Effective Programming Practices for Economists

Scientific Computing

Randomness

Janoś Gabler and Hans-Martin von Gaudecker

Randomness

- All kinds of simulation need random numbers
  - Monte-Carlo exercises
  - Simulating structural models
  - ...
- Computers can only create pseudo random numbers
- They behave like random numbers but are completely deterministic

Creating random numbers

```
>>> rng = np.random.default_rng(5471)
>>> rng.uniform(low=0, high=1, size=3)
array([0.28129558, 0.36638138, 0.51719372])

>>> rng.uniform(low=0, high=1, size=3)
array([0.5964197 , 0.53583563, 0.66671704])

>>> rng = np.random.default_rng(5471)
>>> rng.uniform(low=0, high=1, size=3)
array([0.28129558, 0.36638138, 0.51719372])
```

- Create a **R**andom **N**umber **G**enerator (RNG) using a **seed**
- Use the `rng` with the distribution of your choice
- Provide size and parameters of the distribution
- See docs for list of available distributions

What is the seed?

- Seed can be any integer between 0 and 2 ** 32
- Seeds enumerate all possible states of a random number generator
- Two neighboring seeds (e.g. 999 and 1000) produce independent random numbers

Rules for dealing with randomness

- Never use the old global seeds (via `np.random.seed`)
- Never use `np.random.default_rng()` without a seed
- Generate your seeds randomly (don't use 123, 42, ...)
- Make sure that your main results do not change when you change the seed