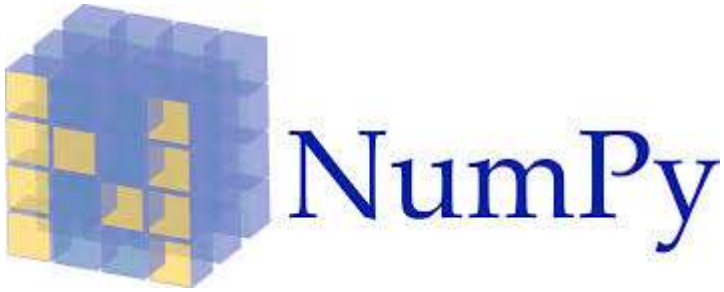
LISBON
SCHOOL OF
ECONOMICS &
MANAGEMENT
UNIVERSIDADE DE LISBOA



CARLOS J. COSTA

Numpy

- Numpy is a fundamental Python library for scientific computing.
- Provides array related functionality
- Has higher level of performance

Numpy

```
import numpy as np
c = np.array([1,2,3,4])
print(type(c))
```

Numpy

- Create array

```
a = np.array([1,2,3,4])
```

- Shape, rank and size:

```
shape = a.shape
rank = np.ndim(a)
size = a.size
```

Numpy

- The bidimensional array (matrix):
- `b = np.array([[1,2,3],[4,5,6]])`
- What information can be obtained about this array:
- `shape = b.shape`

  `rank = np.ndim(b)`

  `size = a.size`

Numpy

- Change value to array:

```
a[2]=50
print(a)
```

Numpy

- Create an array with zeros only

```
a = np.zeros((2,2))
print(a)
```

Numpy

- Create array with only “one”

```
b = np.ones((1,2))
print(b)
```

Numpy

- create 3x3 identity array

```
d = np.eye(3)
print(d)
```

Numpy

- Create array filled with random numbers

```
e = np.random.random((4,4))
print(e)
```

Numpy

```
a = np.array([[1,2,3,4], [5,6,7,8],
[9,10,11,12]])
print(a)
```

Numpy

- slicing arrays
- number of rows, number of columns
- start line to 3, columns from 1 (second) to 3

```
b = a[:3, 1:3]

print(b)
```

Numpy

```
b[0, 0] = 99
```

- what happens to array a?

Numpy

- Another example handles indexes

```
import numpy as np
#create a new array
a = np.array([[1,2,3], [4,5,6], [7,8,9],
[10, 11, 12]])
print(a)
```

Numpy

- create an array of indexes

```
b = np.array([0, 2, 0, 1])
```

Numpy

- Select an element from each column using the b indexes, that is, selects:
- element with index 0 on the first row
- element with index 2 on the second row
- element with index 0 on the third line
- element with index 1 on the fourth row

```
print(a[np.arange(4), b])
```

Numpy

- For example, you can change the elements of each column according to the array of indexes.
- E.g. add 10, but only to the values found

```
a[np.arange(4), b] += 10

print(a)
```

Bibliografia

- http://www.numpy.org/