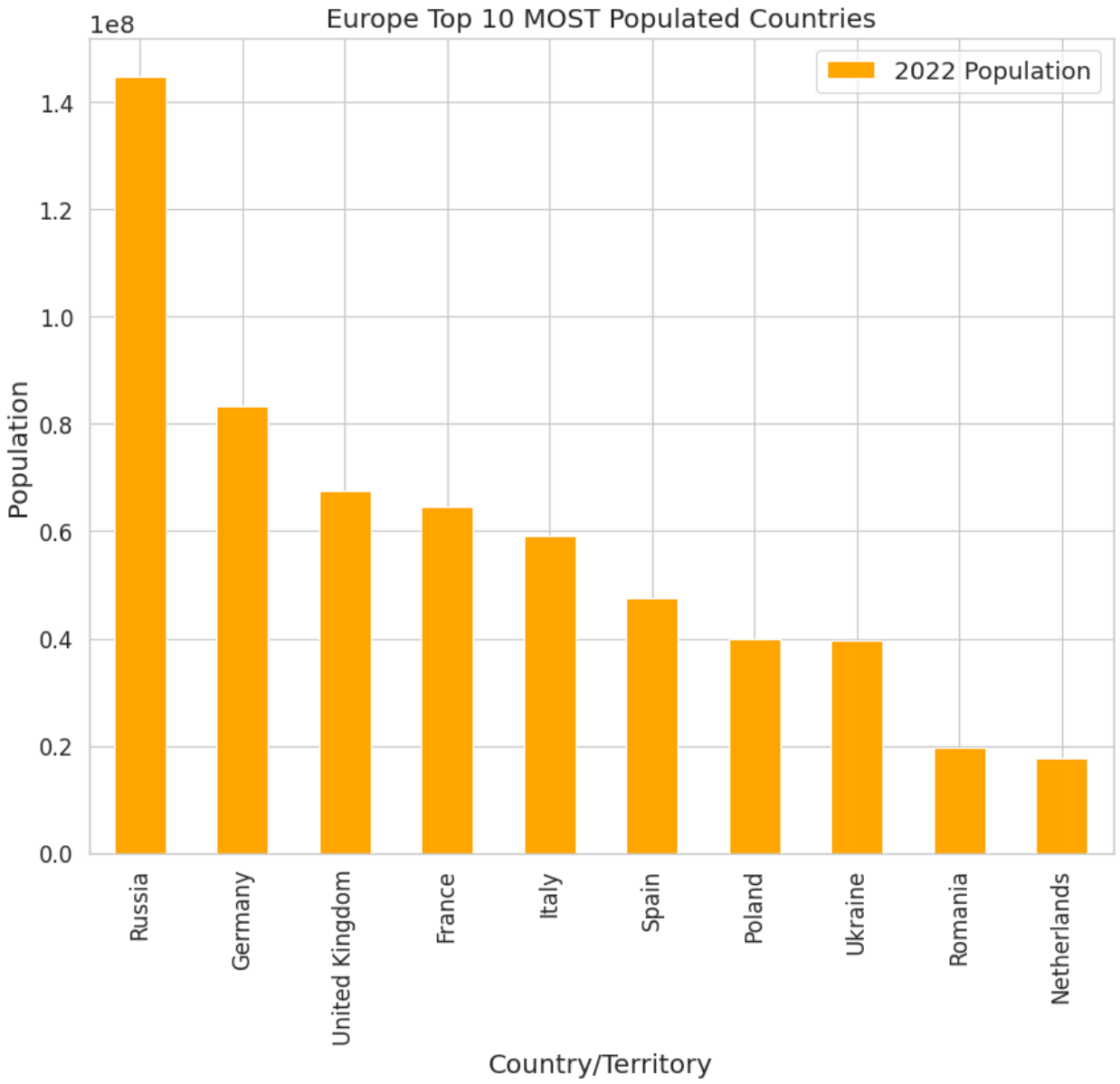
In [ ]:
```
# Africa
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)
african_countries = df.loc[df["Continent"]=="Africa"].sort_values(by=["2022 Po
african_countries[["Country/Territory", "2022 Population"]].sort_values(by="20
```

Out[ ]:
```
<Axes: title={'center': 'Africa Top 10 MOST Populated Countries'}, xlabel='Co
untry/Territory', ylabel='Population'>
```



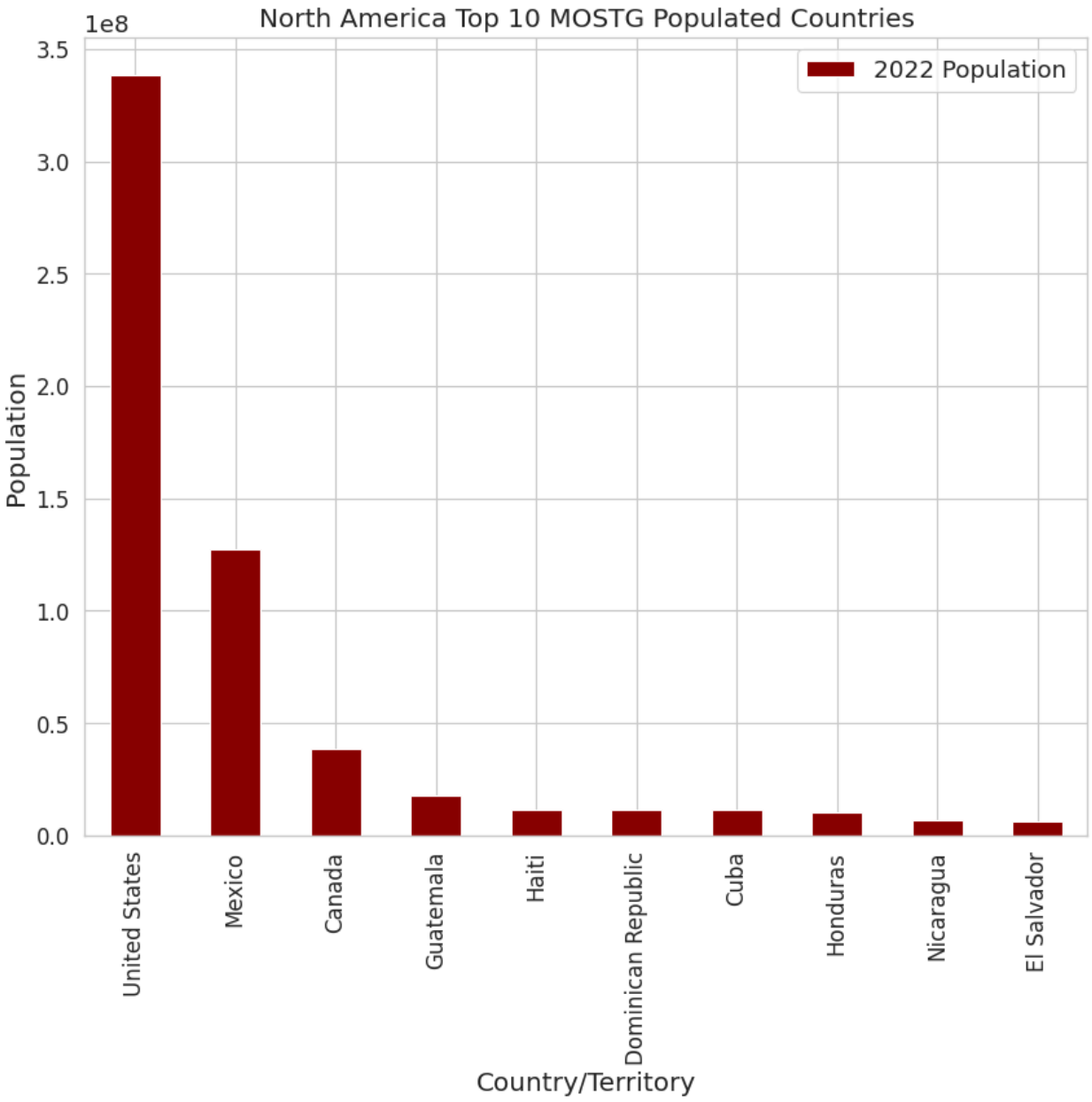
In [ ]:
```
# Europe
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)
european_countries = df.loc[df["Continent"]=="Europe"].sort_values(by=["2022 P
european_countries[["Country/Territory", "2022 Population"]].sort_values(by="2
```

Out[ ]:
```
<Axes: title={'center': 'Europe Top 10 MOST Populated Countries'}, xlabel='Co
untry/Territory', ylabel='Population'>
```



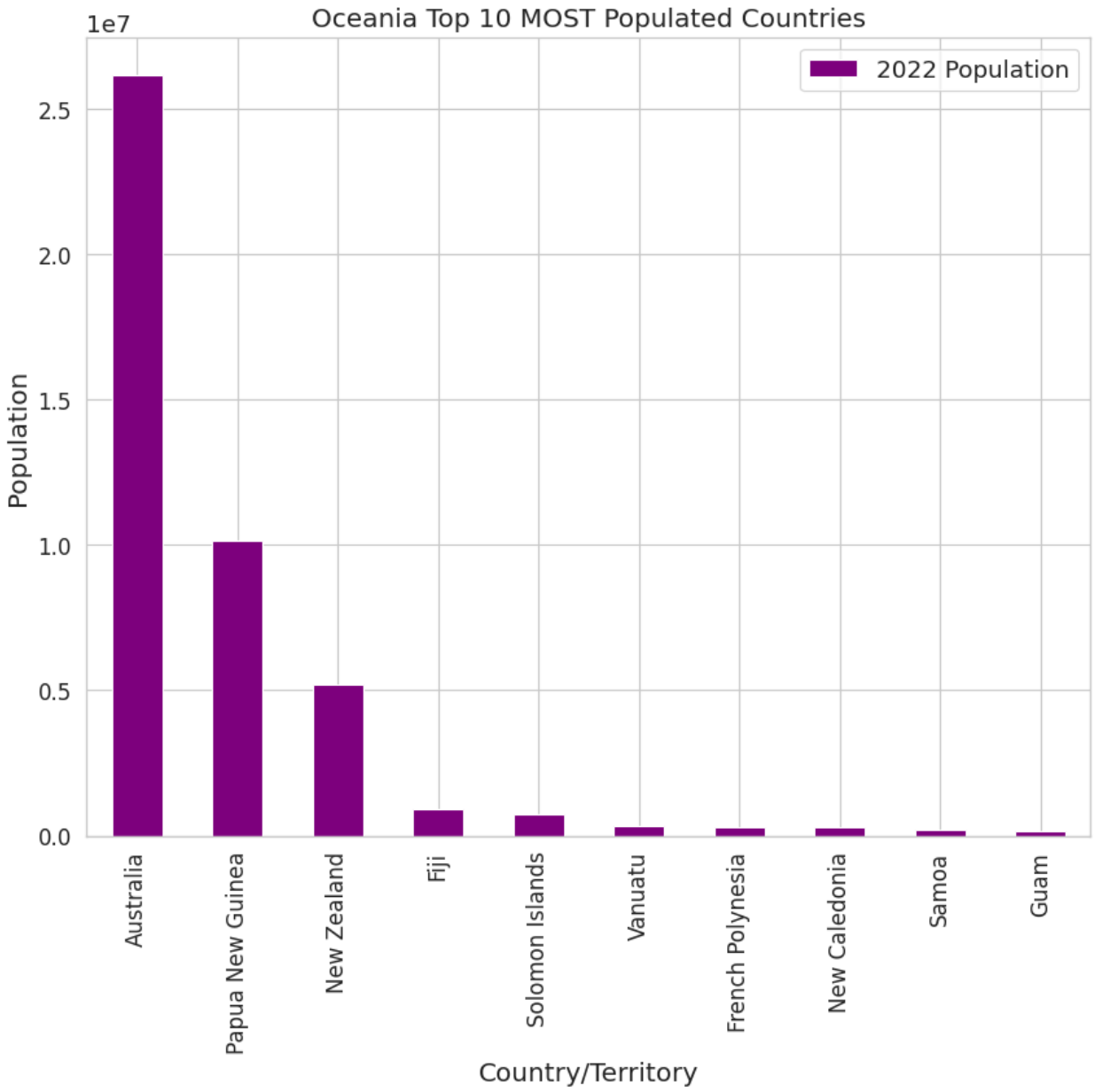
In [ ]:
```
# North America
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)
na_countries = df.loc[df["Continent"]=="North America"].sort_values(by=["2022
na_countries[["Country/Territory", "2022 Population"]].sort_values(by="2022 Pc
```

Out[ ]:
```
<Axes: title={'center': 'North America Top 10 MOSTG Populated Countries'}, xl
abel='Country/Territory', ylabel='Population'>
```



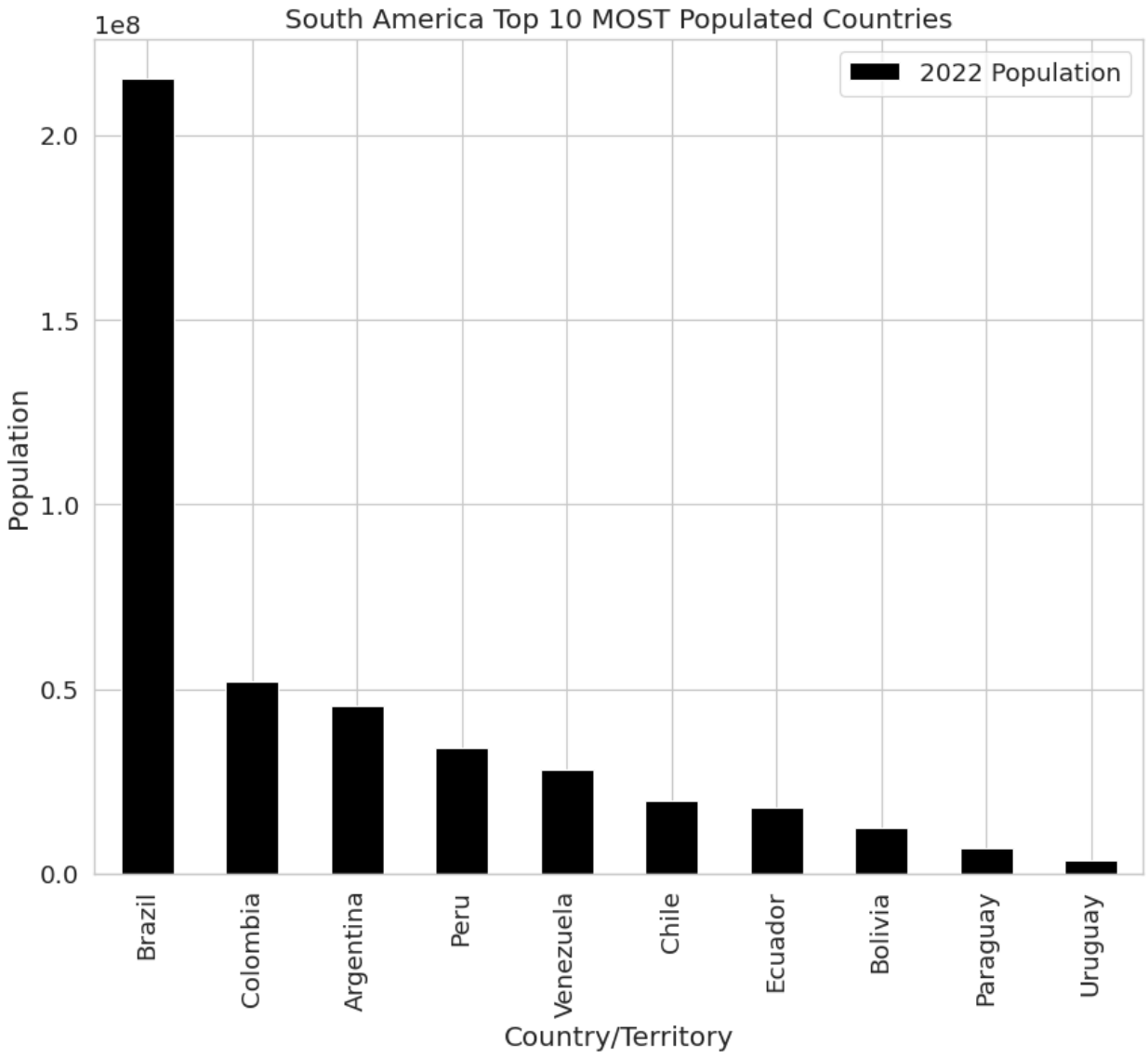
In [ ]:
```
# Oceania
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)
oc_countries = df.loc[df["Continent"]=="Oceania"].sort_values(by=["2022 Popula
oc_countries[["Country/Territory", "2022 Population"]].sort_values(by="2022 Po
```

Out[ ]:
```
<Axes: title={'center': 'Oceania Top 10 MOST Populated Countries'}, xlabel='C
ountry/Territory', ylabel='Population'>
```



In [ ]:
```
# South America
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)
sa_countries = df.loc[df["Continent"]=="South America"].sort_values(by=["2022
sa_countries[["Country/Territory", "2022 Population"]].sort_values(by="2022 Pc
```

Out[ ]:
```
<Axes: title={'center': 'South America Top 10 MOST Populated Countries'}, xla
bel='Country/Territory', ylabel='Population'>
```

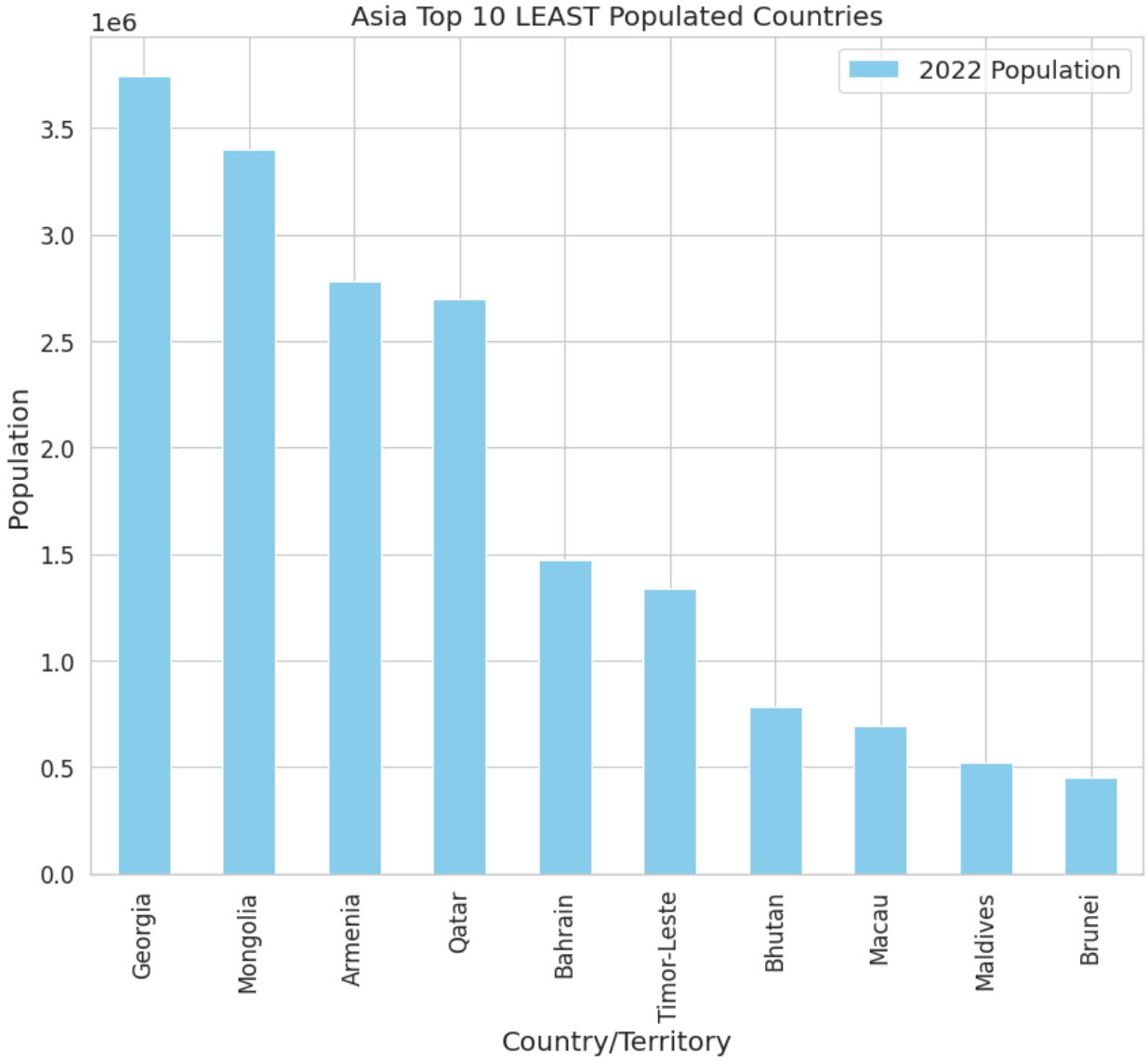


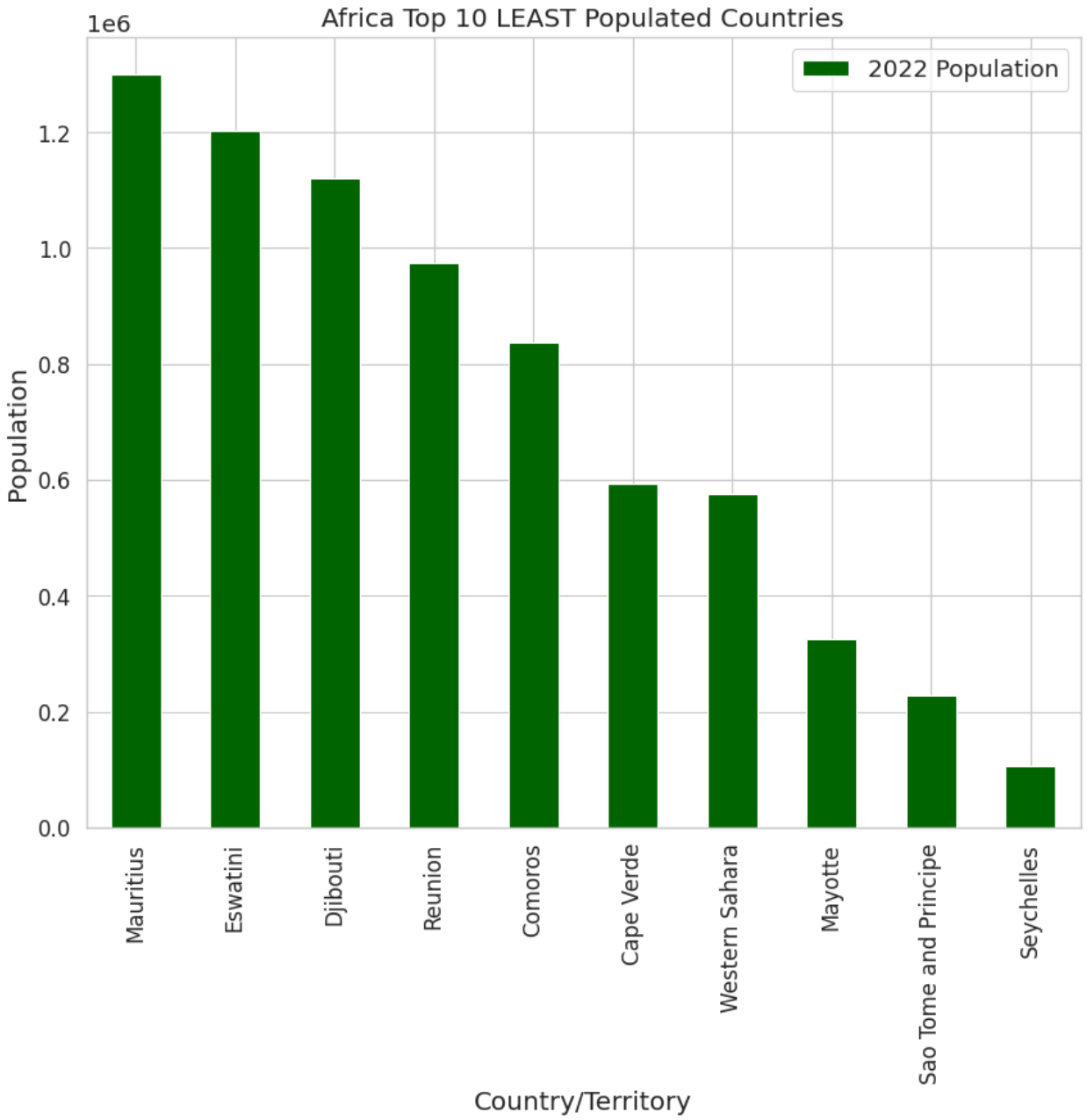
Top 10 Least Populated Countries By Continents

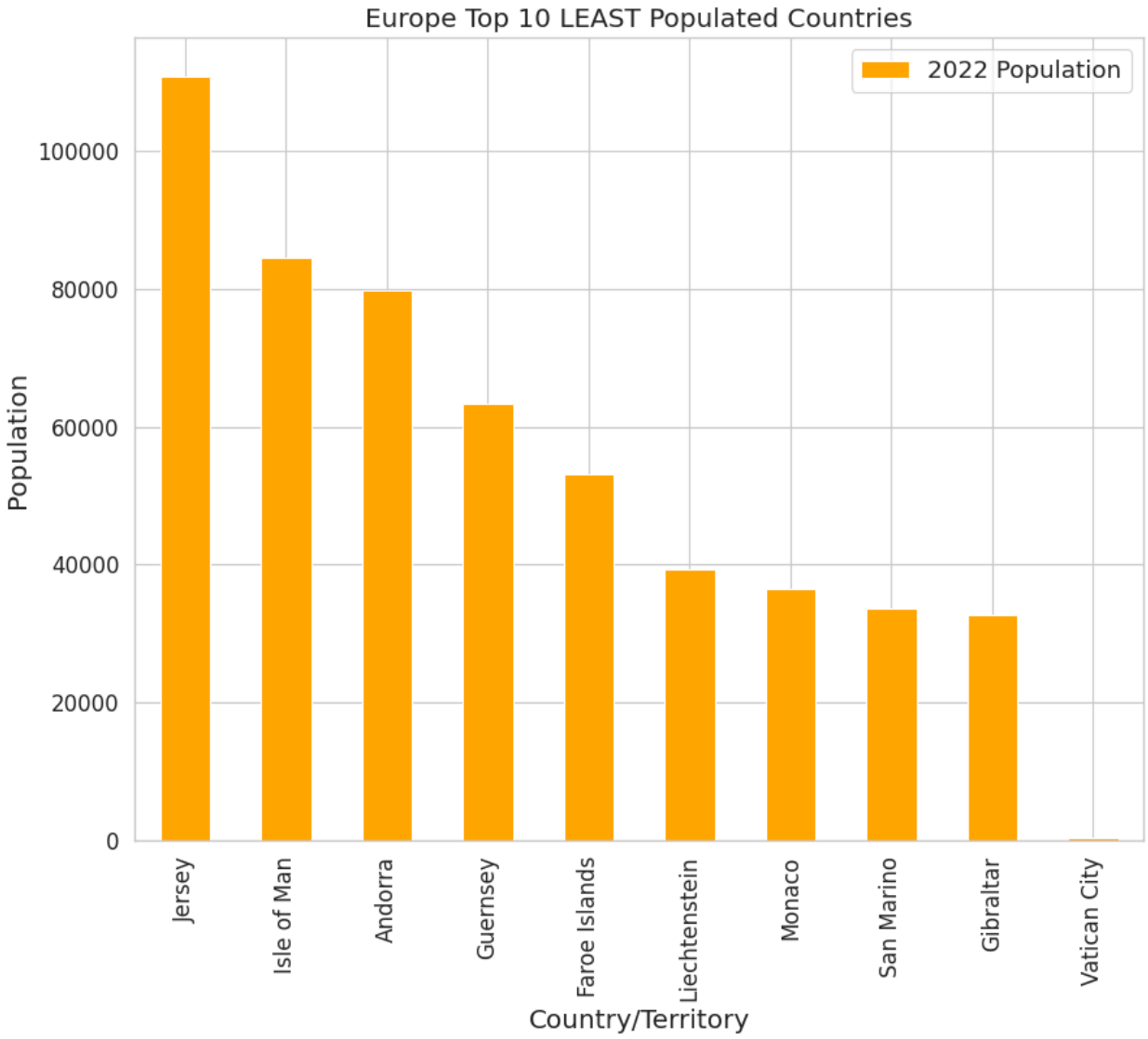
In [ ]:
```
# plotting top 5 LEAST populated countries by continent
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)

# Asian countries
asian_countries[["Country/Territory", "2022 Population"]].sort_values(by="2022

# African countries
african_countries[["Country/Territory", "2022 Population"]].sort_values(by="20

# European countries
european_countries[["Country/Territory", "2022 Population"]].sort_values(by="2
```

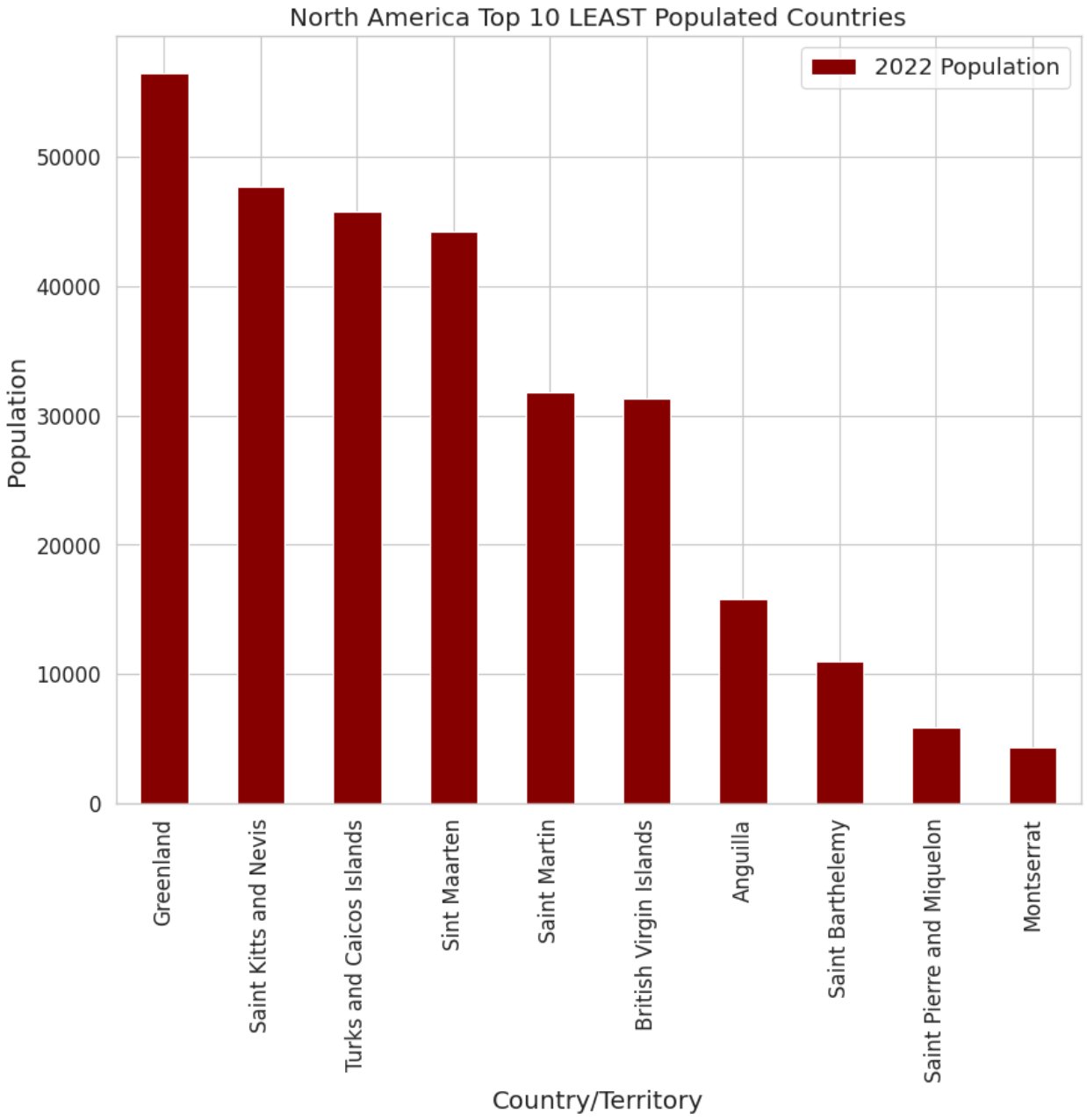
```
# North American countries
na_countries[["Country/Territory", "2022 Population"]].sort_values(by="2022 Pc

# Oceanian countries
oc_countries[["Country/Territory", "2022 Population"]].sort_values(by="2022 Pc

# South American countries
sa_countries[["Country/Territory", "2022 Population"]].sort_values(by="2022 Pc
```

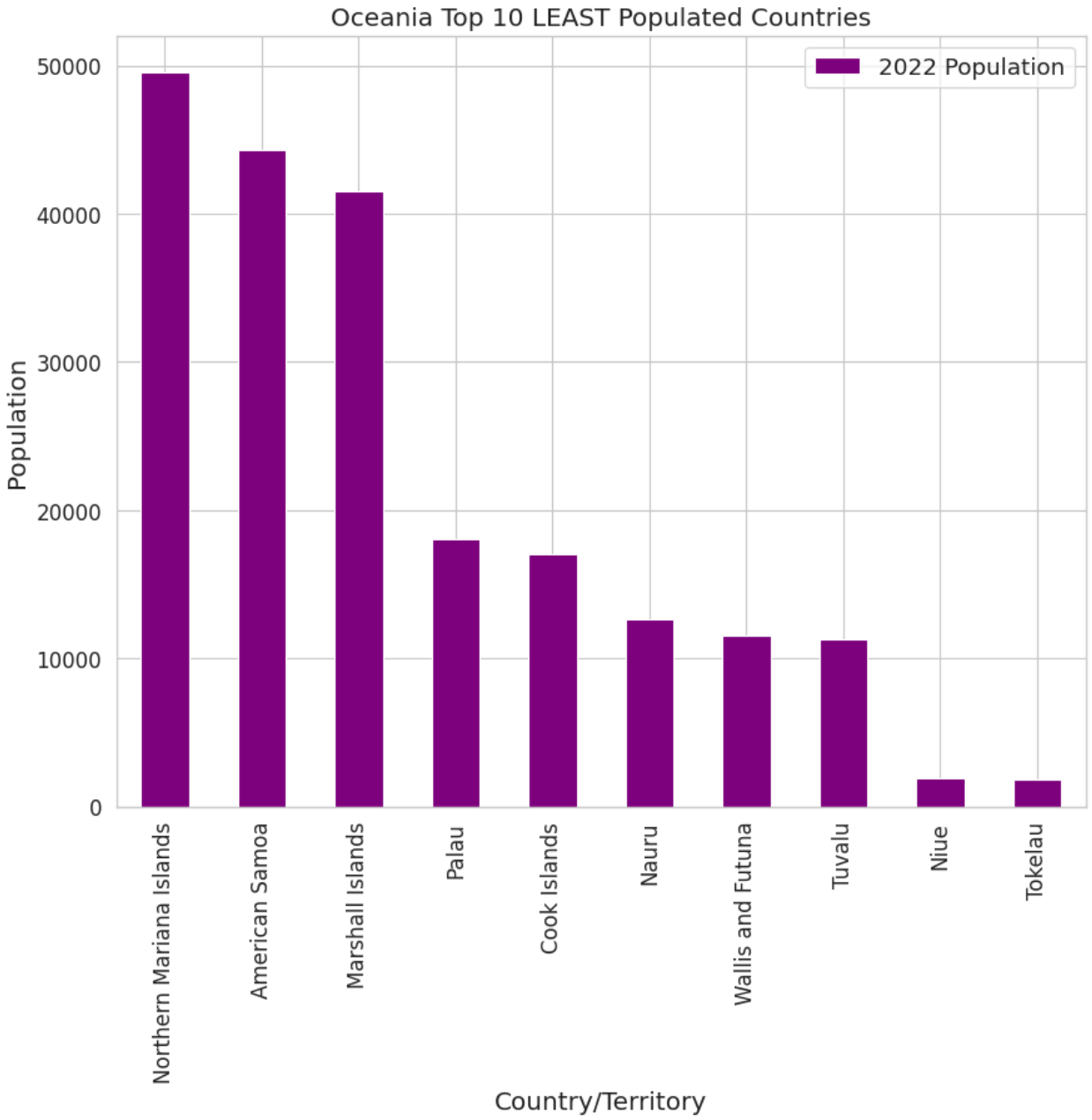
Out[ ]: <Axes: title={'center': 'South America Top 10 LEAST Populated Countries'}, xlabel='Country/Territory', ylabel='Population'>

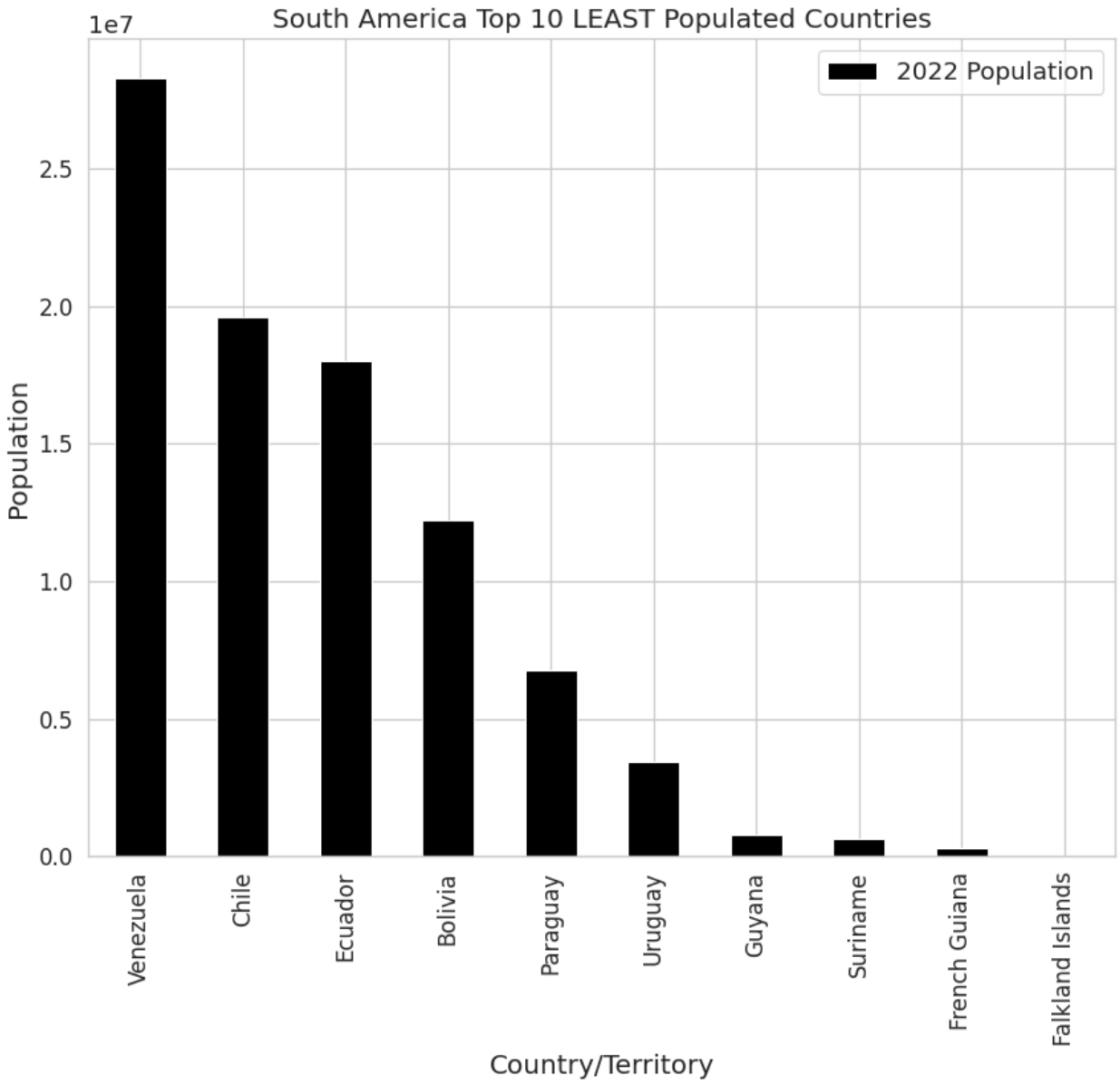



Africa Top 10 LEAST Populated Countries
1e6
2022 Population
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Population
Mauritius
Eswatini
Djibouti
Reunion
Comoros
Cape Verde
Western Sahara
Mayotte
Sao Tome and Principe
Seychelles
Country/Territory


Europe Top 10 LEAST Populated Countries
2022 Population
Population
100000
80000
60000
40000
20000
0
Jersey
Isle of Man
Andorra
Guernsey
Faroe Islands
Liechtenstein
Monaco
San Marino
Gibraltar
Vatican City
Country/Territory

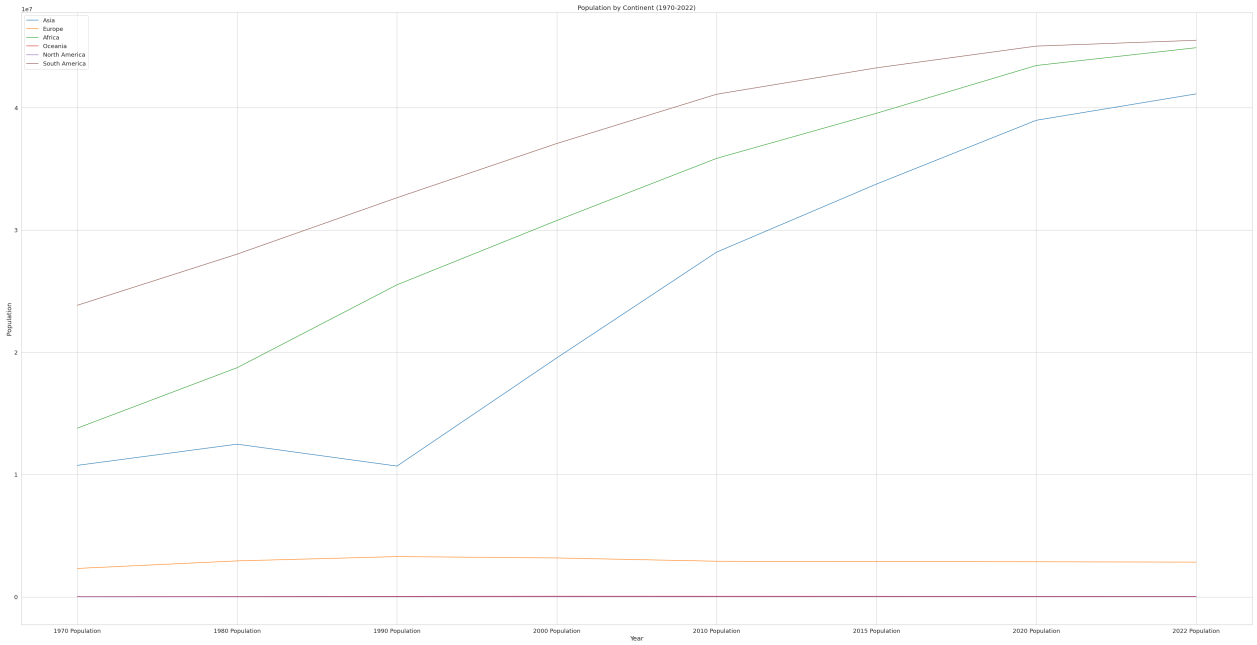

North America Top 10 LEAST Populated Countries
2022 Population
Population
50000
40000
30000
20000
10000
0
Greenland
Saint Kitts and Nevis
Turks and Caicos Islands
Sint Maarten
Saint Martin
British Virgin Islands
Anguilla
Saint Barthelemy
Saint Pierre and Miquelon
Montserrat
Country/Territory


Oceania Top 10 LEAST Populated Countries
2022 Population
Population
50000
40000
30000
20000
10000
0
Northern Mariana Islands
American Samoa
Marshall Islands
Palau
Cook Islands
Nauru
Wallis and Futuna
Tuvalu
Niue
Tokelau
Country/Territory



In [ ]:
```
# 'Population by Continent (1970-2022)'

data = {
    "Continent": df['Continent'],
    "1970 Population": df['1970 Population'],
    "1980 Population": df['1980 Population'],
    "1990 Population": df['1990 Population'],
    "2000 Population": df['2000 Population'],
    "2010 Population": df['2010 Population'],
    "2015 Population": df['2015 Population'],
    "2020 Population": df['2020 Population'],
    "2022 Population": df['2022 Population']
}

df_pop_by_continent = pd.DataFrame(data)

# Plotting
```

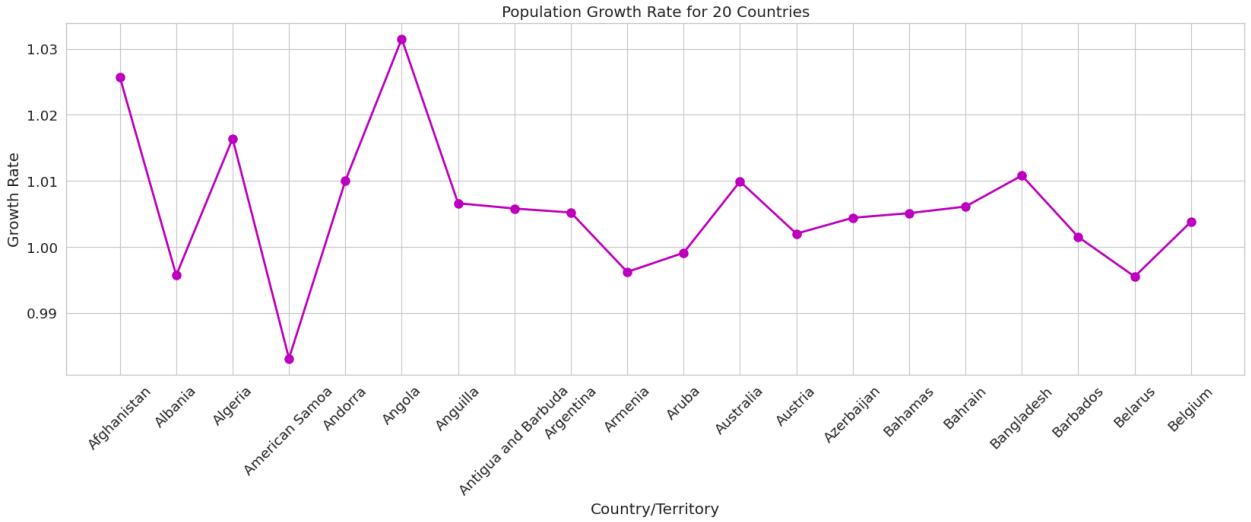
```
years = df_pop_by_continent.columns[1:]
plt.figure(figsize=(50,25))
for continent in df_pop_by_continent["Continent"].unique():
    continent_data = df_pop_by_continent[df_pop_by_continent["Continent"] == c
    plt.plot(years, continent_data, label=continent)

plt.title('Population by Continent (1970-2022)')
plt.xlabel('Year')
plt.ylabel('Population')
plt.legend()
plt.grid(True)
plt.show()
```



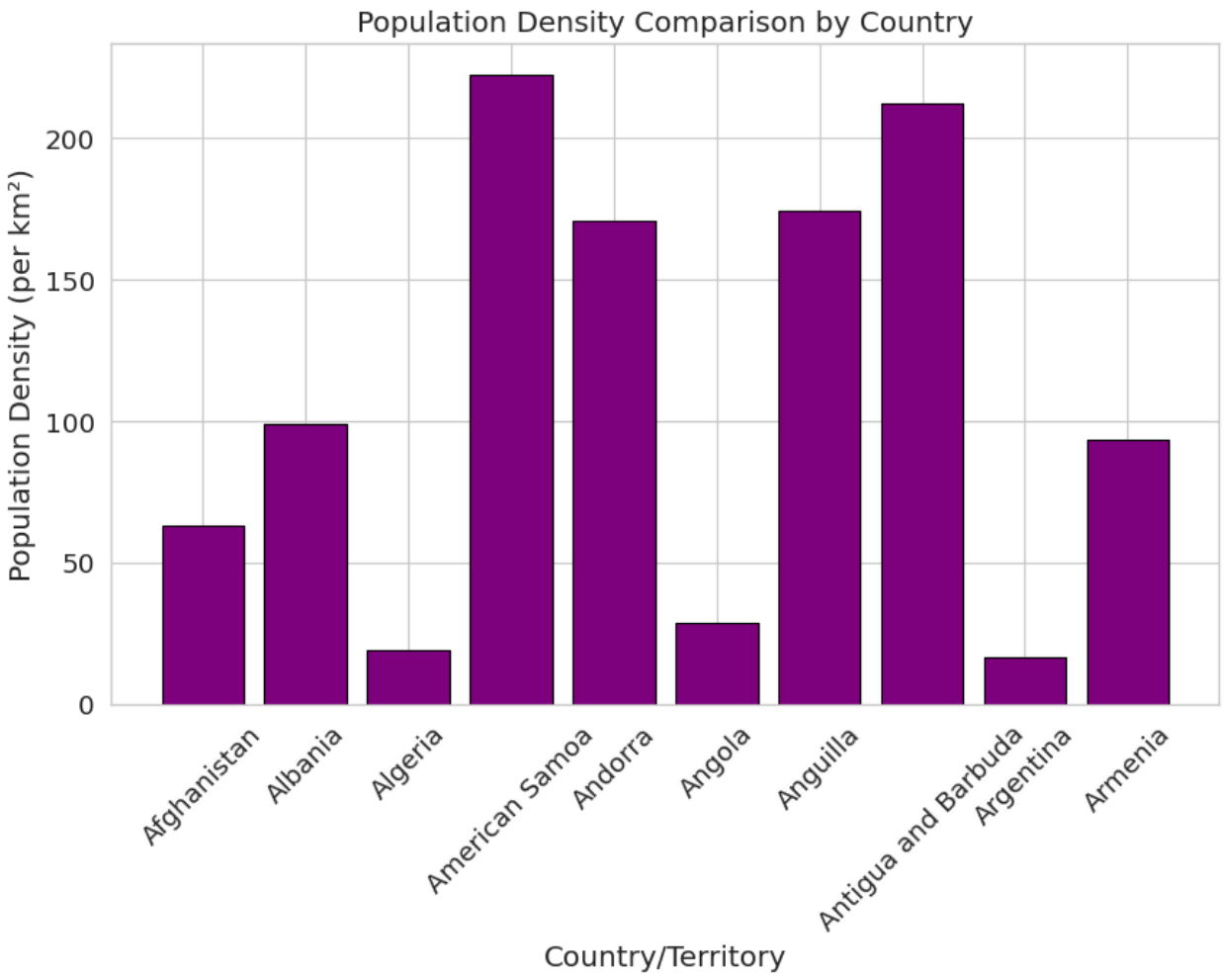
Population Growth Rate for 20 Countries

In [ ]:
```
# Plot a line chart of the population growth rate for individual countries to
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)


x = df['Country/Territory'].head(20)
y = df['Growth Rate'].head(20)
plt.figure(figsize=(20, 6))
plt.plot(x, y, marker='o', markersize=8, color='m', linewidth=2)
plt.xlabel('Country/Territory')
plt.ylabel('Growth Rate')
plt.title('Population Growth Rate for 20 Countries')
plt.xticks(rotation=45)
plt.grid(True)
```

```
plt.show()
```



In [ ]:
```
# Visualize the top 10 most densely populated countries on a world map
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)

x = df['Country/Territory'].head(10)
y = df['Density (per km²)'].head(10)
plt.figure(figsize=(10, 6))
plt.bar(x, y, color='purple', edgecolor='black')
plt.xlabel('Country/Territory')
plt.ylabel('Population Density (per km²)')
plt.title('Population Density Comparison by Country')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

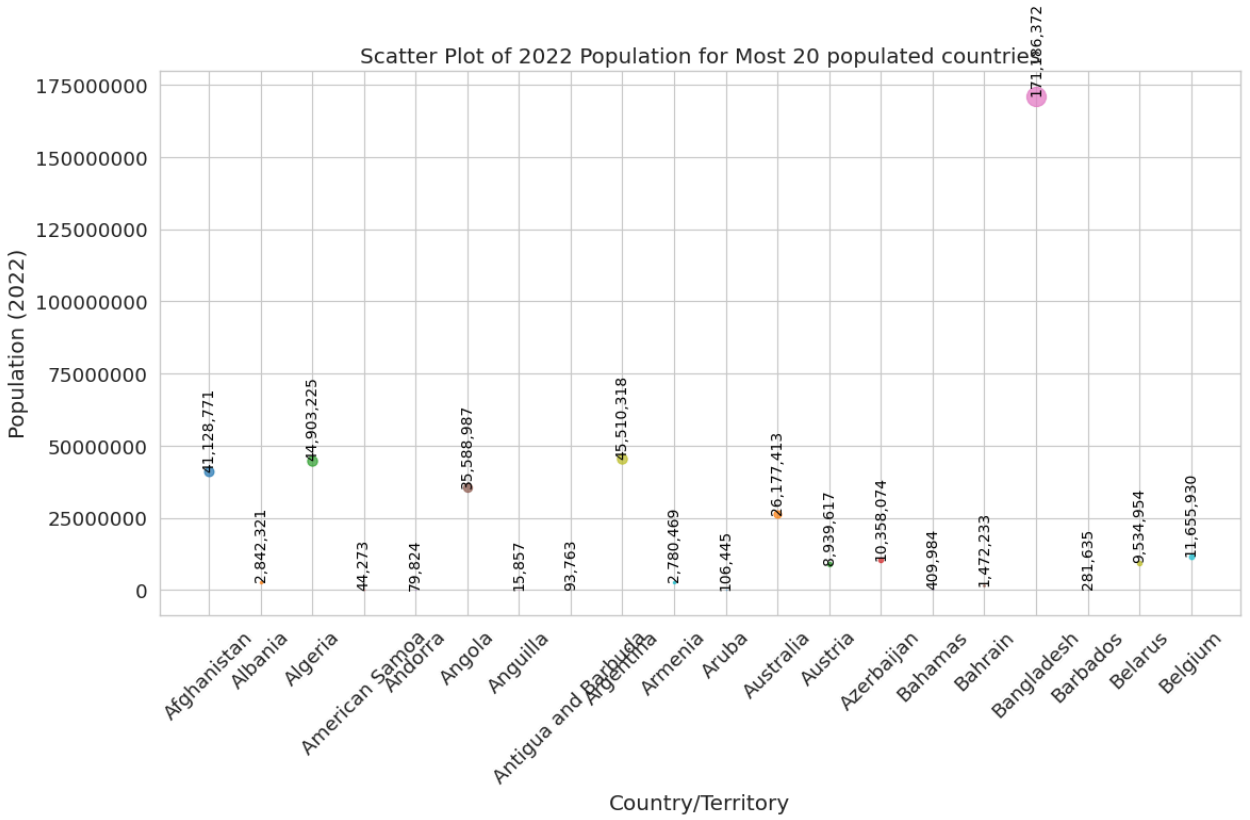


In [ ]:
```python
import plotly.express as px

fig = px.choropleth(df, locations="Country/Territory", color="Density (per km²
                    locationmode='country names',
                    range_color=[0,300],
                    color_continuous_scale=[(0, "#F0F0F0"), (1, '#FFD700')],
                    template='seaborn'
                   )

fig.update_layout(
    title="World Map Visualization from Density",
    font=dict(
        family="Monospace",
        size=14
    )
)

fig.show()
```

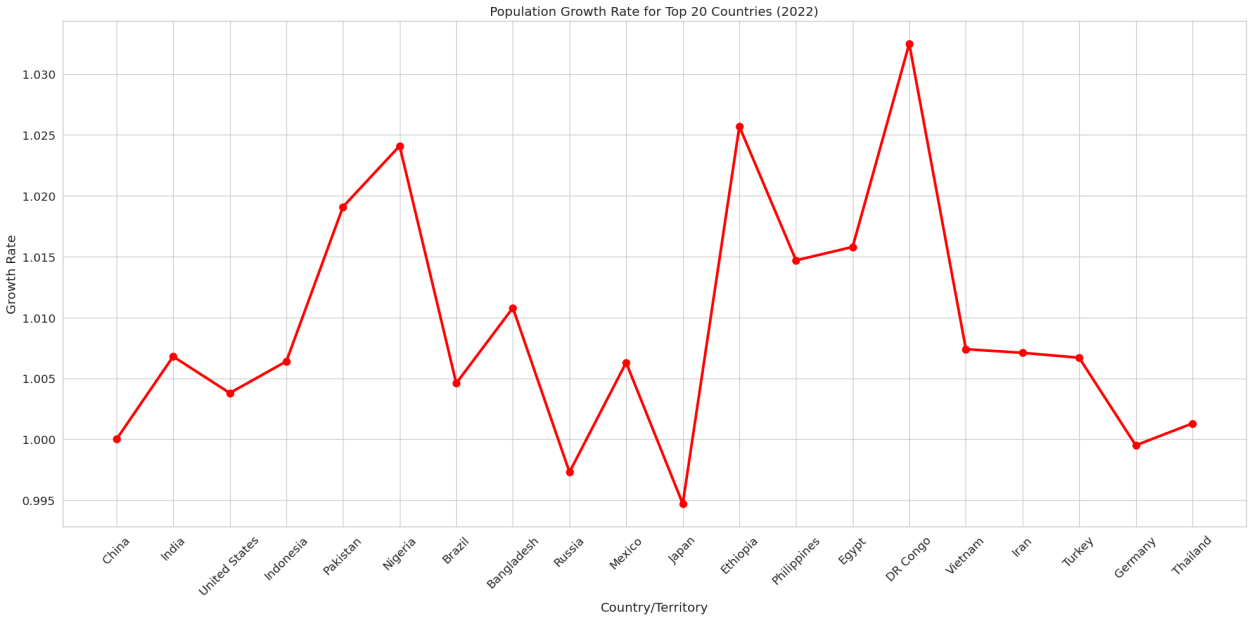
In [ ]:

```
plt.figure(figsize=(12, 8))

x = df['Country/Territory'].head(20)
y = df['2022 Population'].head(20)

# Plot the scatter plot with country names and numbers on y-axis
marker_sizes = df['2022 Population'] / 50000
for i, country in enumerate(x):
    plt.scatter(country, y.iloc[i], s=(marker_sizes.iloc[i])/20, label=country
    plt.text(country, y.iloc[i], f'{y.iloc[i]:,.0f}', ha='center', va='bottom'

# Set y-axis to display numbers in billions
plt.ticklabel_format(style='plain', axis='y', useOffset=False, scilimits=(9, 9

plt.xlabel('Country/Territory')
plt.ylabel('Population (2022)')
plt.title('Scatter Plot of 2022 Population for Most 20 populated countries')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
```

```
plt.show()
```



In [ ]:
```
plt.figure(figsize=(20, 10))

N = 20
top_countries = df.sort_values(by='2022 Population', ascending=False).head(N)

plt.plot(top_countries['Country/Territory'], top_countries['Growth Rate'], mar
plt.xlabel('Country/Territory')
plt.ylabel('Growth Rate')
plt.title(f'Population Growth Rate for Top {N} Countries (2022)')
plt.xticks(rotation=45)
plt.grid(True)

plt.tight_layout()
plt.show()
```

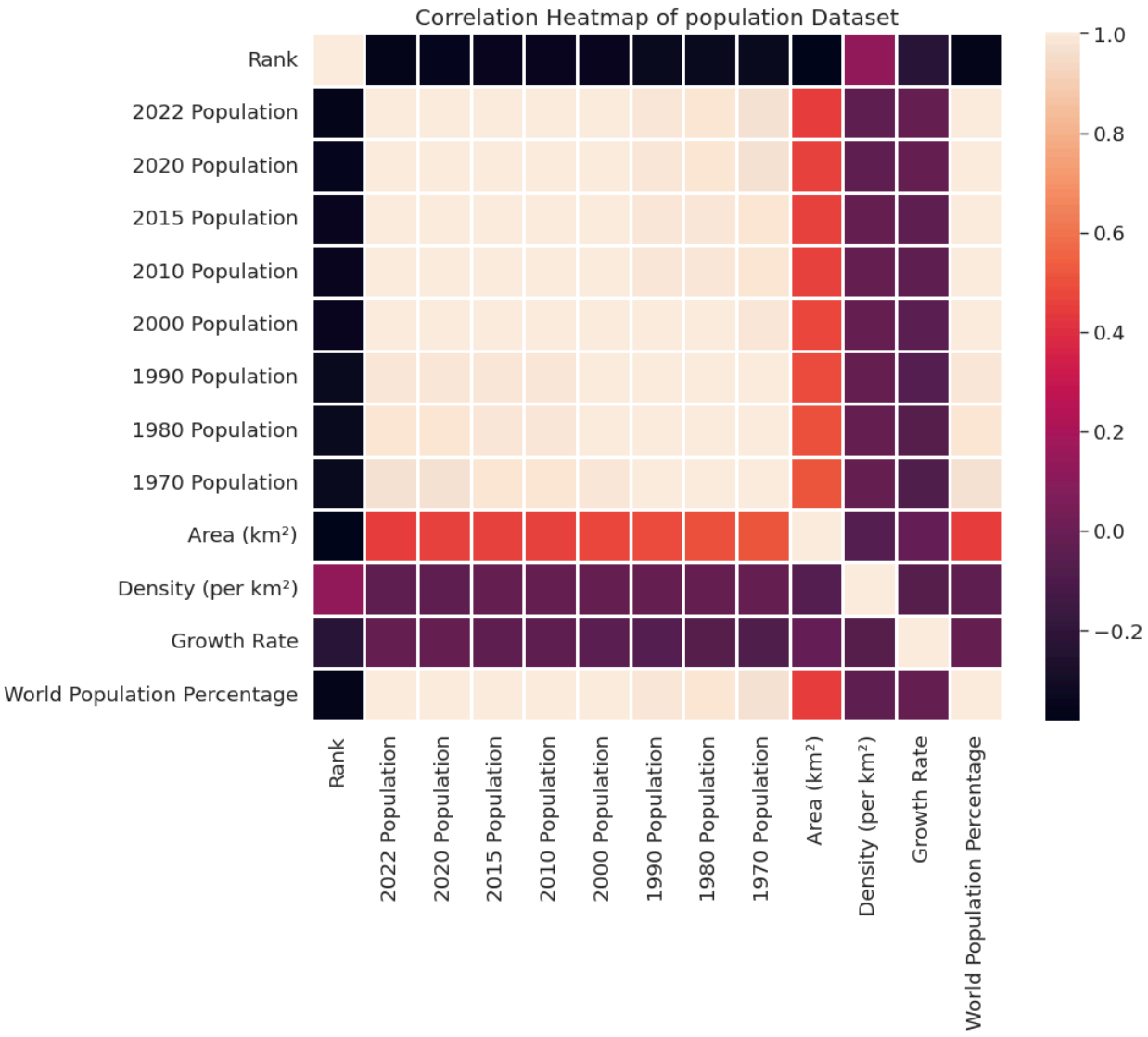


Data Correlations

In [ ]:
```
sns.set_style('whitegrid')
plt.figure(figsize=(10,8))
sns.set_context('paper', font_scale=1.5)

correlation_matrix=df.corr()
sns.heatmap(correlation_matrix,linewidth=1)
plt.title('Correlation Heatmap of population Dataset')
```

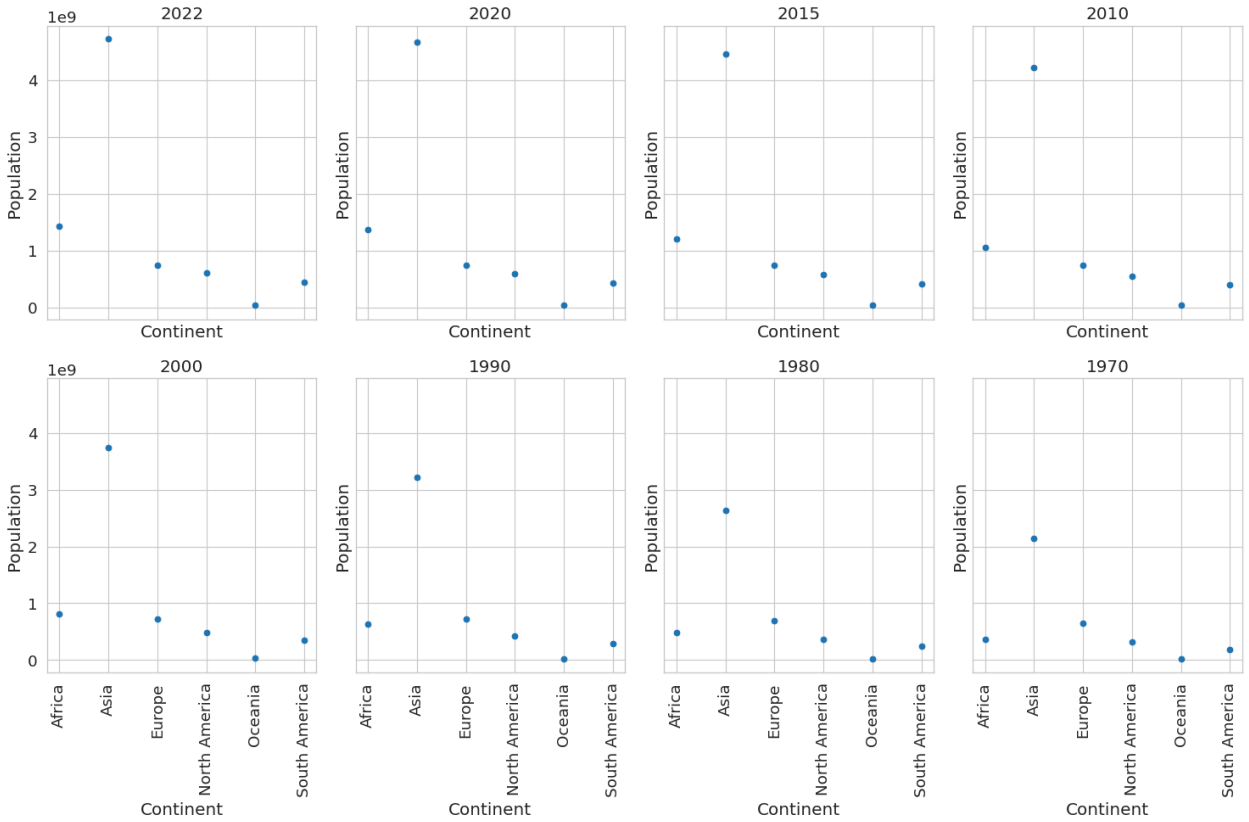
```
<ipython-input-20-02319782aec5>:5: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the val
ue of numeric_only to silence this warning.
```

Out[ ]: Text(0.5, 1.0, 'Correlation Heatmap of population Dataset')



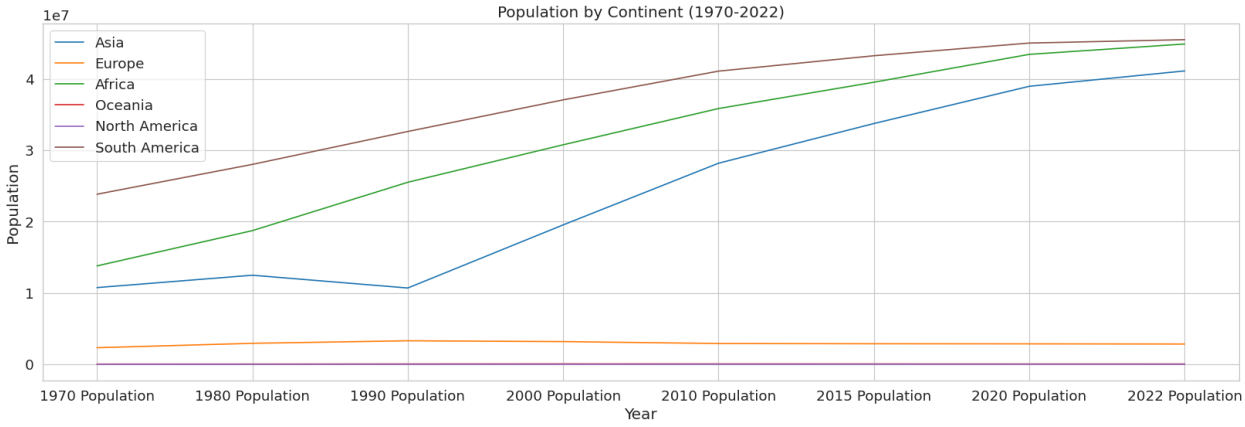
In [ ]:
```python
# Group the data by continent and calculate the sum of population for each yea
grouped_data = df.groupby('Continent').agg({
    '2022 Population': 'sum',
    '2020 Population': 'sum',
    '2015 Population': 'sum',
    '2010 Population': 'sum',
    '2000 Population': 'sum',
    '1990 Population': 'sum',
    '1980 Population': 'sum',
    '1970 Population': 'sum'
}).reset_index()

years = ['2022', '2020', '2015', '2010', '2000', '1990', '1980', '1970']

fig, axes = plt.subplots(2, 4, figsize=(15, 10), sharex=True, sharey=True)
axes = axes.ravel()

# Loop through the years and create a scatter plot for each continent
for i, year in enumerate(years):
```

```
    axes[i].scatter(grouped_data['Continent'], grouped_data[year + ' Populatic
    axes[i].set_title(year)
    axes[i].set_xlabel('Continent')
    axes[i].set_ylabel('Population')
    axes[i].set_xticks(range(len(grouped_data['Continent'])))
    axes[i].set_xticklabels(grouped_data['Continent'], rotation=90)

plt.tight_layout()
plt.show()
```



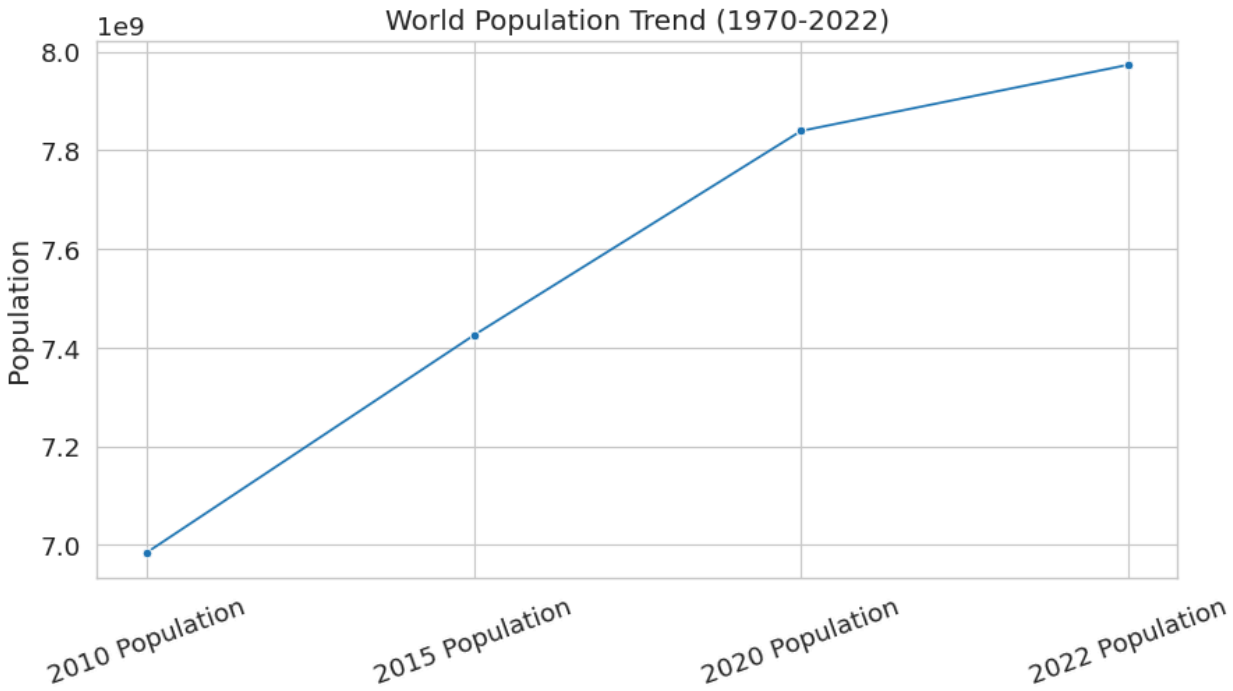
In [ ]:
```
# 'Population by Continent (1970-2022)'

data = {
    "Continent": df['Continent'],
    "1970 Population": df['1970 Population'],
    "1980 Population": df['1980 Population'],
    "1990 Population": df['1990 Population'],
    "2000 Population": df['2000 Population'],
    "2010 Population": df['2010 Population'],
    "2015 Population": df['2015 Population'],
    "2020 Population": df['2020 Population'],
    "2022 Population": df['2022 Population']
}

df = pd.DataFrame(data)

# Plotting
years = df.columns[1:]  # Extract years from columns
```

```
plt.figure(figsize=(20, 6))
for continent in df["Continent"].unique():
    continent_data = df[df["Continent"] == continent].iloc[0, 1:]
    plt.plot(years, continent_data, label=continent)

plt.title('Population by Continent (1970-2022)')
plt.xlabel('Year')
plt.ylabel('Population')
plt.legend()
plt.grid(True)
plt.show()
```



In [ ]:

```
sns.set_style('whitegrid')
sns.set_context('paper', font_scale=1.5)

plt.subplots(figsize=(10,5))
trend = df.iloc[:,5:13].sum().sort_values(ascending=True)

sns.lineplot(x=trend.index, y=trend.values, marker="o")
plt.xticks(rotation=20)
plt.ylabel("Population")
plt.title("World Population Trend (1970-2022)")
plt.show()
```


World Population Trend (1970-2022)
1e9
8.0
7.8
7.6
7.4
7.2
7.0
Population
2010 Population
2015 Population
2020 Population
2022 Population