Effective Programming Practices for Economists

Scientific Computing

Calculations between arrays

Janoś Gabler and Hans-Martin von Gaudecker

Two types of multiplication

```
>>> a = np.arange(4).reshape(2, 2)
>>> b = np.linspace(0.1, 0.4, 4).reshape(2, 2)
>>> b
array([[0.1, 0.2],
       [0.3, 0.4]])
```

```
>>> a * b
array([[0. , 0.2],
       [0.6, 1.2]])
```

```
>>> a @ b
array([[0.3, 0.4],
       [1.1, 1.6]])
```

```
>>> a.dot(b)
array([[0.3, 0.4],
```

- `*` means elementwise multiplication
- `@` and `.dot` mean matrix multiplication
- Both generalize to high dimensions
- They have different requirements on array shapes

Other operations between arrays

```
>>> a = np.arange(3)
>>> b = np.ones(3)
>>> a + b
array([1., 2., 3.])
```

```
>>> a - b
array([-1.,  0.,  1.])
```

```
>>> a / b
array([0., 1., 2.])
```

```
>>> b / a
RuntimeWarning: divide by zero encountered
array([inf, 1. , 0.5])
```

```
>>> (b * 2) ** a
array([1., 2., 4.])
```

- Addition, subtraction and division work as expected
- Division by zero does not raise an error
- Exponentiation uses `**`