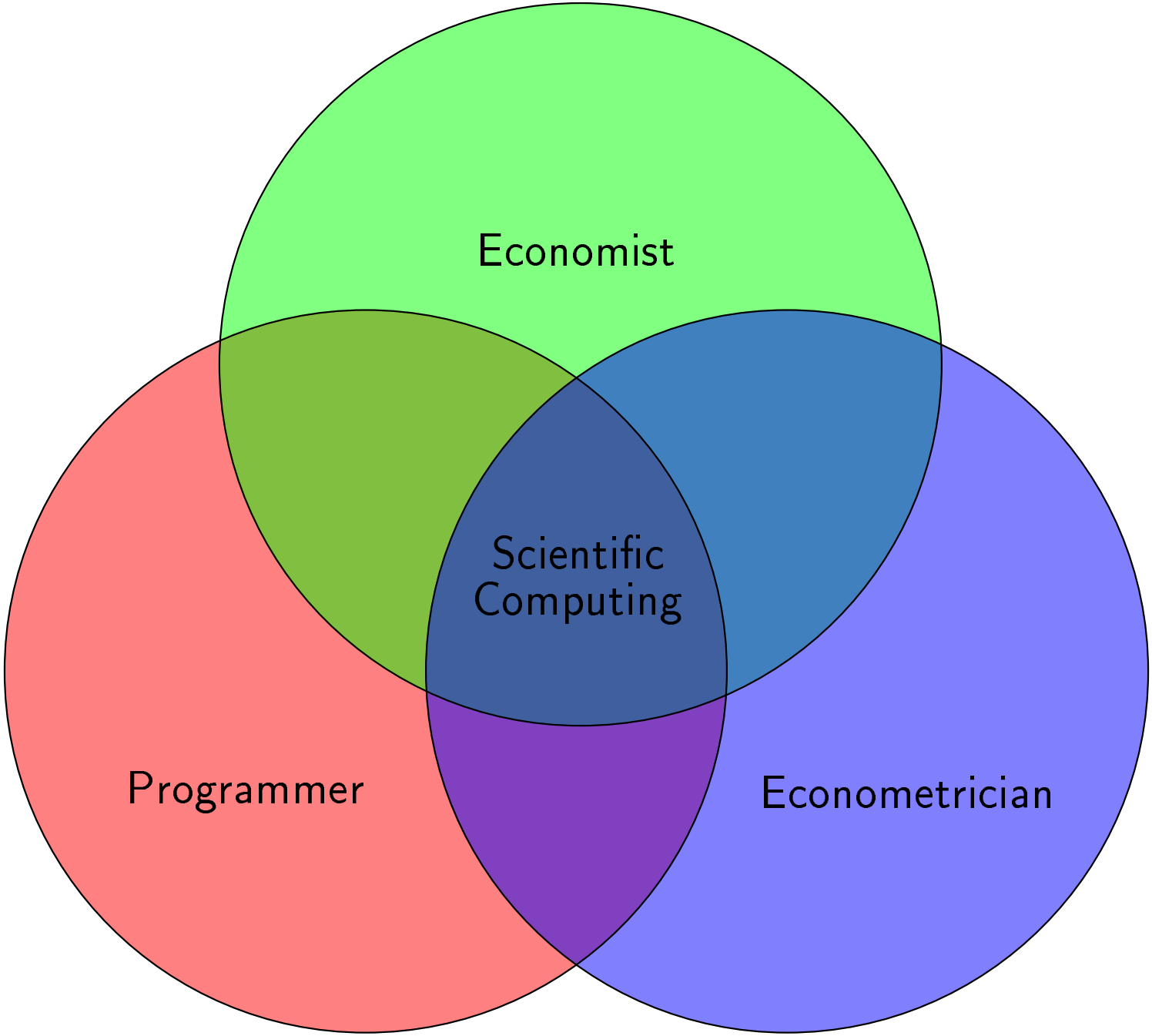
Practical Computing for Economists

Software Engineering

Philipp Eisenhauer


Economist
Scientific
Computing
Programmer
Econometrician

Best Practices

Reference

Greg Wilson, & al. (2014): Best Practices for Scientific Computing, *PLOS Biology*.

(1) Write programs for people, not computers.

- A program should not require its readers to hold more than a handful of facts in memory at once.
- Make names consistent, distinctive, and meaningful.
- Make code style and formatting consistent.

(2) Let the computer do the work.

- Make the computer repeat tasks.
- Save recent commands in a file for re-use.
- Use a build tool to automate workflows.

(3) Make incremental changes.

- Work in small steps with frequent feedback and course correction.
- Use a version control system.
- Put everything that has been created manually in version control.

(4) Don't repeat yourself (or others).

- Every piece of data must have a single authoritative representation in the system.
- Modularize code rather than copying and pasting.
- Re-use code instead of rewriting it.

(5) Plan for mistakes.

- Add assertions to programs to check their operation.
- Use an off-the-shelf unit testing library.
- Turn bugs into test cases.
- Use a symbolic debugger.

(6) Optimize software only after it works correctly.

- Use a profiler to identify bottlenecks.
- Write code in the highest-level language possible.

(7) Document design and purpose, not mechanics.

- Document interfaces and reasons, not implementations.
- Refactor code in preference to explaining how it works.
- Embed the documentation for a piece of software in that software.

(8) Collaborate.

- Use pre-merge code reviews.
- Use pair programming when bringing someone new up to speed and when tackling particularly tricky problems.
- Use an issue tracking tool.

Tools

Why **R**?

- Open Source
- Scientific and Statistical Computing
- Graphics
- Education

Integrated Development Environment

Code Documentation

Roxygen

Profiling

RProf

Unit Testing

RUnit

Build Management

Task Management System

Version Control System

Dissemination

Examples

(1) `grmToolbox`

A Python package for the estimation of marginal, average, and conditional effects of treatment in the generalized Roy model.

(2) `structToolbox`

A Python package to illustrate the reliable structural estimation of a simple dynamic labor supply model.

grmToolbox

Reference

Eisenhauer, Philipp and James Heckman, Edward Vytlacil (2014): The Generalized Roy Model and the Cost-Benefit Analysis of Social Programs, *Journal of Political Economy*, resubmitted.

Generalized Roy Model

Potential Outcomes

$$Y_1 = \mu_1(X) + U_1$$

$$Y_0 = \mu_0(X) + U_0$$

Choice

$$D = \mathbb{1}\{S > 0\}$$

$$S = Y_1 - Y_0 - C$$

$$C = \mu_C(Z) + U_C$$

Observed Outcome

$$Y = DY_1 + (1 - D)Y_0$$

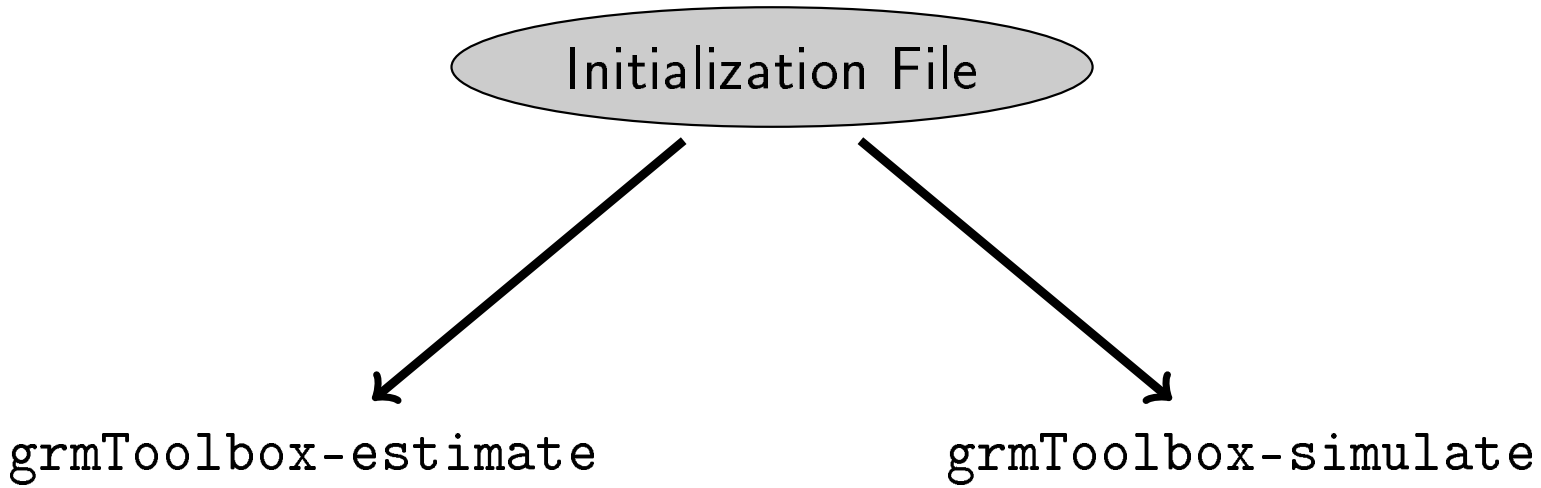
Maximum Likelihood Estimation

$$
\begin{aligned}
\mathcal{L}(\psi) = & \prod_{0} \frac{1}{\sigma_{U_0}} \phi\left(\frac{Y_0 - x'\beta_0}{\sigma_{U_0}}\right) \\
& \times \left(1 - \Phi\left(\frac{z'\gamma - \rho_{U_0,V}(Y_0 - x'\beta_0)/\sigma_{U_0}}{\sqrt{1-\rho_{U_0,V}}}\right)\right) \\
& \times \prod_{1} \frac{1}{\sigma_{U_1}} \phi\left(\frac{Y_1 - x'\beta_1}{\sigma_{U_1}}\right) \\
& \times \Phi\left(\frac{z'\gamma - \rho_{U_1,V}(Y_1 - x'\beta_1)/\sigma_{U_1}}{\sqrt{1-\rho_{U_1,V}}}\right)
\end{aligned}
$$

Transparency, Recomputability, and Extendibility

http://www.policy-lab.org/grmToolbox

- Documentation
  - Source Code
  - Test Suite
- Download


Initialization File
grmToolbox-estimate
grmToolbox-simulate

Additional Commands

- `grmToolbox-perturb`
- `grmToolbox-clean`
- `grmToolbox-terminate`

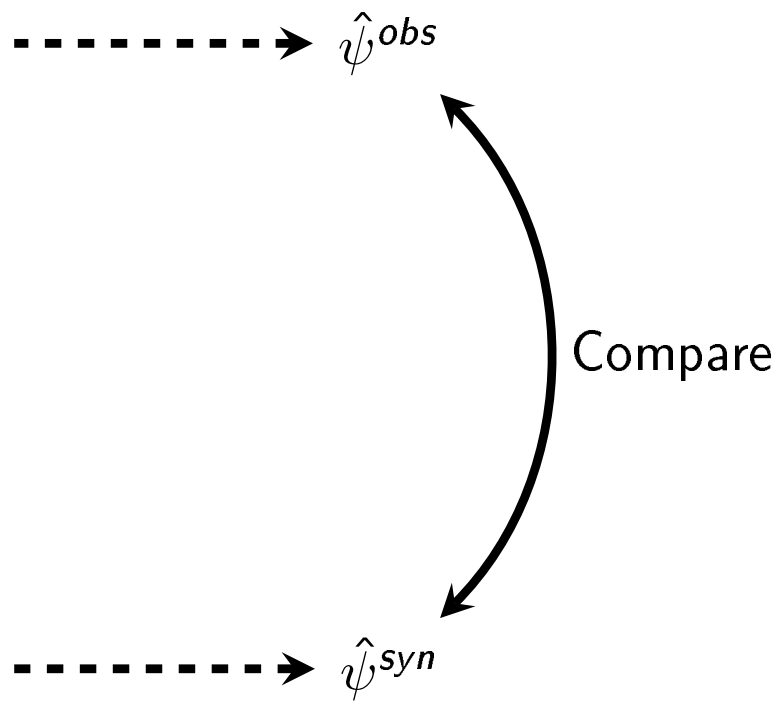
Fit economic model to *observed* data.

↓

Simulate synthetic dataset using estimates.

↓

Fit economic model to *synthetic* data.



structToolbox

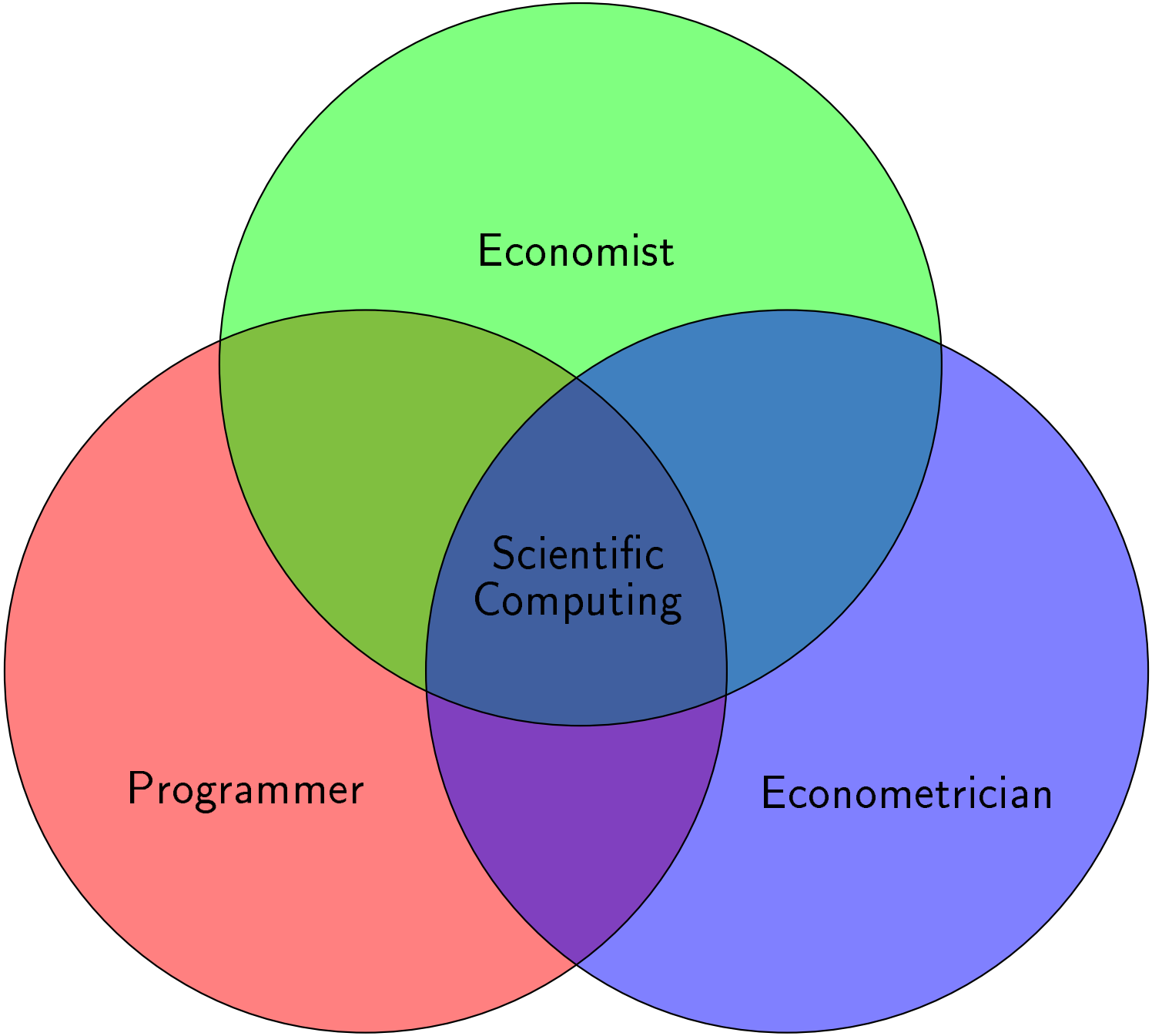
(under construction)

Transparency, Recomputability, and Extendibility

http://www.policy-lab.org/structToolbox

- Documentation
  - Source Code
  - Test Suite
- Download

Conclusion


Economist
Scientific
Computing
Programmer
Econometrician

Software Engineering for Economists

Mail	info@policy-lab.org
Web	http://www.policy-lab.org/teaching/soft-engineering
Repository	https://github.com/practComp2014/softwareEngineering

Philipp Eisenhauer

Mail	eisenhauer@policy-lab.org
Web	http://www.policy-lab.org/peisenha
Repository	https://github.com/peisenha

Appendix

References

Wilson, G., Aruliah, D. A., Brown, C. T., Hong, N. P. C., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K., Mitchell, I., Plumbley, M., Waugh, B., White, E. P., and Wilson, P. (2014). Best Practices for Scientific Computing. *PLOS Biology*.

Links

structToolbox http://www.policy-lab.org/structToolbox

grmToolbox http://www.policy-lab.org/grmToolbox

Git	http://git-scm.com
GitHub	https://github.com
R	http://www.r-project.org
rStudio	https://www.rstudio.com
Roxygen	http://roxygen.org
asana	https://app.asana.com
waf	https://code.google.com/p/waf