Fleet Analytics Take Home Challenge

Problem Statement

The Fleet Analaytics team frequently works with timeseries signal data coming from vehicles across our fleet. This signal data can be torque values, RPM values, or even temperature within the cabin. We frequently need to analyze these signals by extracting specific metrics. Some of these metrics can be the most or least common values seen, the most or least time spent at given values, or the largest or smallest deltas seen in the signal.

In this challenge, you will write code to analyze sample timeseries provided in the `data` folder.

Question 1

This question should take ~40% of your time

Write a module that provides functionality to operate on a timeseries and allows for the following queries:

- The top N most/least common values by total occurrence
- The top N most/least common values by total time spent
- The top N largest/smallest 'cycles' - we will define a cycle as a progression from a given local minima/maxima to the next immediate local minima/maxima. The amplitude of a cycle is the difference between the maxima and the minima.

Using this module, compute the following **for car_0**:

- The 5 least common values by total occurence
- The 3 most common values by total time spent at that value
- The 3 cycles with the largest amplitude. When returning a cycle, please provide the value of the minima, the value of the maxima, and the amplitude of the cycle. Bonus if you can also provide the duration of the cycle.

Question 2

This question should take ~60% of your time

The `data` folder contains drive unit torque data for 10 vehicles in the form of 10 timeseries. Two of these vehicles have experienced higher damage accrual than the other 8. Determine which 2 vehicles are the ones behaving differently and develop a generalized method to programmatically differentiate these 2 vehicles from the other 8.

Deliverable:

Provide directions to run your code and to recreate the solutions to the above questions. This would include installing dependencies, specifying a path, or running the executable. Python is the standard expected language for this coding challenge. You may use any library. However, if using a library outside those available in the standard python installation (numpy, pandas, collections), make a note of it and explain why it is necessary.

Provide solutions along with runtimes for each question. For any modules or funtions that you provide, document the Big-O complexity of each API call in your class as a function of the length of a given timeseries N.