36601   HW6: Python version of process_files()   Due 10:00 PM October 9.

You are already familiar with the weird format of the Energy Monitor, Inc. data files. Briefly, we process all ".txt" files in one folder. A valid header line has the form "# integerCode yyyy-mm-dd". Files without this header are ignored (with no message). The remainder of the file is data lines of the form "location; v1, …, vN; t1, …, tN" where N varies by line and must be less than 'maxTimes'. Some data lines may be preceded by a multiplier line of the form "mult realNumber", in which case the values (but not the times) of the next line are multiplied by the real number.

The process_files() function is given a folder and maxTimes (or uses appropriate defaults) and processes all of the valid files in that folder. For the Python version, the output is printed to stdout, so that we can see the output or redirect it to a ".csv" file. The full details of the output are shown in the file. In Python, we use list comprehensions or maps where appropriate, and are free to use "for" loops. Note that because the output is text, it is not always necessary to convert the (text) input to numeric.

Work in Code Studio, and assure that the "pep8" (not "pylint") linter is turned on.

Use and modify the code in "process_files.py". Fill in any missing code indicated by ">>code<<". Read all comments! Read the code that I wrote, and learn from it. Test your code at the operating system prompt with no arguments, with a folder argument, and with both a folder and 'maxTimes' argument. Be sure to test different data in a different folder, too.

Note that if you redirect your output to, say, "out.csv", then you should be able to read it into R using:
`read.csv("out.csv", na.strings=c("N/A", " N/A"))`

Turn in **process_files.py**. If you resubmit, use the same file name, and Canvas will add a suffix.

As always, ask for help if you are unsure what to do or how to proceed. Make at least one first attempt at least two days before the assignment is due to be sure you have time to ask for help.