

DSC 430: Python Programming  
Assignment 0601: Avocados

Find on the D2L a datafile named “avocados.csv” -- Retrieved from Kaggle (9/26/2018). This data was downloaded from the Hass Avocado Board website in May of 2018 and saved as a single CSV.

- a) Define a function that takes a variable name in the form of a string (e.g. “Total Volume”), reads into memory the values for that variable (but just that variable) and computes the mean using the statistics module.

```
mean_SM = readAndComputeMean_SM("Total Volume")
```

- b) Define a function that takes a variable name in the form of a string (e.g. “Total Volume”), reads into memory the values for that variable (but just that variable) and computes the standard deviation using the statistics module.

```
sd_SM = readAndComputeSD_SM("Total Volume")
```

- c) Define a function that takes a variable name in the form of a string (e.g. “Total Volume”), reads into memory the values for that variable (but just that variable) and computes the median using the statistics module.

```
median_SM = readAndComputeMedian_SM("Total Volume")
```

- d) Repeat a-c, but instead of using the statistics module write your own “homegrown” code to compute the mean, standard deviation and median.

```
mean_HG = readAndComputeMean_HG("Total Volume")
sd_HG = readAndComputeSD_HG("Total Volume")
median_HG = readAndComputeMedian_HG("Total Volume")
```

- e) Repeat a-c, but your functions must be memoryless – **you can hold in memory only a single line from the file at any given time**. You may need to keep track of min, max, sum and a handful of counters.

```
mean_MML = readAndComputeMean_MML("Total Volume")
sd_MML = readAndComputeSD_MML("Total Volume")
median_MML = readAndComputeMedian_MML("Total Volume")
```

Write test code to demonstrate that the means, standard deviations and medians are the same across all three techniques. The first 8 functions are trivial. You will not be awarded points for completing them (though you may lose points if you do not.) The 9th function, “readAndComputeMedian\_MML”, is the challenging part.

Record a three minute video in which you run the code. Then, present your code. Specifically, answer the following questions:

- What variables, counters, etc did you use in your `readAndComputeMedian_MML`?
- Discuss how your `readAndComputeMedian_MML` maintains a small memory footprint regardless of the size of the data file.?

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date, video link, and the honor statement, "I have not given or received any unauthorized assistance on this assignment." Each function must include a docstring and be commented appropriately.