**NAME : Tejaswini Gokanakonda**

**Roll No : DE142**

**DATE : 16-11-2024**

**TOPIC : Population Analysis Report - Malaysia**

**Introduction**

This report provides a comprehensive analysis of Malaysia's population trends over the years. Using real-world data sourced from reliable databases, we explore various demographic metrics such as total population, distribution by age and gender, and year-over-year changes. The study demonstrates data preparation, cleaning, and analysis using Python libraries like Pandas and Matplotlib.

## **1. Dataset Selection and Source**

For this analysis, the **Malaysian Population Dataset** was chosen

Link : <https://data.gov.my/data-catalogue/population_malaysia>

## **2. Data Preparation & Cleaning**

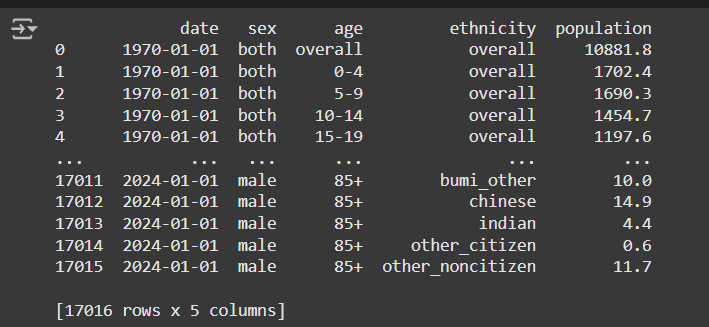
### **2.1 Loading the Dataset**

**import pandas as pd**

**df = pd.read\_csv("Downloads/population\_malaysia.csv")**

**df.head()**

**print(df)**

****

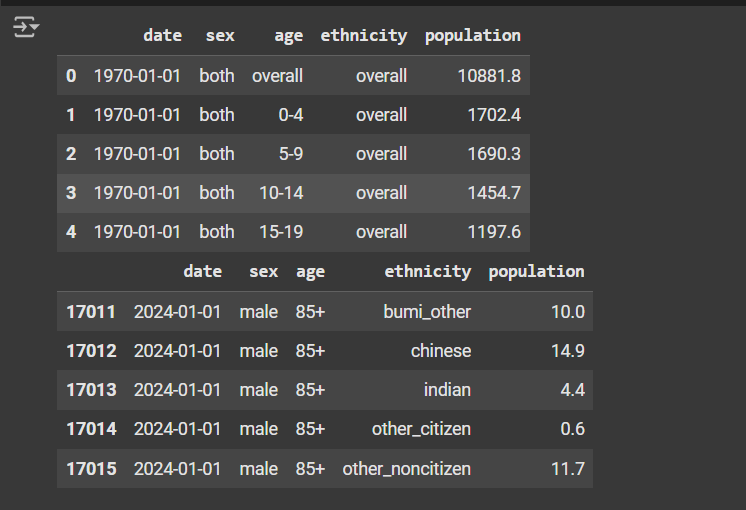
**Explanation:**

* **Pandas library is imported for data manipulation.**
* **The dataset is loaded into a DataFrame df using pd.read\_csv.**
* **df.head() provides a preview of the first 5 rows.**
* **print(df) prints the entire DataFrame (caution: use for small datasets only).**

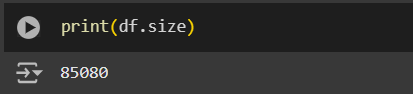
**2.2 Exploring Basic Information**

**display(df.head())**

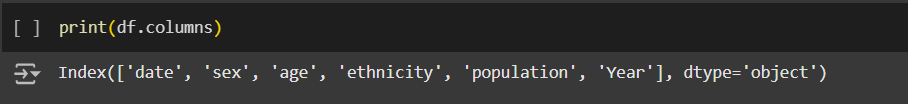
**display(df.tail())**

****

**print(df.size) # Total elements (rows × columns)**

****

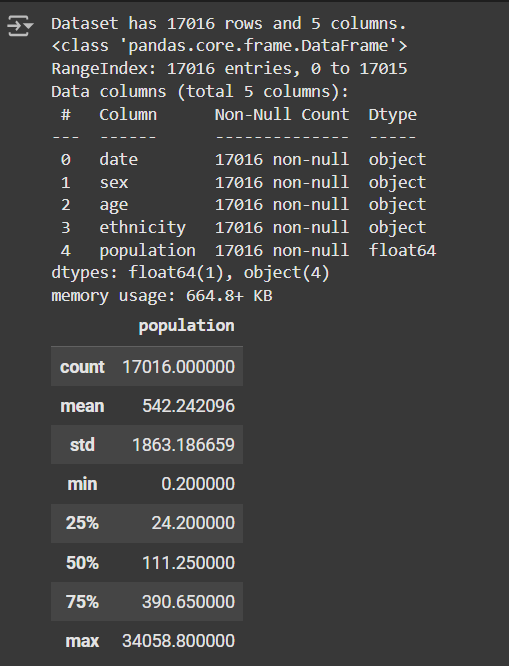
**print(df.columns) # Lists column names**

****

**print(f"Dataset has {df.shape[0]} rows and {df.shape[1]} columns.")**

**df.info() # Provides data types and null count**

**df.describe() # Descriptive statistics**

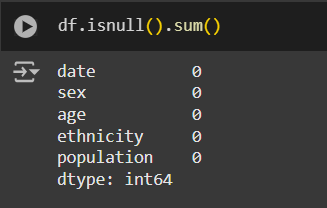
****

**Explanation:**

* **df.size returns total elements.**
* **df.columns lists all column names.**
* **df.info() provides a summary, including column data types, null values, and memory usage.**
* **df.describe() gives statistical summaries for numeric columns.**

### **2.3 Handling Missing Values**

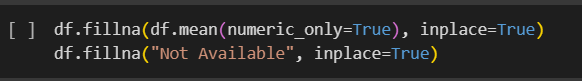
**df.isnull().sum() # Checks for missing values in each column**

****

**# Fill missing values in numeric columns with mean and in non-numeric columns with "Not Available"**

**df.fillna(df.mean(numeric\_only=True), inplace=True)**

**df.fillna("Not Available", inplace=True)**

****

**Explanation:**

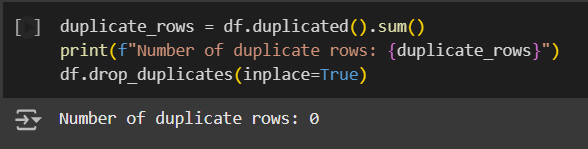
* **df.isnull().sum(): Lists the count of missing values per column.**
* **df.fillna(): Fills missing values in numeric columns with the mean, and non-numeric columns with "Not Available".**

### **2.4 Removing Duplicates**

**duplicate\_rows = df.duplicated().sum()**

**print(f"Number of duplicate rows: {duplicate\_rows}")**

**df.drop\_duplicates(inplace=True)**

****

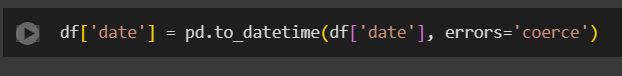
**Explanation:**

* **df.duplicated() checks for duplicates.**
* **df.drop\_duplicates(inplace=True) removes duplicates from the dataset.**

### **2.5 Converting Dates**

**df['date'] = pd.to\_datetime(df['date'], errors='coerce')**

**df['Year'] = df['date'].dt.year**

****

**Explanation:**

* **Date Conversion: Converts date column to a datetime format to ease date manipulation.**
* **Extract Year: Creates a new column Year to represent the year part of the date.**

**pd.to\_datetime(df['date']):**

Converts the 'date' column to a datetime object, enabling easier handling of dates (e.g., extracting year, month, day).

**errors='coerce':**

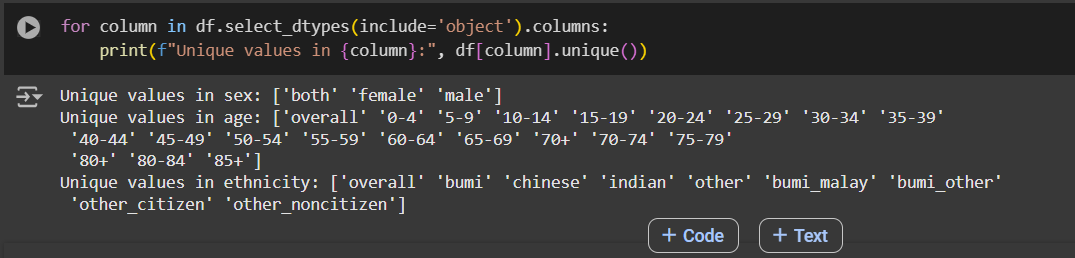
Ensures that any invalid date entries (e.g., improperly formatted strings or null values) are converted to NaT (Not a Time), representing missing or invalid datetime values.

## **Step 3: Exploratory Data Analysis (EDA)**

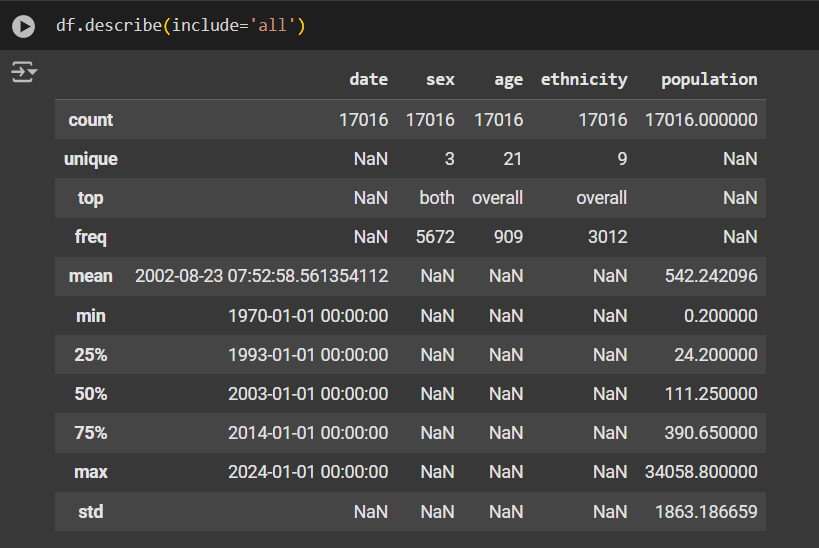
### **3.1 Summary Statistics and Unique Values**

**for column in df.select\_dtypes(include='object').columns:**

**print(f"Unique values in {column}:", df[column].unique())**

****

**df.describe(include='all')**

****

**Explanation:**

* **Lists unique values for categorical columns.**
* **Provides summary statistics for all columns, including non-numeric.**

### 

### **Distributions and Value Counts**

### **for column in df.select\_dtypes(include='object').columns:**

### **print(f"\nValue counts for {column}:\n", df[column].value\_counts())**

### 

### 

### 

### 

### 

### **3.2 Population Distributions and Visualizations**

#### **Year-wise Population Trend**

**import matplotlib.pyplot as plt**

**yearly\_population = df.groupby('Year')['population'].sum().reset\_index()**

**plt.figure(figsize=(12, 6))**

**plt.plot(yearly\_population['Year'], yearly\_population['population'], marker='o', linestyle='-')**

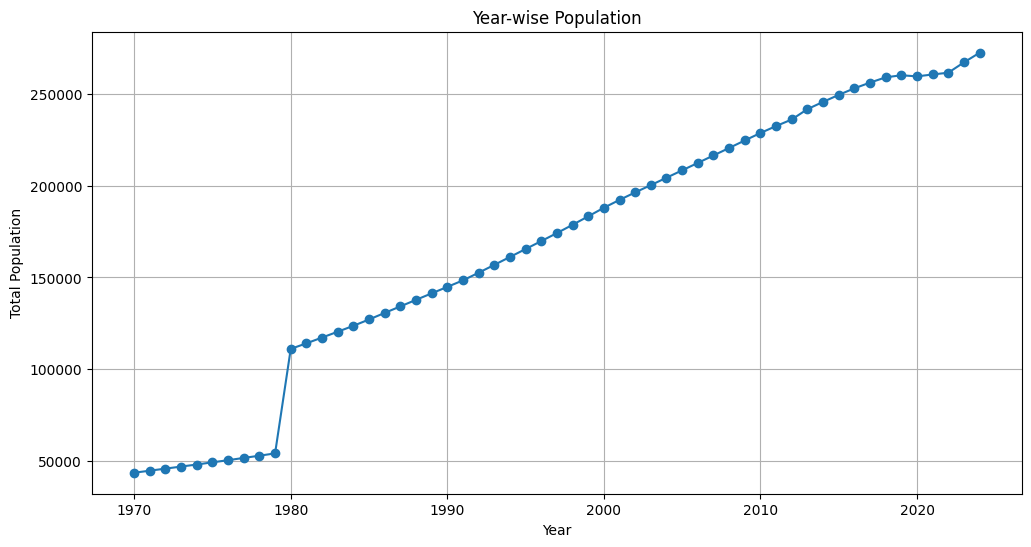
**plt.title('Year-wise Population')**

**plt.xlabel('Year')**

**plt.ylabel('Total Population')**

**plt.grid(True)**

**plt.show()**

****

**Explanation:**

* **Grouping by Year: Calculates total population for each year.**
* **Line Plot: Visualizes year-over-year population trends.**

#### 

#### 

#### **Population Distribution by Age Group (1970 vs 2023)**

**df\_1970 = df[df['Year'] == 1970]**

**df\_2023 = df[df['Year'] == 2023]**

**plt.figure(figsize=(12, 6))**

**plt.bar(df\_1970['age'], df\_1970['population'], width=0.4, label='1970', color='skyblue', align='center')**

**plt.bar(df\_2023['age'], df\_2023['population'], width=0.4, label='2023', color='lightcoral', align='edge')**

**plt.title('Population Distribution by Age Group in 1970 and 2023')**

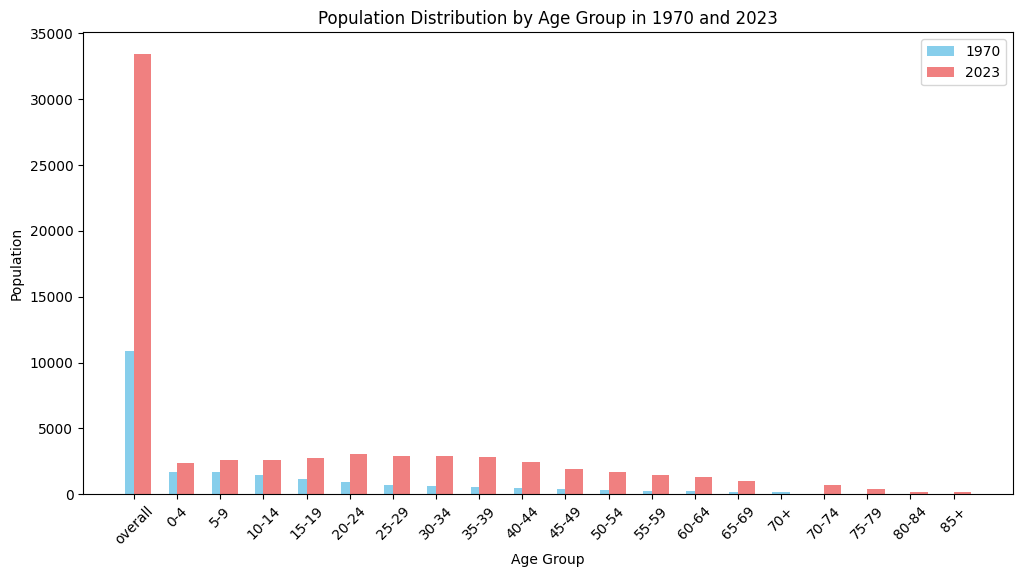
**plt.xlabel('Age Group')**

**plt.ylabel('Population')**

**plt.xticks(rotation=45)**

**plt.legend()**

**plt.show()**

****

**Explanation:**

* **Compares population distribution by age groups for 1970 and 2023 using a bar chart.**

#### **Male vs Female Population Comparison (1970 to 2024)**

**df\_filtered = df[df['Year'].isin([1970, 1980, 1990, 2000, 2010, 2023]) & df['sex'].isin(['male', 'female'])]**

**df\_pivot = df\_filtered.pivot\_table(index='Year', columns='sex', values='population', aggfunc='sum')**

**plt.figure(figsize=(10, 6))**

**df\_pivot.plot(kind='bar', color=['blue', 'pink'], width=0.8)**

**plt.title('Male vs Female Population Comparison (1970 to 2023)')**

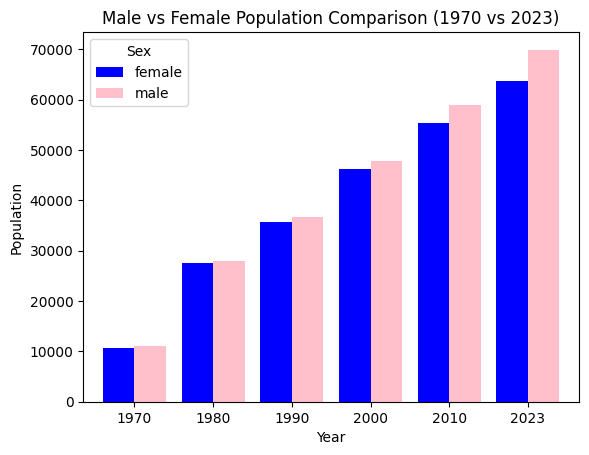
**plt.xlabel('Year')**

**plt.ylabel('Population')**

**plt.xticks(rotation=0)**

**plt.legend(title='Sex')**

**plt.show()**

****

**Explanation:**

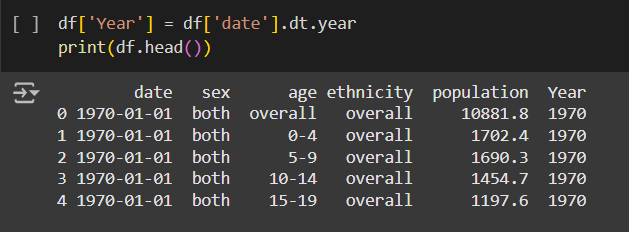
* **Filters data for selected years and sex.**
* **Creates a bar chart showing male and female population distributions across specified years.**

## **Step 4: Data Transformation**

### **Creating Additional Columns**

**df['Year'] = df['date'].dt.year**

**print(df.head())**

****

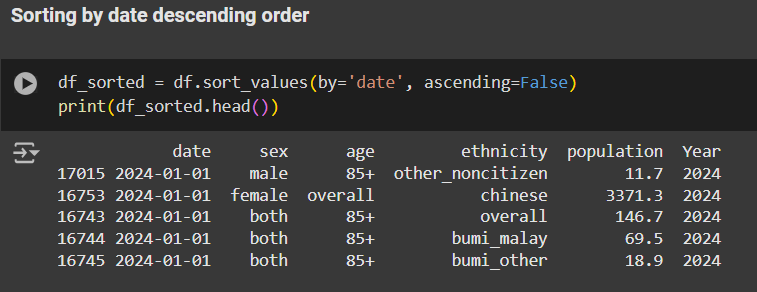
**Explanation:**

* **Creates a new Year column from the date column for easier year-based analysis.**

### **Sorting by Date**

**df\_sorted = df.sort\_values(by='date', ascending=False)**

**print(df\_sorted.head())**

****

**Explanation:**

* **Sorts df by the date column in descending order.**
* **df.sort\_values(by='date', ascending=False): This line sorts the DataFrame df by the 'date' column in descending order (ascending=False).**
* **df\_sorted = ...: The sorted DataFrame is stored in the variable df\_sorted.**
* **print(df\_sorted.head()): This prints the first five rows of the sorted DataFrame to verify the sorting.**

## **Summary**

**The analysis of Malaysia's population dataset highlights significant trends over time:**

* Population Growth: A steady increase in population, especially notable in recent decades.
* Demographic Changes: Age distribution patterns reveal shifts, with a larger aging population observed in 2023 compared to 1970.
* Gender Ratio Trends: Male and female populations vary over time, with a relatively balanced gender ratio but some fluctuations in certain years.

### **Insights and Future Work**

* Year-over-Year Growth: The trend analysis indicates consistent population growth, signaling demographic and economic shifts.
* Age Group Analysis: The age distribution changes suggest that Malaysia is experiencing an aging population, which may have future implications for workforce and healthcare.
* Gender-Specific Trends: Analyzing gender differences in population distribution could reveal more about social and economic shifts affecting gender demographics.

**Future work can extend to:**

* Comparative Studies: Including data from other Southeast Asian countries for broader regional trends.
* Socioeconomic Indicators: Integrating economic, education, and health data to provide a holistic demographic-economic analysis.
* Additional Data Sources: Incorporate data from resources like [CERN Open Data Portal](https://opendata.cern.ch/) or [Earthdata](https://earthdata.nasa.gov/) for multi-dimensional studies.

### 

### **Resources**

* **Pandas Documentation :** [**Pandas Official Documentation**](https://pandas.pydata.org/docs/)
* **Matplotlib Documentation :** [**Matplotlib Official Documentation**](https://matplotlib.org/stable/contents.html)
* **Data Source :** [**https://data.gov.my/data-catalogue/population\_malaysia**](https://data.gov.my/data-catalogue/population_malaysia)

This report illustrates a detailed population analysis process using Python's data analysis tools, providing a foundation for demographic studies and insights into Malaysia's changing population landscape.