Data Analysis Report Using Julius

Phyo Wai Thaung

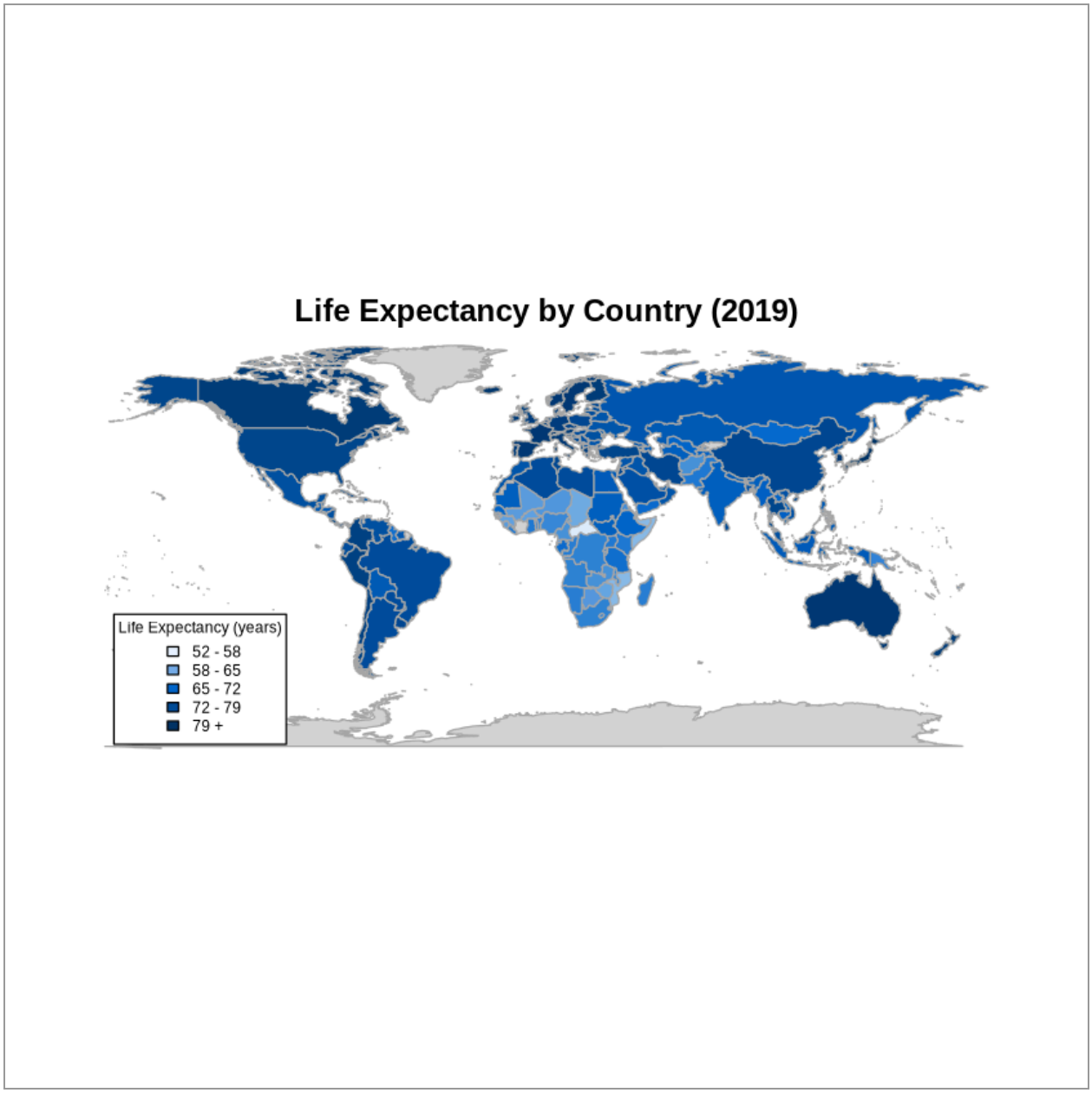
May 21, 2025

Introduction

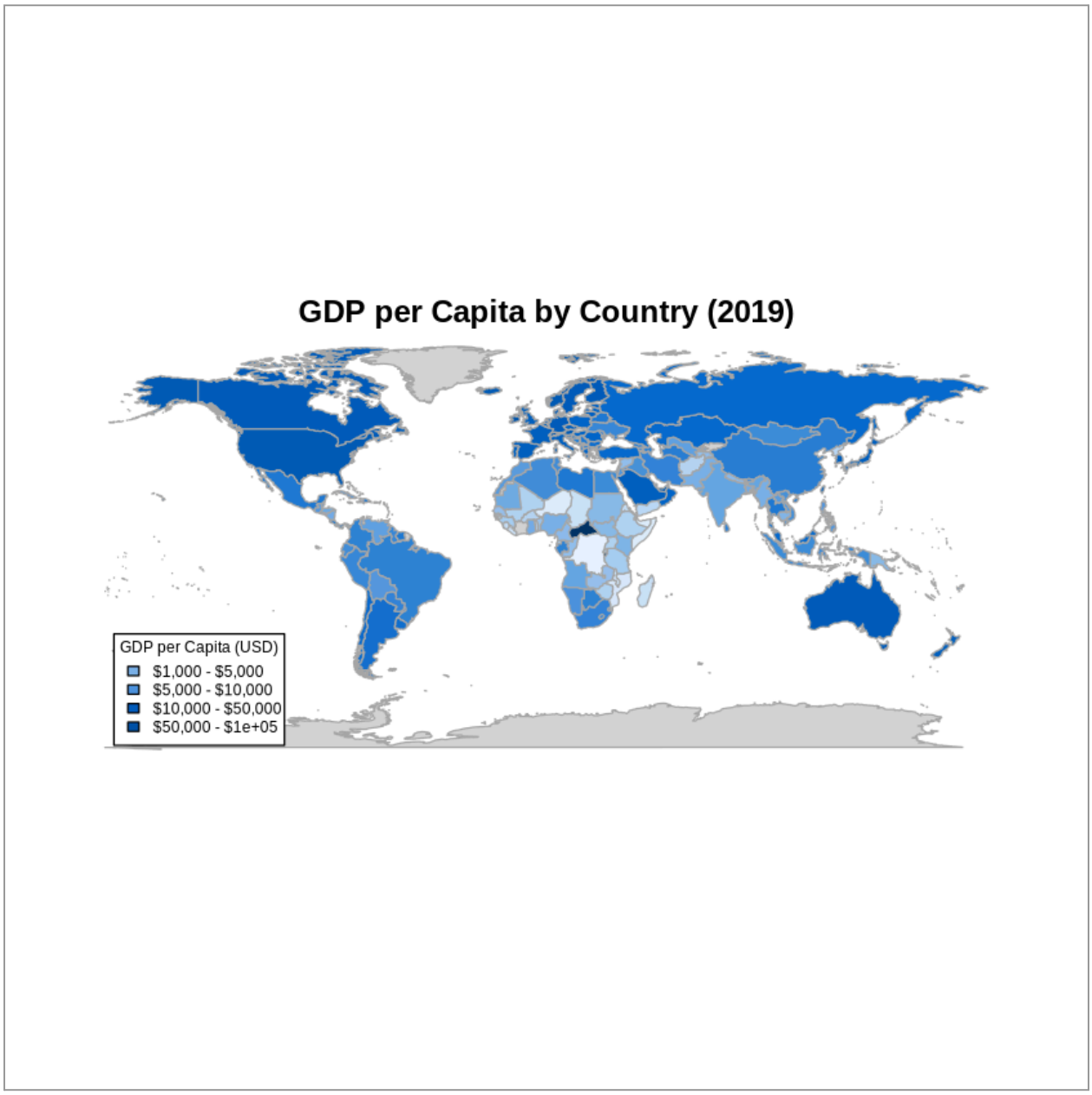
This report will discuss the relationship between Life Expectancy and GDP per Capita using the 2019 world data from the Gapminder's data repository: https://gapminder.org/data. For data analysis, Julius, an AI data analysis chat-based tool, is used.

Results

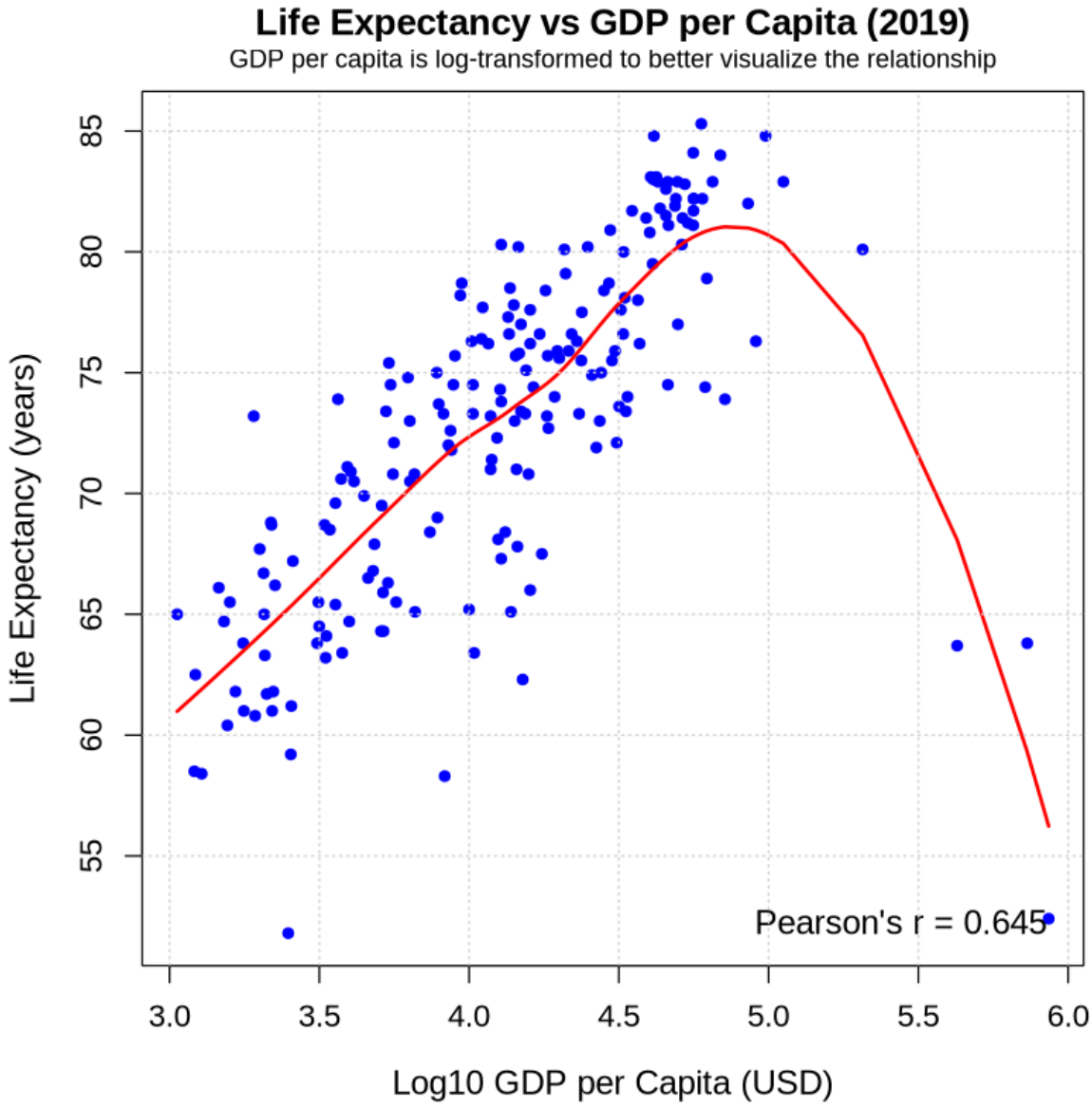
World Map of Life Expectancy



World Map of GDP per Capita



Relationship between Life Expectancy and GDP per Capita



The visualization reveals a strong positive correlation between log GDP per capita and life expectancy. This suggests that as a country's wealth increases, life expectancy tends to increase as well, but this relationship is logarithmic rather than linear - meaning that additional income has diminishing returns on life expectancy after a certain point.

Reflection

I am quite surprised by AI's ability to do quick debugging and data cleaning. In the dataset I provided, some numeric data is mixed with text (eg. 230k), which AI agent diagnosed and transformed it into an analysis-ready format. Another thing that impressed me is AI's statistical knowledge. After producing the first scatterplot, it realized that GDP per capita needs to be log-transformed to get the correct r value and interpretation.

On the other hand, I noticed that the AI agent was not able to absorb all my requests. In my case, it forgot to provide two world maps for the selected indicators. So, I had to remind the AI agent in the end. When further prompts were provided, it first generated bar plots with the reason that there were some issues with mapping packages. But when I insisted providing world maps by installing necessary mapping packages, it then generated maps which were still of poor quality (eg. a lot of NULL values, random colors, etc.) Therefore, I had to provide further detailed instructions to improve the maps. Similarly, the scatterplot AI first produced was short of proper labelling and without Pearson's coefficient value right in the chart. So, I had to give another prompt to improve this.

Overall, the AI agent is very supportive for data analysis, which saves a lot of time in data cleaning and debugging. If I perform the whole work by myself, it may take around 2-3 hours to complete the report. However, it seems that AI still needs to improve in visualization that required me to give follow-up prompts. But, I believe this will get better very soon.