1 Arrays

Arrays are the base data type in NumPy, are are arrays in some ways similar to lists since they both contain collections of elements. The focus of this section is on homogeneous arrays containing numeric data – that is, an array where all elements have the same numeric type (heterogeneous arrays are covered in Chapters 16 and 17).

Additionally, arrays, unlike lists, are always rectangular so that all rows have the same number of elements.

Arrays are initialized from lists (or tuples) using array. Two-dimensional arrays are initialized using lists of lists (or tuples of tuples, or lists of tuples, etc.), and higher dimensional arrays can be initialized by further nesting lists or tuples.

1.1 Matrix

Matrices are essentially a subset of arrays, and behave in a virtually identical manner. The two important differences are:

- Matrices always have 2 dimensions
- Matrices follow the rules of linear algebra for $*$

1.2 Accessing Elements of an Array

Four methods are available for accessing elements contained within an array: scalar selection, slicing, numerical indexing and logical (or Boolean) indexing. Scalar selection and slicing are the simplest and so are presented first. Numerical indexing and logical indexing both depends on specialized functions and so these methods are discussed in Chapter 12

4.4 2-dimensional Arrays Matrices and 2-dimensional arrays are rows of columns, and so x = 2 64 1 2 3 4 5 6 7 8 9 3 75 , is input by enter the matrix one row at a time, each in a list, and then encapsulate the row lists in another list. ¿¿¿ x = array([[1.0,2.0,3.0],[4.0,5.0,6.0],[7.0,8.0,9.0]]) ¿¿¿ x array([[1., 2., 3.], [4., 5., 6.], [7., 8., 9.]])

1.3 Multidimensional Arrays

Higher dimensional arrays are useful when tracking matrix valued processes through time, such as a timevarying covariance matrices. Multidimensional (N -dimensional) arrays are available forN up to about 30, depending on the size of each matrix dimension. Manually initializing higher dimension arrays is tedious and error prone, and so it is better to use functions such as zeros((2, 2, 2)) or empty((2, 2, 2)).

1.4 Concatenation

Concatenation is the process by which one vector or matrix is appended to another. Arrays and matrices can be concatenation horizontally or vertically 4.7 Accessing Elements of an Array Four methods are available for accessing elements contained within an array: scalar selection, slicing, numerical indexing and logical (or Boolean) indexing. Scalar selection and slicing are the simplest and so are presented first. Numerical indexing and logical indexing both depends on specialized functions and so these methods are discussed in Chapter 12.

1.5 The `import` function

Python, by default, only has access to a small number of built-in types and functions. The vast majority of functions are located in modules, and before a function can be accessed, the module which contains the function must be imported.

For example, when using ipython –pylab (or any variants), a large number of modules are automatically imported, including NumPy and matplotlib. This is style of importing useful for learning and interactive use, but care is needed to make sure that the correct module is imported when designing more complex programs.

import can be used in a variety of ways. The simplest is to use from module import * which imports all functions in module. This method of using import can dangerous since if you use it more than once, it is possible for functions to be hidden by later imports. A better method is to just import the required functions. This still places functions at the top level of the namespace, but can be used to avoid conflicts.

```
from pylab import log2 # Will import log2 only
from scipy import log10 # Will not import the log2 from SciPy
```

The only difference between these two is that import scipy is implicitly calling import scipy as scipy. When this formof import is used, functions are used with the “as” name. For example, the load provided byNumPy is accessed using sp.log2, while the pylab load is pl.log2 – and both can be used where appropriate. While this method is the most general, it does require slightly more typing.