

# Life of a Particle : Assignment 2

Sam Meehan

Due Date : 13 January 2017

## How to Submit

This assignment should be submitted by replying to the email sent out by the tutors - **Use the online form!** - with the link to your GitHub repository. Contained within this repository should be a markdown README.md file which provides a guide to the contents of the repository, along with instructions of how to run the code, where necessary. For the full assignment, also include a well-written summary in the form of a latex document with one section per question.

**You are to work with the partner assigned to you as a team.**

## Coding Like a Hipster - (Just Directions)

You don't need to do any work here, these are just directions.

For many common tasks, there are built in functions already existing in python, or which can be imported from the `math` or `numpy` modules. In particular, the ones you may have used before are `min(LIST)`, `max(LIST)`, `sum(LIST)`, `len(LIST)`. However, these are not necessary for “good” programmers. Nearly all programs can be written using the following small set of operations

- `+` (add)
- `*` (multiply)
- `/` (divide)
- `=` (assignment)
- `==` (equals comparison)
- `<` (less than comparison)
- `>` (greater than comparison)

In addition to these operators, it is taken for granted in programming that you can use the following features as well

- variables and lists - for storing the initial dataset
- for loops
- if statements
- print statements - for viewing your code

In this quiz, these are the only things that may appear. If you determine that you absolutely need some other function or operator, then include a comment clearly describing why this is the case.

Finally, you are not allowed to “hard code” in your program, meaning that there cannot be code that you must manually change each time you run it. An example of this is the length of a list. If I have a list `[1,4,2,5,3,6]`, and you want to use the number of items in the list in your code, then the number “6” may not appear in your code.

## Standard Deviation Like a Hipster

For this question, follow the directions to code like a “hipster”. Note that you can use the ROOT package to make histograms or graphs if you like.

Given a set of numbers  $Q$  (example  $Q = [2, 3, 4, 2]$ ), write a program which calculates the standard deviation of this set. For the purposes here, the definition of the standard deviation of a set of numbers is

$$\sigma(Q) = \sqrt{\frac{\sum_{i=0}^{N-1} (x_i - x_{avg})^2}{N}}$$
$$x_{avg} = \frac{\sum_{i=0}^{N-1} x_i}{N}$$
$$N = N(Q)$$

## NOTE!!!!

For the following three “hipster” problems, choose one out of three to do. That is, from the problems (Sorting, Dataset Correlation, Numerical Integration) choose to do only one for credit. If you do one or both of the other two problem, they will count as *BONUS*.

### [1] Sorting Like a Hipster

For this question, follow the directions to code like a “hipster”.

Given this set of data (you can copy and paste it into an array if you like)

[71, 51, 32, 62, 84, 109, 43, 92, 72, 41, 102, 80, 72, 69, 46, 94, 52, 95, 90, 72, 63, 70, 34, 80, 78, 34, 31, 37, 26, 41, 42, 107, 33, 108, 108, 75, 66, 23, 90, 53, 24, 70, 26, 41, 93, 24, 71, 39, 48, 66, 97, 107, 77, 71, 67, 39, 38, 107, 96, 92, 84, 46, 60, 95, 87, 90, 92, 63, 78, 78, 84, 107, 70, 108, 32, 36, 93, 108, 49, 72, 56, 43, 30, 56, 51, 97, 45, 92, 40, 43, 49, 83, 98, 28, 99, 97, 102, 89, 58, 87]

write a program in python which sorts the data into an ascending array. By this, I mean that the  $0^{th}$  element in the list is the smallest, and the last element in the list is the largest. If there is more than one element which has the same value, then the order of these two elements does not matter.

### [2] Dataset Correlation Like a Hipster

For this question, follow the directions to code like a “hipster”.

You are given a set of data in the format of  $(x, y)$  pairs as can be found on the course website - Course Website - for assignment 2. This data is formatted as

$$X = (x_0, x_1, x_2, \dots, x_N) \text{ and } Y = (y_0, y_1, y_2, \dots, y_N)$$

Given this dataset, calculate the “linear correlation coefficient” for this set of data as

$$\rho = \frac{V_{xy}}{\sigma_x \sigma_y}, \quad V_{xy} = E[xy] - \mu_x \mu_y$$

where  $E[xy]$  is the expectation value of the dataset  $XY = (x_0 y_0, x_1 y_1, x_2 y_2, \dots, x_N y_N)$  and  $\mu$  and  $\sigma$  are the standard deviation of the respective dataset. I would recommend making a 2D histogram of this data in addition to doing some Googling about how to calculate the linear correlation coefficient as well.

### [3] Numerical Integration Like a Hipster

For this question, follow the directions to code like a “hipster”. Note that you can use the ROOT package to make histograms or graphs if you like.

One of the most powerful tools in math is the integral. In particular, this is powerful because it can be used to compute the area under a curve, a concept which can be generalized to refer to volumes or more complicated quantities, like the amount of work performed on a particle by an electric field as the particle moves through a trajectory. One typically learns to compute (definite) integrals as

$$I_{y(x)}(a, b) = \int_a^b x^n dx = \frac{x^{n+1}}{n+1} \Big|_{x=a}^{x=b} = \frac{1}{n+1} (b^{n+1} - a^{n+1})$$

and this can be used to compute more complicated integrals like

$$I_{y(x)}(a, b) = \int_a^b y(x) dx$$

However, there is another method to compute the definite integral  $I$  computationally called a *Riemann Sum*. Graphically, it proceeds as shown in Figure 1. The integral is **approximated** by a sum of the area of the rectangles of some width ( $\Delta x$ ) with a height equal to  $y(x_{eval})$  where  $y(x_{eval})$  can be evaluated at the (i) left, (ii) right, or (iii) middle of the rectangle. In this case, the value of the integral is

$$I_{y(x)}(a, b) \sim \sum_{i=0}^{N-1} y(x_i) \times (x_{i+1} - x_i) = \sum_{i=0}^{N-1} y(x_i) \times \Delta x$$

Now, typically (but not necessarily) the value of  $\Delta x$  is fixed and taken to be  $\Delta x = \frac{b-a}{N}$ . Now, if you remember integral calculus, then you will remember that

$$\lim_{N \rightarrow \infty} \sum_{i=0}^{N-1} y(x_i) \times \Delta x = \int_a^b y(x) dx$$

Therefore, we can calculate the integral by writing a piece of code that performs this definite sum! So, the task is to write a piece of python code which performs the integral of  $y(x) = 3x^3 - 4x^2 + 3.2198x$  from  $a = 1$  to  $b = 9$ . Perform this calculation analytically and then perform it numerically using your code for  $N = [10, 50, 100, 200]$ . How close does this approximation approach to the analytic result.

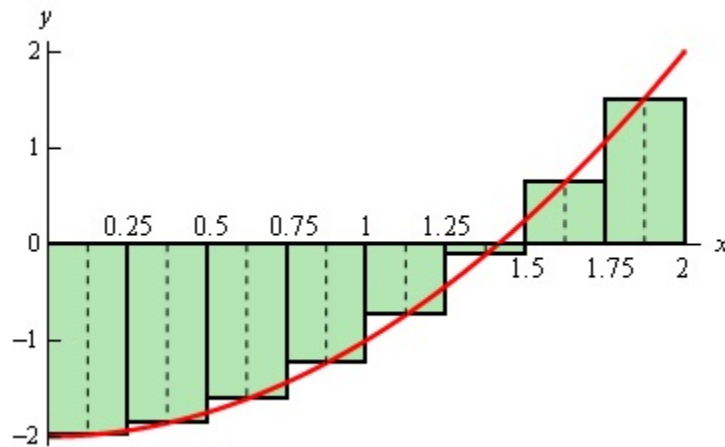


Figure 1: An illustration of the Riemann sum used to approximate the integral of the function  $y(x)$  (the red curve) from  $x = 0$  to  $x = 2$ . In this case, the step size  $\Delta x$  is taken to be 0.25 and the function  $y(x)$  is evaluated at the center point of each interval.

## Fitting a Line

For this question, follow the directions to code like a “hipster”. Note that you can use the ROOT package to make histograms or graphs if you like.

One procedure that is used regularly in computational modelling is that of “fitting”. This itself is a very broad topic and here we will implement one of the more basic procedures called the “minimization of least squares”. To make this concrete, let’s pretend that you have measurements of a particles position at equal time intervals and obtain the following measurements :

x [m]	y [m]
1.0	0.5
2.2	1.1
3.0	1.3
3.9	2.1
4.8	2.6

Now, you have a good reason to believe that this particle is moving on a straight line path, but your eyesight is not good and if you plot these points on a scatter plot, they don’t seem to be perfectly along a straight line. Nonetheless, you would like to determine the functional form of the path followed by the particle, which you know to be  $y = f(x; m, b) = m \times x + b$ . To do this, you can use the “minimization of least squares” as follows :

- Choose values for the parameters  $(m, b)$
- Calculate the value of the fit error which equals  $S = \sum_{points} (f(x_i) - y_i)^2$  as represented graphically in Figure 2.
- Make a list which matches a single choice of  $(m, b)$  to a value of  $S$  and fill it with many  $(m, b, S)$  choices
- Find the entry in this list of  $(m, b, S)$  which has the smallest value of  $S$ . The corresponding  $(m, b)$  values are what we call the “best fit” values of the parameters of our models.

Use this procedure to determine the best fit values of the line which describes the data you observed and create a plot of the data points and your best fit line.

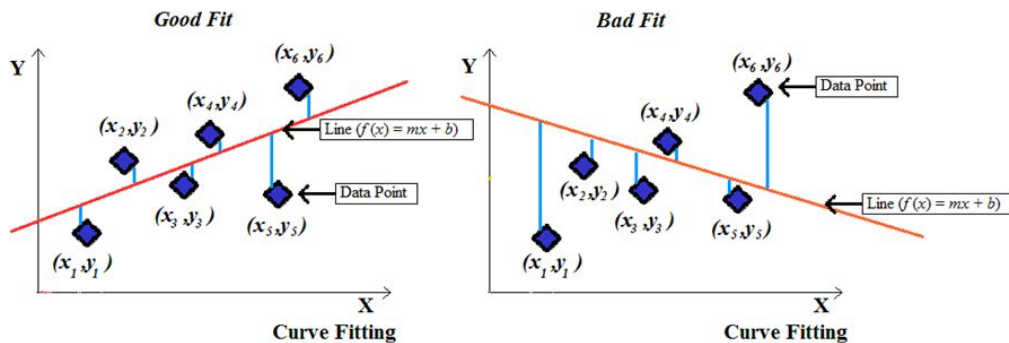


Figure 2: An illustration of the procedure to minimize the fit error and thereby determine the model parameters  $m$  and  $b$  that best describe our data.

## The Logistic Map

This is the completion of what we started in class. You are to work through the exercise of creating code that reproduces the results in this paper - the logistic map paper. In short, you should be recreating the histograms in this paper that demonstrate the behavior of the sequence of  $X$  values produced directly from the logistic map, and the sequence of  $Y$  values that are produced from the transformed logistic map. Use both one-dimensional and two-dimensional histograms to inspect the bias of both  $X$  and  $Y$ .

## Gaussian Random Numbers

For this question, you may import the python random library. However, you may only use the `random.random()` function call which generates a uniformly distributed random number between  $[0,1.0)$ .

Starting from a set of uniformly distributed random numbers from  $[0,1.0)$  (Don't be a hipster, use python for this!) generate a set of 10000 random numbers according to a Gaussian distribution of mean  $\mu = 2.0$  and width  $\sigma = 1.4$ . As a reminder, the Gaussian distribution is also called the "normal distribution" and has the following functional form :

$$Gauss(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} Exponential[-\frac{1}{2}(\frac{x - \mu}{\sigma})^2]$$

where  $Exponential[a] = e^a$ . Use the accept-reject method and limit the range of your generation to  $x \in [-10, 10]$ .