Hands on Tutorial

Scientific Applications:

- Numpy
- Scipy
- Matplotlib
- Ipython environment

By Michael Atambo

Numpy:

What is numpy? in their words:

> *NumPy is the fundamental package for scientific computing with Python. . . . has:*
>
> - *a powerful N-dimensional array object*
> - *sophisticated (broadcasting) functions*
> - *tools for integrating C/C++ and Fortran code*
> - *useful linear algebra, Fourier transform, and random number capabilities*

Homepage: Link

Scipy:

"Scipy Stack" (Partial list)

1. *SciPy library*

Homepage: Link

Scipy:

"Scipy Stack" (Partial list)

1. *SciPy library*
2. *Numpy*

Homepage: Link

Scipy:

"Scipy Stack" (Partial list)

1. *SciPy library*
2. *Numpy*
3. *Matplotlib*

Homepage: Link

Scipy:

"Scipy Stack" (Partial list)

1. *SciPy library*
2. *Numpy*
3. *Matplotlib*
4. *Ipython*

Homepage: Link

Scipy:

"Scipy Stack" (Partial list)

1. *SciPy library*
2. *Numpy*
3. *Matplotlib*
4. *Ipython*
5. *Pandas*

Homepage: Link

Scipy:

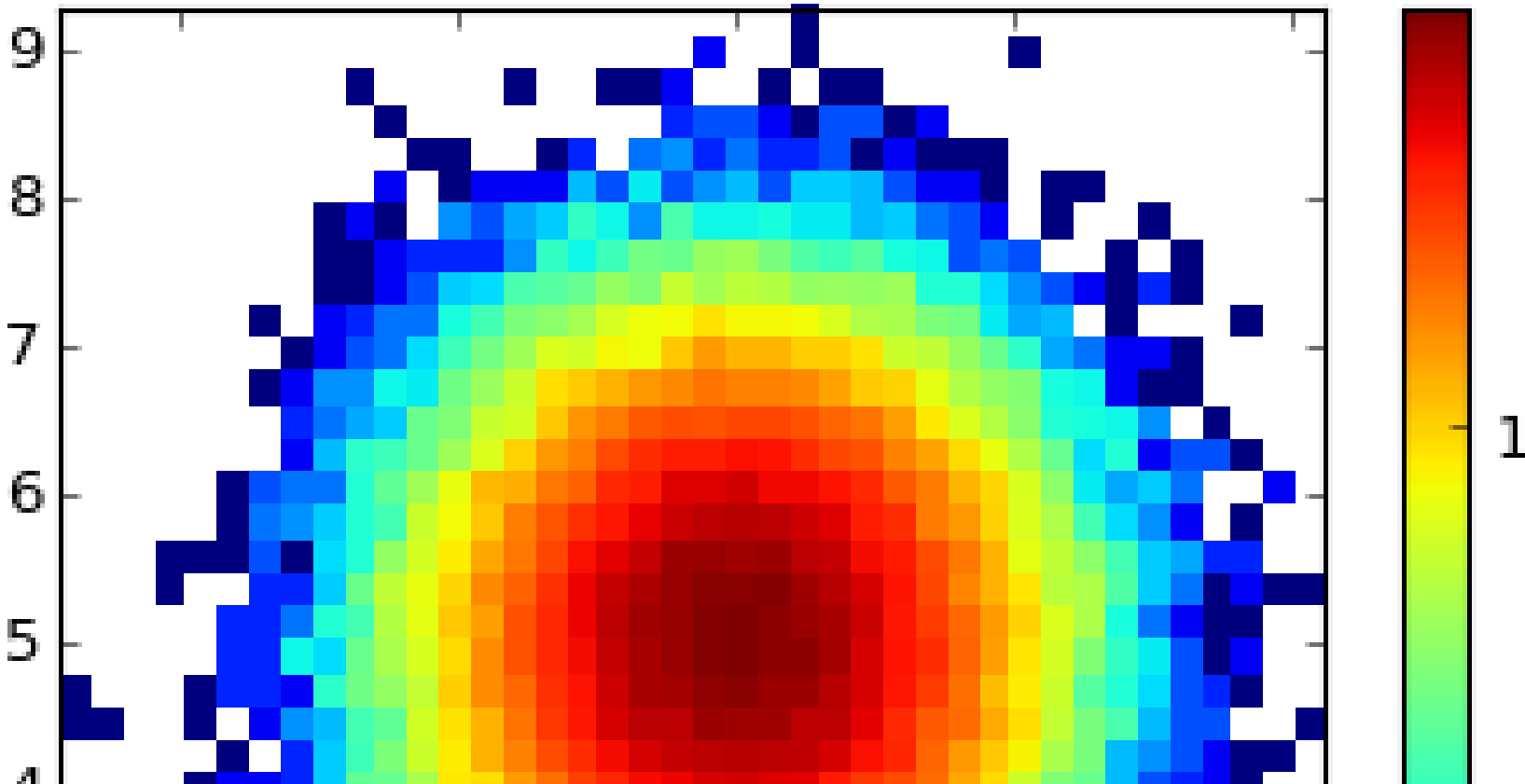
"Scipy Stack" (Partial list)

1. *SciPy library*
2. *Numpy*
3. *Matplotlib*
4. *Ipython*
5. *Pandas*
6. *Sympy*

Homepage: Link

Matplotlib

This is a 2D plotting library, opensource, with excellent documentation and support for multiple formats, can be used interactivly or as a batch utility.



Many others:

- `Ipython` : provides an excellent shell, browser and gui interface with advanced features
- `Pandas` : convenient data structures and analysis tools.
- `Sympy` : symbolic math in python

Excercises

You should (using the aforementioned packages) write and solve problems in python The problems are in the subsequent slides

Vectorization

We can taked advantage of the SIMD architecture of modern cpus, (see wikipedia: Link)

- Here you are to correct a non vectorized procedure. (code provided). You have some provided code in the class repo, please vectorize the scalar implementation, using one of the numpy universal functions. Numpy universal functions provide fast-elementwise array functions. We can effectively get rid of loops (scalar code) that usually run slowly, and apply a function to a numpy array as a whole (vectorization). files are: Github: Link
  - `01_vectorize.py`
  - `02_vectorize.py`

Hint: To do this, substitute whenever possible loops with numpy ufuncs.

Vectorization

- writting a vectorized function from scratch,

  Please create, from scratch the vectorized solutions to these problems:

1. Calculate the probability of the sum for the throw of a pair of dice.
2. Find pi using the ratio of the area of a unit circle to the area of a square is curcumscribed in.

$$\rho = areacircle/areasquare = \frac{\pi r^2}{(2r)^2} = \frac{pi}{4}$$

Approximate this by randomly picking points in a 1.0x1.0 square, and taking the ratio of those that fall in the unit circle to those that fall outside. This should give you rho. $\rho * 4 = \pi$

Plotting with matplotlib

using matplotlib plot these excercises:

- single plots: Using the code for the probability of the sum of a pair of dice worked on previously, plot the results, show the sums on the independent and the probability on the dependent axes.
- multiple plots : On the same graph, draw the approximation of pi vs the number of points from the previous question

Scipy/sympy problems:

- using sympy find the fourier expansion of the square wave
- plot the expansion for an increasing number of terms,
- create a matplotlib animation from this

Riddle (Optional):

For this problem assume the walker is on a line with equally spaced points, one unit distance from each other. The walker can move only one unit left or right

- What is the typical distance from the origin of a random walker after n steps?
- Plot the avarage distance vs the number of steps from the origin.

The solution to this will be provided.