PHY499: Introduction to Python for Scientists

Homework Assignment 4 (14 Oct, 2016)

(NOTE: This assignment will be graded!)

Only use functions, methods, and modules that we already discussed in class to solve these assignments! Send your code to `michael.mommert@nau.edu`.

1 Numpy, numpy, and more numpy...

1.1 Flagstaff Climate - again...

Redo part of assignment 2 - but this time using numpy. Read in the two data files (`flagstaff_climate_2015.dat` and `flagstaff_climate_monthly.dat`), calculate the average high temperatures and low temperatures and the total precitipation for each month in 2015, and determine the residual of each month compared to the average (from the "monthly" data file).
Output: Print the average temperatures/total precipitation per month in 2015, as well as the corresponding residuals, in a nicely formatted table on the screen. You don't have to write your results into a file and you don't have to calculate the average residuals over the whole year.
Can you do this task using only 6 lines of code? This is why you should get familiar with numpy...

1.2 Statistics

1.2.1 Normal Distribution and Small Number Statistics

Small samples can affect statistics. Show this effect by calculating the mean, median, and standard deviation for samples of normally distributed numbers (use `numpy.random.normal`, pick a mean and a standard deviation) of different sizes. Start with a sample of 10 and end with a sample of 10^7 random numbers (use decadal logarithmic steps: 10, 100, 1000, ...). How large does the sample have to be to provide standard deviations that are accurate to within 1%?
Output: Print the sample size, mean, median, and standard deviation as a function of the sample.

1.2.2 Monte Carlo Integration

Monte Carlo integration is a method to integrate a function that is usually too complicated to be integrated analytically numerically. The idea is very simple: you randomly draw numbers from the range over which to integrate and determine the function value at each of those positions, derive the mean of all function values, and multiply that mean with the width of the integration range. This approach is based on the idea that you probe the function at different locations and simply multiply the average function value with the range. Naturally, the accuracy of the result improves for large numbers of random numbers.

Use Monte Carlo integration to integrate the function $f(x) = \sin(x)$ over $0 \leq x \leq 2\pi$ using random samples with 10 to 10^7 numbers (logarithmic steps).
Output: Print the random sample size and the resulting integral as a function of the sample size.

1.3 Sorting

Generate an array of 10 random numbers (any range) and produce an array of indices that would sort the array in ascending order – again: do not simply sort the array, provide the list of indices that would sort the random numbers. Verify your code against `numpy.argsort`.
Output: Print the array of indices generated with your code, the output of `numpy.argsort`, and the sorted random numbers based on either of the index arrays.
Hint: Look into what the numpy functions `sort` and `where` do.

If you want this assignment to count into your final grade, please submit it to `michael.mommert@nau.edu` before 27 Oct, 23:59!