Code 1: Python Data Analysis

- Due Feb 10 by 11:59pm
- Points 80
- Submitting a website url
- Available after Jan 27 at 12am

Read through Newman Chapter 2

Instead of .py file use Jupyter Notebook to create .ipynb file instead following the demonstration from the lecture. This will allow you to write your program in notebook format instead which is useful for learning and research. Once you're an expert you can just copy and paste the code out into a .py file and run it through the terminal. When you're working on your code keep all your files related to PHYS4102 in the same folder. So the program can access your files from the same folder.

1-10 pts) Install Python 3.9 using Anaconda.

https://www.anaconda.com/products/individual

Follow the instruction of the installer and accept the default settings (unless you know better).

Once anaconda is installed on your computer you should be able to open a "Jupyter Notebook" session. Keep that black terminal open but a new tab in your browser should pop-open. This is where you start.

Create a PHYS4102 folder in a directory on your computer. Preferably something like google drive or onedrive would be good as a way to automatically back up your codes. (If you lose your code, it is YOUR responsibility to REDO ALL THE WORK).

Write a program to print out the text "Hello YOUR NAME!". Use the Print Preview and Print your notebook out as a PDF and upload it to this assignment for credit.

2-15pts) Circular Orbits:

a) Starting with Newton's Gravitational Law, use the **Markdown** functionality to derive the altitude, h, of an object orbiting Earth obeys this equation: $h = \left(\frac{GMT^2}{4\pi^2}\right)^{\frac{1}{3}} - R$ where, $G = 6.67 \times 10^{-11}\ \frac{m^3}{kg\ s^2},\ M = 5.97 \times 10^{24} kg,\ R = 6371\ km$ are the mass and radius of Earth, T is the period of the orbit in seconds.

b) In a Python code box, write a program that prompts the user to enter a value of T in minutes. Verify that the code works with the period of the International Space Stations which has an altitude of 410 km and period of 92.68 minutes.

c) In a new code box, re-code the calculation so it makes the calculation using a user defined function instead. Then calculate the altitude of Geosynchronous Orbit.

d) In a new code box, write a while loop which repetitively calculates h from T = 90 minutes to T = 80 minutes, iterating in 1 minute intervals. Is there a minimal orbital period around Earth?

3-10pts) Coordinate Conversions:

a) Write a program which takes x and y as an input and prints out the r and phi (polar angle) of the coordinate. Make sure to utilize if statements to make sure your results are in the correct quadrants.

b) Write a program which takes r and phi and prints out the x and y coordinate. Make sure to utilize if statements to make sure your results are in the correct quadrants.

4-20pt) Lorentz Invariants: In special relativity, there are frame invariant (values which does not change in different frames). These are useful variables since it is independent of the frame of reference of the experiment. The momentum 4-vector and the invariants are,

$$p = \left[\frac{E}{c},\, p_x,\, p_y,\, p_z\right]$$

$$s = \left(\frac{E}{c}\right)^2 - p_x^2 - p_y^2 - p_z^2$$

a) Using a markdown window, show that $\sqrt{s} = mc$, where m is the mass of the particle, c is the speed of light.

b) Write a function which takes a list [E, px, py, pz] as an input and print out the mass*(speed of light) of the particle. Make sure the function returns an obviously wrong answer like -1 for invalid inputs. Extra note, typically in particle physics this energy, momentum and mass all have units of Electrovolt (eV). Mass a proton is 0.938 GeV for example.

c) Verify your function by using a for loop where you loop in the range E from 1-10 [GeV], px = 1-10 [GeV], py = pz =0, with step size of 1 [GeV]. Save the result in a 2D Array and an output file. Make sure to put each energy value in one row and each momentum value in each column. What type of particle would be likely for those on the diagonal of the array?

d) Start a new code box, read your output file from part c) to make sure it is written correctly and print out the result.

5-15pts) Binding Energy of Isotopes. In nuclear physics, there's an semi-empirical mass formula which allows you calculate the binding energy of an isotope given the atomic number Z and atomic mass number A:

$$E_{bind} = a_1 A - a_2 A^{\frac{2}{3}} - a_3 \frac{Z^2}{A^{\frac{1}{3}}} - a_4 \frac{(A-2Z)^2}{A} + \frac{a_5}{A^{\frac{1}{2}}}$$

where a1 = 15.8, a2= 18.3, a3 = 0.714, a4=23.2 and:

$$a_5 = 0 \qquad if\ A\ is\ odd$$

$$a_5 = 12.0 \qquad if\ A\ and\ Z\ are\ both\ even$$

$$a_5 = \ -12.0 \qquad if\ A\ is\ even\ and\ Z\ is\ odd$$

a) Write a program which calculates the binding energy. Check that it works by inputting A = 58 and Z = 28 where E_{bind}~ 500 MeV

b) Add to the program to calculate the binding energy per nucleon, E_{perA} = E_{bind}/A, then calculate both E_{perA} and E_{bin} for values of A = Z to A = 3Z.

c) Loop from Z = 1 to 50 and find the maximum binding energy per nucleon. This is the most stable isotope in the universe. Eventually, all nuclear fusion will result in producing this isotope. Look-up this element and isotope number on wikipedia.

6-10pts) Mystery Data: Download **SampleData1.txt (https://csustan.instructure.com/courses/28240/files/4586089?wrap=1)** **(https://csustan.instructure.com/courses/28240/files/4586089/download?download_frd=1)** . Load it into your python program.

a) Using a while or for loop, print out each row of the data array.

b) Looking at the data make a guess what these two columns represent. Think about what range of values they have and how they are related to each each. Verify your guess but showing a calculation that validates the guess.