The hansl language for the gretl econometrics system

Allin Cottrell Riccardo (Jack) Lucchetti

Wake Forest University
DISES - Università Politecnica delle Marche, Ancona, Italy

Edinburgh – 01/05/2014

Econometrics: a brief history

Quantification of relationships among economic variables.

- "Political Arithmetick" (17th C: William Petty, Gregory King, Charles Davenant)
- Founding of Econometric Society (1930, Ragnar Frisch et al)
- Haavelmo, the Probability Approach (1944)
- Rapid development from 1960s to present

(Frisch's programme: "the unification of statistics, economic theory, and mathematics")

Aims and general character

Main uses of econometric quantification:

- testing of theories
- forecasting
- policy analysis

Application of economic and statistical theory to analysis of socio-economic data.

But also *development* of relevant statistical theory in face of difficult issues: dependence, non-homogeneity, problems related to the usage of non-experimental data.

Dual status of econometrics

Econometrics not a "sub-field" of economics in the same sense as, e.g., labour economics or health economics.

Rather it offers a set of *tools* used in almost all sub-fields (other than pure theory) – plus a discipline that is as much mathematical statistics as economics.

However, an econometrician is not a statistician either: an econometrician is an economist with an above-average command of statistics.

Econometric coding

- Pioneers in 1960s mostly used Fortran (some big names still do)
- In early PC era, command-line programs offering canned routines
- Gauss (1984, MS-DOS), Matlab (also 1984), Ox (around 1997)
- 1990s to date: Evolution of command-line programs: added GUIs and also elements of matrix-oriented languages (Stata, 1985; Eviews, 1994)
- recent tendency: packages built on top of matrix-oriented languages (Dynare)

See, *e.g.*, Charles Renfro, "Econometric Software", in Belsley and Kontoghiorghes, *Handbook of Computational Econometrics* (2009).

Main numerical techniques in econometrics

- Widespread use of matrices (mostly real, only rarely complex)
- Classical optimization techniques for smooth functions (fancier stuff like genetic algorithms also used, but sparingly)
- RNG (increasingly popular, especially for Bayesian techniques)
- A few concepts borrowed from engineering literature: spectra, filtering, signal extraction.

Traditionally, the dimensionality of problems is relatively small: a few Kb of RAM often suffice. This is rapidly changing nowadays with "big data" problems.

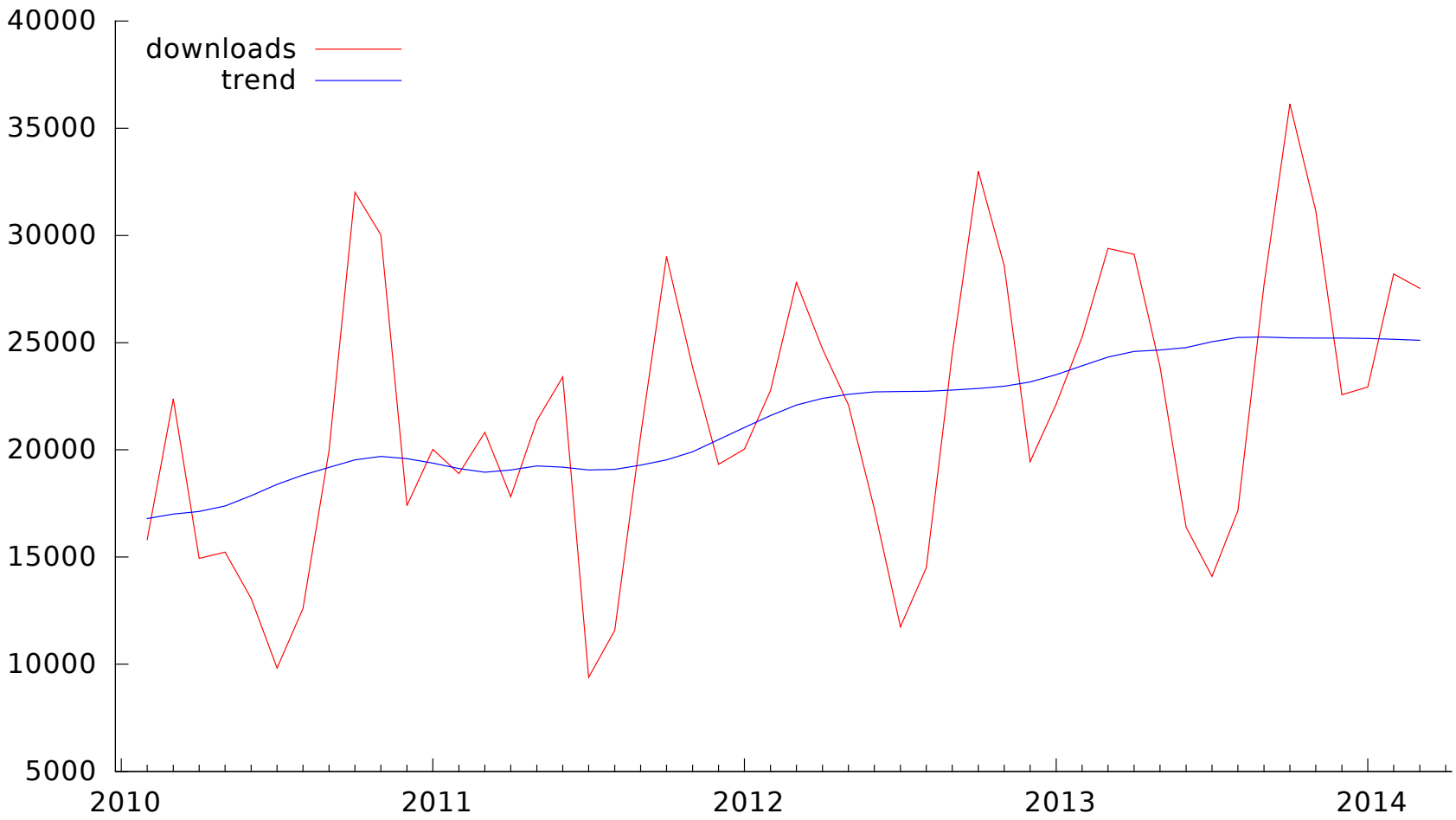
The "market" /uses for econometrics-related software

1. Undergraduate teaching (a big industry)
2. Professional applied work; this could be split into
   1. academic use (universities, mostly)
   2. corporate use (traditionally, large financial institutions but nowadays large retailers such as Amazon)
   3. policy (central banks, other government/supranational institutions)
3. Development of new estimators/tools

Currently dominated by proprietary software – packages such as Stata and Eviews (for 1 and 2), plus Matlab (for 2.3 and 3). But also incursion of R and gretl, and some use of lower-level languages for use 3.

Very few people use compiled languages, even for CPU-intensive tasks.

Monthly gretl downloads from sourceforge



Key abstractions in the econometric domain (besides those found in general matrix-oriented software):

- "the dataset" (plus series, lists of series)
- "the bundle" (as a means of avoiding the need for functions with an excessive number of parameters)

Duality:

- Matrices, scalars, strings, functions *versus*
- Datasets, series, commands

Softening of duality achieved via "accessors" that can be used following commands. (Non-trivial commands become somewhat function-like.)

Key abstractions in the econometric domain (besides those found in general matrix-oriented software):

- "the dataset" (plus series, lists of series)
- "the bundle" (as a means of avoiding the need for functions with an excessive number of parameters)

Duality:

- Matrices, scalars, strings, functions *versus*
- Datasets, series, commands

Softening of duality achieved via "accessors" that can be used following commands. (Non-trivial commands become somewhat function-like.)

Hansl

- Path of gretl development similar to proprietary software (DOS command line program → cross-platform → add GUI → add Matlab-like matrix functionality → add advanced scripting)
- "Feel" of scripting language owes something to bash shell
- C back-end (of course, with a little help from friends: netlib, BLAS, lapack, FFTW and others)
- Transition to development of gretl via hansl (function packages, with GUI integration)
- Some "legacy" formulations and inconsistencies, but hansl much cleaner and easy to learn than many others (Stata, Eviews, even R); OK, we may be a little biased here ;-)

Hansl

- Path of gretl development similar to proprietary software (DOS command line program → cross-platform → add GUI → add Matlab-like matrix functionality → add advanced scripting)
- "Feel" of scripting language owes something to bash shell
- C back-end (of course, with a little help from friends: netlib, BLAS, lapack, FFTW and others)
- Transition to development of gretl via hansl (function packages, with GUI integration)
- Some "legacy" formulations and inconsistencies, but hansl much cleaner and easy to learn than many others (Stata, Eviews, even R); OK, we may be a little biased here ;-)

Hansl

- Path of gretl development similar to proprietary software (DOS command line program → cross-platform → add GUI → add Matlab-like matrix functionality → add advanced scripting)
- "Feel" of scripting language owes something to bash shell
- C back-end (of course, with a little help from friends: netlib, BLAS, lapack, FFTW and others)
- Transition to development of gretl via hansl (function packages, with GUI integration)
- Some "legacy" formulations and inconsistencies, but hansl much cleaner and easy to learn than many others (Stata, Eviews, even R); OK, we may be a little biased here ;-)

Hansl

- Path of gretl development similar to proprietary software (DOS command line program → cross-platform → add GUI → add Matlab-like matrix functionality → add advanced scripting)
- "Feel" of scripting language owes something to bash shell
- C back-end (of course, with a little help from friends: netlib, BLAS, lapack, FFTW and others)
- Transition to development of gretl via hansl (function packages, with GUI integration)
- Some "legacy" formulations and inconsistencies, but hansl much cleaner and easy to learn than many others (Stata, Eviews, even R); OK, we may be a little biased here ;-)

Hansl by example: estimating a VAR

A *VAR* is a commonly-used tool in macroeconomics: in its most basic incarnation:

$$y_t = \mu + \sum_{i=1}^{p} \Phi_i y_{t-i} + \varepsilon_t \tag{1}$$

- y_t is an n-dimensional vector of observable quantities at time t; typically, macroeconomic variables: unemployment rate, inflation, interest rates, etc.
- The μ vector and the Φ_i matrices are unknown and must be estimated from the data.
- ε_t is an n-dimensional vector white noise process.

Suppose that (1) is a valid representation for the observed data: the econometrician's job is to estimate the unknown quantities so that the model can be used for policy analysis and forecasting.

`var-script.pdf`

Parallelization in Hansl

Via OpenMP: been there for a while. It's hard to beat farming this out to OpenBLAS, apart from some specialized routines.

Via MPI: work in progress. But we have

`scalar mpisend(object x, int dest)`
- send object `x` to node `dest`

`object mpirecv(int src)`
- receive an object from node `src`

`scalar mpibcast(object *x [,int root])`
- broadcast object `x`

`scalar mpireduce(object *x, string op [,int root])`
- reduce object `x` via `op`

`scalar mpiallred(object *x, string op)`
- reduce object `x` via `op`, all nodes

`scalar mpiscatter(matrix *m, string op [,int root])`
- scatter matrix `m` using `op`