PH2150 Scientific Computing Skills

Andrew Casey

October 8, 2012

1 Exercise Wk2Ex1:

Create a list containing the first ten elements of the periodic table, create a second list containing the next ten elements of the periodic table. Concatenate (add) the lists together. Use the append function to add the next ten elements to your list. Use the len() function to confirm the new list has 30 elements. Print the 23rd element in your list. We will use this list in later exercises.

1.1 Exercise Wk2Ex1b:

Write a program to create and print a new list containing the elements from your list (from Ex1) that begin with the letter "s" Write a program to create and print a new list containing the elements in your list (from Ex1) that consist of only 4 characters.

2 Exercise Wk2Ex2:

2.1 Exercise Wk2Ex2a:

Write a *function* to compute the area of an arbitrary triangle. An arbitrary triangle can be described by the coordinates of its three vertices: $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. The area of the triangle is given by the formula:

$$A = \frac{1}{2}[x_2y_3 \ - \ x_3y_2 \ - \ x_1y_3 \ + \ x_3y_1 \ + \ x_1y_2 \ - \ x_2y_1]$$

Write a *function* **area(vertices)** that returns the area of a triangle whose vertices are specified by the argument vertices, which is a nested list of the vertex coordinates. For example, vertices can be [[0,0], [1,0], [0,2]] if the three corners of the triangle have coordinates (0, 0), (1, 0) and (0, 2). Test the area function on a triangle with known area. Name of program file: *area_triangle.py*.

2.2 Exercise Wk2Ex2b:

Compute the length of a path. Some object is moving along a path in the plane. At n points of time we have recorded the corresponding (x, y) positions of the object:$(x_0, y_0), (x_1, y_2), .., (x_{n-1}, y_{n-1})$. The total length L of the path from (x_0, y_0) to (x_{n-1}, y_{n-1}) is the sum of all the individual line segments $((x_{i-1}, y_{i-1})to(x_i, y_i)$, $i = 1, ..., n-1$:

$$L = \sum_{i=1}^{n-1} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

Make a function *pathlength(x, y)* for computing L according to the formula. The arguments x and y hold all the $x_0, ..., x_{n-1}$ and $y_0, ..., y_{n-1}$ coordinates, respectively. Test the function on a triangular path with the four points (1, 1), (2, 1), (1, 2), and (1, 1). Name the program file: pathlength.py.

2.3 Exercise Wk2Ex2c:

Approximate pi. The value of *pi* equals the circumference of a circle with radius 1/2. Suppose we approximate the circumference by a polygon through $N + 1$ points on the circle. The length of this polygon can be found using the *pathlength* function from Exercise Wk2Ex2b. Compute $N + 1$ points (x_i, y_i) along a circle with radius 1/2 according to the formulas:

$$x_i = \frac{1}{2}\cos(2\pi i/N), \quad y_i = \frac{1}{2}\sin(2\pi i/N), \quad i = 0, ..., N.$$

Call the pathlength function and write out the error in the approximation of π for $N = 2^k, k = 2, 3, ..., 10$. Name the program file: pi_approx.py.

3 Exercise Wk2Ex3:

The bell-shaped Gaussian function is widely used in science and technology. The parameters m and s are real numbers , where s must be greater than zero.

$$f(x) = \frac{1}{\sqrt{2\pi s}} \exp[-\frac{1}{2}(\frac{x-m}{s})^2]$$

Write a *Python* function *gauss(x, m=0, s=1)* for computing the Gaussian function. Call gauss and print out the results for x equal to -5,-4,-3,-2,-1,0,1,2,3,4,5, using the default values for m and s. Name the program Gaussian_function.py

We will be looking in more detail at plotting in later weeks but to show a Gaussian distribution do the following

```
#PH2150 example code
# This should work within the namespace of Pylab, if using IDLE you will need
    to import modules
x= randn(10000) # generates 10,000 random numbers, distributed around zero
hist (x, 100) #plots a histogram of x with 100 bins
```

gaussian_plot.py

Add labels to the x and y axis and give the plot a title, for example:

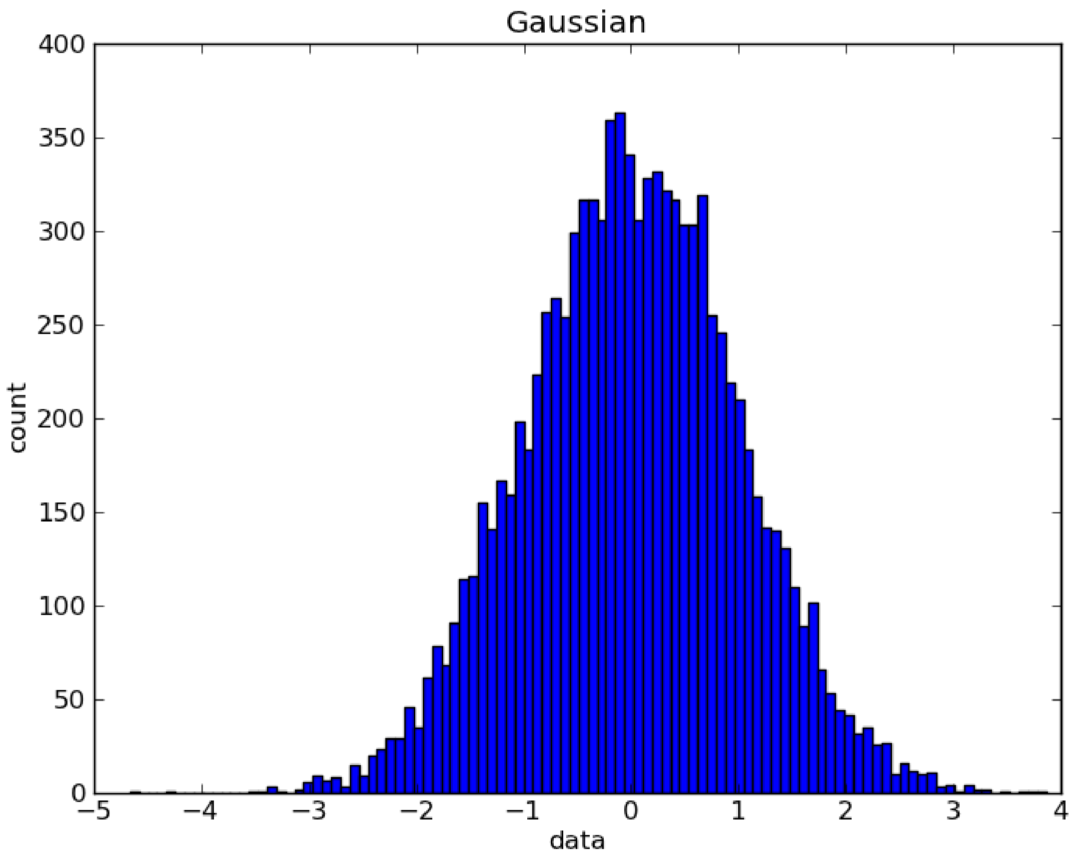

Fig. 1: This is output of the *hist* function from *PyLab*