CS 234 Data, Analytics, and Visualization (Fall 2017)

09/15/2017 Class Worksheet Nr. 1 - Python Review

Here is a list of problems to review your Python knowledge. Create a notebook and work on each problem. Upload the notebook and the HTML page of notebook by next week.

Read & write files / User Menu

Problem 1: Write a program that generates a list of 10000 random integer numbers (between 0 and 10000), writes them into a TEXT file, one per line, then, opens the file, reads the numbers, and prints out the mean and median values for the list of numbers. What is your expectation for these two values? How will the distribution of values look like?

Problem 2: Solve problem 1 using a JSON file instead of a TEXT file. Remember, there is no notion of "one per line" in a JSON file. What is the difference between the two solutions?

Problem 3: Rewrite parts of the solution of Problem 1 and Problem 2 to become modular, using three functions: 1) a function that generates a list of random integers, where the parameter is the size of the list; 2) a function that writes the list on a file, with two parameters: the list of numbers and a filename (whose ending might be "txt" or "json". 3) a function that reads from a file, given as parameter a filename (whose ending might be "txt" or "json").

Problem 4: Create a Python application that uses `if __name__ == "__main__"` and does the following: it shows a menu that asks the user to choose one of the three things: a) generate a file of random integers with a desired length and desired file type; b) print out mean/median values **for all files of integers** that it has stored; c) quit the program.

Note: A program that generates many files will have to deal with the issue of how to store the files with different names, so that they don't override one another. There are many ways to deal with this issue:

a) One can create a file info.data that keeps track of the total number of files in a folder (or their names) and can be queried (and updated) every time you write a new file.
b) One can use the system time: `time.time()` to get the number of seconds since Jan 1, 1970 (read about system time: https://en.wikipedia.org/wiki/Unix_time) and use the rounded value as part of the filename you'll create.
c) One can use the `datetime` module, to capture date and time and use that as part of the filename.

Whichever strategy you use, you'll need to write a specific function that performs the task of generating the filename for the files you'll be creating.