Coding and Interacting with the Gapminder Dataset

Anaconda Workspace Assignment • Interactive Plotting inside Jupyter Notebooks

Participant Name: _____ **Date:** _____

Section 1: Interactive Exploration Paradigm

Traditional static data visualizations offer static presentation snapshots. Utilizing browser-backed render tools like **Plotly Express** inside your Jupyter Notebook workspace transforms static indicators into highly responsive exploration matrices.

This exercise uses data tracking records from the global **Gapminder** registry to evaluate public health markers and economic output metrics across dynamic dimensional vectors.

Section 2: Guided Lab Tasks

Follow this multi-tiered progression loop to set up and deploy your interactive notebook application:

Step 1: Get the Data

Import Plotly Express and load the built-in Gapminder dataset using: `px.data.gapminder()`.

Step 2: Plot the Metrics

Build a scatter plot relating GDP per Capita to Life Expectancy, sizing dots by population, and mapping continent groupings to distinct categorical colors.

Step 3: Play with Filters

Use zoom boundaries to explore clusters, hover over points to instantly identify specific country names, and click/toggle continents inside the legend to filter macro-regions dynamically.

Section 3: Practical Code Workspace

Paste and execute the initialization script below inside an empty code cell in your Jupyter Notebook environment:

```
import plotly.express as px

# Step 1: Ingest the integrated global tracking metrics array
df = px.data.gapminder()

# Filter the structural records to examine a specific modern slice
df_latest = df[df['year'] == 2007]

# Step 2: Build the multi-dimensional interactive canvas configuration
```

```
fig = px.scatter(
    df_latest,
    x="gdpPercap",
    y="lifeExp",
    size="pop",
    color="continent",
    hover_name="country",
    log_x=True,
    size_max=60,
    title="Global Gapminder Audit Metric Matrix (2007)"
)

# Render your dynamic plot dashboard frame directly inside your cell viewport
fig.show()
```

Section 4: Active Analysis & Verification

Execute the code cell in Section 3 inside your local Jupyter sandbox, play with the canvas filters, and resolve the observation points below.

Question 1: Interactive Inspection (Hover Feature)

Hover over the exceptionally large data bubble located within the East Asia region of your plot canvas. Which country does this point map to, and what was its exact registered life expectancy in 2007?

Question 2: Dimensional Mechanics (Logarithmic Scaling)

In Step 2, the code parameters explicitly include the variable option `log_x=True`. Explain why converting the standard linear GDP scale into a logarithmic distribution profile helps optimize chart readability:

Question 3: Legend Filtering Mechanics

Click on the legend indicator labels on the right margin of your canvas viewport to mask and unmask whole continents. What happens to the automatic rendering layout scales of the Y-axis when you filter out all global territories except for **Africa**?

Section 4: Independent Custom Visual Challenge

🚀 **Challenge Assignment:** Pivot your analytical approach to track metrics over time instead of looking at a single-year snapshot.
Modify your plotting script to construct a line chart (`px.line`) tracking the multi-decade migration of `lifeExp` strictly for a single chosen country across all chronological intervals stored within the master `df` array.

Write Your Solution Configuration Code Below:

```
# Construct your custom single-country chronological path line plot below:
```

Interactive Navigation Reference

- **Box Zoom tool** → Click and drag an area on the canvas grid to slice your view into a highly granular, amplified micro-cluster region.
- **Double-Click Reset** → Double-clicking anywhere inside the visualization bounds instantly snaps the coordinates back to default dimensions.