1st Friday Mini-Lab

In Lab - practice plotting and fitting with Python

Megg, a physics major, wants to determine the spring constant k for a small spring. They try to measure this by hanging the spring vertically, adding several different masses M on the end of the spring, and measuring how far the spring stretches or extends, y. Basic physics tells us:

$$ky = mg \tag{1}$$

Sketch this setup on a piece of paper, that you will later insert into your digital lab book.

Megg measures y as a function of M. Their results are

M (g)	y(cm)	y error (cm)
10	8.5	0.4
20	12.8	0.4
30	17.3	0.4
40	20.6	0.4
50	24.6	0.4
60	29.3	0.4

Today's exercise for you will be to analyze this data and extract k from the slope, with uncertainty, using `Python`. Suggested steps include:

1. Start a text editor (ie `notepad` (windows), `TextEdit` (Mac), or `gedit` (Linux). Type in the data using tabs between columns. Make sure to label the top of the columns of the data, putting a # in front of this text. `Python` (and other languages) can be set to ignore any line that begins with #, so you can put comments or column headers in by making the first character of the line #. Save the file with an extension `.txt`. (thus: `springdata.txt`)

   Note: Make sure the file is in the plain text and not rich text format. If you are not sure what format you are in ask your instructor. In TextEdit for a Mac go to "Format" → "Make Plain Text" at the top taskbar.

2. Open Python (either in `Spider` via `Anaconda` if using a local install, or with `JupyterHub` if using the course server) and plot the data (including error bars!)

   To do this, download the script shell "`mini-lab-shell.py`" from myCourses to get started. Save it in the same folder as your data (or upload if using `Jupyer`) and give it a new name like "`McLane-mini-lab.py`". Typically you want a different folder for each experiment. Within the code, change the name of the file to plot to your file name, and then to execute this script, from within `Python`. If it doesn't do quite what you wanted edit the script, save it again, run it, and see if that works. The great thing about a script is you then have a file with all your commands that worked to make your nice plot. You will need to upload a copy of this script to your notebook, and save it to use as a starting point for the next thing you have to plot or fit! In reality no one ever starts each script from scratch, they use one that has worked in the past and modify it for each new use.

3. "Beautify" your plot by adding x and y axis labels, a title, etc. These commands should already be in the script, you just have to uncomment them and add your specifics. Get in the habit of running your python script (or just the individual code cell you changed) every time you make a change. It will make your life a lot easier when it comes to trouble shooting.

4. Fit a function of the form $f(x) = s * x$ to your data. Record the value of the slope s and its uncertainty from the fit. Replot your data with the fit line as well, and since we now have two different things on the plot add a legend.

5. Now export this image so you can put it into your digital `OneNote` notebook. This can be as simple as right clicking on the image, then copy and pasting it into your notebook. There are also (better) ways to save the image but for now this will suffice.

6. Express k as a function of the slope s. Compute it from the fitted slope, with units. For this lab you may assume $g = 9.8$ m/s^2

7. Find the uncertainty in your k from the uncertainty in the slope, using the "rules" discussed in lecture (skip this for now if we haven't had lecture). What is your final value for k with uncertainty?

8. Look at your plot. Does it "look like" a good fit? Try a new fit, with a linear function $f(x) = s*x+b$ instead of the simple proportionality you did before. Does this look better?

9. Again, record the slope s, intercept b, and the uncertainties in both. Use these values to extract k with its uncertainty as before.

10. Since you've created a new plot make sure to also put this one in your notebook. Basically any time you make a new plot it should go in your notebook!

11. Save your `.py` file and insert it into your `OneNote` notebook. If you are using a desktop application this is as simple as navigating to the folder where your file is located, then drag and drop the file into `OneNote`. For `Jupyter` you will need to first export the file to your hard drive. At the taskbar within Jupyter click on "Download As" → "Python (.py)".

Real springs, with actual mass, do not follow the simple form of Equation 1. In fact, the spring's mass contributes an additional term, an effective mass for the spring m_{eff}, so that:

$$ky = (m + m_{eff})g \tag{2}$$

From your slope and intercept, determine the effective mass of the spring, with uncertainty.

This first exercise is part of the "pre-lab" for the first week's lab, measuring g. Insert this as the "pre-lab" section in your measure g `OneNote` notebook. This will how the other experiments will work, where in those cases the pre-lab will match the in person component.

The following link contains various derivations for the effective mass of a spring; for a spring of uniform mass density that isn't stretched beyond the linear Hooke's law regime, $m_{eff} = M/3$ where M is the actual mass of the real spring.