CARNEGIE MELLON UNIVERSITY - AFRICA

DATA, INFERENCE & APPLIED MACHINE LEARNING

(COURSE 18-785)

Professor Patrick McSharry

Assignment 2

MUNYANEZA Kenny Roger

18th September 2023

Introduction

This is the final report for DIAML assignment 2, comprising of five solutions for the problems asked in the assignment.

Tools Used

- **Python:** It was used as the programming language to implement the solution.
- **Jupyter Notebook:** Used the Jupyter notebook plugin in vs code as a development environment.
- **Quandl:** Used as a source of economic datasets used in question 2

Question 1

Third party libraries used

- **pandas**: It was used to manipulate the data-frame provided which involved reading the dataset in csv format.
- **Matplotlib**: Used the Pyplot submodule of Matplotlib to plot the 2D scatter plots required in the question

The kind of relationship I expected

I expected an negative relationship between GDPs per capita of a country and the prevalence of malnutrition in that country, whereby as the GDP per capita increases the malnutrition prevalence decreases.

Implementation process Used

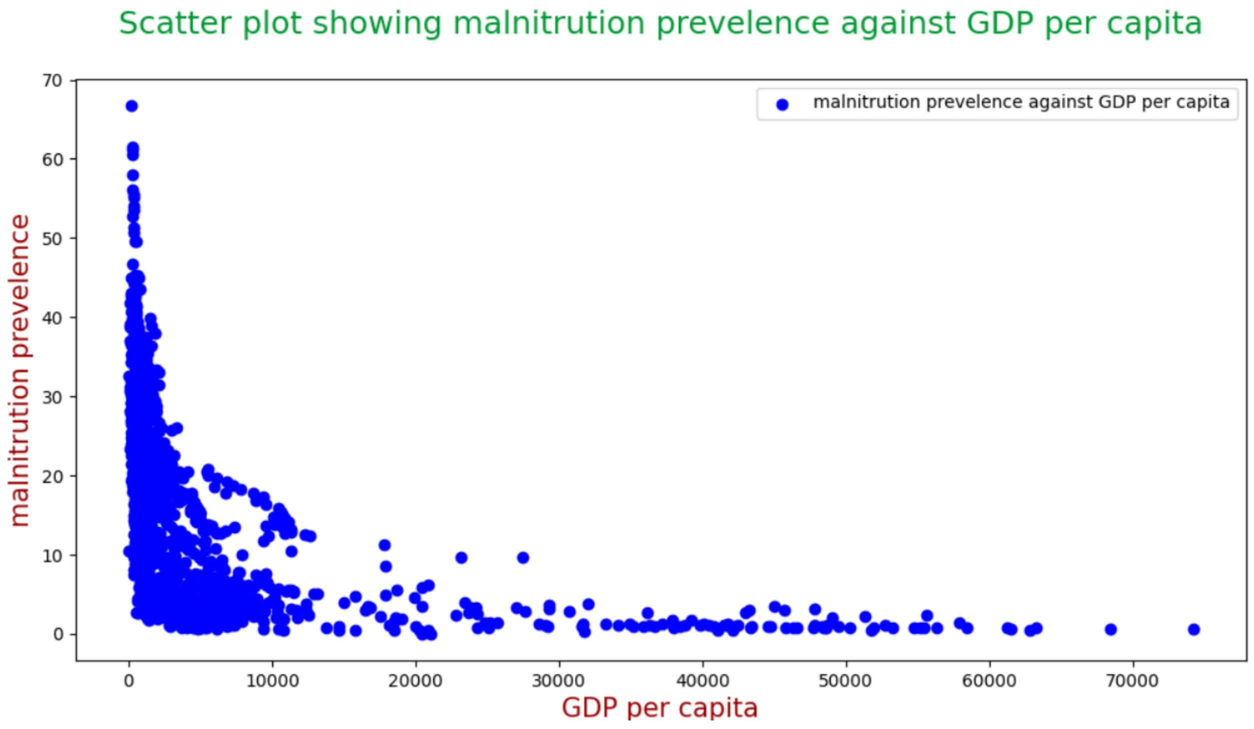
To solve the problem in question 1, I mainly used three functions that all plot a scatter plot of the distribution of malnutrition prevalence in countries against the countries' GDP per capita but the three functions plot the scatter plot in different styles. The first function simply plots the distribution showing malnutrition prevalence and GDP per capita for all years and all countries without the ability of the viewer to distinguish the distribution according to any attribute. The second function does the same process of plotting the distribution but facilitates the viewer of the graph to distinguish the distribution into six graphical regions. Whereby, each coordinate belonging to a particular region has a differently colored marker. The six geographical regions are Latin America & Caribbean, South Asia, Sub-Saharan Africa, Europe & Central Asia, Middle East & North Africa, East Asia & Pacific. The third function also plots the mentioned distribution but allows the viewer of the graph the ability to distinguish the distribution into four income levels. Whereby, each coordinate belonging to a particular income level has a differently colored marker. The four income levels are High income, Low income, Lower middle income, Upper middle income.

In the main execution of the program, I start by reading the two datasets of GDP per capita and Malnutrition prevalence, respectively. Next, I drop the unnamed columns in both datasets, which are the unnamed and unlabeled columns in the datasets. These columns are unnecessary and can potentially cause run time errors hence I dropped them. Next, I called the first function to plot the distribution mentioned without any visualization facility to distinguish the coordinates plotted according to any attribute. This function leads to a graph that shows the relationship between two quantitative variables plotted, which I will discuss in the next section of this report. The second function visually depicted how the distribution of the two quantitative variables can be grouped into geographical regions I called it. Lastly, I called the third function that visually depicted how the distribution of the two quantitative variables can be grouped into the four income levels mentioned earlier.

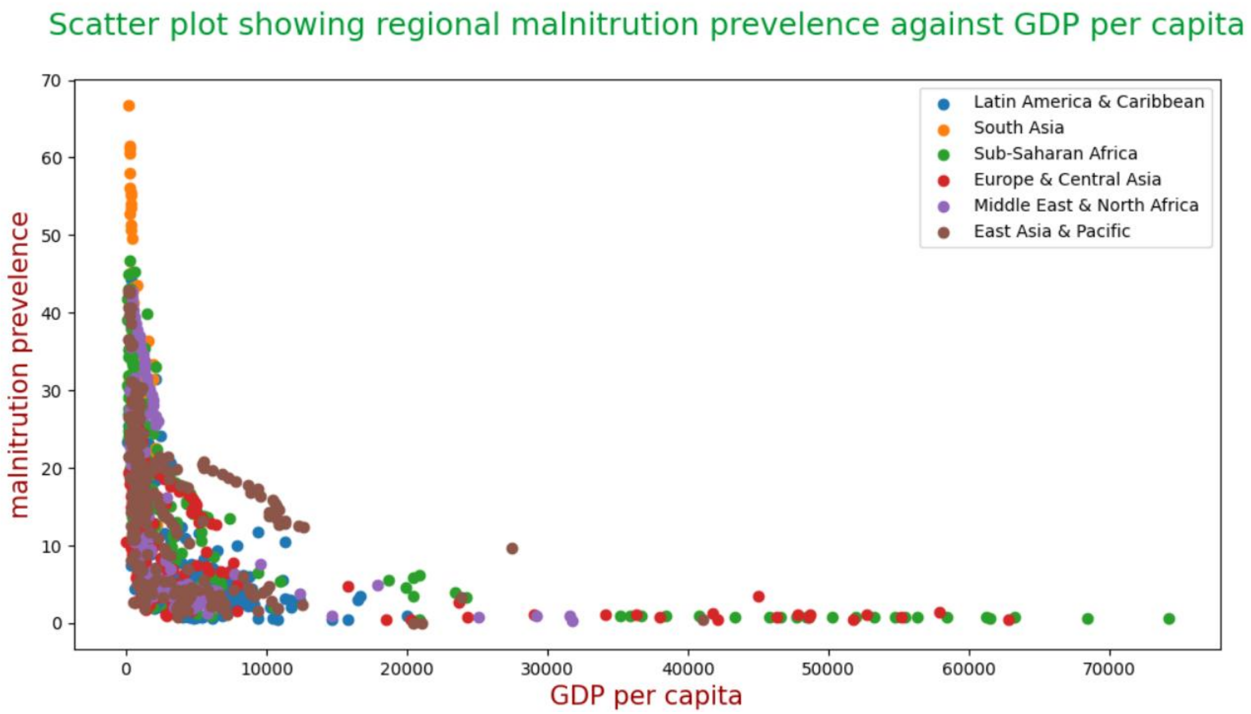
The actual relationship observed

The relationship I observed between GDP per capita and malnutrition prevalence is a J-Shaped relationship with a very strong association.

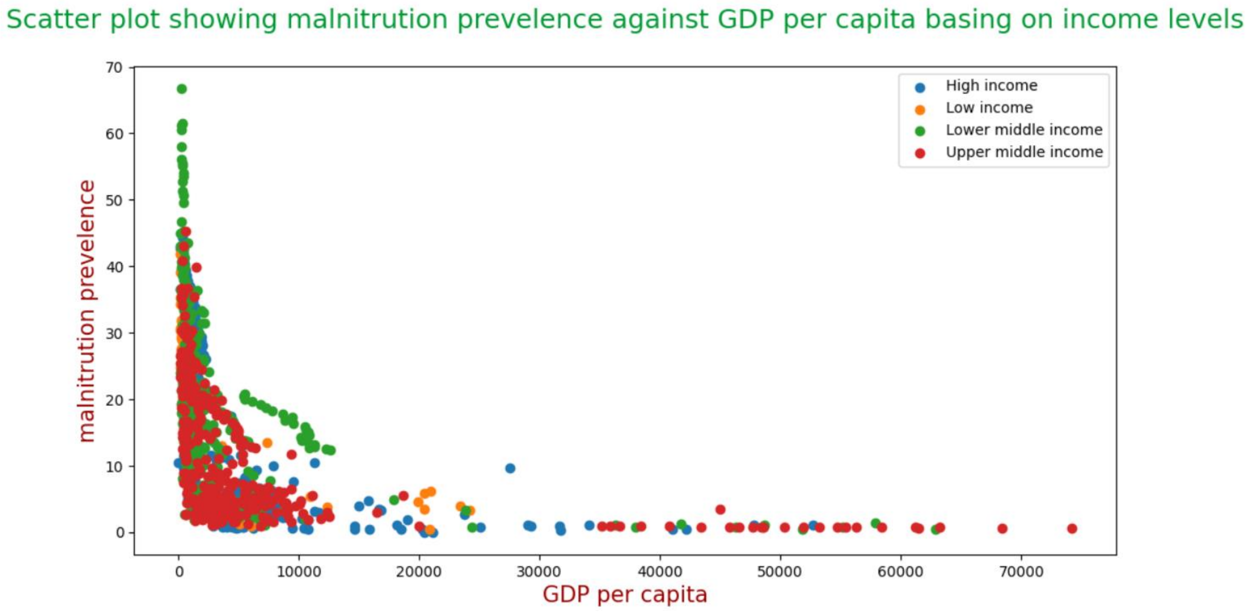
Results



Scatter plot showing malnitrution prevelence against GDP per capita



Scatter plot showing regional malnitrution prevelence against GDP per capita



Insights (Observations)

In all regions and all income levels, malnutrition is higher when the GDP per capita is low and it tends to decrease as the GDP per capita increases.

Question 2

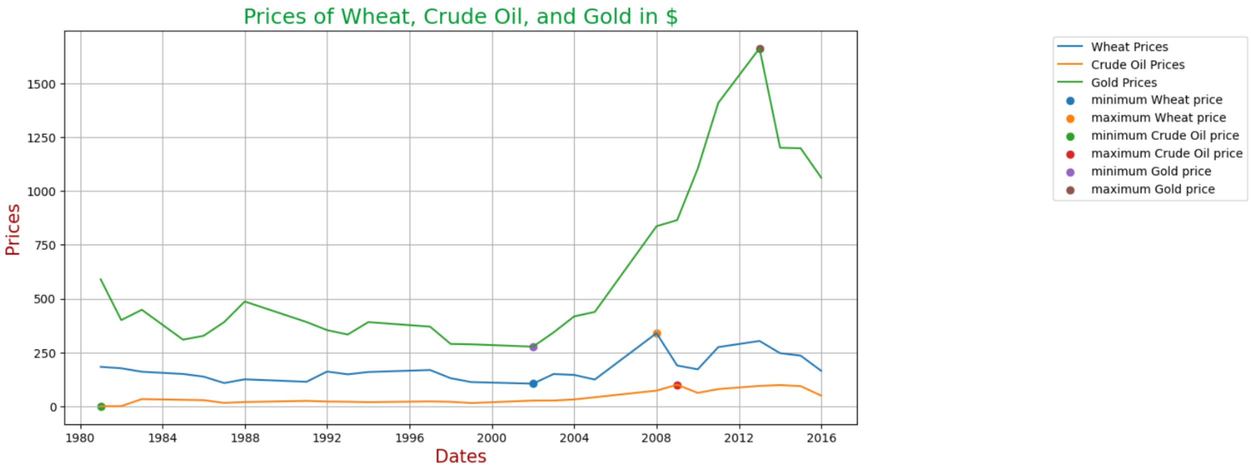
Third party libraries used

- **pandas**: It was used to manipulate the data-frame provided which involved reading the dataset in csv format, as well as merging the dataset into a synchronized dataset.
- **Matplotlib**: Used the Pyplot submodule of Matplotlib to plot the time series required in the question
- **NumPy:** Used to determine the index of the maximum and minimum values in each of the three datasets provided.
- **Quandl:** Used to acquire the three datasets of Wheat, Crude Oil, and gold prices needed to solve this problem.

Implementation process used

To solve this problem, I designed a program that reads the three datasets of interest, namely Wheat prices, Crude oil prices, and gold prices using the Quandl library. After, I renamed the prices column for each dataset since they all had a common naming “Values”. This would have led to random naming of the columns of the new dataset with the three prices columns combined. After, I merged the three data frames on the “Date” column using the inner merge to make sure that all dates that are not consistent throughout the three datasets are dropped. Later, I plotted the three columns containing the prices of the three commodities mentioned earlier, each against the date. This gave me three time series line graphs, and then I proceeded to plotting the minimum and maximum prices for each commodity. To do that I looped through the merged dataset's columns and for each column I was using NumPy to determine the maximum and the minimum prices' indices and then plotting those values by using the index as x and y coordinates. I used differently colored dots on each graph to depict the minimum and maximum values and a legend to explain those colored plots.

Results



Insights

- Gold has the highest prices across all years compared to the two other commodities.
- Wheat has the lowest prices across all years and is most consistent with time compared to the two other commodities.
- The prices of all three commodities were declining in 2016.
- All commodities reached their peak price value from 2008 onwards but before the year 2016.

Question 3

Third party libraries used

- **pandas**: It was used to manipulate the data-frame provided which involved reading the dataset in csv format, as well as computing the summary statistics of both dataset's 2010 column.
- **PrettyTable**: This library serves to display data in a tabular format using ASCII table format.

Implementation Process

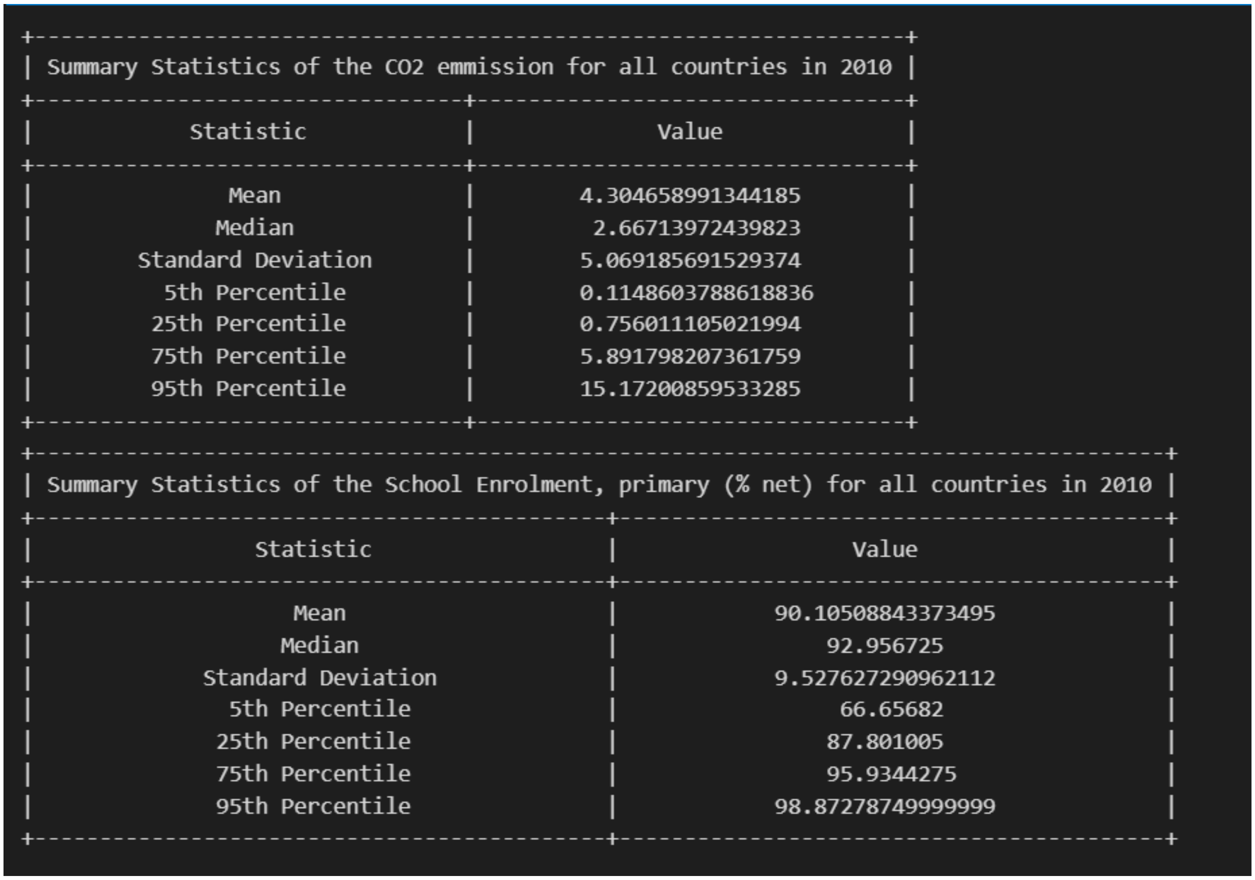
To solve this problem is started by reading the CO2 emission dataset downloaded from the World Bank Indicators using the read_csv function in pandas. Next, I extracted the 2010 column from the whole data frame into a series, which I used to compute the mean, median, standard deviation, 5th percentile, 25th percentile, 75th percentile, and 95th percentile. Then, I created an object of the PrettyTable to tabulate the summary statistics and I appended the statistics row by row in format which specifies the statistics type and the statistics value.

After I repeated the same procedure mentioned above for the School enrollment dataset to compute the summary statistics and tabulating them using ASCII code table format. [1]

Handling Naan Values

To handle Naan values, I used the functions provided by the pandas library to calculate the summary statistics instead of using the functions provided by NumPy to calculate the summary statistics. This is because the functions provided by NumPy are sensitive to Naan values whereas the function provided by pandas are not. The function I used are mean(), median(), std(), quartile() to name a few.

Results



```
+-------------------------------------------------------------------+
| Summary Statistics of the CO2 emmission for all countries in 2010 |
+---------------------------------+---------------------------------+
|            Statistic            |              Value              |
+---------------------------------+---------------------------------+
|              Mean               |        4.304658991344185        |
|             Median              |         2.66713972439823        |
|        Standard Deviation       |        5.069185691529374        |
|          5th Percentile         |        0.1148603788618836       |
|         25th Percentile         |        0.756011105021994        |
|         75th Percentile         |        5.891798207361759        |
|         95th Percentile         |        15.17200859533285        |
+---------------------------------+---------------------------------+
+-----------------------------------------------------------------------------------------+
| Summary Statistics of the School Enrolment, primary (% net) for all countries in 2010 |
+-------------------------------------------+-------------------------------------------+
|                 Statistic                 |                   Value                   |
+-------------------------------------------+-------------------------------------------+
|                   Mean                    |             90.10508843373495             |
|                  Median                   |                 92.956725                 |
|             Standard Deviation            |             9.527627290962112             |
|               5th Percentile              |                  66.65682                 |
|              25th Percentile              |                 87.801005                 |
|              75th Percentile              |                 95.9344275                |
|              95th Percentile              |             98.87278749999999             |
+-------------------------------------------+-------------------------------------------+
```

Insights

- The distribution of CO2 emission data is positively skewed since the mean is greater than the median.
- The distribution of School enrollment data is negatively skewed because the median is greater than the mean.

- The school enrollment data have a higher spread or variability compared to the CO2 emission data since it has a higher standard deviation than CO2 emission data.
- For CO2 emission data, 5 % of the data fall under 0.11 tons per capita, 25% of the data fall under 0.76 tons per capita, 75% of the data fall under 5.89 tons per capita, and 95% of the data fall under 15.17 tons per capita.
- For School enrollment data, 5 % of the data fall under 66.6 School enrolment, primary (% net), 25% of the data fall under 87.80 School enrolment, primary (% net), 75% of the data fall under 95.93 School enrolment, primary (% net), and 95% of the data fall under 98.87 School enrolment, primary (% net).

Question 4

Third party libraries used

- **pandas**: It was used to manipulate the data-frame provided which involved reading the dataset in csv format.
- **Matplotlib**: Used the Pyplot submodule of Matplotlib to plot the graphs required in the question.
- **NumPy:** Used to calculate the Cumulative Distribution Function.

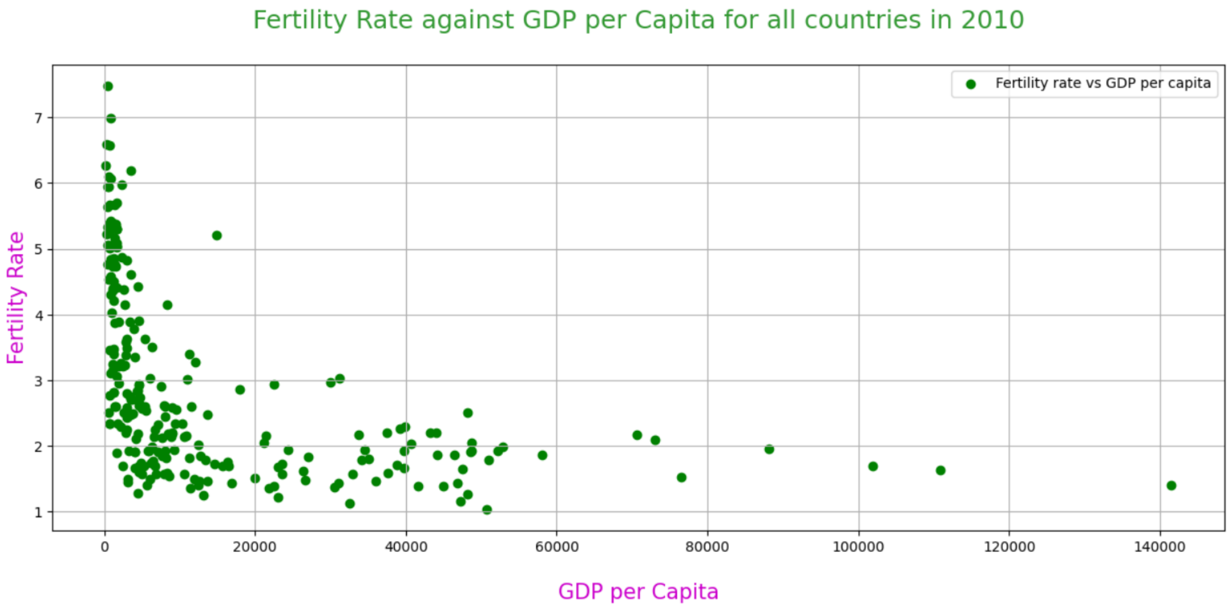
Implementation Process

The first problem in question four required to plot a scatter plot of fertility rate versus GDP per capita hence I started by reading the two datasets namely fertility rate and GDP per capita downloaded from the World Bank Indicators and I extracted the distribution of these variables in the year 2010 and assigned those two distributions to two different lists. After, I plotted the two lists into scatter plot using matplotlib's submodule called Pyplot. This module allowed me to set the graph size, the title of the graph, the labels for x and y coordinates and many more styling features.

For the second question I designed a function with will plot the cumulative distribution function for the fertility rate in the year 1990 and recall it to plot the cumulative distribution function (CDF) for the year 2010. That function expects three parameters whereby the first parameter is a 'pandas' series of the fertility rate for a particular year, the second parameter is the color for the mean and the third parameter is the color for the median. The function first sorts the dataframe in ascending order using the 'Sort' function of NumPy, then it creates a collection containing the probability of each value in the sorted dataframe and stores the correction into a list. Then, the function computes the mean and the median using the .mean() and .median() functions of pandas. Finally, the CDF will be plotted showing the probability of fertility rate variables against the fertility rate variables. Also, for each CDF two lines are plotted one showing the mean and the other showing the median.

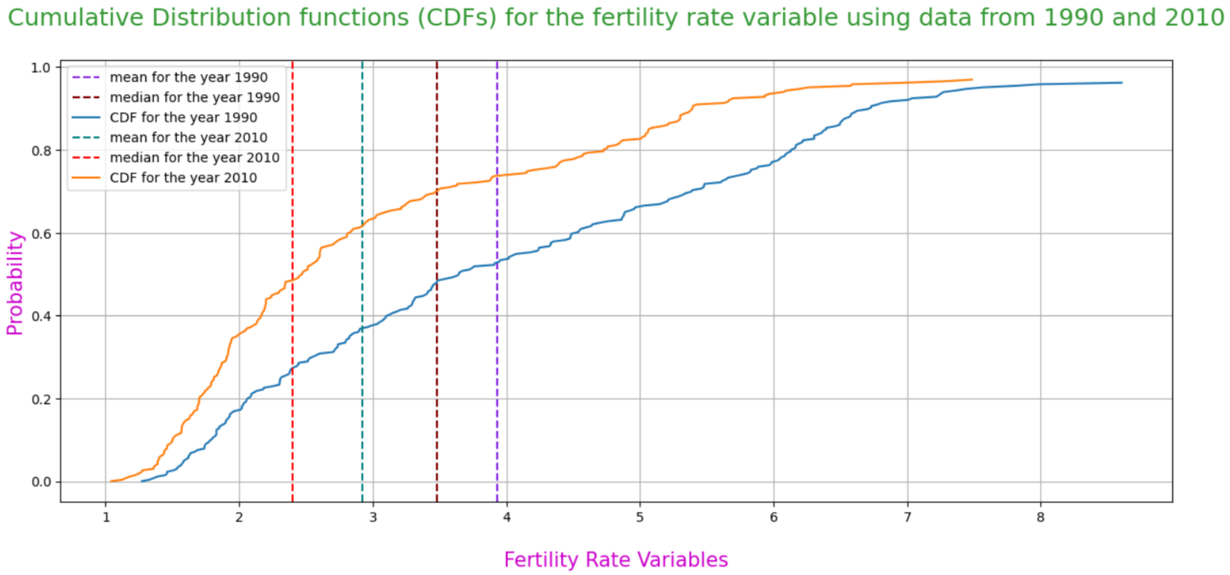
I called the above function twice in the main execution to plot the fertility rate for both 1990 and 2010.

Results



Insights

- There is a correlation between the GDP per capita and the fertility rate.
- The association between GDP per capita and fertility rate is negative and fertility rate decreases as the GDP per capita increases.
- The association between GDP per capita and the fertility rate is not linear and has potential outliers.



Changes in the fertility rate over the 20 years

Over the 20 years, the fertility rate has decreased, whereby for instance the peak fertility rate was between 8 and 9 in 1990 but in 2010 the peak fertility rate is between 7 and 8. Therefore, it can be inferred that the fertility rate has decreased in the past 20 years.

Question 5

Third party libraries used

- **pandas**: It was used to manipulate the data-frame provided which involved reading the dataset in csv format.
- **Xlrd:** Used to read Excel files in the older binary format.

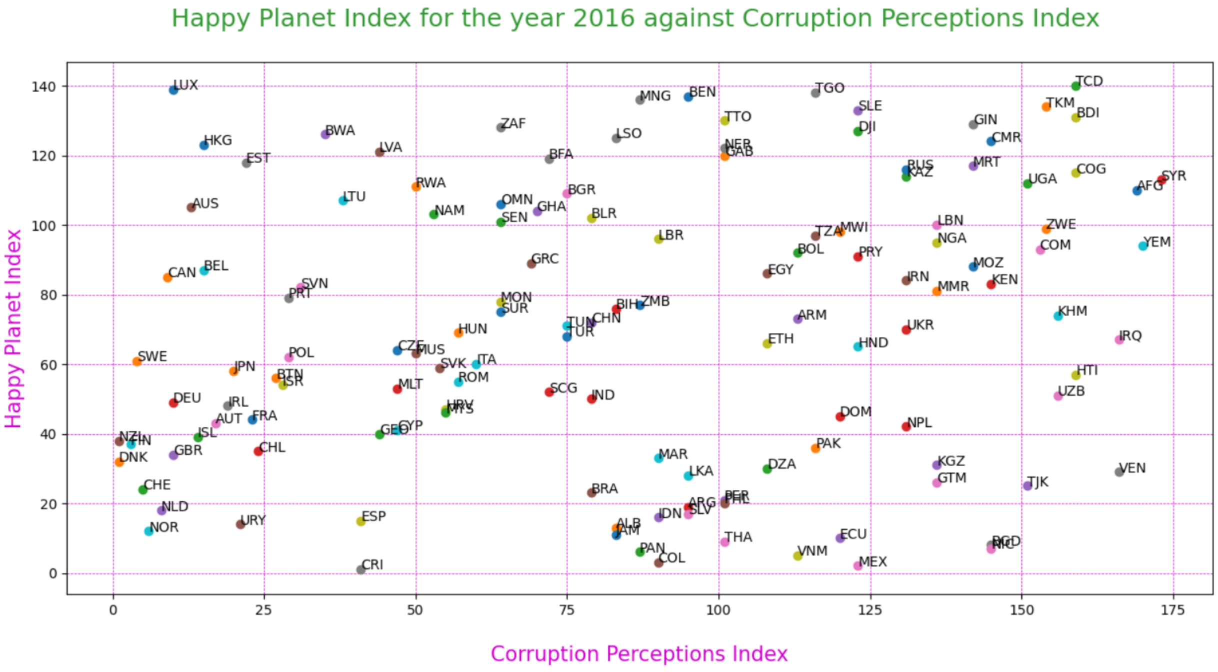
Other Libraries

- **Warnings:** To suppress warning messages generated by openpyxl.

Implementation Process

To solve this problem, I downloaded two datasets from the World Bank Indicators namely the Happy Planet Index (HPI) and the Corruption Perception Index (CPI). Analyzing the two datasets, I realized that HPI dataframe is harder to read because the sheet I was reading contained two tables, yet I was interested in in the topmost table. This meant that I cannot read the table rows until end of file without overlapping the data of the table of interest with the data of the next table. Therefore, I looped through the dataset until the iteration reaches a row with NaaN in all cells, which I assume would be the end of the table. Successfully, I was having all the indices in the HPI dataset as needed at the end of the loop and I extracted the HPI rank and the Country columns into a separate dataframe. Next, I read the second dataset of CPI which was easier to read, and I extracted the Country, CPI 2016 Rank, and WB code (country code) columns into a separate dataframe. To determine the matching countries that match both CPIs and HPIs, I performed an inner merge based on the country column for the two extracted data frames into a single dataset. Lastly, I used a loop to plot the merged dataset rows using Pyplot submodule. The submodule allowed me to mark each HPI against CPI coordinate with a colored marker and annotate the coordinate with the country code.

Results



Unusual Countries:

- Chad seems unusual because it has approximately the same happy index as Luxembourg, yet the two country Corruption indices differ by more than 125 index units. This contradicts the fact that HPI is negatively associated with CPI.
- Also, countries like Norway, Uruguay, and Netherlands have a very low happy index yet their Corruption index as well. This implies that lower corruption leads to lower happiness.