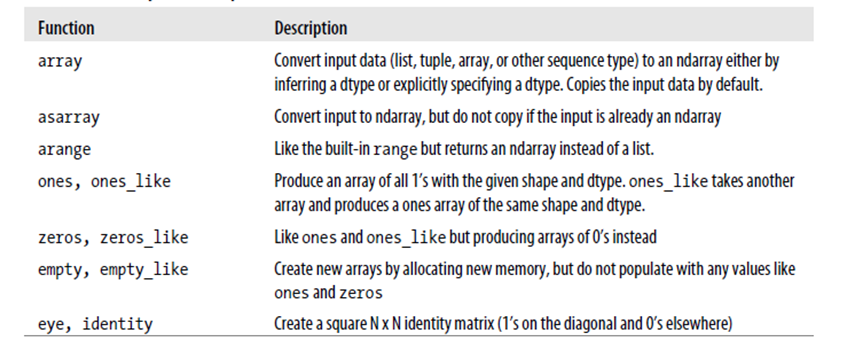
**Week 8 – numerical and scientific packages**

**Common packages for python**

* NumPy/SciPy – numerical and scientific function libraries.
* Numba – Python compiler that support JIT compilation.
* ALGLIB – numerical analysis library.
* Pandas – high-performance data structures and data analysis tools.
* PyGSL – Python interface for GNU Scientific Library.
* ScientificPython – collection of scientific computing modules.
* Matplotlib – plotting library.
* IPython – interactive computing.
* SymPy – symbolic computation library.

NumPy

The key to NumPy is the ndarray object, an *n*-dimensional array of homogeneous data types, with many operations being performed in compiled code for performance.

* + NumPy arrays have a fixed size. Modifying the size means creating a new array.
  + NumPy arrays must be of the same data type, but this can include Python objects.
  + More efficient mathematical operations than built-in sequence
  + NumPy supports a wider variety of data types than built-in python does
    - intc (same as a C integer) and intp (used for indexing)
    - int8, int16, int32, int64
    - uint8, uint16, uint32, uint64
    - float16, float32, float64
    - complex64, complex128
    -  bool\_, int\_, float\_, complex\_ are shorthand for defaults.

**SciPy**

SciPy’s functionality is implemented in a number of specific sub-modules. These include:

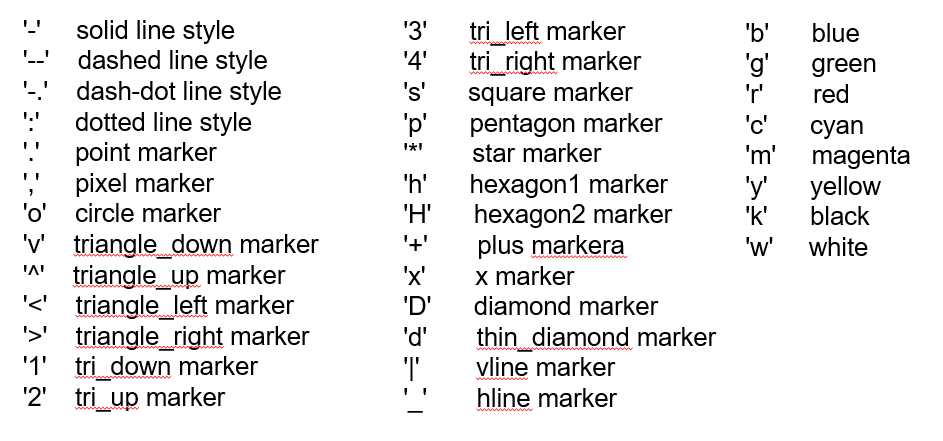
* Linear Algebra (scipy.linalg)
* Compressed Sparse Graph Routines (scipy.sparse.csgraph)
* Spatial data structures and algorithms (scipy.spatial)
* Statistics (scipy.stats)
* Multidimensional image processing (scipy.ndimage)
* Data IO (scipy.io)
* Weave (scipy.weave)

**MATPLOTLIB**

Matplotlib is an incredibly powerful (and beautiful!) plotting library. It’s easy to use and provides a huge number of examples for tackling unique problems. In conjuction with NumPy and SciPy, is comparable to MatLab.

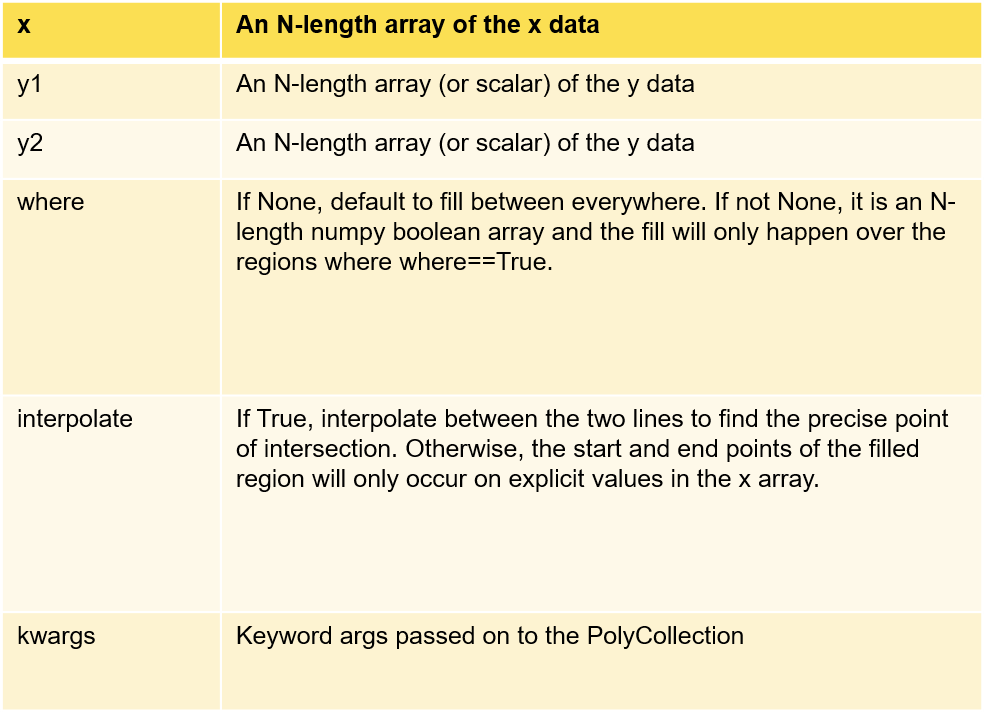
**Pyplot**

At the center of most matplotlib scripts is pyplot. The pyplot module is stateful and tracks changes to a ‘figure’. All pyplot functions revolve around creating or manipulating the state of a figure.



Commands

* Text() adds texts in an arbitrary location
* Xlabel() adds text to the x-axis
* Ylabel() adds text to the y-axis
* Title() adds plot title
* Clear() removes all plots from the axes
* Show() draws the plot
* Axis() given x&y values sets axis limits, without values returns the current axis limits
* Linspace() creates evenly spaced numbers over a specified interval
* Linestyle() specifies line type of plot (- -- : -.)

Shading plots