WPI | Department of Physics

Introduction to computational physics

111X Lab 1R

Last Edited January 10, 2022

Abstract

For today's scientists and engineers, the ability to read and write code is a basic requirement. Though many of your problem sets can be written out by hand, or completed with the assistance of a trusty graphing calculator (or Wolfram Alpha, if that's more your speed), the work of modern research in both academia and industry requires more advanced tools. Python is a free, open source, easy to learn programming language, and very well supported for data science and research applications. A researcher should also be able to translate raw data into measurements, and know how to evaluate the quality of said measurements. Let's kill two birds with one stone.

Lab Objectives

- Basic Python Skills [1]
- Reporting a measurement
- Uncertainty: Accuracy vs Precision

Lab Equipment

- Computer.

Deliverables

- Your Jupyter Notebook with questions answered will be submitted on Canvas as a pdf.

Getting Started

This lab guide is meant to be used as a reference for the in person and remote, synchronous, first lab of PH1110 or PH1111. Make sure you agree to our lab rules on Canvas before your lab section meets, as it is graded and contains important lab information.

During lab, your instructor will give an interactive Python tutorial, followed by a discussion on how to talk about error and ending with semi-guided data set analysis. There will be a few mandatory checkpoints to make sure nobody is falling behind - don't miss them, they are worth points. They are summarized below in the 'Deliverables' section. If you can't attend the live lab section for whatever reason, all materials will be available on your course Canvas page. For any questions, comments, or concerns, please contact your lab instructor or the lab manager; we're on your side, and we want you to succeed!

Learning Python

This lab is almost entirely interactive, guided by the live session with your instructor. Follow along with the Python tutorial given by the instructor during your lab time. All instructions for how to set up your Python workspace are included in the tutorial. Feel free to experiment with any of the example code given and add as many comments as you want or need.

If you are already familiar with Python or computer programming in general, this section is going to be review for you, and may seem a bit slow. This is intentional – you are not the target audience! We want to make sure that everyone taking this class achieves a certain level of skill with Python and, perhaps more importantly, becomes comfortable with the language and knows how and where to get help.

Data Analysis

All instructions for the guided data analysis can be found in, you guessed it, the interactive lab tutorial presentation. This is just a summary of what we're expecting you to do, but feel free to skip ahead to here if you're already comfortable with Python.

Download the file 'lab1data.csv' from Canvas and upload it to your JupyterHub. Before doing any quantitative

[1] No pre-knowledge required!

analysis, graph the data (we use Matplotlib.pyplot) to visualize the set. Find the means and standard deviations, and think about how many significant figures they should have. Finally, reflect on how you could conduct a future, follow up experiment to improve the precision and/or accuracy of these measurements.

What to turn in

At the end of the lab, all you need to do is submit the code file that you have been working on to Canvas. If you've been following the tutorial, it will contain lots of example code plus the results from several dark gray TRY IT OUT boxes and the two group (or solo, if asynchronous) work exercises in function design and data analysis. You are only required to have the 'Deliverables' that is summarized below. Make sure your name and the date are written in a comment at the top of your file.

```
import math

# u_area: propagation of uncertainty for area calculation
# -> x: width measured (cm)
# -> dx: error of width (cm)
# -> y: length measured (cm)
# -> dy: error of length (cm)
# <- uncertainty of calculated area
def u_area(x, dx, y, dy):
    area = x * y
    x_component = (dx / abs(x)) ** 2
    y_component = (dy / abs(y)) ** 2
    answer = area * math.sqrt(x_component + y_component)
    return answer

width, length = 5.3, 8.5
u_width = u_length = 0.1

area = width * length
print("Area =", area, "+/-", u_area(width, u_width, length, u_length))
```

```
Area = 45.05 +/- 1.0016985574512922
```

Figure 1: Propagation of error for area equation that is essential for this lab, and others.

Deliverables

1. GROUP WORK: Fun with functions. Introduce yourself! Name, prospective major, fun facts? Work together to create functions for each of these equations: Volume of a cylinder, Pythagorean theorem, Quadratic Formula [2]

2. TRY IT OUT: On Canvas, find the file lab1data.csv in the lab module. Download this file, then upload it to Jupyter as shown in the slide deck.

3. TRY IT OUT: Follow the example code to graph the data you just uploaded. Don't forget axis labels!

4. TRY IT OUT: Find the mean and standard deviation of your data set and print the values as shown here. How many digits to the left of the decimal should you keep? Think about it, then check your answer.

5. GROUP WORK: Your data set represents the estimated mass (in M*) and distance (in parsecs) of a newly discovered planet, as calculated by some amateur astronomers. What would you recommend they do next to improve this measurement? Why? Discuss using error terms from this section! Summarize your group's findings (in your own words) in a 2-3 sentence comment at the end of your notebook.

For Lab 2R, **if you are a remote student**, you need to download Tracker to a computer which you can use to analyze video's.

[2] Quadratic formula is extra credit