Data Visualization

Source Link:

* Google Colab: <https://colab.research.google.com/drive/1VNRIIOdUCkpcMLIpW4OSCZEKaYFvBbqm?usp=sharing>
* Github: <https://github.com/Wang-Kevin900/DataVisualization>

Main Sources used:

* For bokeh related widget: <https://bokeh.org/>
* Method for storing data: <https://pandas.pydata.org/>
* For troubleshooting: <https://stackoverflow.com/>
* For general purpose: <https://www.w3schools.com/>

Functions:

* **Installing additional libraries** in case the user hasn’t them installed on their machine (by using “!pip!”)
* **Capable of reading** any **data from a datasheet** file that has the following format: Date, Time, value; and that at every couple of data, it starts a new line (we can also add a title at the start of the “group” of data if we want, but it is optional)
* **Organizing the data** read from the datasheet file in a single organized list (format: title, date\_list, value\_list)
* **Ability to “ask” the user to choose** which group of data he wants to visualize
* **Show various graphical widgets** to the user and allowing him to use various tool that will help him to easily visualize certain data
* **Capable of having interaction with the user** by using the “box select” (the data selected be updated into a table and a bar widget)
* **Ability to save Various info in a file txt**, the table widget will store all the action done by the user (that has used the “box select” tool) plus some additional data, and those can be downloaded in a txt file

Notes, Problems and choices

* Bokeh library: the reason why i used this library is because it is easily customizable and it supports the callback event, in wich though, you have to write the code in JS; there is a method to convert some data from JS data kernel to Python data kernel by using IPython, but because of google colab environment (which do not support the Jupyter Notebook environment very well), it is not possible.
* It was really tricky adding interactivity with the user since there isn’t really a solid way to let the 2 different kernel (google colab and console of the broser) communicate, fortunately, the bokeh library provide the ColumnDataSource object though which we can “communicate” from js code (callback event) to python code
* Note that Google Colab does not currently allow arbitrary websocket or notebook comms to be opened, making advanced Bokeh features unusable there, with the consequence of greatly restricting the interactivity with the user (for example we are unable to create a bokeh server with the consequences of inability to use curdoc, that is important to add interactive feature to the program (for example adding/removing dynamically widgets)).
* Another valid way to have interactivity with the user using a plot, is by using the Dash framewok (note that though, there is no direct way, and that you will technically “exit” from the google colab enviroment)