

Introduction:

A world population dataset is critical for government organisations in deciding policies for their populations, whether in a country, continent, or the entire planet. Each region must be aware of the data gathered from this dataset, which can take the form of population counts, growth rates, and population density over time. The extraction and analysis of this dataset will aid in the planning of how each area will be administered, including health, economy, education, and others.

```
import numpy as np
import pandas as pd

import plotly
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots

import warnings
warnings.filterwarnings("ignore")
import jovian
```

About the Dataset

In this dataset, we have historical population data for every country/territory in the world by different parameters. This dataset is created from 2022 World Population by Country.

Rank: Rank by Population.

CCA3: 3 Digit Country/Territories Code.

Country: 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.

Data Preprocessing

```
df_population = pd.read_csv("world_population.csv")
display(df_population.head())
```

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	1990 Population
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	33753499	28189672	19542982	106947
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	2882481	2913399	3182021	32950
2	34	DZA	Algeria	Algiers	Africa	44903225	43451666	39543154	35856344	30774621	255180
3	213	ASM	American Samoa	Pago Pago	Oceania	44273	46189	51368	54849	58230	478
4	203	AND	Andorra	Andorra la Vella	Europe	79824	77700	71746	71519	66097	535

```
df_population.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                              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
```

There are 17 columns and 234 rows. The unique CCA3, nation, and capital are listed in each row.

This dataset includes statistics from the following 8 population years: 1970, 1980, 1990, 2000, 2010, 2015, 2020, and 2022.

Country, Capital, Continent, and CCA3 are the 4 categorical features. The remaining features are numerical.

No values are missing.

Feature Engineering

On the basis of the current dataset, we will construct demographic estimates to indicate how the population might increase in the future. In addition, in order to properly understand how the population changes from year to year, we will also estimate the rate of population increase in the years where this data is unknown.

Ref:

https://www.gerhardbechtold.com/LUPMIS/Manual/a151_population_projected_and_planned_population_.html

```
def cal_population_projection(df, start_year, target_year):
    start_year_pop = str(start_year) + " Population"
    target_year_pop = str(target_year) + " Population"
    start_year_gr = str(start_year) + " Growth Rate"
    df[target_year_pop] = df[start_year_pop] * ((1 + (df[start_year_gr]/100)) ** (target_year - start_year))
    df[target_year_pop] = df[target_year_pop].astype(int)
    return df

def cal_gr_estimation(df, start_year, target_year):
    start_year_pop = str(start_year) + " Population"
    target_year_pop = str(target_year) + " Population"
    target_year_gr = str(target_year) + " Growth Rate"
    df[target_year_gr] = ((df[target_year_pop]/df[start_year_pop]) ** (1/(target_year - start_year)))
    return df

def cal_density(df, year):
    den_year = str(year) + " Density (per km²)"
    pop_year = str(year) + " Population"
    df[den_year] = df[pop_year] / df["Area (km²)"]
    return df

def cal_pop_percentage(df, year):
    pop_percentage_year = str(year) + " World Population Percentage"
    pop_year = str(year) + " Population"
    df[pop_percentage_year] = (df[pop_year] / df[pop_year].sum()) * 100
    return df
```

```

df_population = df_population.rename(columns={"Growth Rate": "2022 Growth Rate"})
df_population["1970 Growth Rate"] = "Unknown"
df_population["2022 Growth Rate"] = (df_population["2022 Growth Rate"] - 1) * 100
df_population = df_population.rename(columns={"Density (per km²)": "2022 Density (per km²)"})
df_population = df_population.rename(columns={"World Population Percentage": "2022 World Population Percentage"})

# Calculate Population Projection
target_year = [2030, 2040, 2050]
for year in target_year:
    df_population = cal_population_projection(df_population, 2022, year)

# Calculate Growth Rate
start_year = [1970, 1980, 1990, 2000, 2010, 2015, 2022, 2022, 2022]
target_year = [1980, 1990, 2000, 2010, 2015, 2020, 2030, 2040, 2050]
for i in range(len(target_year)):
    df_population = cal_gr_estimation(df_population, start_year[i], target_year[i])

target_year = [1970, 1980, 1990, 2000, 2010, 2015, 2020, 2030, 2040, 2050]
# Calculate Density
for i in range(len(target_year)):
    df_population = cal_density(df_population, target_year[i])

# Calculate World Population Percentage
for i in range(len(target_year)):
    df_population = cal_pop_percentage(df_population, target_year[i])

```

```
display(df_population.head())
```

	Rank	CCA3	Country	Capital	Continent	2022 Population	2020 Population	2015 Population	2010 Population	2000 Population	...	Pop Perc
0	36	AFG	Afghanistan	Kabul	Asia	41128771	38972230	33753499	28189672	19542982	...	0.2
1	138	ALB	Albania	Tirana	Europe	2842321	2866849	2882481	2913399	3182021	...	0.0
2	34	DZA	Algeria	Algiers	Africa	44903225	43451666	39543154	35856344	30774621	...	0.3
3	213	ASM	American Samoa	Pago Pago	Oceania	44273	46189	51368	54849	58230	...	0.0
4	203	AND	Andorra	Andorra la Vella	Europe	79824	77700	71746	71519	66097	...	0.0

5 rows × 50 columns

```
df_population.columns
```

```

Index(['Rank', 'CCA3', 'Country', 'Capital', 'Continent', '2022 Population',
      '2020 Population', '2015 Population', '2010 Population',
      '2000 Population', '1990 Population', '1980 Population',
      '1970 Population', 'Area (km²)', '2022 Density (per km²)',
      '2022 Growth Rate', '2022 World Population Percentage',
      '1970 Growth Rate', '2030 Population', '2040 Population',

```

```
'2050 Population', '1980 Growth Rate', '1990 Growth Rate',
'2000 Growth Rate', '2010 Growth Rate', '2015 Growth Rate',
'2020 Growth Rate', '2030 Growth Rate', '2040 Growth Rate',
'2050 Growth Rate', '1970 Density (per km²)', '1980 Density (per km²)',
'1990 Density (per km²)', '2000 Density (per km²)',
'2010 Density (per km²)', '2015 Density (per km²)',
'2020 Density (per km²)', '2030 Density (per km²)',
'2040 Density (per km²)', '2050 Density (per km²)',
'1970 World Population Percentage', '1980 World Population Percentage',
'1990 World Population Percentage', '2000 World Population Percentage',
'2010 World Population Percentage', '2015 World Population Percentage',
'2020 World Population Percentage', '2030 World Population Percentage',
'2040 World Population Percentage', '2050 World Population Percentage'],
dtype='object')
```

```
pop_col = [x for x in df_population.columns.tolist() if x[5:]=="Population"]
clean_pop_col = [int(x.replace(" Population", "")) for x in pop_col]
country_pop_col = [x for x in df_population.columns.tolist() if x=="Country" or x[5:]=="

# Growth Rate Column
gr_col = [x for x in df_population.columns.tolist() if x[5:]=="Growth Rate"]
clean_gr_col = [int(x.replace(" Growth Rate", "")) for x in gr_col]
country_gr_col = [x for x in df_population.columns.tolist() if x=="Country" or x[5:]=="
```

Exploratory Data Analysis

World Population

```
total_population = df_population[pop_col].sum(axis=0)
total_population.index = clean_pop_col
total_population = total_population.sort_index()

display(total_population)

x = [1970, 2022, 2050]
y = [total_population[v] for v in x]
```

```
1970      3694136661
1980      4442400371
1990      5314191665
2000      6147055703
2010      6983784998
2015      7424809761
2020      7839250603
2022      7973413042
2030      8561775145
2040      9465265091
2050     10592457303
dtype: int64
```

```

fig = px.area(total_population, color_discrete_sequence=[ '#3455EB' ])

fig.update_traces(patch={"line": {"width": 3}})

fig.add_trace(
    go.Scatter(x=x,
               y=y,
               mode='markers',
               marker=dict(
                   color="#0926AD",
                   size=12,
               )
    )
)

fig.add_vrect(x0=2023, x1=2055,
              fillcolor="#dbd9d9", opacity=0.5,
              layer="below", line_width=0,
)

fig.add_annotation(yref='paper',
                   x=1994, y=-0.15,
                   text="<b>Actual<b>",
                   showarrow=False,
)

fig.add_annotation(yref='paper',
                   x=2037, y=-0.15,
                   text="<b>Forecast<b>",
                   showarrow=False,
)

fig.update_xaxes(title_text="",
                 range=[1968,2052],
                 showgrid=False,
                 linecolor='black',
                 ticks='outside',
)

fig.update_yaxes(title_text="",
                 range=[3000000000,11000000000],
                 showgrid=False,
                 linecolor='black'
)

fig.update_layout(title_text='World Population Growth, 1970-2050',
                  width=1000,
                  height=450,
                  plot_bgcolor='white',
                  showlegend=False,
)

```

```
fig.show()
```

Observations:

The population of the planet is growing yearly.

By 2050, it is anticipated that there will be 10 billion people on the planet.

```
def draw_subplots(list_data, title):
    color = ["red", "orange", "green", "blue", "purple", "indigo"]
    list_mark_year = [1970, 2022, 2050]
    list_mark_year_gr = [1980, 2022]

    # Create Subplots
    fig = make_subplots(
        rows=2, cols=2,
        specs=[[{"colspan": 2}, None],
               [{}, {}]],
        subplot_titles=("Population Growth, 1970-2050", title+" in 2022", "Growth Rate,
    )

    # Population Growth
    for i, col in enumerate(list_data[0].columns):
        fig.add_trace(go.Scatter(
            x=list_data[0][col].index.tolist(),
            y=list_data[0][col].values,
            name=col,
            mode='lines',
            marker_color=color[i]
        ),
        row=1, col=1,
    )

    fig.add_trace(go.Scatter(
        x=list_mark_year,
        y=list_data[0].loc[list_mark_year, col].values,
        mode='markers',
        marker=dict(
            color=color[i],
            size=8,
        ),
        showlegend=False
    ),
    row=1, col=1
    )

    # Add Gray Rectangular
    fig.add_vrect(x0=2023, x1=2055,
                  fillcolor="#dbd9d9", opacity=0.5,
                  layer="below", line_width=0,
```

```

        xref='x1', yref='y1'
    )

    # Add Annotation Actual and Forecast
    fig.add_annotation(xref='x1', yref='paper', x=1994, y=0.5, text="<b>Actual<b>", showlegend=False)
    fig.add_annotation(xref='x1', yref='paper', x=2037, y=0.5, text="<b>Forecast<b>", showlegend=False)

    # Bar Chart
    fig.add_trace(go.Bar(x=list_data[1].index, y=list_data[1].values, marker_color=colors,
                        row=2, col=1)
    )

    # Growth Rate
    for i, col in enumerate(list_data[2].columns):
        fig.add_trace(go.Scatter(
            x=list_data[2][col].index.tolist(),
            y=list_data[2][col].values,
            name=col,
            mode='lines',
            marker_color=colors[i],
            showlegend=False
        ),
        row=2, col=2,
    )

    fig.add_trace(go.Scatter(
        x=list_mark_year_gr,
        y=list_data[2].loc[list_mark_year_gr, col].values,
        mode='markers',
        marker=dict(
            color=colors[i],
            size=7,
        ),
        showlegend=False
    ),
    row=2, col=2
    )

    # Update X Axis
    fig.update_xaxes(range=[1968, 2052], showgrid=False, linecolor='black', ticks='outside')
    fig.update_xaxes(showgrid=False, linecolor='black', ticks='outside', row=2, col=1)
    fig.update_xaxes(range=[1978, 2024], showgrid=False, linecolor='black', ticks='outside')

    # Update Y Axis
    fig.update_yaxes(domain=[0.575, 1.0], showgrid=False, linecolor='black', row=1, col=1)
    fig.update_yaxes(domain=[0.0, 0.325], row=2, col=1)
    fig.update_yaxes(domain=[0.0, 0.325], showgrid=False, linecolor='black', row=2, col=1)

    # Update Layout
    fig.update_layout(title_text=title,
                      width=1000,
                      height=800,

```



```

        plot_bgcolor='white',
        showlegend=True,
        legend={"itemsizing": "constant"}
    )

    fig.show()

```

Continent Population

```

continent_population = df_population.groupby("Continent")[pop_col].sum()
continent_population = continent_population.transpose()
continent_population.index = clean_pop_col
continent_population = continent_population.sort_index()
continent_population = continent_population[["Asia", "Africa", "Europe", "North America", "South America", "Oceania"]]

population_2022 = continent_population.loc[2022, :]

df_gr = df_population.groupby("Continent")[gr_col].mean()
df_gr.insert(1, "1970 Growth Rate", [0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
df_gr = df_gr.transpose()
df_gr.index = clean_gr_col
df_gr = df_gr.sort_index()
df_gr = df_gr[["Asia", "Africa", "Europe", "North America", "South America", "Oceania"]]
df_gr = df_gr[1:8]

list_data = [continent_population, population_2022, df_gr]

draw_subplots(list_data, "Continent Population")

```

Observations:

Year after year, Asia holds the top spot for the continent with the highest population. By 2050, the population is anticipated to reach 5 billion.

Africa's population will more than treble from 2022 to 2050.

Other continents typically maintain populations of no more than 1 billion people.

Top 7 Most Populous Countries in Consecutive Years

```

year = [1970, 1980, 1990, 2000, 2010, 2015, 2020, 2022, 2030, 2040, 2050]

fig = make_subplots(rows=4, cols=3, vertical_spacing=0.105)

df_high_pop_year_0 = None
j = 0
most = 7
for i in range(12):
    if(i==2):
        continue

```

```

pop_year = str(year[j]) + " Population"
j += 1

df_high_pop_year_1 = df_population.sort_values(by=pop_year, ascending=False).reset_
colors = ["darkgray"] * most
if(df_high_pop_year_0 is not None):
    for x in range(most):
        country_name_1 = df_high_pop_year_1.loc[x, "Country"]
        if(country_name_1 in df_high_pop_year_0["Country"].tolist()):
            country_row_0 = df_high_pop_year_0[df_high_pop_year_0["Country"]==count
            country_index_0 = country_row_0.index.values[0]
            if(x < country_index_0):
                colors[x] = "#22F028" # Increase
            elif(x > country_index_0):
                colors[x] = "#F03D22" # Decrease
        else:
            colors[x] = "#22F028" # Increase

df_high_pop_year_0 = df_high_pop_year_1.copy()

fig.add_trace(
    go.Bar(x=df_high_pop_year_1['CCA3'],
           y=df_high_pop_year_1[pop_year],
           marker_color=colors,
           ),
    row=(i//3)+1, col=(i%3)+1
)

fig.update_xaxes(title_text=year[j-1],
                 showgrid=False,
                 linecolor='black',
                 ticks='outside',
                 row=(i//3)+1, col=(i%3)+1
)

fig.update_layout(height=1100, width=1000,
                 title_text="Top " + str(most) + " Most Populous Countries in Consecuti
                 plot_bgcolor='white',
                 showlegend=False
)

fig.show()

```

Observation:

China and India are the two nations with the largest populations, respectively. India only replaces China as the top nation starting in 2030.

The United States consistently holds the third-most populous country position. USA's position dropped to rank 5 in 2050.

Russia was placed fourth in 1970. Every ten years, the rating drops one spot. From 2010 on, Russia was no longer included in the top 7 rankings.

Indonesia was rated fifth in 1970. Its position has moved up to position 4 from Russia and will remain there until 2040. After being moved down the list by Pakistan and Nigeria in 2040, Indonesia will be placed sixth.

Japan came in sixth place. Its position was bumped up to seventh in 1980 by Brazil. Then, starting in 2020, Japan was no longer listed among the top 7.

Brazil initially came in at number seven in 1970. In 1990, it replaced Japan and Russia to rise steadily in the rankings to fifth place. Brazil's position decreased to seventh in 2015 and 2020, behind Pakistan and Nigeria. According to estimates, Ethiopia won't be in the top 7 ranks until 2050.

Pakistan quickly climbed the rankings in 2000, moving up to sixth place. Pakistan moved up to fifth place in 2015, passing Brazil. The ranking then increased to position 4 in 2050.

In 2010, Nigeria has only recently risen to the top 7. Nigeria's population is increasing quickly from year to year, which is causing other nations' rankings to change. By 2050, Nigeria is anticipated to overtake China as the third-most populated nation.

In 2050, Ethiopia is anticipated to replace Brazil in the top 7.

Top 5 Most Populous Countries

```
df_most_population = df_population.sort_values(by="Rank")[:5][country_pop_col]
df_most_population = df_most_population.set_index("Country").transpose()
df_most_population.index = clean_pop_col
df_most_population = df_most_population.sort_index()

population_2022 = df_most_population.loc[2022, :]

df_gr = df_population.sort_values(by="Rank")[:5][country_gr_col]
df_gr = df_gr.set_index("Country").transpose()
df_gr.index = clean_gr_col
df_gr = df_gr.sort_index()
df_gr = df_gr[1:8]

list_data = [df_most_population, population_2022, df_gr]

draw_subplots(list_data, "Top 5 Most Populous Countries")
```

Observations:

China, India, the United States, Indonesia, and Pakistan will be the top 5 most populous nations by 2022.

Significantly, China and India have substantially larger populations than other nations. In actuality, they now number 1 billion individuals.

In the future, India will surpass China to win the top spot.

Top 5 Least Populous Countries

```

df_least_population = df_population.sort_values(by="Rank", ascending=False)[:5][country]
df_least_population = df_least_population.set_index("Country").transpose()
df_least_population.index = clean_pop_col
df_least_population = df_least_population.sort_index()

population_2022 = df_least_population.loc[2022, :]

df_gr = df_population.sort_values(by="Rank", ascending=False)[:5][country_gr_col]
df_gr = df_gr.set_index("Country").transpose()
df_gr.index = clean_gr_col
df_gr = df_gr.sort_index()
df_gr = df_gr[1:8]

list_data = [df_least_population, population_2022, df_gr]

draw_subplots(list_data, "Top 5 Least Populous Countries")

```

Observations:

By 2022, Vatican City, Tokleau, Niue, the Falkland Islands, and Montserrat will be the 5 least populous nations in a row.

There are currently no more than 5000 people living there. In actuality, there are only a little more than 1000 inhabitants of Vatican City.

More than 10,000 people used to live in Montserrat. But the population shrank to less than 5000 individuals.

Top 5 Countries with the Highest Growth Rates

```

df_highest_gr = df_population.sort_values(by="2022 Growth Rate", ascending=False)[:5][c
df_highest_gr = df_highest_gr.set_index("Country").transpose()
df_highest_gr.index = clean_pop_col
df_highest_gr = df_highest_gr.sort_index()

df_gr = df_population.sort_values(by="2022 Growth Rate", ascending=False)[:5][country_g
df_gr = df_gr.set_index("Country").transpose()
df_gr.index = clean_gr_col
df_gr = df_gr.sort_index()
df_gr = df_gr[1:8]

gr_2022 = df_gr.loc[2022, :]

list_data = [df_highest_gr, gr_2022, df_gr]

draw_subplots(list_data, "Top 5 Countries with the Highest Growth Rates")

```

Observations:

Moldova, Poland, Niger, Syria, and Slovakia will have the five greatest growth rates by 2022.

If this trend persists, Moldova's population will increase fivefold by 2050.

If the growth rate does not alter, Poland's population will increase quickly and reach 100 million people in the future.

Top 5 Countries with the Lowest Growth Rate

```
df_lowest_gr = df_population.sort_values(by="2022 Growth Rate")[:5][country_pop_col]
df_lowest_gr = df_lowest_gr.set_index("Country").transpose()
df_lowest_gr.index = clean_pop_col
df_lowest_gr = df_lowest_gr.sort_index()

df_gr = df_population.sort_values(by="2022 Growth Rate")[:5][country_gr_col]
df_gr = df_gr.set_index("Country").transpose()
df_gr.index = clean_gr_col
df_gr = df_gr.sort_index()
df_gr = df_gr[1:8]

gr_2022 = df_gr.loc[2022, :]

list_data = [df_lowest_gr, gr_2022, df_gr]

draw_subplots(list_data, "Top 5 Countries with the Lowest Growth Rates")
```

Observations:

By 2022, Ukraine, Lebanon, American Samoa, Bulgaria, and Lithuania will have the five lowest growth rates.

Future predictions indicate that this nation's population will decrease overall.

Based on current growth rates, Ukraine's population will decline significantly to reach 3 million people.

Top 5 Countries with the Highest Density

```
df_highest_density = df_population.sort_values(by="2022 Density (per km²)", ascending=False)
df_highest_density = df_highest_density.set_index("Country").transpose()
df_highest_density.index = clean_pop_col
df_highest_density = df_highest_density.sort_index()

df_gr = df_population.sort_values(by="2022 Density (per km²)", ascending=False)[:5][country_gr_col]
df_gr = df_gr.set_index("Country").transpose()
df_gr.index = clean_gr_col
df_gr = df_gr.sort_index()
df_gr = df_gr[1:8]

density_2022 = df_population.sort_values(by="2022 Density (per km²)", ascending=False)[
density_2022.index = df_highest_density.columns

list_data = [df_highest_density, density_2022, df_gr]

draw_subplots(list_data, "Top 5 Countries with the Highest Density")
```

Observations:

By 2022, Macao, Monaco, Singapore, Hong Kong, and Gibraltar will have the greatest densities.

Currently, there are more than 5 million people living in Singapore and Hong Kong. Singapore's population will eventually converge with Hong Kong's ageing population.

As opposed to Macau, Monaco, and Gibraltar, this. Less than a million people live there.

It will be challenging to identify in visualisation if there is a wide population range because the population difference between Monaco and Gibraltar is extremely modest from year to year and the population is not more than 50000 people.

Top 5 Countries with the Lowest Density

```
df_lowest_density = df_population.sort_values(by="2022 Density (per km²)")[:5][country_
df_lowest_density = df_lowest_density.set_index("Country").transpose()
df_lowest_density.index = clean_pop_col
df_lowest_density = df_lowest_density.sort_index()

df_gr = df_population.sort_values(by="2022 Density (per km²)")[:5][country_gr_col]
df_gr = df_gr.set_index("Country").transpose()
df_gr.index = clean_gr_col
df_gr = df_gr.sort_index()
df_gr = df_gr[1:8]

density_2022 = df_population.sort_values(by="2022 Density (per km²)")[:5]["2022 Density
density_2022.index = df_lowest_density.columns

list_data = [df_lowest_density, density_2022, df_gr]

draw_subplots(list_data, "Top 5 Countries with the Lowest Density")
```

Observations:

By 2022, Greenland, the Falkland Islands, Western Sahara, Mongolia, and Namibia will be the five nations with the lowest densities.

Greenland, the nation with the lowest population density, only has less than 100000 inhabitants and will keep doing so in the future.

Falkland Islands also contain the top 5 nations with the lowest populations, in addition to the top 5 nations with the lowest densities.

The populations of Namibia and Mongolia are expanding quickly, and they are likely to leave the top 5 countries with the lowest densities.

```
data_slider = []
years = [1970, 1980, 1990, 2000, 2010, 2015, 2020, 2022, 2030, 2040, 2050]
```

```
colors = [[0, '#61fa70'],[0.2, '#235927'],[0.4, '#19401b'], [0.6, '#143317'],[0.8, '#0f1907'],[1, '#003300']]
df_population['1970 Growth Rate'] = 0.0
```

```
# Data Object
```

```
for year in years:
    str_year_pop = str(year) + " Population"
    str_year_gr = str(year) + " Growth Rate"
    str_year_den = str(year) + " Density (per km2)"
    str_year_pop_per = str(year) + " World Population Percentage"

    df_population['Text'] = 'Capital: ' + df_population['Capital'] + '<br>' + \
        'Continent: ' + df_population['Continent'] + '<br>' + \
        'Area (km2): ' + df_population['Area (km2)'].apply(str) + ' ' + \
        'Density (per km2): ' + df_population[str_year_den].apply(str) + ' ' + \
        'Growth Rate: ' + df_population[str_year_gr].apply(str) + ' ' + \
        'World Population Percentage: ' + df_population[str_year_pop_per].apply(str)

    data_year = dict(
        type='choropleth',
        colorscale = colors,
        locations = df_population['Country'],
        z = df_population[str_year_pop].astype(float),
        locationmode = 'country names',
        text = df_population['Text'],
        colorbar = dict(
            title = "Populations",
        ),
        zmin = 0.0,
        zmax = 1800000000.0
    )
```

```
data_slider.append(data_year)
```

```
# Steps for the Slider
```

```
steps = []

for i in range(len(data_slider)):
    step = dict(method='restyle',
        args=['visible', [False] * len(data_slider)],
        label=str(years[i]))
    step['args'][1][i] = True
    steps.append(step)
```

```
sliders = [dict(active=10, pad={"t": 1}, steps=steps)]
```

```
layout = dict(
    title="World Population Based on Countries",
    sliders=sliders
)
```

```
fig = dict(data=data_slider, layout=layout)
```

```
plotly.offline.iplot(fig)
```

```
data_slider = []
years = [1980, 1990, 2000, 2010, 2015, 2020, 2022]
colors = [[0, '#543005'], [0.34, '#8C510A'], [0.3913, '#F5F5F5'], [0.46, '#80CDC1'], [0.5

# Data Object
for year in years:
    str_year_pop = str(year) + " Population"
    str_year_gr = str(year) + " Growth Rate"
    str_year_den = str(year) + " Density (per km2)"
    str_year_pop_per = str(year) + " World Population Percentage"

    df_population['Text'] = 'Capital: ' + df_population['Capital'] + '<br>' + \
        'Continent: ' + df_population['Continent'] + '<br>' + \
        'Population: ' + df_population[str_year_pop].apply(str) + ' ' + \
        'Area (km2): ' + df_population['Area (km2)'].apply(str) + ' ' + \
        'Density (per km2): ' + df_population[str_year_den].apply(s
        'World Population Percentage: ' + df_population[str_year_pc

    data_year = dict(
        type='choropleth',
        colorscale = colors,
        locations = df_population['Country'],
        z = df_population[str_year_gr].astype(float),
        locationmode = 'country names',
        text = df_population['Text'],
        colorbar = dict(
            title = "Growth Rates"
        ),
        zmin = -9.0,
        zmid = 0.0,
        zmax = 14.0
    )

    data_slider.append(data_year)

# Steps for the Slider
steps = []

for i in range(len(data_slider)):
    step = dict(method='restyle',
                args=['visible', [False] * len(data_slider)],
                label=str(years[i]))
    step['args'][1][i] = True
    steps.append(step)

sliders = [dict(active=6, pad={"t": 1}, steps=steps)]
```



```
layout = dict(  
    title="World Growth Rate Based on Countries",  
    sliders=sliders  
)  
  
fig = dict(data=data_slider, layout=layout)  
  
plotly.offline.iplot(fig)
```

Conclusion

After researching and examining this information, we have a better understanding of population datasets, how the population changes in a region of a country, continent, or the world on the basis of specific criteria from year to year, and other topics. Of course, including other information in the study, such as age, gender, life expectancy, and gross domestic product, will make it more fascinating.