Essential documentation for the software used in *Long run relations in European electricity prices*[*]

Matteo Pelagatti

November 2008

This software is Ox code (Doornik 2007), so you need either the commercial version (OxMetrics) or the console version you may find at `www.doornik.com`.

1 lucas_test.ox

This is the library for computing Johansen LR test and Lucas PLR test based on Student's distribution and for simulating its p-values using the bootstrap.

`johansen(mDy, mYx, mUx, avEval, amAlpha, amBeta, ...)`

Performs Gaussian ML for ECM model with cCoint cointegration relations.

`mDy` $(T \times K_1)$ matrix of first difference of original series.

`mYx` $(T \times K_2 \geq K_1)$ matrix of 1-lagged levels of original series and restricted regressors.

`mUx` $(T \times K_3)$ matrix of unrestricted regressors (delays of `mDy` and unrestricted dummies).

`avEval` (address) out: $(K_1 \times 1)$ vector of eigenvalues.

`amAlpha` (address) out: adjustment coefficient matrix.

`amBeta` (address) out: cointegration matrix.

optional parameters (in case cointegration rank known).

`cCoint` scalar rank of cointegration.

`amGamma` (address) out: unrestricted regressors coefficients matrix.

[*]Use at your own risk.

`amV` (address) out: covariance matrix of errors.

`amErr` (address) out: error series.

Returns: vector with Johansen trace statistics if computations succeeded and only standard parameters are present, log-likelihood if optional parameters are present, 0 if computational problems arose.

`series_builder(mY, mX, mU, cP, amDy, amYx, amUx)`

It makes the `mDy` and `mYx` series for `johansen()`, starting from the endogenous variables `mY`, the restricted regressors `mX` and the unrestricted regressors `mU`.

`mY` $(T \times K_1)$ matrix of endogenous variables.

`mX` $(T \times [K_2 - K_1])$ matrix of restricted regressors.

`mU` $(T \times K_3)$ matrix of unrestricted regressors.

`cP` (scalar) number of lags of differenced variables in the ECM.

`amDy` (address) out: $(T \times K_1)$ matrix of first differences of `mY`.

`amYx` (address) out: $(T \times K_2 \geq K_1)$ matrix of 1-lagged levels of original series and restricted regressors.

`amUx` (address) out: $(T \times [K_3 + cP * K_1])$ matrix of 1 to `cP` lags of differenced `mY` and unrestricted regressors.

Returns: 1.

`tecm(mDy, mYx, mUx, dDF, cCoint, amAlpha, amBeta, amGamma, amV, amErr, avW)`

Performs Student's t ML for ECM model with `cCoint` cointegration relations using the EM algorithm (Lange *et al.* 1989, Little 1988).

`mDy` $(T \times K_1)$ matrix of first difference of original series.

`mYx` $(T \times K_2 \geq K_1)$ matrix of 1-lagged levels of original series and restricted regressors.

`mUx` $(T \times K_3)$ matrix of unrestricted regressors (delays of `mDy` and unrestricted dummies).

`dDF` (scalar) degree of freedom of Student's t.

`cCoint` (scalar) rank of cointegration.

`amAlpha` (address) out: adjustment coefficient matrix.

`amBeta` (address) out: cointegration matrix.

`amGamma` (address) out: unrestricted regressors coefficient matrix.

`amV` (address) out: covariance matrix of errors.

`amErr` (address) out: error series.

`avW` (address) out: $(T \times 1)$ vector with weights for WLS.

Returns: log-likelihood.

`lucas_plr(mY, cLags, cDetType, dDF, ...)`

Lucas' Pseudo LR test with Student's t with `dDF` degrees of freedom.

`mY` $(T \times K_1)$ matrix of endogenous series.

`cLags` (scalar) number of lags of differenced `mY`.

`cDetType` (scalar) type of deterministic part: `NONE` = no deterministic, `RCONST` = restricted constant, `CONST` = unrestricted constant, `RTREND`, restricted trend, `TREND` = unrestricted linear trend (this may lead to deterministic quadratic trends).

`dDF` (series) degrees of freedom of Student's t.

Optional arguments:

`[0]` restricted regressors.

`[1]` unrestricted regressors (if only restricted regressors needed, pass <> as first optional argument).

Returns: vector of PLR statistics.

`sim_ecm(mEps, mYstart, cCoint, cP, mX, mU, mAlpha, mBeta, mGamma, amY)`

It simulates from an ECM(`cP`) model with cointegration rank `cCoint` using `mYstart` as initial values, `mEps` as shocks, `mX` as restricted regressors, `mUx` as unrestricted regressors, parameters `mAlpha`, `mBeta` and `mGamma` the simulated time series is written in the address `amY` (for formats see `series_builder()`). Notice: `mYstart` must be of rows `cP+1`.

`boot_plr(cIter, mY, cLags, cDetType, dDF, ...)`

It bootstraps Lucas' Pseudo LR test with Student's t with `dDF` degrees of freedom with the method of Swensen (2006).

`mY` ($T \times K_1$) endogenous series.

`cLags` (scalar) number of lags of differenced `mY`.

`cDetType` (scalar) type of deterministic part: `NONE` = no deterministic, `RCONST` = restricted constant, `CONST` = unrestricted constant, `RTREND` = restricted trend, `TREND` = unrestricted linear trend (may lead to deterministic quadratic trends).

`dDF` (scalar) degrees of freedom of Student's t.

Optional arguments:

`[0]` restricted regressors.

`[1]` unrestricted regressors (if only restricted regressors needed, pass <> as first optional argument).

The function returns no output, but prints the PLR tests with p-values.

2 NewKPSS.ox

This is the library for computing KPSS and IKPSS tests. It needs the library `rq.ox` by Roger Koenker (`http://www.econ.uiuc.edu/ roger/`).

The two functions for the end-user are

```
kpss(vY, bTrend, cTrunc)
ikpss(vY, bTrend, cTrunc)
```

They compute the KPSS and IKPSS statistics.

`vY` ($T \times 1$) time series to test for stationarity.

`bTrend` (boolean) 0 = de-mean (resp. de-median), 1 = OLS de-trending (resp. LAD de-trending).

`cTrunc` (scalar) set the bandwidth (or truncation) parameter. If set to -1 the function computes it automatically (see below)

These functions use three global variables that may be changed any time before the function call:

`M_KERNEL` may be assign equal to the values `QUADRATIC` or `BARTLETT`.

`M_BANDWIDTH` may be assigned to the values `ANDREWS` or `NEWEYWEST`.

`M_VERB` 0 = textual output off, 1 = textual output on.

3 nh.ox

This is the library for computing the original and the robust versions of Nyblom and Harvey's (2000) cointegration test. It needs the library `rq.ox` by Roger Koenker (`http://www.econ.uiuc.edu/ roger/`).

`nhstat(mX, cTrend, cBandWidth)inhstat(mX, cTrend, cBandWidth)`

They perform the multivariate generalization of KPSS test by Nyblom and Harvey (2000) and the robust version thereof based on signs.

`mX` $(T \times N)$ data matrix.

`cTrend` (scalar) 0 = nothing, 1 = de-mean/median, 2 = de-trend (linear trend).

`cBandWidth` (scalar) negative = automatic bandwidth, integer = bandwidth of Bartlett window.

Return: the vector of NH/INH statistics.

4 fmlad.ox

This library contains functions for computing the FM-LAD regression by Phillips (1995). The only relevant function for the end user is the following. It needs the library `rq.ox` by Roger Koenker (`http://www.econ.uiuc.edu/ roger/`).

`fmlad(vY, mX, iBandWidth)`

`vY` $(T \times 1)$ vector of dependent variable.

`mX` $(T \times N)$ matrix of regressors (a constant is automatically included).

`iBandWidth` (scalar) bandwidth parameter. If negative automatic bandwidth selection $(4(T/100)^(2/9))$.

References

Doornik JA (2007). *An Object-oriented Matrix Programming Language - Ox 5*. Timberlake Consultants Ltd, London.

Lange KL, Little JA, Taylor JMG. 1989. Robust Statistical Modeling Using the t distribution. *Journal of the American Statistical Association* **84**: 881-896.

Little RJA. 1988. Robust estimation of the mean and covariance matrix from data with missing values. *Applied Statistics* **37**: 23-38.

Nyblom J, Harvey A. 2000. Test of common stochastic trends. *Econometric Theory* **16**: 176-199.

Phillips PCB. 1995. Robust nonstationary regression. *Econometric Theory* **11**: 912-951.

Swensen AR. 2006. Bootstrap algorithms for testing and determining the cointegration rank in VAR models. *Econometrica* **74**: 16991714. DOI: 10.1111/j.1468-0262.2006.00723.x