Effective Programming Practices for Economists

Scientific Computing

Array indexing

Janoś Gabler and Hans-Martin von Gaudecker

What is indexing?

- Remember: we can select elements of a list using something like `` `list[1:3]` ``
- Numpy uses the same syntax `` `arr[1:3]` ``
- Numpy generalizes it in several ways:
  - Indexing into multi-dimensional arrays
  - Boolean indexing

Indexing 1d

```
>>> a = np.array([0, 1, 2, 3, 4])
>>> a[2]
2

>>> a[-1]
4

>>> a[1:3]
array([1, 2])

>>> a[2:]
array([2, 3, 4])

>>> a[[0, 3]]
array([0, 3])

a[[True, False, True, False, True]]
array([0, 2, 4])
```

- Indexing starts at 0
- Single elements are not returned as array
- Slices include lower bound and exclude upper bound
- Lower or upper bound can be omitted
- Last example: Boolean indexing

Indexing 2d

```
# select a row
>>> b = np.arange(12).reshape(4, 3)
>>> b[2]
array([6, 7, 8])

# select a column
>>> b[:, 2]
array([ 2,  5,  8, 11])

# select an element
>>> b[1, 2]
5

# select a slice
>>> b[:2, :2]
array([[0, 1],
       [3, 4]])
```

- Indexing in multiple dimensions is just the same as in one!
- Separate the indexing for the dimensions by commas
- Omit later dimensions if you do not want to restrict them