CS 207 | Systems Development for Computational Science

A time series class, continued

Part 1 Courtney volunteered! (Oct 11 commit)

This question continues your implementation of the time series class you started last week. Then, you stored your time series as a python list.

Add methods `__iter__` to your project Time Series class to iterate over values, a method `itertimes` to iterate over times, and a method `iteritems` to iterate over time-value pairs. (This is a similar interface to python dictionaries). To test these, check both the types of the results and the answers you expect.

↑ issue #4, assigned to Courtney by Sarah

Part 2 Lu Shen

Next, you will store the data as a numpy 1-D array.

Please implement a new class `ArrayTimeSeries` which inherits your `TimeSeries` class and uses `numpy.array` to store the data internally.

At this point you will notice a fundamental mismatch: unlike python `list` s or `array.array` s which are dynamic, `numpy` expects you to provide a length of the sequence. This means that you cant process each element of the time series as it comes in, and must calculate the length of the input sequence to allocate space for the time series in your constructor. (This makes it hard to write a "direct from file" without intermediate storage implementation; but we shall worry about this later).

Because your class inherits from your `TimeSeries` class, you'll notice you get some functionality automatically. You may choose to reimplement `__getitem__` , `__setitem__` , `__len__` , and the iteration functions at your discretion. Please do NOT implement a `__str__` or `__repr__` function. Note that this means the `__str__` and `__repr__` functions will need to work with both classes, and you may need to fix your original versions.

Make sure that any doctests and tests you wrote before to test all kinds of sequences as input still pass. (If you didnt write any, how do you know your time series Class from last time is any good? We'll be testing your code by running our own tests as well!) Laura Ware (issue #21)

Next weeks we'll refactor these classes and add support for synthetic time series, and incoming data that is streaming!