Effective Programming Practices for Economists

18. Programming environments and programming languages

Hans-Martin von Gaudecker

Department of Economics, Universität Bonn

universitätbonn

Development environments

A development environment is the whole collection of tools that people use to create software, not just a few tools that are specific to a particular programming language. Examples of these tools are version control tools [...], build tools [...], and bug tracking tools.

Doar (2005)

- My guess is that most academic economists never thought conciously more than a couple of hours about these issues
  - So you're at the top of the distribution already
- Yet the choices, made conciously or not, can make a project break or succeed
- Doar (2005) is slightly outdated on specific tools, but still highly recommended!

Research / development environments

Task	Naïve choice	Alternatives
Typesetting	Word	LaTeX, Scientific Workplace, LyX
Bibliography management	None or edit .bib-file directly	`JabRef`, `Bibdesk`
Source code editing	Word, Notepad	`SublimeText`, `TextMate`, `WinEdt`, `Emacs`, `Vim`, IDEs
Version control	None / manually	`Git`, `Subversion`, `Mercurial`
Debugging	None / `print`	GDB, PDB, IDEs
Issue tracking	None / pen+paper	`Redmine`, `Sourceforge`, `GoogleCode`, `github`, `bitbucket`
Testing	None	nose, Matlab `xUnit` / `built-in` (R2013a+)
Building the project	None / master do file or similar	Waf, SCons, make
Documenting the code	None	`Sphinx`, `m2html`, `Sweave`

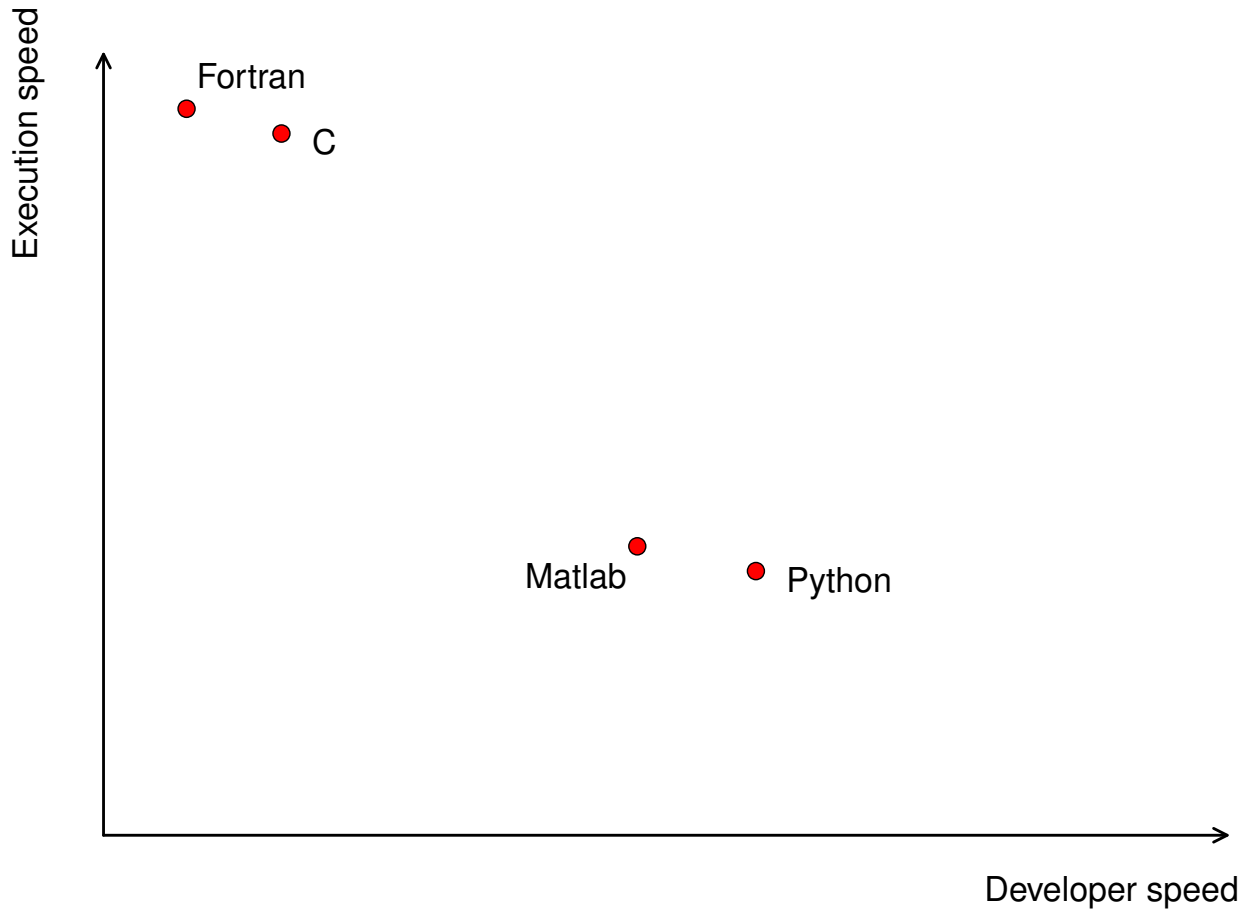
Choice of tools for mundane tasks

Task	Naïve choice	Alternatives
Create a simple figure	MS Excel	Stata, Matplotlib, Matlab
Plot a function	MS Excel	Matplotlib, Matlab, (Stata)
Quickly calculate sample statistics	MS Excel	Stata, Python, . . .
Quickly clean some data	MS Excel	Stata, Python, . . .
Check some theoretical result	Pen & paper (i.e. repeat)	Mathematica, Matlab, `SimPy`

Why are we programming at all?

- Software developer: Creates tools to be used by others
- End user: Applies tools developed by others
- As researchers, we are often in the middle
- Developing a new estimator means that no professional software developer had the chance of implementing the corresponding code before
- Much of structural econometrics / macro: Model specifics and solution strategy are closely intertwined
  - Makes it hard to rely on generic programs
  - Code sharing would help, should improve (Barnes, 2010)

Choice of programming language
Speed in two dimensions



Reminder of how the typical analysis proceeds

1. Data management
2. Actual analysis (estimation / simulation)
3. Visualisation and results formatting
4. Writing the paper

Scenario 1
Empirical project, red. form / “experiment”

- E.g. estimating the good old Mincer equation once more, using some novel covariate
- All estimators you’ll ever need implemented in Stata (or R)
- Use it from beginning to end
  - Potentially Python for formatting of tables, if `estout` does not give you **directly** what you want
- Consider Python + SQL or SAS for data management . . .
  - . . . if your data do not fit into memory;
  - . . . if you need to describe complicated relations between observational units;
  - . . . or if they have multiple dimensions

Scenario 2
Developing an estimator

- Deriving the theoretical properties of an estimator will hardly be enough for publication nowadays
- Monte Carlo simulations are the absolute minimum
  - You really should provide a function which can be used by applied researchers (Koenker and Zeileis, 2009)
  - Especially if you want the estimator to be used
- Develop your code in Matlab, Python, or R
  - Fast development cycles (b/c of debugging and testing)
- When you're done (paper published? working paper polished?), port it to Stata
  - Close to trivial if you avoided too many higher-level constructs of languages (e.g. objects, dictionaries)

Scenario 3
(Macro) simulation / calibration

- You'll need some real data and lots of assumptions on distributions leading to simulated data
- Then you have a (potentially) computationally intensive component in the main analysis
- Do steps 1 and 3 in Python or Matlab
- Tool choice for Step 2 depends on computation time
  - Fairly small problem: Use Python or Matlab
  - Larger problem: Use Fortran or C
  - In some cases, a mix of Fortran or C with Python (f2py, Cython, etc.) or Matlab (.mex) might be useful

Scenario 4
Structural micro estimation

- The core problem is more or less the same as in scenario 3
  - Generate predictions about behaviour from economic theory, find the “right” parameter values for the theory
- Main differences: Data management is more involved, you must allow for heterogeneity, and you need standard errors
- Suggestions from previous scenario apply
  - Probably more arguments for mixing languages in step 2, as opposed to standalone Fortran or C
- Separate the estimation of parameters and the calculation of standard errors
  - Often you can’t afford to wait for the estimation to complete before you debug your standard error code
  - Good tests might be a substitute, of course

References I


Barnes, Nick (2010). “Publish Your Computer Code: It is Good Enough”. In: *Nature* 467, p. 753.


Doar, Matthew B. (2005). *Practical Development Environments*. Sebastopol, CA, USA: O’Reilly Media.


Koenker, Roger and Achim Zeileis (2009). “On Reproducible Econometric Research”. In: *Journal of Applied Econometrics* 24.5, pp. 833–847.

Acknowledgements

- This course is designed after and borrows a lot from the `Software Carpentry` course designed by `Greg Wilson` for scientists and engineers.

- The Software Carpentry course material is made available under a `Creative Commons Attribution License`, as is this course's material.

License for the course material

[Links to the full `legal` text and the source text for `this` page.] You are free:

- **to Share** to copy, distribute and transmit the work
- **to Remix** to adapt the work

Under the following conditions:

- **Attribution** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

With the understanding that:

- **Waiver** Any of the above conditions can be waived if you get permission from the copyright holder.
- **Public Domain** Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
- **Other Rights** In no way are any of the following rights affected by the license:
  - Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
  - The author's moral rights;
  - Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

Notice For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to `this` web page.