| | |
|---|---|
| Project Title | **World Population Analysis** |
| language | Machine learning, python, SQL, Excel |
| Tools | VS code, Jupyter notebook |
| Domain | Data Analyst |
| Project Difficulties level | Advance |

Dataset : Dataset is available in the given link. You can download it at your convenience.

Click here to download data set

# About Dataset

### Context

The current US Census Bureau world population estimate in June 2019 shows that the current global population is 7,577,130,400 people on earth, which far exceeds the world population of 7.2 billion in 2015. Our own estimate based on UN data shows the world's population surpassing 7.7 billion.

China is the most populous country in the world with a population exceeding 1.4 billion. It is one of just two countries with a population of more than 1 billion, with India being the second. As of 2018, India has a population of over 1.355 billion people, and its population growth is expected to continue through at least 2050. By the year 2030, the

country of India is expected to become the most populous country in the world. This is because India's population will grow, while China is projected to see a loss in population.

The following 11 countries that are the most populous in the world each have populations exceeding 100 million. These include the United States, Indonesia, Brazil, Pakistan, Nigeria, Bangladesh, Russia, Mexico, Japan, Ethiopia, and the Philippines. Of these nations, all are expected to continue to grow except Russia and Japan, which will see their populations drop by 2030 before falling again significantly by 2050.

Many other nations have populations of at least one million, while there are also countries that have just thousands. The smallest population in the world can be found in Vatican City, where only 801 people reside.

In 2018, the world's population growth rate was 1.12%. Every five years since the 1970s, the population growth rate has continued to fall. The world's population is expected to continue to grow larger but at a much slower pace. By 2030, the population will exceed 8 billion. In 2040, this number will grow to more than 9 billion. In 2055, the number will rise to over 10 billion, and another billion people won't be added until near the end of the century. The current annual population growth estimates from the United Nations are in the millions - estimating that over 80 million new lives are added each year.

This population growth will be significantly impacted by nine specific countries which are situated to contribute to the population growing more quickly than other nations. These nations include the Democratic Republic of the Congo, Ethiopia, India, Indonesia, Nigeria, Pakistan, Uganda, the United Republic of Tanzania, and the United States of America. Particularly of interest, India is on track to overtake China's position as the most populous country by 2030. Additionally, multiple nations within Africa are expected to double their populations before fertility rates begin to slow entirely.

## Content

In this Dataset, we have Historical Population data for every Country/Territory in the world by different parameters like Area Size of the Country/Territory, Name of the Continent, Name of the Capital, Density, Population Growth Rate, Ranking based on Population, World Population Percentage, etc.

## Dataset Glossary (Column-Wise)

- **Rank**: Rank by Population.
- **CCA3**: 3 Digit Country/Territories Code.
- **Country/Territories**: Name of the Country/Territories.
- **Capital**: Name of the Capital.
- **Continent**: Name of the Continent.
- **2022 Population**: Population of the Country/Territories in the year 2022.
- **2020 Population**: Population of the Country/Territories in the year 2020.
- **2015 Population**: Population of the Country/Territories in the year 2015.

- **2010 Population**: Population of the Country/Territories in the year 2010.
- **2000 Population**: Population of the Country/Territories in the year 2000.
- **1990 Population**: Population of the Country/Territories in the year 1990.
- **1980 Population**: Population of the Country/Territories in the year 1980.
- **1970 Population**: Population of the Country/Territories in the year 1970.
- **Area (km²)**: Area size of the Country/Territories in square kilometer.
- **Density (per km²)**: Population Density per square kilometer.
- **Growth Rate**: Population Growth Rate by Country/Territories.
- **World Population Percentage**: The population percentage by each Country/Territories.

## Structure of the Dataset

| 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 km²) | Growth Rate | World Population Percentage |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 139 | JAM | Jamaica | Kingston | North America | 2827377 | 2820436 | 2794445 | 2733896 | 2612205 | 2392030 | 2135546 | 1859091 | 10991 | 257.2447 | 0.9999 | 0.04 |
| 107 | LBY | Libya | Tripoli | Africa | 6812341 | 6653942 | 6192235 | 6491988 | 5154790 | 4236983 | 2962720 | 1909177 | 1759540 | 3.8717 | 1.0114 | 0.09 |
| 22 | TZA | Tanzania | Dodoma | Africa | 65497748 | 61704518 | 52542823 | 45110527 | 34463704 | 26206012 | 19297659 | 13618192 | 945087 | 69.3034 | 1.0300 | 0.82 |
| 183 | PYF | French Polynesia | Papeete | Oceania | 306279 | 301920 | 291787 | 283788 | 250927 | 211089 | 163591 | 117891 | 4167 | 73.5011 | 1.0074 | 0.00 |
| 89 | HND | Honduras | Tegucigalpa | North America | 10432860 | 10121763 | 9294505 | 8450933 | 6656725 | 5053234 | 3777990 | 2782753 | 112492 | 92.7431 | 1.0150 | 0.13 |
| 178 | GLP | Guadeloupe | Basse-Terre | North America | 395752 | 395642 | 399089 | 403072 | 424067 | 391951 | 334234 | 318310 | 1628 | 243.0909 | 0.9992 | 0.00 |
| 26 | MMR | Myanmar | Nay Pyi Taw | Asia | 54179306 | 53423198 | 51483949 | 49390988 | 45538332 | 40099553 | 33465781 | 27284112 | 676578 | 80.0784 | 1.0071 | 0.68 |
| 221 | VGB | British Virgin Islands | Road Town | North America | 31305 | 30910 | 29366 | 27556 | 20104 | 15617 | 11109 | 9581 | 151 | 207.3179 | 1.0059 | 0.00 |
| 94 | HUN | Hungary | Budapest | Europe | 9967308 | 9750573 | 9844246 | 9986825 | 10202055 | 10375989 | 10698679 | 10315366 | 93028 | 107.1431 | 1.0265 | 0.12 |
| 136 | PRI | Puerto Rico | San Juan | North America | 3252407 | 3271564 | 3497335 | 3717922 | 3827108 | 3543776 | 3214568 | 2737619 | 8870 | 366.6750 | 0.9989 | 0.04 |

## Acknowledgement

This Dataset is created from **https://worldpopulationreview.com/**. If you want to learn more, you can visit the Website.

Cover Photo by: **People symbol vector created by rawpixel.com - www.freepik.com**

## World Population Analysis Machine Learning Project

### Project Overview

The goal of this project is to analyze global population trends using historical data and predict future population growth. This involves using machine learning techniques to

explore demographic data, identify key factors influencing population changes, and build predictive models.

#### Dataset

You can use datasets from sources like:

- **United Nations (UN) World Population Prospects**
- **World Bank**
- **Kaggle** (e.g., the "World Population Data" dataset)

#### Steps and Implementation

1. **Data Collection**
2. **Data Preprocessing**
3. **Exploratory Data Analysis (EDA)**
4. **Feature Engineering**
5. **Model Building**
6. **Model Evaluation**
7. **Visualization**
8. **Report Generation**

### Implementation Code

Here is a sample implementation in Python:

```
# Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Data Collection
# Load the dataset (example: world_population.csv)
data = pd.read_csv('world_population.csv')

# Display basic info about the dataset
print(data.info())
print(data.head())

# Data Preprocessing
# Handle missing values
data = data.dropna()

# Feature Engineering
# Create additional features if necessary (e.g., population growth rate)
data['GrowthRate'] = data['Population'].pct_change() * 100
data = data.dropna()

# Define features and target variable
features = ['Year', 'BirthRate', 'DeathRate', 'NetMigration', 'FertilityRate']
X = data[features]
y = data['Population']

# Splitting the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Feature Scaling
scaler = StandardScaler()
```

```
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Model Building
# Train a Linear Regression model
model = LinearRegression()
model.fit(X_train_scaled, y_train)

# Predict on the test set
y_pred = model.predict(X_test_scaled)

# Model Evaluation
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("R^2 Score:", r2_score(y_test, y_pred))

# Visualization of Results
plt.figure(figsize=(14,7))
plt.plot(data['Year'], data['Population'], label='Actual Population')
plt.plot(X_test['Year'], y_pred, label='Predicted Population', linestyle='--')
plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Prediction')
plt.legend()
plt.show()
```

## Project Report Outline

#### 1. Introduction

- Overview of the project
- Importance of population analysis

#### 2. Data Collection

- Source of data
- Description of dataset features

#### 3. Data Preprocessing

- Handling missing values
- Feature selection and engineering

#### 4. Exploratory Data Analysis (EDA)

- Summary statistics
- Visualization of population trends
- Analysis of key factors affecting population growth

#### 5. Model Building

- Description of the machine learning model used
- Training and testing dataset split
- Feature scaling

#### 6. Model Evaluation

- Performance metrics (MSE, $R^2$ score)
- Comparison of predicted vs actual population

#### 7. Results and Discussion

- Interpretation of results
- Key insights from the analysis
- Limitations and potential improvements

#### 8. Conclusion

- Summary of findings
- Future work and recommendations

### Sample Report Excerpt

### Introduction

The objective of this project is to analyze historical world population data and predict future population trends. Understanding population dynamics is crucial for planning and policy-making in various sectors such as healthcare, education, and infrastructure.

### Data Collection

The dataset used in this project is sourced from the United Nations World Population Prospects and includes features such as year, birth rate, death rate, net migration, fertility rate, and population.

### Data Preprocessing

To prepare the data for analysis, we handled missing values by removing rows with NA values. Additional features, such as the population growth rate, were created to enhance the model's predictive power.

### Exploratory Data Analysis

Exploratory analysis revealed significant trends in population growth over the years. Key factors such as birth rate, death rate, and fertility rate were visualized to understand their impact on population changes.

### Model Building

A Linear Regression model was trained using the historical population data. The features were scaled to ensure uniform contribution to the model, and the dataset was split into training and testing sets.

### Model Evaluation

The model's performance was evaluated using Mean Squared Error (MSE) and $R^2$ score. The results showed a reasonable prediction accuracy, with the predicted population trends closely following the actual data.

### Results and Discussion

The analysis indicated that factors such as birth rate and fertility rate significantly influence population growth. However, the model could be further improved by incorporating more features and using advanced machine learning techniques.

#### Conclusion

This project successfully demonstrated the use of machine learning in analyzing and predicting world population trends. Future work could involve exploring more sophisticated models and considering additional demographic factors.

### Additional Resources

- UN World Population Prospects
- World Bank Population Data
- Kaggle World Population Dataset

This project framework provides a comprehensive approach to analyzing and predicting world population trends using machine learning. You can further enhance it by experimenting with different algorithms, fine-tuning hyperparameters, and incorporating additional data sources.

## Sample code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.subplots as sp
import plotly.graph_objects as go
```

```
from plotly.subplots import make_subplots
import warnings
# Suppress FutureWarning messages
warnings.simplefilter(action='ignore', category=FutureWarning)
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
# Graph
```

## World Population Dataset

- Rank: Rank by Population.
- CCA3: 3 Digit Country/Territories Code.
- Country/Territories: Name of the Country/Territories.
- Capital: Name of the Capital.
- Continent: Name of the Continent.
- 2022 Population: Population of the Country/Territories in the year 2022.
- 2020 Population: Population of the Country/Territories in the year 2020.
- 2015 Population: Population of the Country/Territories in the year 2015.
- 2010 Population: Population of the Country/Territories in the year 2010.
- 2000 Population: Population of the Country/Territories in the year 2000.
- 1990 Population: Population of the Country/Territories in the year 1990.
- 1980 Population: Population of the Country/Territories in the year 1980.
- 1970 Population: Population of the Country/Territories in the year 1970.
- Area (km²): Area size of the Country/Territories in square kilometer.
- Density (per km²): Population Density per square kilometer.
- Growth Rate: Population Growth Rate by Country/Territories.
- World Population Percentage: The population percentage by each Country/Territories

In [2]:

```
df = pd.read_csv('/kaggle/input/world-population-dataset/world_population.csv')
```

In [3]:

```
df.head()
```

Out[3]:

| | 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 km²) | Growth Rate | World Population Percentage |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 36 | AFG | Afghanistan | Kabul | Asia | 41128771 | 38972230 | 33753499 | 28189672 | 19542982 | 10694796 | 12486631 | 10752971 | 652230 | 63.0587 | 1.0257 | 0.52 |
| 1 | 138 | ALB | Albania | Tirana | Europe | 2842321 | 2866849 | 2882481 | 2913399 | 3182021 | 3295066 | 2941651 | 2324731 | 28748 | 98.8702 | 0.9957 | 0.04 |
| 2 | 34 | DZA | Algeria | Algiers | Africa | 44903225 | 43451666 | 39543154 | 35856344 | 30774621 | 25518074 | 18739378 | 13795915 | 2381741 | 18.8531 | 1.0164 | 0.56 |
| 3 | 213 | ASM | American Samoa | Pago Pago | Oceania | 44273 | 46189 | 51368 | 54849 | 58230 | 47818 | 32886 | 27075 | 199 | 222.4774 | 0.9831 | 0.00 |
| 4 | 20 | AN | Andorra | Andor | Eur | 7982 | 7770 | 7174 | 7151 | 6609 | 5356 | 3561 | 1986 | 468 | 170.56 | 1.01 | 0.00 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 3 | D | | ra la Vel la | ope | 4 | 0 | 6 | 9 | 7 | 9 | 1 | 0 | | 41 | 00 | |

## Data Cleaning

In [4]:

```
df.shape
```

Out[4]:

```
(234, 17)
```

In [5]:

```
df.isna().sum()
```

Out[5]:

```
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
```

No missing values!

In [6]:

```
print(f"Amount of duplicates: {df.duplicated().sum()}")
```

```
Amount of duplicates: 0
```

In [7]:

```
df.columns
```

Out[7]:

```
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 km²)',
       'Growth Rate', 'World Population Percentage'],

      dtype='object')
```

In [8]:

```
# Drop 'CCA3' and 'Capital' columns since we won't be using them in the analysis

df.drop(['CCA3', 'Capital'], axis=1, inplace=True)
```

In [9]:

```
df.head()
```

Out[9]:

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 213 | American Samoa | Oceania | 44273 | 46189 | 51368 | 54849 | 58230 | 47818 | 32886 | 27075 | 199 | 222.4774 | 0.9831 | 0.00 |
| 4 | 203 | Andorra | Europe | 79824 | 77700 | 71746 | 71519 | 66097 | 53569 | 35611 | 19860 | 468 | 170.5641 | 1.0100 | 0.00 |

In [10]:

```
df.tail()
```

Out[10]:

| | Rank | Country/Territory | Continent | 2022 Population | 2020 Population | 2015 Population | 2010 Population | 2000 Population | 1990 Population | 1980 Population | 1970 Population | Area (km²) | Density (per km²) | Growth Rate | World Population Percentage |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 229 | 226 | Wallis and Futuna | Oceania | 11572 | 11655 | 12182 | 13142 | 14723 | 13454 | 11315 | 9377 | 142 | 81.4930 | 0.9953 | 0.00 |
| 23 | 7 | Zimbabw | Afric | 1632 | 1566 | 1415 | 1283 | 1183 | 1011 | 7049 | 5202 | 390 | 41.766 | 1.020 | 0.20 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 4 | e | a | 0537 | 9666 | 4937 | 9771 | 4676 | 3893 | 926 | 918 | 757 | 5 | 4 | |

## Visualizations

In [11]:

```
custom_palette = ['#0b3d91', '#e0f7fa', '#228b22', '#1e90ff', '#8B4513', '#D2691E',
'#DAA520', '#556B2F']
```

In [12]:

```
countries_by_continent = df['Continent'].value_counts().reset_index()
```

In [13]:

```
# Create the bar chart
fig = px.bar(
    countries_by_continent,
    x='Continent',
    y='count',
    color='Continent',
    text='count',
    title='Number of Countries by Continent',
    color_discrete_sequence=custom_palette
)

# Customize the layout
fig.update_layout(
    xaxis_title='Continents',
    yaxis_title='Number of Countries',
    plot_bgcolor='rgba(0,0,0,0)',  # Set the background color to transparent
    font_family='Arial',  # Set font family
    title_font_size=20  # Set title font size
)
```

```
# Show the plot
fig.show()
```

575050402314AfricaAsiaEuropeNorth AmericaOceaniaSouth America0102030405060

ContinentAfricaAsiaEuropeNorth AmericaOceaniaSouth AmericaNumber of Countries by ContinentContinentsNumber of Countries

In [14]:

```
continent_population_percentage = df.groupby('Continent')['World Population Percentage'].sum().reset_index()
```

In [15]:

```
# Create the pie chart
fig = go.Figure(data=[go.Pie(labels=continent_population_percentage['Continent'], values=continent_population_percentage['World Population Percentage'])])

# Update layout
fig.update_layout(
    title='World Population Percentage by Continent',
    template='plotly',
    paper_bgcolor='rgba(255,255,255,0)',  # Set the paper background color to transparent
    plot_bgcolor='rgba(255,255,255,0)'    # Set the plot background color to transparent
)

# Update pie colors
fig.update_traces(marker=dict(colors=custom_palette, line=dict(color='#FFFFFF', width=1)))
```

```
# Show the plot
fig.show()
```

59.2%17.9%9.34%7.52%5.48%0.55%

AsiaAfricaEuropeNorth AmericaSouth AmericaOceaniaWorld Population Percentage by Continent

In [16]:

```
# Melt the DataFrame to have a long format
df_melted = df.melt(id_vars=['Continent'],
                    value_vars=['2022 Population', '2020 Population', '2015
Population',
                                '2010 Population', '2000 Population', '1990
Population',
                                '1980 Population', '1970 Population'],
                    var_name='Year',
                    value_name='Population')

# Convert 'Year' to a more suitable format
df_melted['Year'] = df_melted['Year'].str.split().str[0].astype(int)

# Aggregate population by continent and year
population_by_continent = df_melted.groupby(['Continent',
'Year']).sum().reset_index()
```

In [17]:

```
fig = px.line(population_by_continent, x='Year', y='Population', color='Continent',
              title='Population Trends by Continent Over Time',
              labels={'Population': 'Population', 'Year': 'Year'},
              color_discrete_sequence=custom_palette)

fig.update_layout(
```

```
    template='plotly_white',
    xaxis_title='Year',
    yaxis_title='Population',
    font_family='Arial',
    title_font_size=20,
)

fig.update_traces(line=dict(width=3))

fig.show()
```

19701980199020002010202001B2B3B4B

ContinentAfricaAsiaEuropeNorth AmericaOceaniaSouth AmericaPopulation Trends by Continent Over TimeYearPopulation

## World Population Comparison: 1970 to 2020

In [18]:

```
features=['1970 Population' ,'2020 Population']
for feature in features:
    fig = px.choropleth(df,
                        locations='Country/Territory',
                        locationmode='country names',
                        color=feature,
                        hover_name='Country/Territory',
                        template='plotly_white',
                        title = feature)
    fig.show()
```

0100M200M300M400M500M600M700M800M1970 Population1970 Population

00.2B0.4B0.6B0.8B1B1.2B1.4B2020 Population2020 Population

Over the past 50 years, India has experienced a remarkable population growth. In 1970, the population of India was approximately 557 million. By 2020, this number had surged to around 1.4 billion, reflecting the significant demographic changes the country has undergone in just over half a century.

In [19]:

```
growth = (df.groupby(by='Country/Territory')['2022
Population'].sum()-df.groupby(by='Country/Territory')['1970
Population'].sum()).sort_values(ascending=False).head(8)
```

In [20]:

```
fig=px.bar(x=growth.index,
       y=growth.values,
       text=growth.values,
       color=growth.values,
       title='Growth Of Population From 1970 to 2020 (Top 8)',
       template='plotly_white')
fig.update_layout(xaxis_title='Country',
                  yaxis_title='Population Growth')
fig.show()
```


8596718726033528871765339901629719481602729451379615171189436231036445 12IndiaChi naPakistanNigeriaIndonesiaUnited StatesBrazilBangladesh0100M200M300M400M500M600M700M800M900M

200M300M400M500M600M700M800McolorGrowth Of Population From 1970 to 2020 (Top 8)CountryPopulation Growth


In [21]:

```
top_8_populated_countries_1970 = df.groupby('Country/Territory')['1970
Population'].sum().sort_values(ascending=False).head(8)
top_8_populated_countries_2022 = df.groupby('Country/Territory')['2022
Population'].sum().sort_values(ascending=False).head(8)
```

In [22]:

```
features = {'top_8_populated_countries_1970': top_8_populated_countries_1970,
            'top_8_populated_countries_2022': top_8_populated_countries_2022}

for feature_name, feature_data in features.items():
    year = feature_name.split('_')[-1]  # Extract the year from the feature name
    fig = px.bar(x=feature_data.index,
                 y=feature_data.values,
                 text=feature_data.values,
                 color=feature_data.values,
                 title=f'Top 8 Most Populated Countries ({year})',
                 template='plotly_white')
    fig.update_layout(xaxis_title='Country',
                      yaxis_title='Population Growth')
    fig.show()
```


82253445055750130120032834013009301011522839410541683996369875782945 83ChinaIndia United StatesRussiaIndonesiaJapanBrazilGermany0100M200M300M400M500M600M700M800M

100M200M300M400M500M600M700M800McolorTop 8 Most Populated Countries (1970)CountryPopulation Growth

1425887337141717317333828985727550133923582486221854121221531349817118 6372ChinaIndiaUnited StatesIndonesiaPakistanNigeriaBrazilBangladesh00.2B0.4B0.6B0.8B1B1.2B1.4B

0.2B0.4B0.6B0.8B1B1.2B1.4BcolorTop 8 Most Populated Countries (2022)CountryPopulation Growth


## World Population Growth Rates: The Fastest Growing Countries

In [23]:

```
sorted_df_growth = df.sort_values(by='Growth Rate', ascending=False)
```

```
top_fastest = sorted_df_growth.head(6)
top_slowest = sorted_df_growth.tail(6)
```

In [24]:

```
def plot_population_trends(countries):
    # Calculate the number of rows needed
    n_cols = 2
    n_rows = (len(countries) + n_cols - 1) // n_cols

    # Create subplots
    fig = sp.make_subplots(rows=n_rows, cols=n_cols, subplot_titles=countries,
                           horizontal_spacing=0.1, vertical_spacing=0.1)

    for i, country in enumerate(countries, start=1):
        # Filter data for the selected country
        country_df = df[df['Country/Territory'] == country]

        # Melt the DataFrame to have a long format
        country_melted = country_df.melt(id_vars=['Country/Territory'],
                                value_vars=['2022 Population', '2020 Population', '2015
Population',
                                            '2010 Population', '2000 Population', '1990
Population',
                                            '1980 Population', '1970 Population'],
                                var_name='Year',
                                value_name='Population')

        # Convert 'Year' to a more suitable format
        country_melted['Year'] =
country_melted['Year'].str.split().str[0].astype(int)

        # Create a line plot for each country
```

```
        line_fig = px.line(country_melted, x='Year', y='Population',
color='Country/Territory',
                           labels={'Population': 'Population', 'Year': 'Year'},
                           color_discrete_sequence=custom_palette)

        # Update the line plot to fit the subplot
        row = (i - 1) // n_cols + 1
        col = (i - 1) % n_cols + 1
        for trace in line_fig.data:
            fig.add_trace(trace, row=row, col=col)

    # Update the layout of the subplots
    fig.update_layout(
        title='Population Trends of Selected Countries Over Time',
        template='plotly_white',
        font_family='Arial',
        title_font_size=20,
        showlegend=False,
        height=600*n_rows,  # Adjust height for bigger plots
    )

    fig.update_traces(line=dict(width=3))
    fig.update_xaxes(title_text='Year')
    fig.update_yaxes(title_text='Population')

    fig.show()
```

In [25]:

```
fastest = top_fastest[['Country/Territory', 'Growth Rate']].sort_values(by='Growth
Rate', ascending=False).reset_index(drop=True)
fastest
```

Out[25]:

| | Country/Territory | Growth Rate |
|---|---|---|
| 0 | Moldova | 1.0691 |
| 1 | Poland | 1.0404 |
| 2 | Niger | 1.0378 |
| 3 | Syria | 1.0376 |
| 4 | Slovakia | 1.0359 |
| 5 | DR Congo | 1.0325 |

In [26]:

```
plot_population_trends(['Moldova', 'Poland', 'Niger', 'Syria', 'Slovakia', 'DR
Congo'])
```

19701980199020002010202033.2M3.4M3.6M3.8M4M4.2M4.4M19701980199020002010202033M34M35M36M37M38M39M40M19701980199020002010202205M10M15M20M25M197019801990200020

1020206M8M10M12M14M16M18M20M22M19701980199020002010202O4.6M4.8M5M5.2M5.4M5.6M19701980199020002010202020M30M40M50M60M70M80M90M100M

Population Trends of Selected Countries Over TimeYearYearYearYearYearYearPopulationPopulationPopulationPopulationPopulationPopulationMoldovaPolandNigerSyriaSlovakiaDR Congo

## World Population Growth Rates: The Slowest Growing Countries

In [27]:

```
slowest = top_slowest[['Country/Territory', 'Growth Rate']].sort_values(by='Growth Rate', ascending=False).reset_index(drop=True)
slowest
```

Out[27]:

| | Country/Territory | Growth Rate |
|---|---|---|
| 0 | Latvia | 0.9876 |
| 1 | Lithuania | 0.9869 |
| 2 | Bulgaria | 0.9849 |
| 3 | American Samoa | 0.9831 |

| | | |
|---|---|---|
| 4 | Lebanon | 0.9816 |
| 5 | Ukraine | 0.9120 |

In [28]:

```
plot_population_trends(['Latvia', 'Lithuania', 'Bulgaria', 'American Samoa',
'Lebanon', 'Ukraine'])
```

19701980199020002010202019.9M2M2.1M2.2M2.3M2.4M2.5M2.6M2.7M19701980199020002010202028.8M3M3.2M3.4M3.6M3.8M19701980199020002010202077.5M8M8.5M9M19701980199020002010202030k35k40k45k50k55k19701980199020002010202025M3M3.5M4M4.5M5M5.5M6M6.5M19701980199020002010202040M42M44M46M48M50M52M

Population Trends of Selected Countries Over TimeYearYearYearYearYearYearPopulationPopulationPopulationPopulationPopulationPopulationLatvia LithuaniaBulgariaAmerican SamoaLebanonUkraine

## Land Area by Country

In [29]:

```
land_by_country = df.groupby('Country/Territory')['Area
(km²)'].sum().sort_values(ascending=False)
most_land = land_by_country.head(5)
least_land = land_by_country.tail(5)
```

In [30]:

```
# Create subplots
fig = sp.make_subplots(rows=1, cols=2, subplot_titles=("Countries with Most Land",
"Countries with Least Land"))
```

```
# Plot countries with the most land
fig.add_trace(go.Bar(x=most_land.index, y=most_land.values, name='Most Land',
marker_color=custom_palette[0]), row=1, col=1)

# Plot countries with the least land
fig.add_trace(go.Bar(x=least_land.index, y=least_land.values, name='Least Land',
marker_color=custom_palette[1]), row=1, col=2)

fig.update_layout(
    title_text="Geographical Distribution of Land Area by Country",
    showlegend=False,
    template='plotly_white'
)

fig.update_yaxes(title_text="Area (km²)", row=1, col=1)
fig.update_yaxes(title_text="Area (km²)", row=1, col=2)

fig.show()
```

RussiaCanadaChinaUnited StatesBrazil02M4M6M8M10M12M14M16MSaint BarthelemyTokelauGibraltarMonacoVatican City05101520

Geographical Distribution of Land Area by CountryArea (km²)Area (km²)Countries with Most LandCountries with Least Land

## Land Area Per Person by Country

In [31]:

```
df['Area per Person']=df['Area (km²)'] / df['2022 Population']
country_area_per_person = df.groupby('Country/Territory')['Area per Person'].sum()
most_land_available = country_area_per_person.sort_values(ascending=False).head(5)
least_land_available = country_area_per_person.sort_values(ascending=False).tail(5)
```

In [32]:

# linkcode

```
# Create subplots
fig = sp.make_subplots(rows=1, cols=2, subplot_titles=("Countries with Most Land Available Per Capita", "Countries with Least Land Available Per Capita"))

# Plot countries with the most land
fig.add_trace(go.Bar(x=most_land_available.index, y=most_land_available.values, name='Most Land', marker_color=custom_palette[2]), row=1, col=1)

# Plot countries with the least land
fig.add_trace(go.Bar(x=least_land_available.index, y=least_land_available.values, name='Least Land', marker_color=custom_palette[3]), row=1, col=2)

fig.update_layout(
    title_text="Distribution of Available Land Area by Country Per Capita",
    showlegend=False,
    template='plotly_white'
)

fig.update_yaxes(title_text="Land Available Per Person", row=1, col=1)
fig.update_yaxes(title_text="Land Available Per Person", row=1, col=2)


fig.show()
```

Reference link