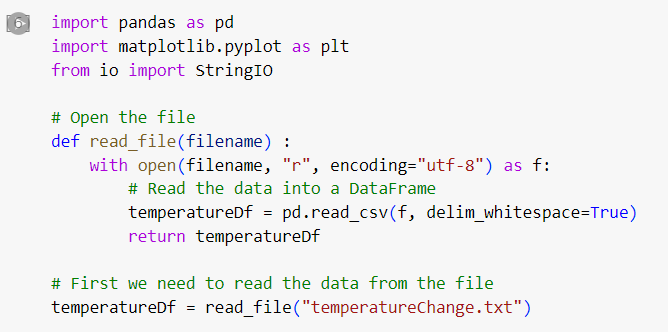
Code Walkthrough

Use this document to help you understand the code and for step by step explanations of what is happening.



 **Imports Tools:**

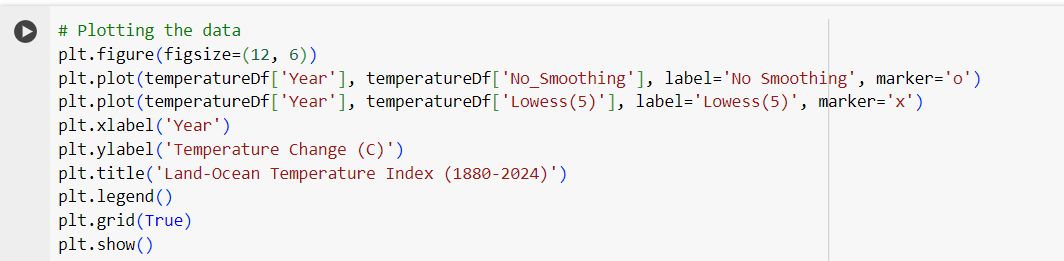
* pandas (pd) helps organize data in tables.
* matplotlib (plt) is for making graphs.

 **Reads the File:**

* The read\_file function opens a file and reads the data into a table (DataFrame).
* The file’s data is separated into columns based on spaces.

 **Uses the Data:**

* The last line reads the file "temperatureChange.txt" and stores the data in a table called temperatureDf.



 **Set Up the Graph:**

* plt.figure(figsize=(12, 6)): This line sets the size of the graph.

 **Plot the Data:**

* plt.plot(...) is used to plot two lines on the graph:
  + The first line ('No Smoothing') shows temperature changes over time without any smoothing (using circles as markers).
  + The second line ('Lowess(5)') shows a smoothed version of the temperature changes (using "x" markers).

 **Label and Customize the Graph:**

* plt.xlabel('Year'): Adds a label for the x-axis (years).
* plt.ylabel('Temperature Change (C)'): Adds a label for the y-axis (temperature change in degrees Celsius).
* plt.title('Land-Ocean Temperature Index (1880-2024)'): Gives the graph a title.
* plt.legend(): Shows a legend explaining what the lines represent.
* plt.grid(True): Adds a grid to the graph for easier reading.

 **Show the Graph:**

* plt.show(): Finally, this line displays the graph.



In the last section of the code this is the new code, your task is to fix the mistakes.

 **Set Up a Loop to Plot Top 10 Countries:**

* count = 0: This line sets up a counter to start from 0. The counter keeps track of how many countries have been plotted so far.

 **Loop Through the Top 10 Countries:**

* while count < a:: This loop will run 10 times to process the top 10 countries in the dataset.
  + country = countries[b]: For each loop iteration, it selects the country at the current position (count) from the list of countries.
  + emissions = ghgDf.loc[ghgDf['Country/Region'] == country, ghgDf.columns[3:]].values.flatten().astype(float): This line extracts the emissions data for the selected country. It looks at the data from the 4th column onwards, which typically represents yearly emissions data. The data is converted into a format that can be easily plotted.
  + plt.plot(ghgDf.columns[3:], emissions, label=country, marker='o'): This line plots the emissions data for the selected country on the chart, marking each data point with a circle.

 **Increment the Counter:**

* count += c: This line adds 1 to the counter, moving on to the next country in the next loop iteration.