The Economic Fitness of Countries

This analysis uses global trade data to explore the industrial development of countries through a measure called fitness, which captures the diversification and sophistication of their export capabilities. Fitness serves as an indicator of a country's industrial development and is compared against GDP per capita. Additionally, we reveal the nested structure of the world trade bipartite network.

1. Data Import

Objective:

Import the required datasets: the export matrix, the list of countries, and the GDP data.

Steps:

1. **Export Matrix**:
    - Import the matrix as a NumPy array using `pandas.read_csv`. Each row represents a country, and each column represents a product. The values correspond to export quantities.
2. **Country List**:
    - Import the list of countries as a Python list to map the rows of the export matrix to their respective country codes (3-letter ISO codes).
3. **GDP Data**:
    - Import GDP per capita data for 2021. Use the `Country Code` column for mapping and `2021 [YR2021]` for GDP values.

Verification:

- Ensure the dimensions of the export matrix match the number of countries and products.
- Verify that all country codes are consistent across datasets.

---

2. Binarization of the Matrix and Plotting the Ordered Matrix

Objective:

Compute the Revealed Comparative Advantage (RCA), binarize the export matrix, and plot the reordered matrix to highlight its nested structure.

RCA Definition:

The RCA for a country c and product p is defined as:

$$\mathrm{RCA}(c,p) = \frac{X(c,p)/X_c}{X_p/X_{\text{total}}}$$

where:

- $X(c,p)$: Export value of product p by country c.
- X_c: Total exports of country c.
- X_p: Total exports of product p.
- X_{total}: Total exports globally.

Steps:

1. **Compute RCA**: Use the formula above to compute the RCA for the matrix.

```
def rca_matrix(mat):
    u = np.sum(mat, axis=0)  # Product totals
    d = np.sum(mat, axis=1)  # Country totals
    t = np.sum(mat)          # Global total
    RCA = np.nan_to_num((mat * t / u).T / d).T
    return RCA
```

2. **Binarize RCA**:
    - Values ≥1 are set to 1 (comparative advantage), and values <1 are set to 0.
3. **Order the Matrix**:
    - Sort countries by diversification (sum of binary matrix rows) in ascending order.
    - Sort products by ubiquity (sum of binary matrix columns) in descending order.
4. **Plot Nested Structure**:
    - Visualize the reordered matrix using a grayscale heatmap.

---

3. Fitness and Complexity Algorithm

Objective:

Compute the fitness of countries and complexity of products using an iterative algorithm, then identify and plot the top and bottom 10 countries by fitness along with their GDP per capita.

Fitness and Complexity Definitions:

The iterative formulas are:

$$F_c^{(n+1)} = \sum_p M(c,p) \cdot Q_p^{(n)}$$

$$Q_p^{(n+1)} = \left(\sum_c M(c,p) \cdot \frac{1}{F_c^{(n)}} \right)^{-1}$$

with normalization:

$$F_c^{(n+1)} = \frac{F_c^{(n+1)}}{\langle F_c^{(n+1)} \rangle}, \quad Q_p^{(n+1)} = \frac{Q_p^{(n+1)}}{\langle Q_p^{(n+1)} \rangle}$$

Steps:

1. **Initialize**:
    - Start with Fc=1 Qp=1 for all countries and products.
2. **Iterate**:
    - Update Fc and Qp using the above equations for 100 iterations.

Implementation:

```
def compute_fitness_complexity(M, iterations=100):
    F = np.ones(M.shape[0])  # Countries
    Q = np.ones(M.shape[1])  # Products
    for _ in range(iterations):
        F_new = M @ Q
        Q_new = 1 / (M.T @ (1 / F_new))
        F, Q = F_new / np.mean(F_new), Q_new / np.mean(Q_new)
    return F, Q
```

3. **Top and Bottom Countries**:
    - Sort countries by fitness and select the top and bottom 10 for display.

---

4. Plotting the Fitness-GDP Plane

Objective:

Visualize the relationship between fitness and GDP per capita on a log-log scale and highlight key countries.

Steps:

1. **Merge Fitness and GDP Data**:
    - Join fitness results with GDP data using the `Country Code` column.
2. **Handle Missing Data**:
    - Exclude countries with missing or invalid GDP data.
3. **Log Transformation**:
    - Plot log(Fitness) vs. log(GDP per capita).
4. **Highlight Countries**:
    - Annotate 15 relevant countries, including India, China, oil producers, underdeveloped countries, middle-income trap countries, and advanced economies.
5. **Plot**:
    - Scatter plot with annotations for highlighted countries.

Visualization:

- Use a grid for clarity and set axis limits to focus on the main region of interest.