**7b.** Write a Python Program for creating maps using Plotly library

## 1. Introduction.

- Plotly is a powerful and versatile graphing library in Python designed for creating interactive, web-based visualizations. It supports a wide range of chart types, including plots, graphs, and maps, making it a popular choice for data scientists and analysts.
- The Gapminder dataset is a comprehensive collection of global development data compiled by the Gapminder Foundation. It includes information about various countries across different years, encompassing key indicators such as population, GDP, life expectancy, and more. This dataset is frequently used in data science and visualization projects to explore trends and patterns in global development.
- In this program, we will use Plotly Express to visualize the Gapminder dataset on a geographic map. The Gapminder dataset contains population data for countries over several years and comes built-in with Plotly Express. The output is an interactive world map with countries represented as points. Each point's color indicates the continent, and additional information is displayed on hover. The size of the points corresponds to the population of each country in the year 2007.

## 2. Program code.

```
# Import necessary libraries
import plotly.express as px
from plotly.offline import plot

# Load the Gapminder dataset
gapminder_data = px.data.gapminder()

# Display the first 10 rows of the gapminder_data DataFrame as a Markdown table
print(gapminder_data.head(10).to_markdown())

# Filter the dataset for the chosen year (2007 in this case)
df = gapminder_data.query("year==2007")

# Create a scatter plot on a world map
fig = px.scatter_geo(df, locations="iso_alpha", color="continent",
                     hover_name="country", size="pop", projection="natural earth")
```

```
# Save the plot to an HTML file
plot(fig, filename='map.html')
```

## 3. Explanation of the code

### 3a. Importing Libraries

***import plotly.express as px***

- This statement imports the Plotly Express library and assigns it the alias 'px'. The alias is a shorthand reference to the library, making it easier to use.
- Plotly Express is a high-level interface built on top of Plotly that simplifies the creation of interactive visualizations. In the program, it is specifically used for generating a scatter plot on a world map.
- Plotly Express provides a convenient and user-friendly syntax for creating complex plots with minimal code, making it a popular choice for data visualization tasks.

***from plotly.offline import plot***

- This statement imports the 'plot' function from the Plotly offline module. The 'plot' function is used to save the interactive plot as an HTML file for offline viewing.
- After creating the interactive map using Plotly Express, the 'plot' function is called to save the plot as an HTML file named 'Map'. This allows users to view and interact with the map without an internet connection.
- The offline module in Plotly is useful when you want to generate static HTML files for sharing or embedding interactive plots in documents without the need for an internet connection.

### 3b. Loading Gapminder Dataset

***gapminder_data = px.data.gapminder()***

- Loads the Gapminder dataset using the px.data.gapminder() function and assigns it to a variable named gapminder_data. The dataset contains information about various countries over different years, including indicators such as population, GDP, life expectancy, and more.

- Plotly Express (px) provides several sample datasets through the px.data module.

px.data include

***gapminder()***

- *The Gapminder dataset, as discussed in the provided Python program. It includes information about various countries over multiple years.*

***tips()***

- *A dataset containing information about restaurant tips, total bill amounts, and other details.*

***iris():***

- *The famous Iris dataset, which includes measurements of sepal length, sepal width, petal length, and petal width for different species of iris flowers.*

***wind():***

- *A dataset with wind speed and direction measurements at different locations.*

***election():***

- *Election results dataset, providing information about votes in a fictional election.*

- The Gapminder dataset is a well-known and widely used resource in data science and visualization. The Gapminder dataset is a key component of the program, serving as the source of data for creating the world map. By loading this dataset, the program gains access to a rich set of global development data that can be visualized and analyzed.

**Dataset Structure**

- The dataset is typically structured as a DataFrame with columns representing different variables (e.g., country, year, population) and rows representing individual observations for each country in a specific year.

**First few rows of the DataFrame**

| | country | continent | year | lifeExp | pop | gdpPercap | iso_alpha | iso_num |
|---|---|---|---|---|---|---|---|---|
| 0 | Afghanistan | Asia | 1952 | 28.801 | 8425333 | 779.445 | AFG | 4 |
| 1 | Afghanistan | Asia | 1957 | 30.332 | 9240934 | 820.853 | AFG | 4 |
| 2 | Afghanistan | Asia | 1962 | 31.997 | 10267083 | 853.101 | AFG | 4 |
| 3 | Afghanistan | Asia | 1967 | 34.02 | 11537966 | 836.197 | AFG | 4 |
| 4 | Afghanistan | Asia | 1972 | 36.088 | 13079460 | 739.981 | AFG | 4 |

***print(gapminder_data.head(10).to_markdown())***

- Outputs a formatted Markdown table displaying the first 10 rows of the gapminder_data DataFrame.

## 3c. Filtering Gapminder Dataset

***df = gapminder_data.query("year==2007")***

- This line filters the Gapminder dataset (gapminder_data) to include only the rows where the 'year' column is equal to 2007. The result is stored in a new DataFrame called df.
- The filtering is performed to focus on a specific year (2007) within the Gapminder dataset. This allows the program to create a world map visualization based on data from that particular year.
- DataFrame 'df':The resulting DataFrame, df, now contains only the rows where the year is 2007. It retains the structure of the original dataset but focuses on a specific time frame.

## 3d. Creating and Showing the Map

***fig = px.scatter_geo(df, locations="iso_alpha", color="continent", hover_name="country", size="pop", projection="natural earth")***

- This line creates a scatter plot on a world map using Plotly Express (px). The df DataFrame, filtered for the year 2007, is used as the source of data. The resulting plot is stored in a fig (figure) object.
- This is the core step where the actual map visualization is generated. The parameters provided to px.scatter_geo determine how the data is represented on the map, such as location, color, size, and projection.
- The scatter plot on the world map visually represents the filtered data from 2007. Each point on the map corresponds to a country, and the position, color, and size of the points convey information about the country's location, continent, and population, respectively.

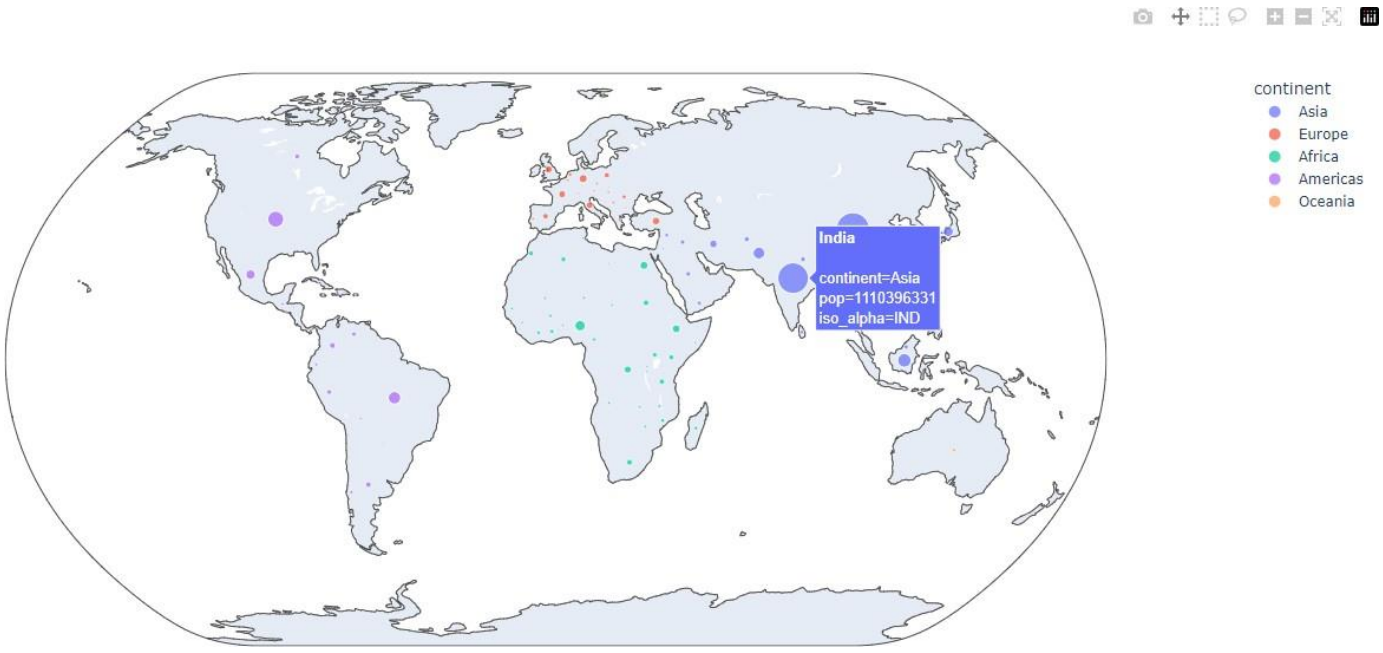
**Parameters:**

- locations="iso_alpha": Specifies the column in the DataFrame (df) containing ISO alpha-3 country codes, which are used to identify the locations on the map.
- color="continent": Determines the color of each point based on the continent to which the country belongs.
- hover_name="country": Displays the name of the country when hovering over its corresponding point on the map.

- size="pop": Adjusts the size of the points based on the population of each country.
- projection="natural earth": Chooses the projection style for the world map, in this case, "natural earth."

## 4. Output

- The output is an interactive world map with countries represented as points. Each point's color indicates the continent, and additional information is displayed on hover. The size of the points corresponds to the population of each country in the year 2007.




## 5. Real-Time Applications of Creating Maps using Plotly

Plotly's interactivity and ease of use make it a powerful tool for creating real-time maps across various domains. Here are some applications:

### 1. Tracking and Monitoring

- Logistics and Supply Chain: Visualize the real-time location of vehicles, packages, and assets, optimizing routes and monitoring delivery progress.
- Public Transportation: Track buses, trains, and other public transport in real-time, helping commuters plan journeys and manage congestion.

- Emergency Response: Monitor wildfires, floods, or other natural disasters in real-time, aiding response efforts and evacuation planning.
- Animal Tracking: Track the movement of endangered species or livestock, gaining insights into their behavior and habitat use.

**2. Data Visualization and Analysis**

- Weather Forecasting: Display real-time weather data like temperature, precipitation, and wind speed on interactive maps, providing crucial information for decision-making.
- Traffic Monitoring: Visualize real-time traffic congestion and flow, helping drivers navigate efficiently and manage traffic management systems.
- Air Quality Monitoring: Track real-time air quality data across regions, enabling public health agencies to communicate risks and take necessary actions.
- Social Media Analysis: Visualize the real-time spread of information, trends, and sentiment on social media platforms across geographic locations.

**3. Interactive Dashboards and Public Engagement**

- Election Results: Display real-time election results on interactive maps, engaging citizens and providing transparency in the process.
- COVID-19 Tracking: Track the spread of the virus in real-time, showing cases, vaccinations, and other relevant data for public awareness.
- Real Estate Market Analysis: Visualize real-time housing prices and trends on interactive maps, aiding buyers and sellers in making informed decisions.
- Citizen Science Projects: Engage the public in collecting and visualizing data on environmental issues, such as water quality or air pollution.

**Plotly's strengths in these applications include**

- Interactivity: Users can zoom, pan, and hover over data points for deeper insights.
- Customization: Maps can be tailored to specific needs with various markers, colors, and overlays.
- Live Updates: Data can be streamed in real-time, reflecting the latest changes dynamically.
- Accessibility: Maps can be shared easily online or embedded in web applications.