PYTHON Tool Box

Every student should prepare a "Tool Box" of PYTHON programs addressing each of the topics listed below. Your programs should be designed to operate on a general set of input data, and provide useful numerical and/or graphical output. Be sure to test each program with your own input data.

The PYTHON routines are to be used ONLY FOR ARITHMETIC (such as matrix inversion, summing computations, and the like), as well as for plotting functions. You may not, for example, use the 'canned' fitting or statistical functions built into the program. Instead, you should program these functions yourself.

Save your program scripts as files and put them in a folder named TOOLBOX on a thumb drive. **You must use the input and program file names for each problem as they are given below.** *You need not be concerned that the output from your programs abides by the rules for quoting significant figures.*

Each program file should include a header in which the program's functions and the format of its input file are summarized. Your Tool Box will be evaluated based on how well your programs are able to run and analyze data sets provided by the instructor, and if they abide by these instructions.

Your thumb drive will be returned. Then, when you enroll in PHY 435 or 535 you will then be expected to use your own programs to analyze the data collected from your own experimental projects.

1. Compute the sample mean, the sample variance, the sample standard deviation, and the error in the mean of a set of repeated measurements.

Input: File SS.dat containing an arbitrary number of numerical values (real or integer) arranged in a single-column format.

Output: Screen display of the requested results.

Program File Name: SampleStats.py

2. Compute the weighted mean, and the error in the mean, of a set of repeated measurements.

Input: File WM.dat containing an arbitrary number of numerical values arranged in a 2-column format: (y, σ_y).

Output: Screen display of the requested results.

Program File Name: WeightedMean.py

3. Make a transformation from a measured variable and its uncertainty (y, σ_y) to a new variable

$$z = log(y)$$

and its uncertainty σ_z.

Input: File LD.dat containing an arbitrary number of numerical values arranged in a 2-column format: (y, σ_y).

Output: Screen display of the corresponding values of (z, σ_z).

Program File Name: LogDataM.py

4. Make a linear point plot of a data set, showing the measurement uncertainties as error bars.

Input: File LPD.dat containing an arbitrary number of numerical values arranged in a 3-column format: (x, y, σ_y).

Output: Screen display.

Program File Name: LinearPlotData.py

5. Make a semi-log point plot (ln y vs x) of a data set showing the uncertainties as error bars.

Input: File SLPD.dat containing an arbitrary number of numerical values arranged in a 3-column format: (x, y, σ_y).

Output: Screen display.

Program File Name: SemiLogPlotData.py

6. Make a linear connect-the-points plot of a set of (x,y) values.

Input: File LPP.dat containing an arbitrary number of numerical values arranged in a 2-column format: (x, y).

Output: Screen display.

Program File Name: LinearPlotPoints.py

7. Do a linear-regression fit to a function of the form

$$y = a \times x$$

given a set of data with their associated uncertainties. Make a linear point plot of the data showing the uncertainty as an error bar. Overlay the fitted linear function as a smooth line.

Do not use the statistical packages available in PYTHON in your program. Instead, use PYTHON to do the arithmetic needed to implement the general formulas which you have already derived in a previous homework assignment.

Input: File R1P.dat containing an arbitrary number of numerical values arranged in a 3-column format: (x, y, σ_y).

Output: Best-fit value of a *and its uncertainty,* σ_a*; the value of* χ^2*; and a screen display of the plot.*

Program File Name: Regress1Par.py

8. Do a linear-regression fit to a function of the form

$$y = a + b \times x$$

given a set of data with their associated uncertainties. Make a linear point plot of the data showing the uncertainty as an error bar. Overlay the fitted linear function as a smooth curve.

Do not use the statistical packages available in PYTHON in your program. Instead, use PYTHON to do the arithmetic needed to implement the general formulas which appear in your textbook.

Input: File R2P.dat containing an arbitrary number of numerical values arranged in a 3-column format: (x, y, σ_y).

Output: Best-fit value of a *and* b *and their uncertainties,* σ_a *and* σ_b*; the value of* χ^2*; and a screen display of the plot.*

Program File Name: Regress2Par.py