PHY1112: Assignment 6

> Pretty Pictures and Concise Captions

Assigned: February 13th, 2024

Due: February 27th, 2024

Learning Objectives

1. Learn more about available plotting functionality in matplotlib
2. Learn more about formatting and finalizing your python plots
3. Practice applying mathematical formulas to your NumPy arrays

Grade Breakdown

|  |  |  |  |
| --- | --- | --- | --- |
| Part | 1 | 2 | Total |
| Points | 19 | 15 | 33 |
| Score |  |  |  |

**Question 0: Quality Captions are Key**

In this assignment, you will be generating plots and saving them for submission. The plots should be formatted well, with proper titles, labelled axes, and **figure captions**.

A quality plot is easy for a reader to understand. A big part of this is including a good figure caption that:

* Is **clear and concise**. Your figure captions are not your analysis, they should not be too wordy.
* It should **accurately describe the content** and answer the question “what am I looking at?”. It may point out key elements of the figure.
* It should **include a Figure number** for reference in any analysis or discussion in the main text of the document.
* When relevant, it should **mention the methodology or source** of the data, for example whether this is from theory, experiment, simulation, etc.

Here is an example:

A graph of a graph showing a blue line

Description automatically generated

**Figure 1:** Real (blue) and imaginary (green) parts of the relative permittivity of an optical material described by a Lorentz oscillator model with a resonance at 380 nm.

**Question 1: Dicey Situation**

Referring back to Assignment 4, Question 2:

1. Write a function that takes in a number `number\_of\_dice` and returns all possible combinations of dice rolls[[1]](#footnote-2) as a 2 dimensional NumPy array, using `itertools.product`.

Make it general to any number of sides by adding a `number\_of\_sides` argument to your function.

The number of columns of your output array will equal the number of dice, and the number of rows will equal the number of possible combinations.

**(4 marks)**

1. Now consider 2 dice, each with 6 sides.
   * Write out all the possible combinations of dice rolls by hand.
   * Sum the rolls by hand, and
   * Count (again by hand) how many times each sum value appears.
   * Then, make a table that contains the possible sum value on the left, with how many times it appears on the right.

Provide the answers to each bullet point in your document.

**(4 marks)**

1. The chart you made in part ‘b’ is the data you need for creating a histogram that tells how probable each sum is when you roll two dice. To get the probability, you need to normalize your count data by dividing each number in the right hand column by the total number of possible rolls.

Implement the sums vs probability data as two 1D NumPy arrays. Use the ‘bar’ function to plot a probability histogram. Include appropriate title and axis labels. Take a screenshot of your plot and put it into your report.

**(2 marks)**

1. We are going to repeat part ‘b’ numerically through the following steps:

* Use your function from part ‘a’ to make an array with all the possible combinations of dice rolls. Print out your array and include a screenshot in your answers.
* Next, make a new array that contains, for each possible combination (*i.e*, each row), the sum of both dice. It should be a 1D NumPy array, with a length equal to the number of rows in the “all possible combinations of dice rolls” array. Print out your array and include a screenshot in your answers.
* Make another new 1D NumPy array that holds a sorted list of all the unique values of the sums (equivalent to the left side of your chart of part ‘b’). Print out your array and include a screenshot in your answers.
* Make another new array that holds how many times each value occurs in the “sums of both dice” array (equivalent to the right side of your chart of part ‘b’). Print out your array and include a screenshot in your answers.  
  **(4 marks)**

1. Use the ‘bar’ function to plot the probability histogram from the data you obtained in part ‘d’, properly normalized as before. Include appropriate title and axis labels. Take a screenshot of your plot and put it into your report. Do your results align with what you obtained by hand?

**(2 marks)**

1. Be sure to submit your two plots with appropriate figure captions.  
   **(2 marks)**

**(19 marks total, 1 for docstrings/file header/variable naming/comments)**

**Question 2: Scatterbrained**

This question is a continuation of the Rigid Body problem from Assignment 5, Question 3.

For this assignment, a new data set is being provided that contains the masses of particles, “PHY1112\_A6\_Q2\_Masses.csv”.

Data files containing the particles’ corresponding positions and velocities were given for Assignment 5 (“PHY1112\_A5\_Q3\_Positions.csv” and “PHY1112\_A5\_Q3\_Velocities”)

1. Recycling your previous code, read in the all the positions and velocities for the Rigid Bodies and store them into NumPy arrays.

**(1 mark)**

1. Using the `scatter` plotting function from matplotlib.pyplot, plot the (x,y) positions of the Rigid Bodies as black filled circles on an x-y graph. Don’t forget to include a title, axis labels and a legend. Include a snapshot of your figure.

**(3 marks)**

1. Read in all the masses for the Rigid Bodies and store them in a NumPy array.   
   **(1 mark)**
2. Determine the average speed and average mass of your Rigid Bodies, as well as the standard deviations of speed and mass. Report the values you obtained.

**(4 marks)**

1. Determine the x and y coordinates of the Center of Mass of this particle distribution, using the formulas:

Report your values.

Then, add this point to the figure you created in part ‘b’ as a red square, and ensure it is included in the legend. Include a snapshot of your figure.  
**(3 marks)**

1. Be sure to submit your two plots with appropriate captions  
   **(2 marks)**

**(15 marks total, 1 for docstrings/file header/variable naming/comments)**

1. Different orderings of the same numbers are considered distinct. For example, in the case of 2 dice, a roll of (1,2) is distinct from a roll of (2,1) and are both considered as possible combinations. [↑](#footnote-ref-2)