

Principles of Programming in Econometrics

Introduction, structure, and advanced programming techniques

Charles S. Bos

Vrije Universiteit Amsterdam
Tinbergen Institute

`c.s.bos@vu.nl`

August 2020 – Version Python

Separate lecture slides

Compilation: July 27, 2020

Overview

Principles of Programming in Econometrics

D0: Syntax, example 2⁸

D1: Structure, scope

D2: Numerics, packages

D3: Optimisation, speed

Day 2: Numerics and flow

9.30 Numbers and representation

- ▶ Steps, flow and structure
- ▶ Floating point numbers
- ▶ Practical Do's and Don'ts
- ▶ Packages
- ▶ Graphics

13.30 Practical

- ▶ Cleaning OLS program
- ▶ Loops
- ▶ Bootstrap OLS estimation
- ▶ Handling data: Inflation

Program flow

Programming is (should be) no magic:

- ▶ Read your program. There is only one route the program will take. You can follow it as well.
- ▶ Statements are executed in order, starting at `main()`
- ▶ A statement can call a function: The statements within the function are executed in order, until encountering a `return` statement or the end of the function
- ▶ A statement can be a *looping* or *conditional* statement, repeating or skipping some statements. See below.
- ▶ (The order can also be broken by `break` or `continue` statements. Don't use, ugly.)

And that is all, any program follows these lines.

(Sidenote: Objects/parallel programming etc)

Flow 2: Reading easily

As a general hint:

- ▶ Main .py file:
 - ▶ import packages
 - ▶ import your routines (see next page)
 - ▶ Contains only `main()`
 - ▶ Preferably only contains calls to routines (`Initialise`, `Estimate`, `Output`)
- ▶ Each routine: Maximum 30 lines / one page. If longer, split!

Flow 3: Using modules

A module is a file containing a set of functions

All content from module `incstack.py` in directory `lib` can be imported by

```
from lib.incstack import *
```

Result: Nice short `stackols3.py`

```
#####  
### main  
def main():  
    # Magic numbers  
    ...  
    # Initialisation  
    (vY, mX)= ReadStack(sData, sY, asX, True)  
    ...
```

Q: What would be the difference between `from lib.incstack import *` and `import lib.incstack?`

In Spyder:

- ▶ check current directory (`pwd`), make sure that you are in your working directory (use `cd` if need be)
- ▶ add general directory with modules to the `PYTHONPATH`, using Tools-PYTHONPATH manager

Flow 4: Cleaning out directory structure

Use structure for programming, and for storing results:

```
stack/stackols3.py      # Main routine
stack/lib/incstack.py   # Included functions
stack/data/stackloss.csv # Data
stack/output/           # Space for numerical output
stack/graphs/           # Space for graphs
```

**Ensure you program cleanly, make sure you can find
routines/results/graphs/etc...**