

# Plotting documentation:

## Code explanation:

There are 3 functions that are used in this script.

1. Zero\_to\_nannew: This function converts all the zero values to NaN so that zero values are not plotted on the graphs.
2. Combine: This function retrieves or returns(creates) the merged CSV file of a particular patientID in the mergedCSV folder.
3. Final: This is the function where data formatting and plotting is happening.

## Explanation:

1. In 'final' function, combine function is called to retrieve the CSV file of a patient. Then we start reading that file row by row and store the timestamps and FHR data in their respective variables.
2. In each row, there are 4 samples for each FHR, MHR, TOCO values. And we are storing all the four samples in particular variables with their timestamp. For example, we are storing it as a tuple :(time,mhr0),(time,mhr1),(time,mhr2),(time,mhr3).
3. Then X datapoints, and Y datapoints are collected for each of the three signals and stored in separate lists . In our code,  
listx is storing FHR values and listy is storing their corresponding timestamps  
listx1 is storing MHR values and listy1 is storing their corresponding timestamps  
listx2 is storing TOCO values and listy2 is storing their corresponding timestamps

As listx,listx1,listx2 is storing timestamps, we need to convert those timestamps into seconds relative to the starting timestamp. The lists corresponding to listx,listx1,listx2 are list\_x,list\_x1,list\_x2.

We then use these lists (list\_x,list\_x1,list\_x2) as our X datapoints for plotting and (listy,listy1,listy2) as our Y datapoints.

## Now the plotting part,

We are making 2 figures and in each figure there are 2 subplots in which the first subplot will show both FHR and MHR and the second subplot will show TOCO with a default range of 10 mins on x-axis and in second figure, the range is 15 mins.

Each subplot has certain specifications:

For x-axis:

- Vertical line every 10 secs
- Bold vertical every 1 min
- Very bold vertical every 5 min
- Limits: 10 minutes at a time
- Option to pan/scroll along the x-axis

For the y-axis:

- Horizontal line every 10 heart beats
- Bold horizontal line every 30 heart beats
- Limits: 30-240 heart beats
- y-axis ticks on both sides (left and right) at the same time

So, to implement this functionality, In one subplot, we used 3 x-axis and 3 y-axis where one x and y-axis are used to present the ticks or grid lines after every 10 secs. Second x and y-axis are used to present the ticks or grid lines after every 1 min(60 secs) . Third, x and y-axis are used to present the ticks or grid lines after every 5 min(300 secs).

The scrolling functionality is implemented by using `dragmode='pan'` in `plotly.update_layout` method.

### **Implementation of clicking on two points on the graph that will generate a pop up:**

In this part, the requirement was to display a filtered graph on a popup whenever two data points on a graph are clicked.

This has been implemented using dash and callback methods.

*display\_click\_data* is used to collect the points between the two clicked indices in a list.

*Update\_graph* updates the graph on the popup when points are clicked repeatedly and includes the plot specs.

*toggle\_modal* is used to define the behavior of the button which when clicked opens and closes the popup.

### **How to run the plotting script:**

While running the script, pass the patientID, in this case, let's assume 1234 is the patientID. So it should be something like - `python plotting_script.py 1234` (`python plotting_script.py <patientID>`) and after the script runs open a web browser and type 'localhost:8050' in the url.