PHYS 4023
Fall 2025

Intro to Python Plotting Homework
Due September 5th

Introduction: It's important that plotting data is efficient and convenient for you. The python module Matplotlib is an particularly useful plotting environment. This homework is meant to get you started on making a few simple plots, and examining some more sophisticated plots that you might use later in this course (or later in a research project or similar). I have provided some 'cheat sheets' for matplotlib commands on the course GitHub page that may be useful, and you can also reference short discussion of plotting in seciton 1.5 of the textbook.

For this assignment, create a notebook file with code cells dedicated to each problem (be sure to clearly label which problem is which either using code comments, markdown or a combination of the two). Once complete, push your completed file to your GitHub repository for grading. Be sure to check the plots are clearly visible in the online repository after uploading.

1. Generate a list of one hundred data points x_n that are equally spaced in the interval [0,2π]. Use this to evaluate $f(x_n) = \exp[6\sin(x_n)]$, and then produce a plot of x_n vs. $f(x_n)$. Produce two versions of the plot, one where the data are represented by red diamonds that are disconnected, and a second, where the data is not shown as markers, but instead a continuous green curve. In both graphs, label the axes and title the plots "Problem 1 A" and "Problem 1 B".

2. Download the file NoisyFunction.dat provided on the course blackboard page. This file contains data in the form (t_n, $y(t_n)$) pairs, which can be loaded into python. Create a scatter plot of t_n vs. $y(t_n)$ with appropriate axes labels. Next, Calculate $\log(y^2)$ as an array, and plot it versus t also as a scatter plot. Note that you may use any marker and color combination for the plots that you want, but use something other than the default style to get some practice with different options. Make sure the plots have appropriate axes labels and title the graphs "Problem 2 A" and "Problem 2 B".

3. Go to https://matplotlib.org/stable/gallery/index.html, which is a repository of different types of matplotlib plots. Take a moment to look over the different types of plots present. Click on a plot that you think is interesting, and look at the code used to create the plot. Note that there are a wide variety of examples with matplotlib, and these code examples can be an excellent starting point if you're ever not sure how to make a particular type of plot. Next, locate examples dealing with creating a histogram and creating subplots. You will need to reference these for the next question.

4. Go to https://www.kaggle.com/datasets, which is Kaggle's repository of public data sets. Take a moment to explore the different data and categories available on the site. Find a data set that is interesting to you and lends itself to produce an easy plot or two. Create a 2x1 subplot where at least one of the subplots is a histogram using the dataset of your choice from Kaggle. You can pick the style for the second subplot (histogram, scatter, pie, etc.). Label axes on the plots appropriately and set a title on the plots called "Problem 4" and save. Note if you have trouble picking a dataset, https://www.kaggle.com/datasets/alexisbcook/data-for-datavis has some good options.