The PMG package for gretl
Pooled Mean Group panel estimation

by Jack Lucchetti and Sven Schreiber*

v0.9 (March 2020)

This package implements the Mean Group (MG) and Pooled Mean Group\(^1\) (PMG) for possibly nonstationary panel data; in the latter, (only) the long-run coefficients are homogeneous across units. The estimated long-run relationship is a cointegrating equation in the case where the involved variables are I(1).

The package contains the functions \texttt{PMG\_setup}, \texttt{PMG\_estimate} and \texttt{MG\_estimate} for scripting, but a graphical interface is also provided, and the package attaches itself in the Model menu under the Panel sub-menu if so desired.

The underlying model equation is one of the following, where deterministic terms are omitted for clarity:

\[
\Delta y_{it} = \phi_i y_{it-1} + \sum_{k=1}^{K} (\beta_{ki} x_{kit} + g_{ki} \Delta x_{kit}) + \text{lagged diffs} + e_{it} \tag{1}
\]

\[
\Delta y_{it} = \phi_i y_{it-1} + \sum_{k=1}^{K} (\beta_{ki} x_{kit-1} + g_{ki} \Delta x_{kit}) + \text{lagged diffs} + e_{it} \tag{2}
\]

\[
\Delta y_{it} = \phi_i y_{it-1} + \sum_{k=1}^{K} (\beta_{ki} x_{kit-1}) + \text{lagged diffs} + e_{it} \tag{3}
\]

In (1) the levels as well as the differences of the exogenous regressors are included contemporaneously, which we denote by “PSS”. Representation (2) is more conventional in that the levels terms that enter the long-run relationship are lagged just like the \(y_{it-1}\) term. We denote this as “standard”, but note that since the contemporaneous differences \(\Delta x_{kit}\) are still present this representation is fully equivalent to (1); the likelihood and long-run parameter estimates will be identical. In contrast, in model type (3) all contemporaneous terms are removed which does make a difference and which we label “reduced form”.

In all model types above the coefficients are not yet restricted and in principle heterogeneous (notice the \(i\) indices). Estimating each panel unit separately and averaging the estimates yields the MG estimator, whose main purpose is merely to provide a comparison benchmark. The pooling in the PMG approach works by assuming that the long-run coefficients \(\theta = -\beta_{ki}/\phi_i\) are homogeneous across the panel units, i.e. they do not depend on \(i\). The PSS article shows how to estimate such models. The \(\phi_i\) coefficients are still left as heterogeneous and are usually treated by reporting their associated mean-group estimate.

1 Public functions for scripting

All the functions provided in this package assume that the currently loaded dataset is endowed with a panel structure (see the \texttt{setobs} command in the \textit{Gretl Command Reference}).

\*Please send questions and comments and especially bug reports to the gretl mailing list.
Then one sets up the PMG model by calling the `PMG_setup()` function with suitable arguments, followed by a call to `PMG_estimate()`.

The package example script, `PMG_sample.imp`, replicates one of the estimates from the original PSS article.

### 1.1 PMG_setup

Arguments (all arguments with defaults are optional):

1. endogenous variable $y$ (series)
2. regressors $X$ (cointegration partners if non-stationary; list)
3. vector of lag lengths of $y$ and all $X$es (matrix, default all ones)
4. other exogenous covariates $Z$ (list, default null)
5. model type chooser (integer, default PSS-style)

For argument 3, the lag length convention is to give the maximum lag referring to levels. Either give a single number for a common lag length, or for each I(1) variable a value must be specified, so the vector length must then be $K + 1$. A lag order of zero is acceptable for the $X$es, but see also the model type (arg 5).

A constant term is always included, other (unrestricted) deterministics must be given in argument 4.

The model type in argument 5 defines the treatment of the contemporaneous values of the regressors as explained above.

**Returns:** A gretl bundle with the following elements...

- `kx`: number $K$ of I(1) regressors in $X$ (int)
- `kz`: number of exogenous regressors (can differ from $Z$ elements because of constant term; int)
- `Ti`: valid time series lengths of every unit (matrix with $N0$ elements)
- `N0`: number of units before possibly discarding some (int)
- `N`: number of effective units (int)
- `keysN`: units indices in effective dataset (excluding those discarded; matrix)
- `lagorders`: copy of argument 3 (matrix, 1 + $K$ -vector)
- `yname`: name of the endogenous variable in the dataset (string)
- `xnames`: names of $X$es in dataset (string array)
- `mtype`: copy of argument 5 (int)
- `autoinit`: initial values for the long-run coefficients based on a static fixed-effects regression (matrix)

(externally used objects: Hdy, Hylag, HX, mW, matrices arrays; printMGonly, bool)
1.2 PMG_estimate

Arguments:

1. bundle from PMG_setup (pointer to bundle)
2. verbosity switch (0 to 2, default 1)
3. vector of manually supplied initial parameter values (matrix, optional)
4. maximum number of iterations (default 100, int)
5. switch to initialize by the reduced-form model (default off, bool)

Argument 3 may be useful if the likelihood maximization fails for the automatic initial values. Similarly for argument 4, if the default number of iterations turns out to be too small to achieve convergence. Finally, argument 5 can be used for model types “PSS” or “standard” (1 or 2) to first estimate the reduced-form model just to obtain starting values for the long-run coefficients, but normally you would probably not need that.

The verbosity switch can be set to 0 (print nothing), 1 (default, print estimates for long-run coefficients) or 2 (print iterations too – which may only work in the scripting context).

The function adds to the input bundle (in pointer form) the following elements (unless an error occurs):

- warnings: some messages (string)
- verbose: copy of argument 2
- iter: number of needed iterations (int)
- iter1stround: non-zero only if argument 5 was set to on (int)
- loglik: final log likelihood value (scalar)
- phis: group-specific ECT loadings (matrix, N-vector)
- avephi: mean-group estimator of the ECT loadings (scalar)
- Vavephi: variance of that loadings estimator (scalar)
- lrcoeff: the long-run coefficients (matrix, K-vector)
- PMGvcv: covariance matrix of the long-run coeffs (matrix)
- Htestsignal: result code of what happened during the Hausman test (int)
- Hstat: Hausman test stat for homogeneity of long-run coeffs, see below (scalar)
- Hpval: associated p-value (scalar)

Furthermore, since the Hausman test is automatically executed which needs the mean-group estimator, the bundle will also contain the result objects from there, see the MG_estimate function.

(internal objects: param, fs)

The function returns 0 for success, or a non-zero scalar error code.
Hausman test  The test is run automatically and refers to all long-run coefficients.\textsuperscript{2}

If verbosity is switched on, the Hausman test result is printed. If the original test statistic $H_{\text{stat}}$ is negative, the p-value $H_{\text{pval}}$ is based on the absolute value of $H_{\text{stat}}$; this is admissible since under $H_0$ it doesn’t alter the test. In that case also another approach is provided and the results stored under $H_{\text{stat1}}$ and $H_{\text{pval1}}$. (The latter approach uses the average ratio of the residual variance estimates from PMG and MG to ‘inflate’ the VCV of the MG coefficient estimates. The idea is to use a common residual variance estimate for both VCVs, similar in spirit to Stata’s `sigmamore` option. Again, this is justified under $H_0$, but in this PMG context it may not always succeed, in contrast to the trivial solution that takes the absolute value.)

The $H_{\text{testsignal}}$ is 0 if the test proceeds normally; the other values are 1: the VCV matrix difference was not positive definite, 2: in addition, the original test statistic was negative, 3: furthermore, despite the implemented remedy the new VCV matrix difference was still not positive definite, and finally 4: the test statistic was still negative (such that the only workable option is to use the absolute value of the original test statistic).

1.3 MG_estimate

Arguments:

1. bundle from PMG_setup (pointer to bundle)

2. switch to print the MG estimate directly (default off, bool)

Provides mean group estimates, that is simple averages of the unit-specific coefficients. The main purpose is to facilitate the Hausman test. If the function is called directly and not in the context of a PMG estimation you can switch argument 2 to on to get the printout. It adds to the input bundle the following new members:

- MGcoeff: the loading and the long-run coefficients ($1 + k\times$ vector)
- VMGload: variance of the loading (scalar)
- MGvcv: var-cov matrix of the long-run coeffs (simple nonparametric estimate)

(Internal: ‘residi’ matrices array of unit-specific residual vectors)

2 Changelog

- v0.9 March 2020 – integrate Hausman test (including remedies for negative test statistics); as byproduct add the fully heterogeneous mean-group estimator – fix nasty bug that included $\Delta X_t$ and $\Delta X_{t+1}$ in what was supposed to be the reduced-form spec – add the possibility to initialize the model types with contemporaneous regressors through the reduced form – enhance this documentation

- v0.1 April 2019 - initial release

\textsuperscript{2}Testing a subset may be added in the future.