

CH02_Country_Data_Python_Supplement

September 3, 2026

1 Chapter 2 — Country data (Python supplement)

Condensed notebook for the **Python / Plotly** portion of *Chapter 2: Country Data*.

Target graphics:

- GDP vs. life expectancy scatterplots (**Figures 2.75–2.79**)
- Life expectancy choropleth (**Figure 2.80**)
- Rich/poor box plots (**Figures 2.81–2.82**)
- Histograms of country measures (**Figures 2.83–2.84**)

Data: GDP.csv, Inf_mort.csv, Life_exp.csv, and Population.csv in Data For Condensed Notebooks.

Dependencies:

- pandas
- plotly
- numpy
- pycountry

1.1 Imports and display options

```
[1]: import pandas as pd
import plotly.express as px
import numpy as np
```

```
[2]: # Set output options.
import plotly.io as pio
pio.renderers.default = "pdf+jupyterlab+notebook"
```

1.2 Loading the data

```
[3]: gdp = '../Data For Condensed Notebooks/GDP.csv'
inf_mort = '../Data For Condensed Notebooks/Inf_mort.csv'
life_exp = '../Data For Condensed Notebooks/Life_exp.csv'
pop = '../Data For Condensed Notebooks/Population.csv'
gdp_df = pd.read_csv(gdp)
inf_mort_df = pd.read_csv(inf_mort)
life_exp_df = pd.read_csv(life_exp)
```

```
pop_df = pd.read_csv(pop)
```

1.3 Merging datasets

Outer joins on Country; drop rows missing population before sizing dots.

```
[4]: gdp_life_df = pd.merge(gdp_df, life_exp_df, on='Country', how='outer')
```

```
[5]: gdp_life_inf_df = pd.merge(gdp_life_df, inf_mort_df, on='Country', how='outer')
```

```
[6]: all_df = pd.merge(gdp_life_inf_df, pop_df, on='Country', how='outer')
```

```
[7]: all_df
```

```
[7]:
```

	GDP_rank	Country	GDP_per_cap	LE_rank	Life_exp	IM_rank	\
0	207.0	Afghanistan	2000.0	222.0	51.7	1.0	
1	125.0	Albania	12500.0	59.0	78.5	124.0	
2	111.0	Algeria	15200.0	78.0	77.0	81.0	
3	136.0	American Samoa	11200.0	134.0	73.4	128.0	
4	32.0	Andorra	49900.0	8.0	82.9	199.0	
..	...	...	...	...	...	...	
235	176.0	West Bank	4300.0	110.0	75.2	105.0	
236	198.0	Western Sahara	2500.0	192.0	63.4	27.0	
237	222.0	Yemen	1300.0	176.0	65.9	38.0	
238	178.0	Zambia	4000.0	219.0	52.7	16.0	
239	201.0	Zimbabwe	2300.0	203.0	60.4	57.0	

	Inf_mort	Pop_rank	Population
0	110.6	39.0	34124811.0
1	11.9	136.0	3047987.0
2	19.6	33.0	40969443.0
3	11.3	210.0	51504.0
4	3.6	200.0	76965.0
..	...	...	...
235	14.1	141.0	2747943.0
236	51.9	169.0	603253.0
237	46.0	48.0	28036829.0
238	61.1	69.0	15972001.0
239	32.7	72.0	13805084.0

[240 rows x 9 columns]

```
[8]: all_df.dropna(subset=['Population'], inplace=True)
```

1.4 Scatterplots — GDP vs. life expectancy

Figures 2.75–2.79.

Figure 2.75 (basic scatterplot)

```
[9]: px.scatter(all_df, x='GDP_per_cap', y='Life_exp')
```



Figure 2.76 (color by infant mortality)

```
[10]: px.scatter(all_df, x='GDP_per_cap', y='Life_exp', color='Inf_mort')
```

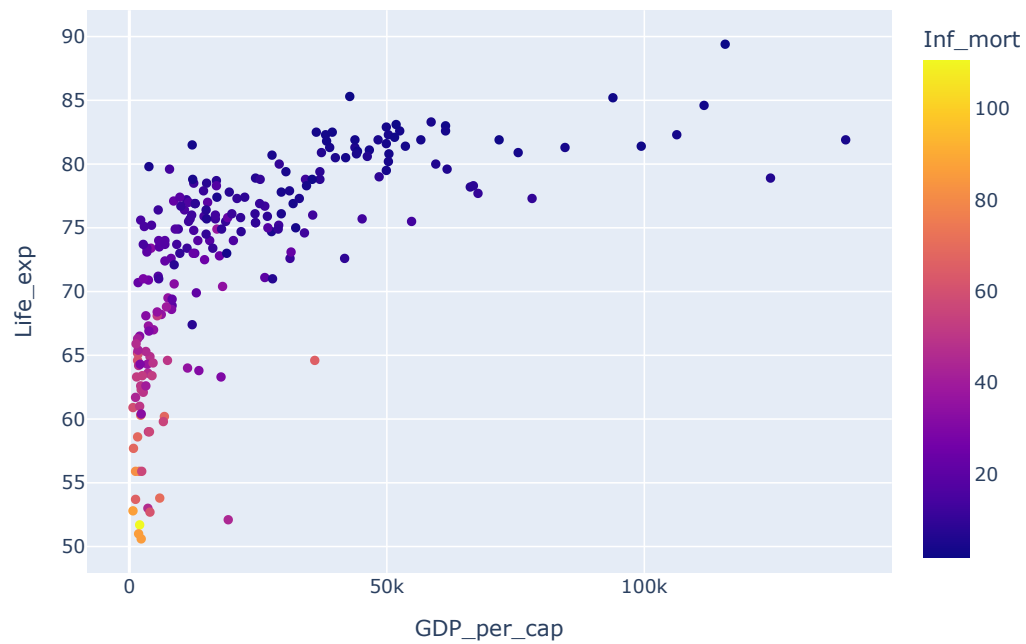


Figure 2.77 (Orange→Blue colorscale)

```
[11]: px.scatter(all_df,  
                x='GDP_per_cap',  
                y='Life_exp',  
                color='Inf_mort',  
                color_continuous_scale=['orange', 'blue'])
```

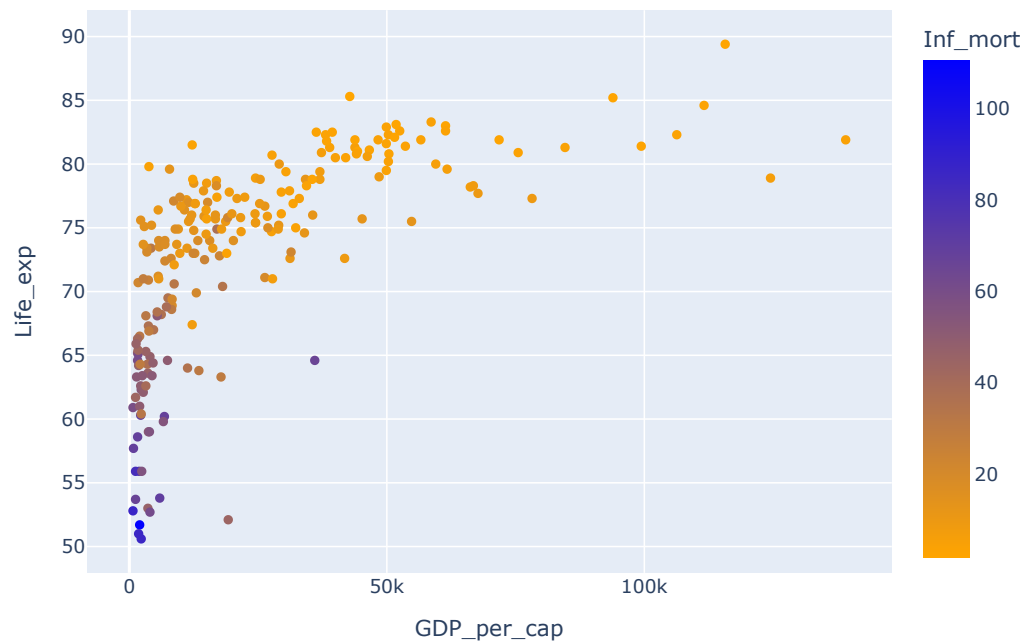


Figure 2.78 (dot size by population)

```
[12]: px.scatter(all_df,
                 x='GDP_per_cap',
                 y='Life_exp',
                 color='Inf_mort',
                 color_continuous_scale=['orange', 'blue'],
                 size='Population')
```

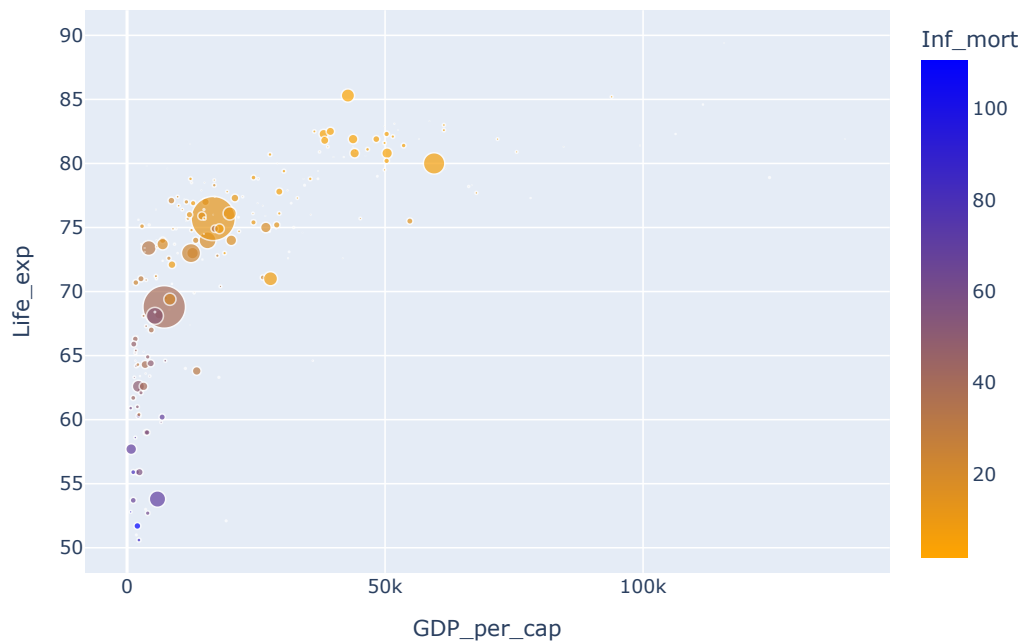
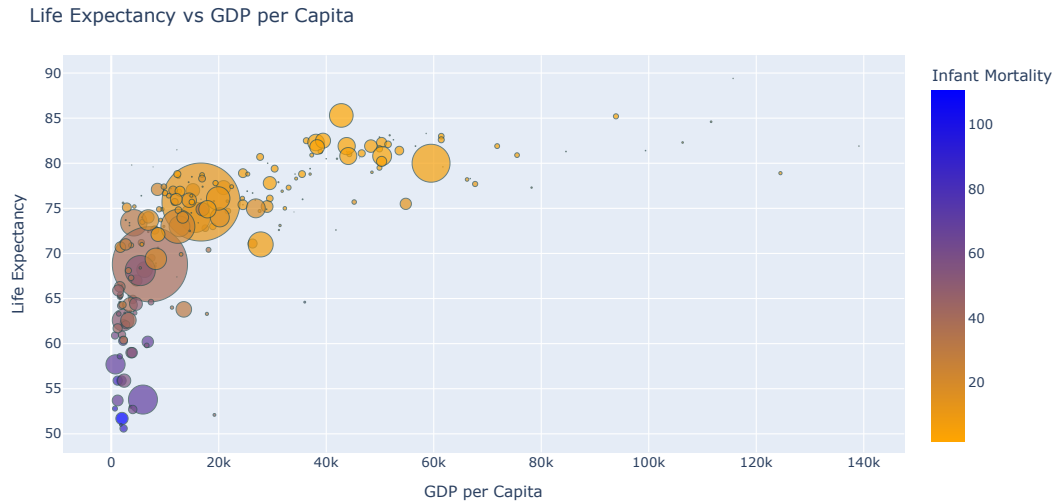


Figure 2.79 — Final scatterplot with custom colors, dot size, borders, and axis titles.

```
[13]: scatterplot5 = px.scatter(all_df,
                                x='GDP_per_cap',
                                y='Life_exp',
                                color='Inf_mort',
                                color_continuous_scale=['orange', 'blue'],
                                size='Population',
                                size_max=50,
                                width=1000)

scatterplot5.update_traces(marker=dict(line=dict(width=0.5,
                                                    color='DarkSlateGrey')),
                             selector=dict(mode='markers'))
scatterplot5.update_layout(xaxis_title='GDP per Capita',
                             yaxis_title='Life Expectancy',
                             title='Life Expectancy vs GDP per Capita',
                             coloraxis_colorbar=dict(title='Infant Mortality'))

scatterplot5
```



1.5 Choropleth — life expectancy

Map ISO alpha-3 codes with pycountry (name overrides for mismatches). **Figure 2.80.**

```
[14]: import pycountry

name_overrides = {
    "Burma": "Myanmar",
    "Korea, South": "Korea, Republic of",
    "Korea, North": "Korea, Democratic People's Republic of",
    "Bahamas, The": "Bahamas",
    "Gambia, The": "Gambia",
    "Congo, Democratic Republic of the": "Congo, The Democratic Republic of",
    "Congo, Republic of the": "Congo",
    "Czechia": "Czech Republic",
    "Cote d'Ivoire": "Côte d'Ivoire",
    "Syria": "Syrian Arab Republic",
    "Iran": "Iran, Islamic Republic of",
    "Russia": "Russian Federation",
    "Vietnam": "Viet Nam",
    "Tanzania": "Tanzania, United Republic of",
    "Laos": "Lao People's Democratic Republic",
    "Bolivia": "Bolivia, Plurinational State of",
    "Venezuela": "Venezuela, Bolivarian Republic of",
    "Moldova": "Moldova, Republic of",
    "Macedonia": "North Macedonia",
    "Turkey": "Türkiye",
}
```

```

}

code_overrides = {
    "Niger": "NER",
}

def get_iso3(name):
    if name in code_overrides:
        return code_overrides[name]
    name = name_overrides.get(name, name)
    try:
        return pycountry.countries.search_fuzzy(name)[0].alpha_3
    except:
        return None

all_df['country_code'] = all_df['Country'].apply(get_iso3)

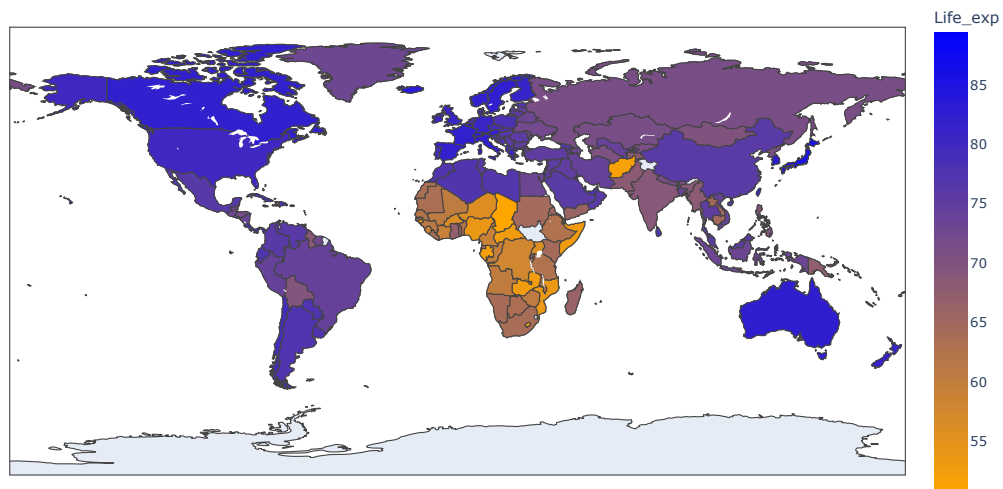
```

```

[15]: choropleth_fig = px.choropleth(all_df, locations='country_code',
                                     color='Life_exp',
                                     hover_name='Country',
                                     color_continuous_scale=['orange', 'blue'],
                                     width=1000,
                                     height=600)

choropleth_fig

```



1.6 Box plots — rich vs. poor countries

Wealth from a GDP-per-capita cutoff; drop rows missing GDP. **Figure 2.81** (color by wealth); **Figure 2.82** (color by infant mortality).

```
[16]: all_df['Wealth'] = np.where(all_df['GDP_per_cap'] > 60000, 'Rich', 'Poor')
```

```
[17]: all_df.dropna(subset='GDP_per_cap', inplace=True)
```

```
[18]: box_plot = px.box(all_df, x='Wealth',  
                        y='Life_exp',  
                        points='all',  
                        color='Wealth',  
                        hover_data=['Country'])  
box_plot.update_layout(legend_title_text='Wealth',  
                        yaxis_title='Life Expectancy',  
                        title='Life Expectancy: Rich vs Poor Countries')  
box_plot
```

Life Expectancy: Rich vs Poor Countries



```
[19]: box_plot = px.box(all_df, x='Wealth',  
                        y='Life_exp',  
                        points='all',
```

```

        color='Inf_mort',
        hover_data=['Country'])
box_plot.update_layout(legend_title_text='Wealth',
                        yaxis_title='Life Expectancy',
                        title='Life Expectancy: Rich vs Poor Countries')
box_plot

```

Life Expectancy: Rich vs Poor Countries



1.7 Histograms — country measures

Four-panel grid with mean and median reference lines (`make_subplots + go.Histogram`). **Figure 2.83.**

```

[20]: from plotly.subplots import make_subplots
import plotly.graph_objects as go

# Create a grid framework in which to place the histograms.
fig = make_subplots(
    rows=2, cols=2,
    subplot_titles=("GDP per Capita", "Population", "Infant Mortality", "Life_
↳ Expectancy"))

# Add the histograms to the grid.

```

```

fig.add_trace(go.Histogram(x=all_df['GDP_per_cap'], marker_color='seagreen',
    ↪showlegend=False),
                row=1, col=1)
fig.add_trace(go.Histogram(x=all_df['Population'], marker_color='seagreen',
    ↪showlegend=False),
                row=1, col=2)
fig.add_trace(go.Histogram(x=all_df['Inf_mort'], marker_color='seagreen',
    ↪showlegend=False),
                row=2, col=1)
fig.add_trace(go.Histogram(x=all_df['Life_exp'], marker_color='seagreen',
    ↪showlegend=False),
                row=2, col=2)

# Add median lines.
fig.add_vline(x=all_df['GDP_per_cap'].median(), line_width=2, line_dash='dash',
               line_color='black', row=1, col=1, name='Median', showlegend=True)
fig.add_vline(x=all_df['Population'].median(), line_width=2, line_dash='dash',
               line_color='black', row=1, col=2)
fig.add_vline(x=all_df['Inf_mort'].median(), line_width=2, line_dash='dash',
               line_color='black', row=2, col=1)
fig.add_vline(x=all_df['Life_exp'].median(), line_width=2, line_dash='dash',
               line_color='black', row=2, col=2)

# Add mean lines.
fig.add_vline(x=all_df['GDP_per_cap'].mean(), line_width=2, line_color='blue',
               row=1, col=1, name='Mean', showlegend=True)
fig.add_vline(x=all_df['Population'].mean(), line_width=2, line_color='blue',
    ↪row=1, col=2)
fig.add_vline(x=all_df['Inf_mort'].mean(), line_width=2, line_color='blue',
    ↪row=2, col=1)
fig.add_vline(x=all_df['Life_exp'].mean(), line_width=2, line_color='blue',
    ↪row=2, col=2)

fig.update_layout(height=500, width=700,
                   title_text="Histograms for Country Measures")
fig.show()

```

Histograms for Country Measures



Figure 2.84 — Single GDP-per-capita histogram with Plotly Express.

```
[21]: # Create a single histogram plot.
gdp_histogram = px.histogram(all_df,
                             x='GDP_per_cap', title='Distribution of GDP per_
                             ↳Capita',
                             height=600,
                             width=800)

# Set the axis titles.
gdp_histogram.update_xaxes(title='GDP per Capita')
gdp_histogram.update_yaxes(title='Number of Countries')

# Set a single color for all bars.
gdp_histogram.update_traces(marker_color='seagreen')

# Add the vertical lines for the mean and median.
gdp_histogram.add_vline(x=all_df['GDP_per_cap'].median(), line_width=2,
↳line_dash='dash',
                             line_color='black', row=1, col=1, name='Median',
↳showlegend=True)
gdp_histogram.add_vline(x=all_df['GDP_per_cap'].mean(), line_width=2,
↳line_dash='solid',
```

```
line_color='blue', row=1, col=1, name='Mean',  
↪showlegend=True)
```

```
gdp_histogram
```

Distribution of GDP per Capita

