Effective Programming Practices for Economists

Scientific Computing

Measuring runtime

Janoś Gabler and Hans-Martin von Gaudecker

Example

```
def array_cobb_douglas(factors, weights, a):
    out = np.empty(len(factors))
    for i in range(len(factors)):
        out[i] = _cobb_douglas(factors[i], weights, a)
    return out

def _cobb_douglas(factors, weights, a):
    return a * np.prod(factors**weights)
```

- Assume we want to evaluate the function multiple times
- Will be the running example for all speedup screencasts
- Possible applications
  - Structural models with production
  - Skill formation models

Setting up representative inputs

```
# number of input factors
k = 5
# number of evaluations
n = 10_000

# set up random inputs
rng = np.random.default_rng(93726483)
factors = rng.uniform(0.1, 3, size=(n, k))
weights = np.array([0.2, 0.1, 0.3, 0.2, 0.2])
a = 1.2
```

- Sizes should be representative of your real application!
  - Not all algorithms scale linearly
  - You want to optimize what you really need!
- Use random numbers for inputs

Timing fast functions

```
%timeit array_cobb_douglas(factors, weights, a)
```

```
25.1 ms ± 488 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

- `%timeit` only works in notebooks!
- It does many things automatically
  - discard outliers
  - determine how often the code is evaluated
  - determine suitable units of time

Timing slow functions

```
from time import perf_counter

start = perf_counter()
array_cobb_douglas(factors, weights, a)
runtime = perf_counter() - start
runtime
```

- Use this if your function takes several seconds and you only want to evaluate it once
- Do not use `time.time` instead of `time.perf_counter` because it has very low resolution on windows (full seconds)
- Only interpret differences between `perf_counter` evaluations

Limitations

- Measured runtime depends on background tasks
  - Try to run few applications in the background
  - Do not run timings in parallel!
- Runtime does not tell you where the time is spent
  - Need profiling