Using Gretl and Hansl for Econometrics

Artur Tarassow[1]

[1] Faculty of Economics and Social Sciences
University of Hamburg

Melbourne, 23.03.2017

Outline

1. Aims and general character
2. Econometric software
3. Some background information on Gretl
4. Datasets and matrices
5. Hansl
    - Models and methods
    - User-contributed provided packages
6. Hansl by example
7. Other aspects
8. Get started with Gretl and Hansl

Aims and general character

Main uses of econometric quantification:[1]

- 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.

[1] The content of this presentation relies heavily on the joint work of Allin Cottrell (Wake Forest University) and Riccardo ”Jack”Lucchetti (Università Politecnica delle Marche).

Dual status of econometrics

Econometrics is 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)

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 to hold data. 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 also large retailers such as Amazon/Google)
    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 (C, Fortran etc.), even for CPU-intensive tasks.

Some background information on Gretl I

- Acronym for: **G**nu **R**egression, **E**conometrics and **T**ime-series **L**ibrary
- URL: `http://gretl.sourceforge.net/`
- Comprises a large shared library, a common command line program and a GUI client.
- Makes use of reliable free software packages, e.g. (multi-threaded) LAPACK/BLAS, fftw, GTK, gnuplot, etc.
- First version released in January 2000 and has been under active development since then → *open-source* and *free*.
- Written in C and available for Windows, OS X and Linux.
- User interface is available in 16 languages.
- Gretl is documented in a *User Guide* of 350+ pages and a *Command Reference* of 160+ pages; a tutorial introduction to hansl is also available.

Some background information on Gretl II

- Gretl comprises a full featured graphical user interface (GUI).
- Functions can be driven either by hansl scripting or by the GUI.

Unique Selling Point

- Gretl offers a **high-level matrix oriented language** similar to Matlab and Gauss ...
- **AND** a high-level language that is attuned to **econometrics**.

Datasets

- A dataset is basically the union of matrix **y** (T by k) and **X** (T by m) with some additional metadata.
- Econometrics deals with the three main forms of data: (i) cross-section, (ii) time-series and (iii) panel.
- Think of a dataset as a big matrix. To make sense of a regression one needs to know:
  - What the columns and rows refer to?
  - Are the rows representing time period: (i) what is the sample beginning and sample end, (ii) at what frequency were data recorded?
- In econometric software the dataset is typically *not* a matrix as such but a richer structure.

  ⇒ part or all of the dataset can be turned into a matrix on demand.

Datasets

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, arrays, bundles, functions (harder but flexible) *versus*
- Datasets, series, commands (easy)

Softening of duality achieved via "accessors" that can be used following commands.

Hansl – “Hansl’s A Neat Scripting Language”

- Path of gretl development similar to proprietary software (DOS command line program → cross-platform → add GUI → add Matlab-like matrix functionality → add advanced scripting → parallelisation)
- “Feel” of scripting language owes something to bash shell (UNIX).
- 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 optional GUI integration)
- Some “legacy” formulations and inconsistencies, but hansl is much cleaner and easier to learn than many others (Stata, Eviews, even R); OK, I may be a little biased here ;-)

Hansl

- Hansl's repertoire includes over 130 commands to date:
  - (hypothesis) Tests
  - (descriptive) Statistics
  - Dataset (manipulation, sorting, etc.)
  - Estimation (OLS, MLE, GMM, single-eq. and systems, etc.)
  - Graphs (scatter plots, boxplot, time-series, etc.)
  - Programming (control flow and debugging)
  - Transformations
  - Printing
  - Utilities
  - Forecasting

Models

Gretl *natively* implements a wide number of models and methods

- Time series methods
  - ARIMA, univariate GARCH-type, (S)VARs and VECMs, unit root and cointegration tests, Kalman filter, MIDAS, real-time datasets
- Limited dependent variables
  - logit, probit, tobit, sample selection, interval regression, models for count and duration data, etc.
- Panel-data estimators, including instrumental variables, probit and GMM-based dynamic panel models

Many more methods are provided by *user-contributed* packages

`http://ricardo.ecn.wfu.edu/gretl/cgi-bin/gretldata.cgi?opt=SHOW_FUNCS`

User-contributed provided packages

Currently about 104 user-contributed packages are provided. Some time-series/panel packages you might be interested in are e.g.:

- `BMA` – Bayesian Model Averaging for the linear regression models with jointness measures
- `BreitungCandelonTest` – Breitung-Candelon test of frequency-wise Granger (non-) causality
- `coint2rec` – Cointegration stability tests (Hansen&Johansen)
- `DIF_panelcoint` – Panel (non)-cointegration tests with bootstrap p-values
- `gig` – An assortment of univariate GARCH models
- `gregory_hansen` – Residual-based tests for cointegration in models with regime shifts
- `johansensmall` – Small-sample Johansen coint. rank tests (bootstrap and Bartlett)
- `StrucBreak` – Autodetection of structural breaks in linear models, Bai-Perron style
- `SVAR` – Structural VARs
- `Threshold_Panel` – Panel threshold model (Hansen, JoE 1999)

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
var-script.inp
```

Supported data formats

Supported formats include:

- Own XML data files
- Comma Separated Values
- Excel
- Gnumeric and Open Document worksheets
- Stata .dta files
- SPSS .sav files
- Eviews workfiles
- JMulTi data files
- Own format binary databases (allowing mixed data frequencies and series lengths)
- RATS 4 databases and PC-Give databases
- Includes a sample US macro database. See also the gretl data page.

Communication with other software

Gretl can interact with other software packages.

Easily send and receive datasets and matrices.

Call other software via Gretl's `foreign-language` block.

List of software supported:

- R
- Ox
- Octave
- Stata
- Python
- Julia

Gretl and R Example

```
function list RStructTS(series myseries)
    smpl ok(myseries) --restrict
    sx = argname(myseries)
    foreign language=R --send-data --quiet
        @sx <- gretldata[, "myseries"]
        strmod <- StructTS(@sx)
        compon <- as.ts(tsSmooth(strmod))
        gretl.export(compon)
    end foreign
    append @dotdir/compon.csv
    rename level @sx_level
    rename slope @sx_slope
    rename sea @sx_seas
    list ret = @sx_level @sx_slope @sx_seas
    return ret
end function
# ------------ main -------------------------
open bjg.gdt
list X = RStructTS(lg)
```

Parallelization in Hansl

Via **OpenMP**

Via **MPI**: Gretl has

```
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

For details and examples see here: `https://sourceforge.net/projects/gretl/files/manual/gretl-mpi-a4.pdf/download`

Get started with Gretl and Hansl

Take a look at the following resources to start

- *Gretl Command Reference* (accessible via the Help menu in the gretl GUI): contains a complete listing of the commands and built-in functions available in hansl, with a full account of their syntax and options.
- *Gretl User's Guide*: Chapters 10 to 16 on data types, loops, the definition and use of functions.
  `http://sourceforge.net/projects/gretl/files/manual/`
- *Sample scripts*: The gretl package comes with a large number of sample or practice scripts (under menu item /File/Script files/Practice file).
- *Function packages*: Ambitious examples of hansl coding (via the gretl menu item /Tools/Function packages/On server.)
- *gretl-users mailing list*: Most well-considered questions get answered quite quickly and fully
  `http://lists.wfu.edu/mailman/listinfo/gretl-users`
- Textbook *Using gretl for Principles of Econometrics* (4th edit.)
  `http://www.learneconometrics.com/gretl/`