Numpy

What is NumPy

NumPy is an extension to the Python programming language, adding support for large, multi-dimensional arrays and matrix's, along with a large library of high-level mathematical functions to operate on these arrays.

- NumPy is the fundamental library needed for **scientific computing** with Python.
- **This contains**:
  - ✓ N-dimensional array object
  - ✓ Array slicing methods
  - ✓ Array reshaping methods
- **Numerical routines in numpy:**
  - ✓ Linear algebra functions
  - ✓ Fourier transform
  - ✓ Random number capabilities

Why NumPy

- Arithmetic Operation can not applied directly on lists

Example:

```
>>> # List 1
>>> list1 = [9,8,7,6,5]
>>> # List 2
>>> list2 = [1,2,3,4,5]
>>> # Trying to Add the corresponding values in lists
>>> listSum=list1+list2
>>> # Instead we receive the Union of list1 and list2
>>> print(listSum)
[9, 8, 7, 6, 5, 1, 2, 3, 4, 5]
```

- Hence we need an efficient arrays with arithmetic and better multidimensional tools
- NumPy package provide arrays which are similar to lists, but much more capable, except fixed size.

NumPy - ndarray

- NumPy's main object is ndarray (homogeneous multidimensional array).
  - ✓ It is a table of elements (usually numbers), all of the same type, indexed by a tuple of integers.
  - ✓ Dimensions → usually called axes.
  - ✓ Rank → number of axes.

- Examples of multidimensional arrays include vectors, matrices, and spreadsheets etc

[9, 1, -1] → An array of rank 1 i.e. A matrix with 1 row and columns

[[10, 0.21, -30], → An array of rank 2 (A matrix with 2 rows and 3 columns)
[1.9, 7.4, 1.9]]

Numpy - Array Creation

There are a couple of mechanisms for creating arrays in NumPy:

- Using a Python list or tuple
- Using functions that are dedicated to generating numpy arrays, such as arange(), ones(), zeros() linspace() ,random(),eye() etc.
- Reading data from files

Numpy - Creating Array using list

https://www.jetbrains.com/help/pycharm/installing-uninstalling-and-upgrading-packages.html

Creating 1-D Array

```
>>> import numpy
>>> # Creating a List
>>> list = [10,20,30,40]
>>> # Creating an array using Numpy
>>> arr = numpy.array(list)
>>> print(arr)
[10, 20, 30, 40]
```

Creating 2-D Array

```
>>> import numpy
>>> # Creating a List
>>> list = [[1,2,3],[5,10,5],[10,20,30]]
>>> # Creating an array using Numpy
>>> arr = numpy.array(list)
>>> print(arr)
[[ 1  2  3]
 [ 5 10  5]
 [10 20 30]]
```

Numpy - Creating Array using built-in function

- zeros(shape) -- creates an array filled with 0 values with the specified shape.
- ones(shape) -- creates an array filled with 1 values.
- arange() -- creates arrays with regularly incrementing values.
- linspace() -- creates arrays with a specified number of elements, and spaced equally between the specified beginning and end values.
- random.random(shape) – creates arrays with random floats over the interval (0,1).
- eye(): Return a 2-D array with ones on the diagonal and zeros elsewhere.

Example:

```
import numpy as np

arr=np.zeros((3,2),int)

arr=np.ones((3,2),int)

arr=np.arange(10)

arr=np.random.random(10)

arr=np.eye(3,3,dtype=int)
```

Numpy – ndarray Attributes

ndarray.ndim: Gives dimension (rank of axes) of the array

ndarray.shape: **S**hape will be tuple (n,m).

ndarray.size: the total number of elements of the array.

ndarray.dtype: an object describing the type of the elements in the array.

ndarray.itemsize: the size in bytes of each element of the array.

ndarray.data: the buffer containing the actual elements of the array.

Numpy Useful Methods

sort()-value sorting

argsort()-index sorting

transpose()-matrix transpose

invert()-matrix inverse

dot()-matrix multiplication

std()

min()

max()

mean()

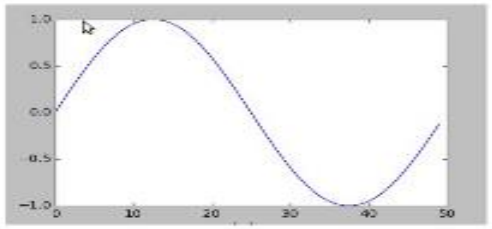
sum()

Matplotlib

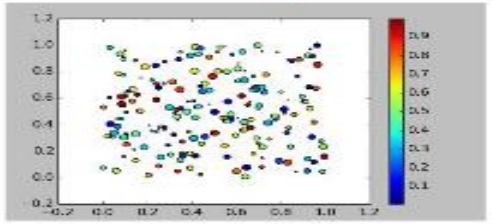
Matplotlib

- ✓ **Matplotlib** is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms
- ✓ **Matplotlib** for day-to-day data exploration.
- ✓ You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code
- ✓ **Matplotlib** has a large community, tone of plot types, and is well integrated into ipython shell.

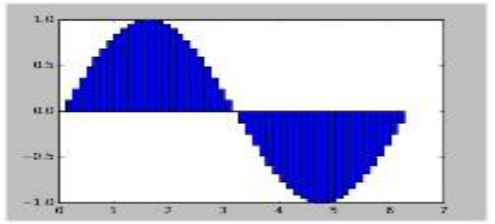
Line plots

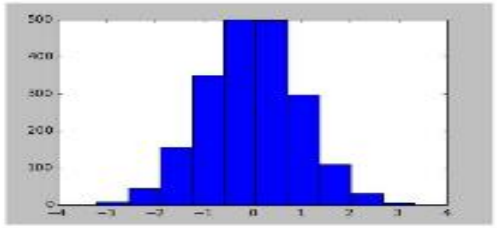


Scatter plot

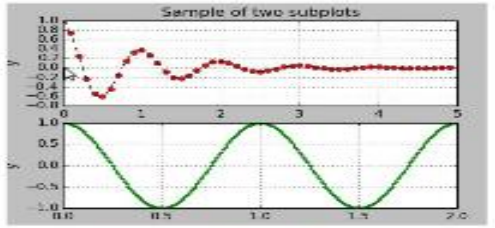


Bar Plot





Histogram



Multiple plots

THANK YOU!!