

Python Programming and Data Visualisation - application

Milestone 1

Karolina Gałczyńska, Yanina Shchaslyva

January 2022

1 Introduction

This document presents the development plan on an application created during the course of Python Programming and Data Visualisation course, with main functionality of gathering and presenting data in a web application, allowing the user to interact with the controls to change the data range on the visualisation charts. Project is developed with use of Python libraries: Plotly and Dash, using Pandas to manage data. Key functionalities and their current development will be presented below.

2 Application content

2.1 Data gathering and management

Data gathering will be implemented with use of a separate thread, running with the application. The thread will execute functions collecting data as JSON files from webpage and transform them, so that they can be placed in a DataFrame (implemented using Pandas library).

Data contains information of 6 patients, each one described with a number of constant variables (firstname, lastname, id, disability) and a number of dynamic values (such as measurement id, timestamp and values of 6 measurement on six sensors placed on both patient's feet along with the alarm flag in case of anomalies). All of that data will be stored in data structures, using one dictionary to store all of the data objects and their features stored statically with one DataFrame for each patient storing dynamic values from sensors. Data will be cleared periodically to only store records not older than 10 minutes from current time in application run.

Data will also contain one additional object with a DataFrame containing information on anomalies, such as patient id, measurement id and timestamp and sensor name with value at that measurement.

2.2 Data visualisation and page layout

The application will contain a series of plots, presenting clue variables in the stored data. We plan on using minimalistic layout with few unnecessary details, maximum of two plots in a row in an ordered grid with an interface allowing to choose one of the patients at a time. In styling, we will operate on simple color schemes, using white as background color and placing on it graphs in neutral colors, separate for each of the chosen graph for clarity.

Most important plots that will be implemented on the page will be a real-time visualisation of walking, plots of sensors values changing in time (separate for every of the sensors) and a chart presenting amount of anomalies occurring for the patient. In page layout we will also provide space for non-graph presentations on main data statistics (like minimal and maximal value) and on anomalies occurrences.

Below we present analysis of application according to Edward Tufte's rules of data visualisation.

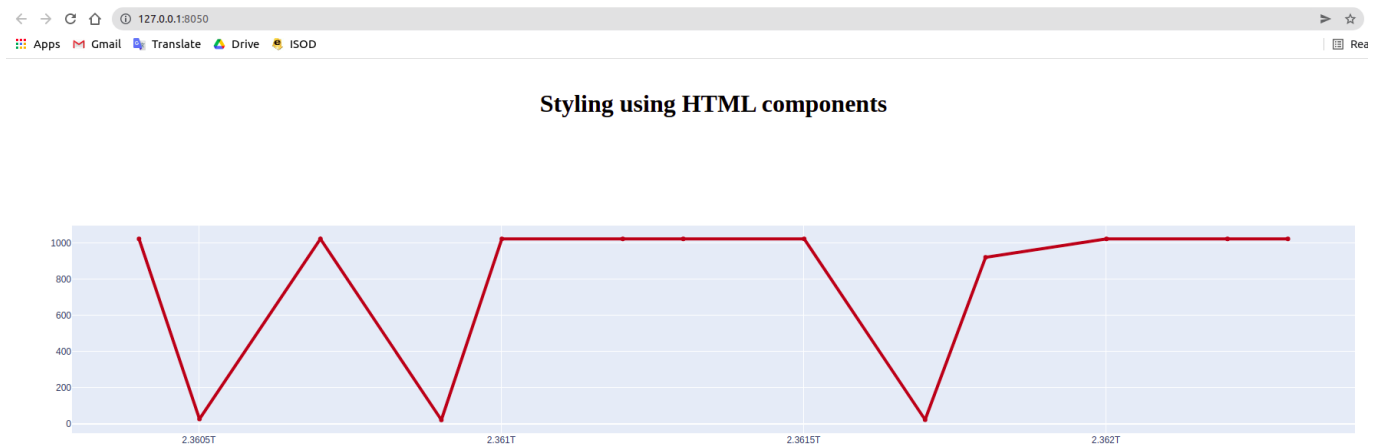
Name	Importance	Comments
Graphical integrity	medium	As this project is aiming to deepen our knowledge on data visualisation and skills in packages usage, we will aim to keep the graphic simple and consistent, though it will not be our priority overall.
Data-Ink ratio	high	The data we are working with is quite valuable. A lot of different plots and their variants can be depicted, so that we want to present only data that carry some important information. However, often it is a challenge to fit a lot of the information on one plot and still for it to remain simple.
Chartjunk	medium	Despite everything, some visual elements that are not even necessary, still remain important as they make graphs more pleasant and user aiming. There are different types of "junk" and it's matter, how to diverse the visual content.
Data Density	medium	For our amount of information it is quite easy to reach high density of the data on the plot. Our goal is to adjust it in simple and clear way to be understandable for observer.
Small Multiples	high	Data will contain of multiple dimensions of similar properties, so we will aim to highlight it separately in similar forms for each dimension.

3 Current development

Current application allows us to collect data with usage of threading, updating continuously data storage and removing unnecessary records. As this still is in development, the values that are currently used in functions will be adjusted to final project requirements in further stages, in order to allow us easier debugging in the process.

Simple plots have been prepared and compiled, though in the course of adding new functionalities (threading), new errors have occurred - once those are resolved, development process will focus fully on delivering high-quality, diverse data visualisation.

Below, there is a screenshot depicting some basic results that we managed to accomplish.



Rysunek 1: A basic plot of loaded data.