About Dataset

## Context

The current US Census Bureau world population estimate in June 2019 shows thatthecurrentglobalpopulationis 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 isthemostpopulouscountryintheworldwithapopulationexceeding1.4billion.Itisoneofjusttwocountries withapopulationofmorethan1billion,withIndiabeingthesecond.Asof2018,Indiahasapopulationofover

1.355billionpeople,anditspopulationgrowthisexpectedtocontinuethroughat least 2050. By the year 2030, the

countryof 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 expectedtocontinuetogrowexceptRussiaandJapan,whichwillsee their populations drop by 2030 before falling again significantly by 2050.

Many other nations have populations of atleastonemillion,whiletherearealsocountriesthathavejustthousands. The smallest population in the world can be found in Vatican City, where only 801 people reside.

In 2018, the world’s population growthratewas1.12%.Everyfiveyearssincethe1970s,thepopulationgrowthrate 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, thisnumberwillgrowtomorethan9billion.In2055,thenumber 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 areinthemillions-estimatingthatover80millionnew lives are added each year.

This population growthwillbesignificantlyimpactedbyninespecificcountrieswhicharesituatedtocontributetothe 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.Particularlyofinterest,IndiaisontracktoovertakeChina'spositionasthemostpopulouscountryby2030. Additionally, multiple nations withinAfricaareexpectedtodoubletheirpopulationsbeforefertilityratesbegintoslow entirely.

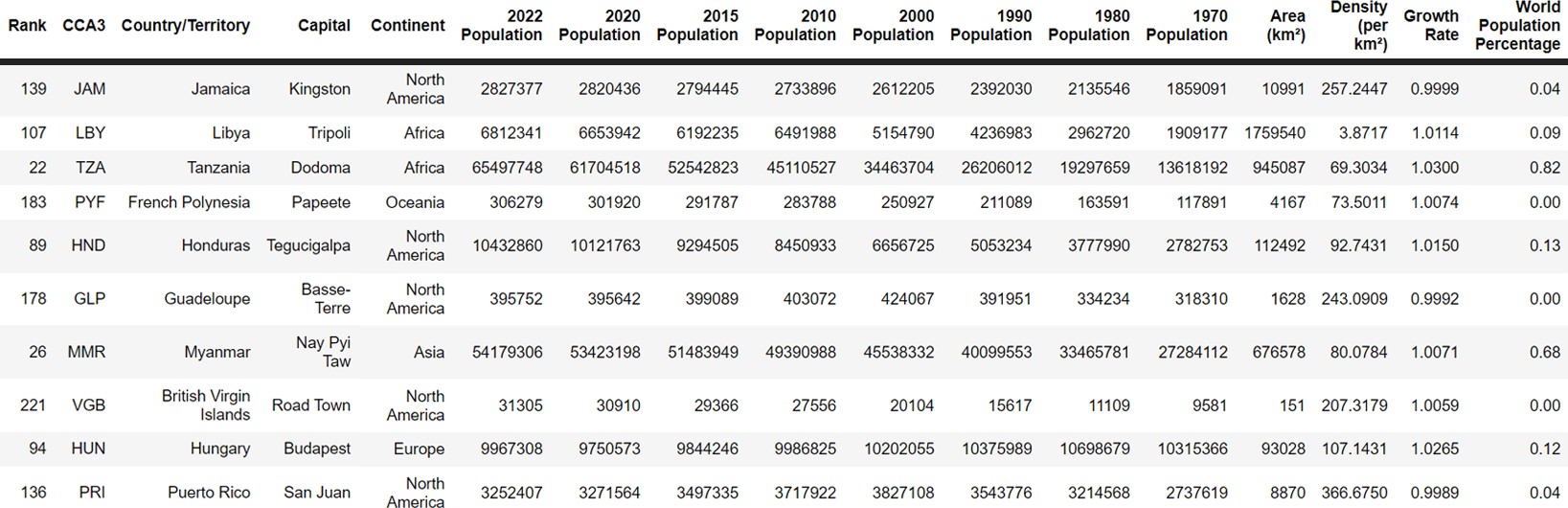
## Content

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**:3DigitCountry/TerritoriesCode.
* **Country/Territories**:NameoftheCountry/Territories.
* **Capital**: Name of the Capital.
* **Continent**: Name of the Continent.
* **2022Population**:PopulationoftheCountry/Territoriesintheyear 2022.
* **2020Population**:PopulationoftheCountry/Territoriesintheyear 2020.
* **2015Population**:PopulationoftheCountry/Territoriesintheyear 2015.
* **2010Population**:PopulationoftheCountry/Territoriesintheyear 2010.
* **2000Population**:PopulationoftheCountry/Territoriesintheyear 2000.
* **1990Population**:PopulationoftheCountry/Territoriesintheyear 1990.
* **1980Population**:PopulationoftheCountry/Territoriesintheyear 1980.
* **1970Population**:PopulationoftheCountry/Territoriesintheyear 1970.
* **Area(km²)**:AreasizeoftheCountry/Territoriesinsquare kilometer.
* **Density(per km²)**: Population Density per square kilometer.
* **Growth Rate**: Population Growth Rate by Country/Territories.
* **WorldPopulation Percentage**:The populationpercentage byeach Country/Territories.

## Structure of the Dataset

****

**Acknowledgement**

ThisDatasetiscreatedfrom[**https://worldpopulationreview.com/**](https://worldpopulationreview.com/).Ifyouwanttolearnmore,youcanvisitthe Website.

Cover Photo by: [**People symbol vector created by rawpixel.com - www.freepik.com**](https://www.freepik.com/vectors/people-symbol)

# WorldPopulationAnalysisMachineLearningProject

**Project Overview**

Thegoalofthisprojectistoanalyzeglobalpopulationtrendsusinghistoricaldataand predictfuturepopulationgrowth.Thisinvolvesusingmachinelearningtechniquesto

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

**Dataset**

Youcanusedatasetsfromsourceslike:

* **UnitedNations(UN)WorldPopulationProspects**
* **WorldBank**
* **Kaggle**(e.g.,the"WorldPopulationData"dataset)

**StepsandImplementation**

1. **DataCollection**
2. **DataPreprocessing**
3. **ExploratoryDataAnalysis(EDA)**
4. **FeatureEngineering**
5. **ModelBuilding**
6. **ModelEvaluation**
7. **Visualization**
8. **ReportGeneration**

**ImplementationCode**

Here is a sample implementation in Python:

#Importingnecessarylibraries import pandas as pd

import numpy as np

importmatplotlib.pyplotasplt import seaborn as sns

fromsklearn.model\_selectionimporttrain\_test\_split from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LinearRegression

fromsklearn.metricsimportmean\_squared\_error,r2\_score

# Data Collection

#Loadthedataset(example:world\_population.csv) data = pd.read\_csv('world\_population.csv')

#Displaybasicinfoaboutthedataset print(data.info())

print(data.head())

# Data Preprocessing

#Handlemissingvalues data = data.dropna()

# Feature Engineering

#Createadditionalfeaturesifnecessary(e.g.,populationgrowthrate) 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

#TrainaLinearRegressionmodel model = LinearRegression() model.fit(X\_train\_scaled, y\_train)

# Predict on the test set

y\_pred = model.predict(X\_test\_scaled)

# Model Evaluation

print("MeanSquaredError:",mean\_squared\_error(y\_test,y\_pred)) print("R^2 Score:", r2\_score(y\_test, y\_pred))

#VisualizationofResults plt.figure(figsize=(14,7))

plt.plot(data['Year'], data['Population'], label='Actual Population') plt.plot(X\_test['Year'],y\_pred,label='PredictedPopulation',linestyle='--') plt.xlabel('Year')

plt.ylabel('Population')

plt.title('WorldPopulationPrediction') plt.legend()

plt.show()

# ProjectReportOutline

1. **Introduction**
   * Overview of the project
   * Importance of population analysis
2. **DataCollection**
   * Source of data
   * Description of dataset features
3. **DataPreprocessing**
   * Handling missing values
   * Feature selection and engineering
4. **ExploratoryDataAnalysis(EDA)**
   * Summary statistics
   * Visualizationofpopulationtrends
   * Analysisofkeyfactorsaffectingpopulation growth
5. **ModelBuilding**
   * Description of the machine learning model used
   * Trainingandtestingdataset split
   * Feature scaling
6. **ModelEvaluation**
   * Performance metrics (MSE, R² score)
   * Comparison of predicted vs actual population
7. **ResultsandDiscussion**
   * Interpretation of results
   * Key insights from the analysis
   * Limitations and potential improvements
8. **Conclusion**
   * Summary of findings
   * Future work and recommendations

# SampleReportExcerpt

**Introduction**

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

**DataCollection**

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

**DataPreprocessing**

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.

**ExploratoryDataAnalysis**

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.

**ModelBuilding**

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.

**ModelEvaluation**

The model's performance was evaluated using Mean Squared Error (MSE) and R²score. The resultsshowedareasonablepredictionaccuracy,withthepredictedpopulationtrends closely following the actual data.

**ResultsandDiscussion**

The analysisindicatedthatfactorssuchasbirthrateandfertilityratesignificantlyinfluence 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 populationtrends.Futureworkcouldinvolveexploringmoresophisticated models and considering additional demographic factors.

# AdditionalResources

* + [UNWorldPopulationProspects](https://population.un.org/wpp/)
  + [WorldBankPopulationData](https://data.worldbank.org/indicator/SP.POP.TOTL)
  + [KaggleWorldPopulationDataset](https://www.kaggle.com/datasets)

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.

**Samplecode**

importpandasaspd import numpy as np

importmatplotlib.pyplotasplt import seaborn as sns

import plotly.express as px import plotly.subplots as sp importplotly.graph\_objectsasgo

fromplotly.subplotsimportmake\_subplots import warnings

*#SuppressFutureWarningmessages*

warnings.simplefilter(action='ignore', category=**FutureWarning**)

fromplotly.offlineimportdownload\_plotlyjs,init\_notebook\_mode,plot,iplot init\_notebook\_mode(connected=True)

*#Graph*

**WorldPopulationDataset**

* + - Rank: Rank by Population.
    - CCA3:3DigitCountry/TerritoriesCode.
    - Country/Territories:NameoftheCountry/Territories.
    - Capital: Name of the Capital.
    - Continent: Name of the Continent.
    - 2022Population:PopulationoftheCountry/Territoriesintheyear 2022.
    - 2020Population:PopulationoftheCountry/Territoriesintheyear 2020.
    - 2015Population:PopulationoftheCountry/Territoriesintheyear 2015.
    - 2010Population:PopulationoftheCountry/Territoriesintheyear 2010.
    - 2000Population:PopulationoftheCountry/Territoriesintheyear 2000.
    - 1990Population:PopulationoftheCountry/Territoriesintheyear 1990.
    - 1980Population:PopulationoftheCountry/Territoriesintheyear 1980.
    - 1970Population:PopulationoftheCountry/Territoriesintheyear 1970.
    - Area(km²):AreasizeoftheCountry/Territoriesinsquare kilometer.
    - Density(per km²): Population Density per square kilometer.
    - Growth Rate: Population Growth Rate by Country/Territories.
    - WorldPopulation Percentage:The populationpercentage byeach Country/Territories

In [2]:

df= pd.read\_csv('/kaggle/input/world-population-dataset/world\_population.csv')

In [3]:

df.head()

Out[3]:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | R  ank | CCA3 | Country  /Territor y | Ca pit al | Con tine nt | 2022  Pop ulati on | 2020  Pop ulati on | 2015  Pop ulati on | 2010  Pop ulati on | 2000  Pop ulati on | 1990  Pop ulati on | 1980  Pop ulati on | 1970  Pop ulati on | Are a (km  ²) | Den sity (per km²  ) | Gr ow th Ra te | Worl d Popu lation Perc enta ge |
| 0 | 3  6 | AFG | Afghani stan | Ka bul | Asia | 4112  8771 | 3897  2230 | 3375  3499 | 2818  9672 | 1954  2982 | 1069  4796 | 1248  6631 | 1075  2971 | 652  230 | 63.  058  7 | 1.  02  57 | 0.52 |
| 1 | 1  3  8 | ALB | Albania | Tir an a | Eur ope | 2842  321 | 2866  849 | 2882  481 | 2913  399 | 3182  021 | 3295  066 | 2941  651 | 2324  731 | 287  48 | 98.  870  2 | 0.  99  57 | 0.04 |
| 2 | 3  4 | DZA | Algeria | Alg ier s | Afric a | 4490  3225 | 4345  1666 | 3954  3154 | 3585  6344 | 3077  4621 | 2551  8074 | 1873  9378 | 1379  5915 | 238  174  1 | 18.  853  1 | 1.  01  64 | 0.56 |
| 3 | 2  1  3 | ASM | America nSamoa | Pa go Pa go | Oce ania | 4427  3 | 4618  9 | 5136  8 | 5484  9 | 5823  0 | 4781  8 | 3288  6 | 2707  5 | 199 | 222  .47  74 | 0.  98  31 | 0.00 |
| 4 | 2  0 | AN | Andorra | An dor | 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 |  |

**DataCleaning**

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! |  |  |
| print(f"Amount of duplicates: | **{**df.duplicated().sum()**}**") | In [6]: |

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',

'1980Population','1970Population','Area(km²)','Density(perkm²)', 'Growth Rate', 'World Population Percentage'],

dtype='object')

In [8]:

*#Drop'CCA3'and'Capital'columnssincewewon'tbeusingthemintheanalysis*

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

In [9]:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| df.head()  Out[9]: | | | | | | | | | | | | | | | | |
|  | 3 | 2  1  3 | American Samoa | Oce ania | 4427  3 | 4618  9 | 5136  8 | 5484  9 | 5823  0 | 4781  8 | 3288  6 | 2707  5 | 199 | 222.  4774 | 0.9  83  1 | 0.00 |
|  | 4 | 2  0  3 | Andorra | Euro pe | 7982  4 | 7770  0 | 7174  6 | 7151  9 | 6609  7 | 5356  9 | 3561  1 | 1986  0 | 468 | 170.  5641 | 1.0  10  0 | 0.00 |
| In [10]:  df.tail()  Out[10]: | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | R  a nk | Country/ Territory | Cont inent | 2022  Popul ation | 2020  Popul ation | 2015  Popul ation | 2010  Popul ation | 2000  Popul ation | 1990  Popul ation | 1980  Popul ation | 1970  Popul ation | Are a (km  ²) | Den sity (per km²  ) | Gr owt h Rat e | World Popul ation Perce ntage |
| 2  2  9 | 2  2  6 | Wallis and Futuna | Oce ania | 1157  2 | 1165  5 | 1218  2 | 1314  2 | 1472  3 | 1345  4 | 1131  5 | 9377 | 142 | 81.  493  0 | 0.9  95  3 | 0.00 |
| 2  3 | 7 | Zimbabw | Afric | 1632 | 1566 | 1415 | 1283 | 1183 | 1011 | 7049 | 5202 | 390 | 41.  766 | 1.0  20 | 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]:

*#Createthebarchart*

fig= px.bar(

countries\_by\_continent, x='Continent', y='count', color='Continent', text='count',

title='NumberofCountriesbyContinent', color\_discrete\_sequence=custom\_palette

)

*#Customizethelayout*

fig.update\_layout( xaxis\_title='Continents', yaxis\_title='NumberofCountries',

plot\_bgcolor='rgba(0,0,0,0)', *#Setthebackgroundcolortotransparent*

font\_family='Arial', *#Setfontfamily*

title\_font\_size=20 *#Settitlefontsize*

)

*#Showtheplot*

fig.show()

575050402314AfricaAsiaEuropeNorth AmericaOceaniaSouth America0102030405060

ContinentAfricaAsiaEuropeNorthAmericaOceaniaSouthAmericaNumberofCountriesby ContinentContinentsNumber of Countries

In [14]:

continent\_population\_percentage=df.groupby('Continent')['WorldPopulation Percentage'].sum().reset\_index()

In [15]:

*#Createthepiechart*

fig=go.Figure(data=[go.Pie(labels=continent\_population\_percentage['Continent'], values=continent\_population\_percentage['World Population Percentage'])])

*#Updatelayout*

fig.update\_layout(

title='WorldPopulationPercentagebyContinent', template='plotly',

paper\_bgcolor='rgba(255,255,255,0)', *#Setthepaperbackgroundcolorto*

*transparent*

plot\_bgcolor='rgba(255,255,255,0)' *#Settheplotbackgroundcolorto transparent*

)

*#Updatepiecolors*

fig.update\_traces(marker=dict(colors=custom\_palette,line=dict(color='#FFFFFF', width=1)))

*#Showtheplot*

fig.show()

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

AsiaAfricaEuropeNorthAmericaSouthAmericaOceaniaWorldPopulationPercentagebyContinent

*#MelttheDataFrametohavealongformat*

df\_melted = df.melt(id\_vars=['Continent'],

value\_vars=['2022 Population', '2020 Population', '2015

In [16]:

Population',

Population',

'2010 Population', '2000 Population', '1990

'1980Population','1970Population'], var\_name='Year',

value\_name='Population')

*#Convert'Year'toamoresuitableformat*

df\_melted['Year']= df\_melted['Year'].str.split().str[0].astype(int)

*#Aggregatepopulationbycontinentandyear* 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

ContinentAfricaAsiaEuropeNorthAmericaOceaniaSouthAmericaPopulationTrendsbyContinentOver TimeYearPopulation

**WorldPopulationComparison:1970to2020**

In [18]:

features=['1970Population','2020Population'] 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()

0100M200M300M400M500M600M700M800M1970Population1970Population

00.2B0.4B0.6B0.8B1B1.2B1.4B2020Population2020Population

Overthepast50years,Indiahasexperiencedaremarkablepopulationgrowth.In1970,thepopulationofIndia 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.

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

In [19]:

fig=px.bar(x=growth.index, y=growth.values, text=growth.values, color=growth.values,

title='GrowthOfPopulationFrom1970to2020(Top8)', template='plotly\_white')

fig.update\_layout(xaxis\_title='Country',

yaxis\_title='Population Growth')

In [20]:

fig.show()

859671872603352887176533990162971948160272945137961517118943623103644512IndiaChi

naPakistanNigeriaIndonesiaUnited

StatesBrazilBangladesh0100M200M300M400M500M600M700M800M900M

200M300M400M500M600M700M800McolorGrowthOfPopulationFrom1970to2020(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] *#Extracttheyearfromthefeaturename*

fig= px.bar(x=feature\_data.index,

y=feature\_data.values, text=feature\_data.values, color=feature\_data.values,

title=f'Top8MostPopulatedCountries(**{**year**}**)', template='plotly\_white')

fig.update\_layout(xaxis\_title='Country',

yaxis\_title='Population Growth')

fig.show()

8225344505575013012003283401300930101152283941054168399636987578294583ChinaIndia

United StatesRussiaIndonesiaJapanBrazilGermany0100M200M300M400M500M600M700M800M

100M200M300M400M500M600M700M800McolorTop8MostPopulatedCountries (1970)CountryPopulation Growth

14258873371417173173338289857275501339235824862218541212215313498171186372ChinaI

ndiaUnited StatesIndonesiaPakistanNigeriaBrazilBangladesh00.2B0.4B0.6B0.8B1B1.2B1.4B

0.2B0.4B0.6B0.8B1B1.2B1.4BcolorTop8MostPopulatedCountries(2022)CountryPopulation Growth

**WorldPopulationGrowthRates:TheFastestGrowingCountries**

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):

*#Calculatethenumberofrowsneeded*

n\_cols = 2

n\_rows = (len(countries) + n\_cols - 1) // n\_cols

*#Createsubplots*

fig = sp.make\_subplots(rows=n\_rows, cols=n\_cols, subplot\_titles=countries,

horizontal\_spacing=0.1, vertical\_spacing=0.1)

fori, country **in** enumerate(countries, start=1):

*#Filterdatafortheselectedcountry*

country\_df = df[df['Country/Territory'] == country]

*#MelttheDataFrametohavealongformat*

country\_melted = country\_df.melt(id\_vars=['Country/Territory'],

value\_vars=['2022 Population', '2020 Population', '2015

Population',

Population',

'2010 Population', '2000 Population', '1990

'1980Population','1970Population'], var\_name='Year',

value\_name='Population')

*#Convert'Year'toamoresuitableformat*

country\_melted['Year'] = country\_melted['Year'].str.split().str[0].astype(int)

*#Createalineplotforeachcountry*

line\_fig=px.line(country\_melted,x='Year',y='Population', color='Country/Territory',

labels={'Population':'Population','Year':'Year'}, color\_discrete\_sequence=custom\_palette)

*#Updatethelineplottofitthesubplot*

row=(i-1)//n\_cols+1 col = (i - 1) % n\_cols + 1 fortrace**in**line\_fig.data:

fig.add\_trace(trace, row=row, col=col)

*#Updatethelayoutofthesubplots*

fig.update\_layout(

title='PopulationTrendsofSelectedCountriesOverTime', template='plotly\_white',

font\_family='Arial', title\_font\_size=20, showlegend=False,

height=600\*n\_rows, *#Adjustheightforbiggerplots*

)

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','GrowthRate']].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'])

1970198019902000201020203.2M3.4M3.6M3.8M4M4.2M4.4M19701980199020002010202033M34 M35M36M37M38M39M40M1970198019902000201020205M10M15M20M25M197019801990200020

1020206M8M10M12M14M16M18M20M22M1970198019902000201020204.6M4.8M5M5.2M5.4M5.6 M19701980199020002010202020M30M40M50M60M70M80M90M100M

PopulationTrendsofSelectedCountriesOver

TimeYearYearYearYearYearYearPopulationPopulationPopulationPopulationPopulationPopulationMoldov aPolandNigerSyriaSlovakiaDR Congo

**WorldPopulationGrowthRates:TheSlowestGrowingCountries**

In [27]:

slowest=top\_slowest[['Country/Territory','GrowthRate']].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','AmericanSamoa', 'Lebanon', 'Ukraine'])

1970198019902000201020201.9M2M2.1M2.2M2.3M2.4M2.5M2.6M2.7M197019801990200020102

0202.8M3M3.2M3.4M3.6M3.8M1970198019902000201020207M7.5M8M8.5M9M197019801990200

02010202030k35k40k45k50k55k1970198019902000201020202.5M3M3.5M4M4.5M5M5.5M6M6.5 M19701980199020002010202040M42M44M46M48M50M52M

PopulationTrendsofSelectedCountriesOver

TimeYearYearYearYearYearYearPopulationPopulationPopulationPopulationPopulationPopulationLatvia LithuaniaBulgariaAmerican SamoaLebanonUkraine

**LandAreabyCountry**

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 [29]:

In [30]:

*#Createsubplots*

fig=sp.make\_subplots(rows=1,cols=2,subplot\_titles=("CountrieswithMostLand", "Countries with Least Land"))

*#Plotcountrieswiththemostland*

fig.add\_trace(go.Bar(x=most\_land.index,y=most\_land.values,name='MostLand', marker\_color=custom\_palette[0]), row=1, col=1)

*#Plotcountrieswiththeleastland*

fig.add\_trace(go.Bar(x=least\_land.index,y=least\_land.values,name='LeastLand', marker\_color=custom\_palette[1]), row=1, col=2)

fig.update\_layout(

title\_text="GeographicalDistributionofLandAreabyCountry", 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()

RussiaCanadaChinaUnitedStatesBrazil02M4M6M8M10M12M14M16MSaint BarthelemyTokelauGibraltarMonacoVatican City05101520

GeographicalDistributionofLandAreabyCountryArea(km²)Area(km²)CountrieswithMost LandCountries with Least Land

**LandAreaPerPersonbyCountry**

In [31]:

df['Areaper 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

*#Createsubplots*

fig=sp.make\_subplots(rows=1,cols=2,subplot\_titles=("CountrieswithMostLand Available Per Capita", "Countries with Least Land Available Per Capita"))

*#Plotcountrieswiththemostland* 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)

*#Plotcountrieswiththeleastland*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="DistributionofAvailableLandAreabyCountryPerCapita", showlegend=False,

template='plotly\_white'

)

fig.update\_yaxes(title\_text="LandAvailablePerPerson",row=1,col=1) fig.update\_yaxes(title\_text="LandAvailable Per Person", row=1, col=2)

fig.show()