World Population Analysis Using Python

Overview: This Jupyter notebook presents an in-depth analysis of the "World Population Analysis" dataset. The dataset contains information on countries, their capitals, continents, and population figures for the years 2010, 2020, and more. The analysis aims to explore population trends, growth rates, and distributions across continents.

Project Process: Data Loading: The project begins with loading the dataset into the notebook using Pandas, allowing for further exploration and analysis.

Data Cleaning: The dataset undergoes thorough data cleaning processes to handle missing values, inconsistencies, and formatting issues. This ensures data integrity and accuracy for subsequent analysis.

Exploratory Data Analysis (EDA): EDA is performed to gain insights into various aspects of the population dataset.

Import Library

In [2]:
```
import pandas as pd
```

In [3]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\Syed Arif\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWar
ning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of Sc
iPy (detected version 1.25.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Uploading Csv fle

In [4]:
```
df = pd.read_csv(r"C:\Users\Syed Arif\Desktop\World Population Analysis\world_p
```

Data Preprocessing

.head()

head is used show to the By default = 5 rows in the dataset

In [5]:
```
df.head()
```

Out[5]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	2 Popula
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	33753499	28189
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	2882481	2913
2	34	DZA	Algeria	Algiers	Africa	44903225	43451666	39543154	35856
3	213	ASM	American Samoa	Pago Pago	Oceania	44273	46189	51368	54
4	203	AND	Andorra	Andorra la Vella	Europe	79824	77700	71746	7

.tail()

tail is used to show rows by Descending order

In [6]:
```
df.tail()
```

Out[6]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	Popu
229	226	WLF	Wallis and Futuna	Mata-Utu	Oceania	11572	11655	12182	
230	172	ESH	Western Sahara	El Aaiún	Africa	575986	556048	491824	4
231	46	YEM	Yemen	Sanaa	Asia	33696614	32284046	28516545	247
232	63	ZMB	Zambia	Lusaka	Africa	20017675	18927715	16248230	137
233	74	ZWE	Zimbabwe	Harare	Africa	16320537	15669666	14154937	128

.shape

It show the total no of rows & Column in the dataset

In [7]:
```
df.shape
```

Out[7]:
```
(234, 17)
```

.Columns

It show the no of each Column

In [8]:
```
df.columns
```

Out[8]:
```
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')
```

.dtypes

This Attribute show the data type of each column

In [9]:
```
df.dtypes
```

Out[9]:
```
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
```

.unique()

In a column, It show the unique value of specific column.

In [10]:
```
df["Continent"].unique()
```

Out[10]:
```
array(['Asia', 'Europe', 'Africa', 'Oceania', 'North America',
       'South America'], dtype=object)
```

.nuique()

It will show the total no of unque value from whole data frame

In [11]:
```
df.nunique()
```

Out[11]:
```
Rank                           234
CCA3                           234
Country/Territory              234
Capital                        234
Continent                        6
2022 Population                234
2020 Population                234
2015 Population                234
2010 Population                234
2000 Population                234
1990 Population                234
1980 Population                234
1970 Population                234
Area (km²)                     233
Density (per km²)              234
Growth Rate                    180
World Population Percentage     70
dtype: int64
```

.describe()

It show the Count, mean , median etc

In [12]:
```
df.describe()
```

Out[12]:

	Rank	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	Po
count	234.000000	2.340000e+02	2.340000e+02	2.340000e+02	2.340000e+02	2.340000e+02	2.340
mean	117.500000	3.407441e+07	3.350107e+07	3.172996e+07	2.984524e+07	2.626947e+07	2.271
std	67.694165	1.367664e+08	1.355899e+08	1.304050e+08	1.242185e+08	1.116982e+08	9.783
min	1.000000	5.100000e+02	5.200000e+02	5.640000e+02	5.960000e+02	6.510000e+02	7.000
25%	59.250000	4.197385e+05	4.152845e+05	4.046760e+05	3.931490e+05	3.272420e+05	2.641
50%	117.500000	5.559944e+06	5.493074e+06	5.307400e+06	4.942770e+06	4.292907e+06	3.825
75%	175.750000	2.247650e+07	2.144798e+07	1.973085e+07	1.915957e+07	1.576230e+07	1.186
max	234.000000	1.425887e+09	1.424930e+09	1.393715e+09	1.348191e+09	1.264099e+09	1.153

.value_counts

It Shows all the unique values with their count

In [13]:
```
df["Continent"].value_counts()
```

Out[13]:
```
Africa           57
Asia             50
Europe           50
North America    40
Oceania          23
South America    14
Name: Continent, dtype: int64
```

.isnull()

It shows the how many null values

In [14]:
```
df.isnull()
```

Out[14]:

	Rank	CCA3	Country/Territory	Capital	Continent	2022 Population	2020 Population	2015 Population	Popu
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
229	False	False	False	False	False	False	False	False	
230	False	False	False	False	False	False	False	False	
231	False	False	False	False	False	False	False	False	
232	False	False	False	False	False	False	False	False	
233	False	False	False	False	False	False	False	False	

234 rows × 17 columns

.info()

To Show Data type of each column

In [15]:
```
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
```

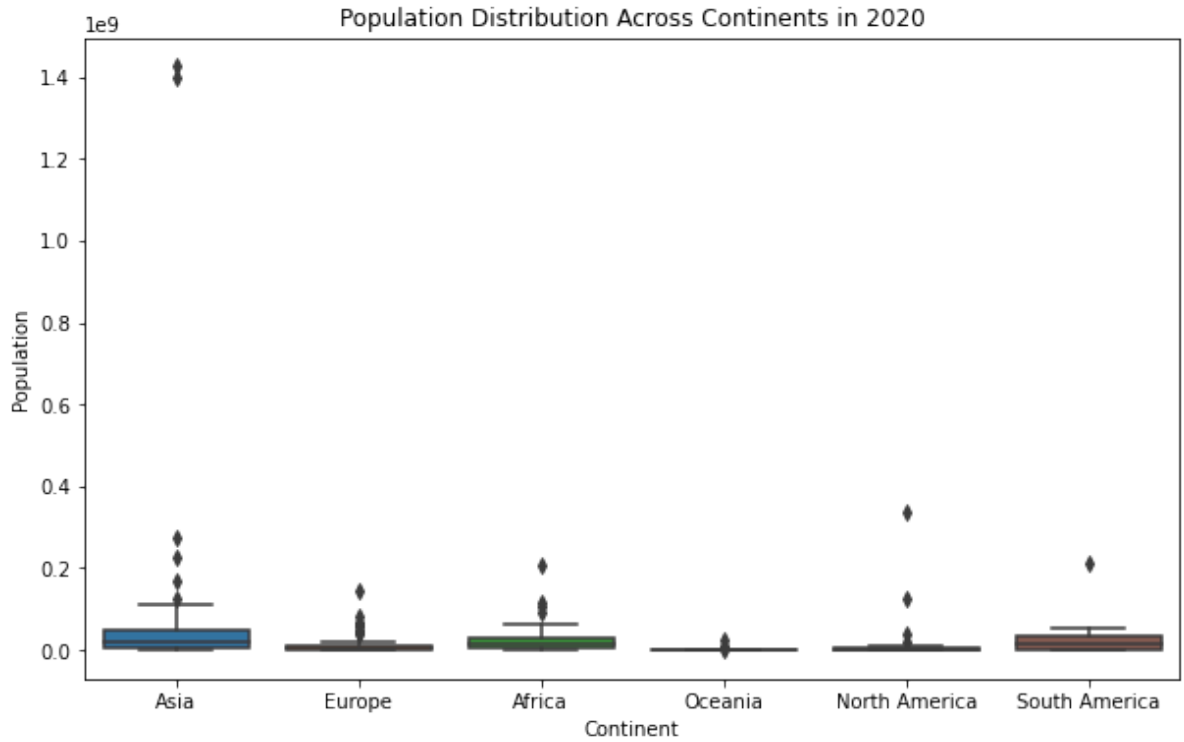
Is there any Null value present in any Column ? Show with heatmap

In [16]:
```
df.isnull().sum()
```

Out[16]:
```
Rank                            0
CCA3                            0
Country/Territory               0
Capital                         0
Continent                       0
2022 Population                 0
2020 Population                 0
2015 Population                 0
2010 Population                 0
2000 Population                 0
1990 Population                 0
1980 Population                 0
1970 Population                 0
Area (km²)                      0
Density (per km²)               0
Growth Rate                     0
World Population Percentage     0
dtype: int64
```

What does the distribution of population look like across different continents?

In [18]:
```
# Plotting population distribution across continents
plt.figure(figsize=(10, 6))
sns.boxplot(x='Continent', y='2020 Population', data=df)
plt.title('Population Distribution Across Continents in 2020')
plt.xlabel('Continent')
plt.ylabel('Population')
plt.show()
```



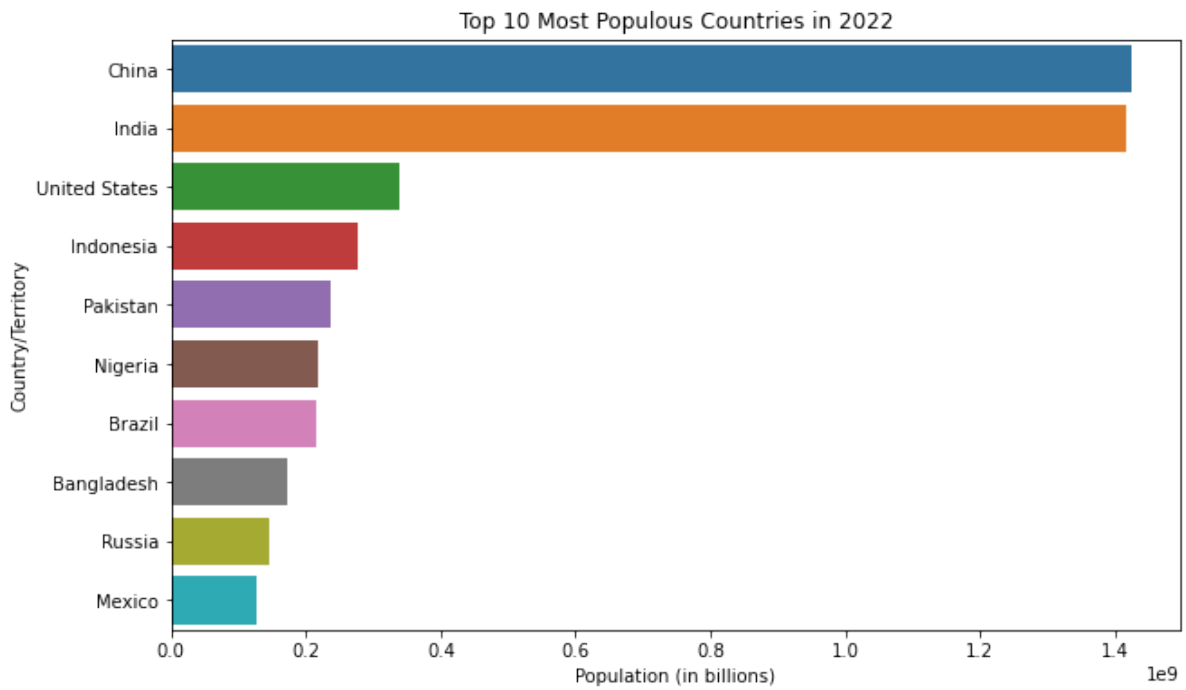
Which countries have the highest and lowest populations in 2020?

In [19]:
```
# Finding countries with the highest population in 2020
highest_population_2020 = df.nlargest(5, '2020 Population')
print("Countries with the highest population in 2020:\n", highest_population_2

# Finding countries with the lowest population in 2020
lowest_population_2020 = df.nsmallest(5, '2020 Population')
print("\nCountries with the lowest population in 2020:\n", lowest_population_2
```

```
Countries with the highest population in 2020:
     Country/Territory  2020 Population
41               China       1424929781
92               India       1396387127
221     United States        335942003
93          Indonesia        271857970
156          Pakistan        227196741

Countries with the lowest population in 2020:
     Country/Territory  2020 Population
226      Vatican City              520
209           Tokelau             1827
150              Niue             1942
64   Falkland Islands             3747
137        Montserrat             4500
```

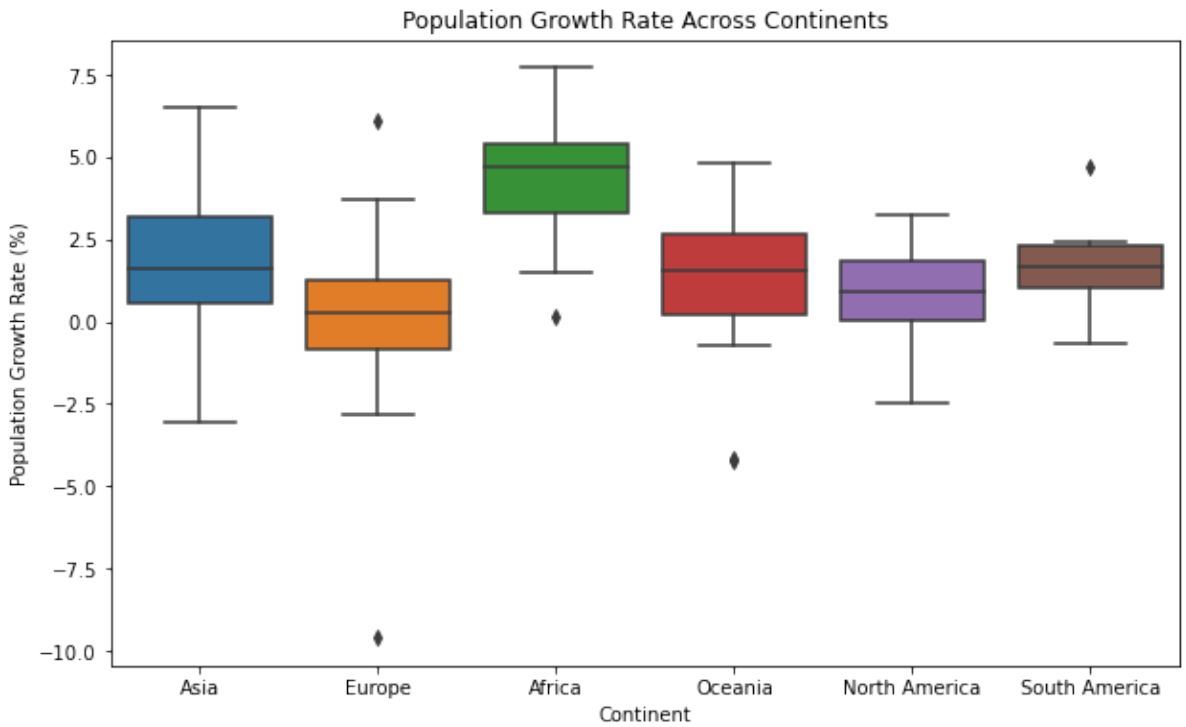
What are the top 10 most populous countries in 2020, and how do their populations compare?

In [26]:
```
# Finding the top 10 most populous countries in 2020
top_10_populous_countries = df.nlargest(10, '2022 Population')

# Plotting population of top 10 countries in billions
plt.figure(figsize=(10, 6))
sns.barplot(x='2022 Population', y='Country/Territory', data=top_10_populous_co
plt.title('Top 10 Most Populous Countries in 2022')
plt.xlabel('Population (in billions)')
plt.ylabel('Country/Territory')
plt.show()
```

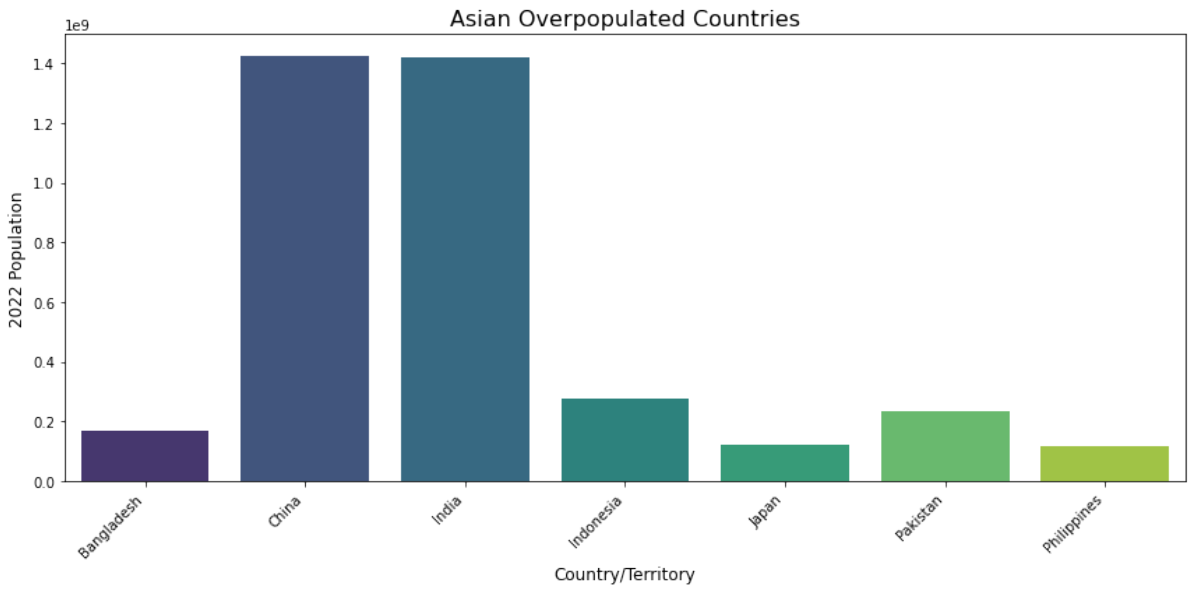


How does the population growth rate vary across continents?

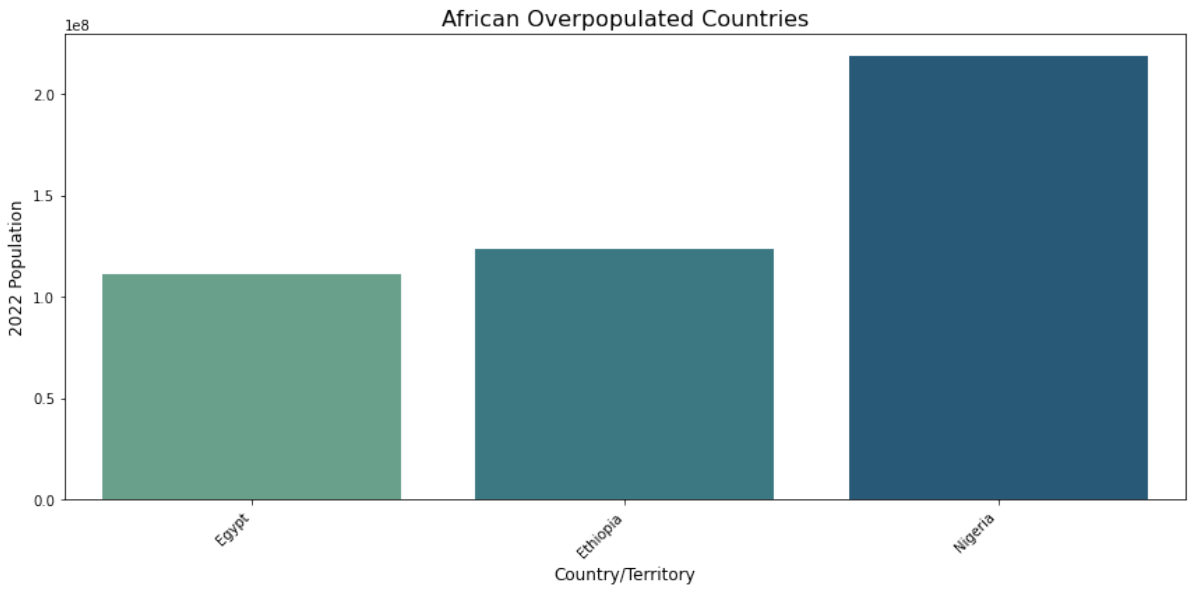
In [27]:
```
# Calculating population growth rate across continents
df['Population_Growth_Rate'] = (df['2022 Population'] - df['2020 Population'])

# Plotting population growth rate across continents
plt.figure(figsize=(10, 6))
sns.boxplot(x='Continent', y='Population_Growth_Rate', data=df)
plt.title('Population Growth Rate Across Continents')
plt.xlabel('Continent')
plt.ylabel('Population Growth Rate (%)')
plt.show()
```



In [33]:
```
# Filter the DataFrame for Asian countries with population > 100 million and de
filtered_df = df[(df["Continent"] == "Asia") & (df["2022 Population"] > 100_000

# Plotting
plt.figure(figsize=(12, 6))
sns.barplot(x="Country/Territory", y="2022 Population", data=filtered_df, palet
plt.title("Asian Overpopulated Countries", fontsize=16)
plt.xlabel("Country/Territory", fontsize=12)
plt.ylabel("2022 Population", fontsize=12)
plt.xticks(rotation=45, ha="right", fontsize=10)  # Rotate x-labels for better
plt.yticks(fontsize=10)
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



In [37]:
```
# Filter the DataFrame for African countries with population > 50 million and c
filtered_df = df[(df["Continent"] == "Africa") & (df["2022 Population"] > 50_00

# Plotting
plt.figure(figsize=(12, 6))
sns.barplot(x="Country/Territory", y="2022 Population", data=filtered_df, palet
plt.title("African Overpopulated Countries", fontsize=16)
plt.xlabel("Country/Territory", fontsize=12)
plt.ylabel("2022 Population", fontsize=12)
plt.xticks(rotation=45, ha="right", fontsize=10)  # Rotate x-labels for better
plt.yticks(fontsize=10)
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



In [42]:
```
# Filter the DataFrame for African countries with population > 50 million and c
filtered_df = df[(df["Continent"] == "Europe") & (df["2022 Population"] > 50_00

# Plotting
plt.figure(figsize=(12, 6))
sns.barplot(x="Country/Territory", y="2022 Population", data=filtered_df)
plt.title("Europe Overpopulated Countries", fontsize=16)
plt.xlabel("Country/Territory", fontsize=12)
plt.ylabel("2022 Population", fontsize=12)
plt.xticks(rotation=45, ha="right", fontsize=10)  # Rotate x-labels for better
plt.yticks(fontsize=10)
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```