UNIVERSITY OF MICHIGAN

EECS (SI) 182: Building Applications for Information Environments

Assignment 6 – Putting Everything Together So far. Spreadsheet Data Analysis

Points: 40 (plus extra credit: 5 points)

Due Date: Wed. Oct. 22, 2008 at 7:00 PM (earlier submission preferred). Note: ctools will accept submissions till 11:55 PM, without penalty, which is very strict.

Your mission, in this assignment, is to bring together various techniques we have learned to analyze the data in spreadsheets using Python's map/reduce/filter functions. You will write some fairly general-purpose functions that you will be able to use beyond this class to analyze data in most spreadsheets. Each of the functions is fairly small (less than 5-6 lines of code), but when put together, this is an ambitious program that is beginning to do a realistic real-world task.

A spreadsheet is essentially a table, or two-dimensional data. We saw one example of a spreadsheet in csv format in an earlier assignment. Below is an example of a few rows of a spreadsheet, representing scores of students in 182. The format is

<studentname>, <assign1 score>, <assign2 score>, <assign3 score>, <assign4 score>

Name4,19.0,24.5,13.0,39.0
Name5,19.0,23.0,11.0,38.5
Name6,18.0,24.0,14.0,39.0
Name7,18.0,25.0,15.0,39.5
Name8,17.0,24.0,14.0,38.5
Name9,20.0,25.0,14.0,40.0

Notice that all the fields in a row, by convention, are comma-separated.

The kind of tasks we may wish to do on the above data:

- Add elements in a column: Compute the average value in a column (e.g., assignment 1)
- Add columns: Add the values for the assignments for each student
- Filter: Find over-achievers in the class: those with above 95 in the total score for the assignments

More generally, if you imagine any spreadsheet, the analysis will usually require:

- Extracting a column
- Manipulating one or more columns: E.g., adding corresponding elements of a column
- Reducing the values in a column to a sum or some statistic

With that in mind, the task in this assignment is to write some functions to help analyze data in spreadsheets. We have given you some code that you can start with. It is in the file assign06.py. The code is fairly-well commented and explains what you need to do. Ask questions if it is not clear. In short, here is what you will need to do:

1. *tocolumns(rlist):* Write the function tocolumns(rlist) that takes the spreadsheet data in row-by-row format and returns the data in column-by-column format. For example, applying tocolumns to

   [['a', 'b', 'c'], ['d', 'e', 'f']]

   returns the following:

   [['a', 'd'], ['b', 'e'], ['c', 'f']]
2. *tonumbers(clist, i):* Fix the code that I wrote incorrectly that converts strings to numbers in column *i* of a spreadsheet. Assume that the *clist* is in column-by-column format. Also generalize it so that if the column contains values that cannot be converted to numbers, they are converted to a default value of 0. I have given partial code.
3. *writerows(outfile, rows):* creates a long string out of rows in the csv format (with each row separated by a newline) and writes it to the file outfile. I have given partial code.
4. *torows(clist):* dual of *tocolumns.* Takes in a clist in the column-by-column format and returns the corresponding version of the data in a row-by-row format.
5. Add in some tests to make sure that your *tonumbers()* function works. As an example of how to write testcases, I have given tests for *tocolumns()*
6. Run the tests and the main program. *The main program will currently give errors because the code is not yet complete.* Once you write the above functions correctly, it should work.
7. Add in code to extract student records for overachivers (those with total score > 95) and write that back to a file.
8. Check the output of allresults.csv and overachievers.csv to make sure they appear correct.

Generally, I use “pass” in the sample file where you have to write new code. “pass” is like an empty statement in Python. It does nothing.

There is an opportunity for extra credit in this assignment. You need to write and test the function generatestudentemail() and generatestudentemails(). The extra credit is worth 5 points and can be used to make-up points on previous assignments. Or do it for fun if you don’t need the points!

What to Submit:

1. Submit the modified file with the name assign06.py.
2. Submit the output files allresults.csv and overachievers.csv
3. Submit a screen snapshot that shows you running assign06.