The PTconf package for \textit{gretl}

Sven Schreiber

this version: 0.99 (March 2020)

This function package is based on my paper “The estimation uncertainty of permanent-transitory decompositions in co-integrated systems”, in \textit{Econometric Reviews} 2019, 38(3), pp. 279-300; DOI: 10.1080/07474938.2016.1235257, see also: http://www.tandfonline.com/doi/full/10.1080/07474938.2016.1235257. Please cite the paper if you use this package for research.

“PTconf” stands for Permanent-Transitory Confidence.

1 Brief account of the underlying methodology

Needless to say, it would be best if you checked out the mentioned paper above. But here’s a brief summary.

1.1 Framework and assumptions

Consider a standard $n$-dimensional VAR with $p$ lags:

$$y_t = A_1 y_{t-1} + ... + A_p y_{t-p} + \mu + \epsilon_t, \quad t = p, ..., T$$ (1)

where the innovations are white noise with covariance matrix $\Theta$. We can re-parameterize this system as a VECM:

$$\Delta y_t = \alpha \beta' y_{t-1} + \sum_{i=1}^{p-1} B_i \Delta y_{t-i} + \mu + \epsilon_t$$ (2)

When co-integration is present, the long-run matrix $\alpha \beta' = -I + \sum_{i=1}^{p} A_i$ has reduced rank $r$ which is the number of linearly independent co-integration relationships (and is also the column rank of the $n \times r$ matrices $\alpha$ and $\beta$). The coefficients of the lagged differences are given by $B_j = - \sum_{i=j+1}^{p} A_i$. We define the lag polynomial $B(L) = I - \sum_{i=1}^{p-1} B_i L^i$. Because it will be repeatedly needed below, we introduce an abbreviation for the following term: $Q \equiv B(1) - \alpha \beta'$.

It is well known that the constant term $\mu$ can serve two purposes: if unrestricted, it may represent a linear drift term in the levels of the variables, as well as balancing the mean of the co-integrating relations. But if it is restricted as $\mu = \alpha \mu_0$, the levels of the data are assumed to be free of linear trend components. In the following, we will deal with the more general case of an unrestricted constant, which is much more popular in economics given the trending behavior of many variables in growing economies. As a further deterministic component it would also be possible for our analysis to allow a linear trend term in the
co-integrating relations, because the convergence rate of its estimator is also greater than \( \sqrt{T} \). (It may be advisable in practical work to normalize the trend term to have mean zero.)

We assume that variables are \( I(0) \) or \( I(1) \) and that the co-integration rank \( r \), \( n > r > 0 \), and the lag order \( p \) are taken to be given and fixed.

### 1.2 Constructing the variances of the transitory components

The idea is to condition on the observed data relevant for the decomposition in period \( \tau \). It will become clear that for the Gonzalo-Granger (GG, see below) decomposition this concerns the data \( y_\tau \), and for the Stock-Watson (SW) decomposition the \( p \) vectors \( y_\tau, \ldots, y_{\tau-p+1} \) are involved. We could implement this idea by actually removing the conditioning data from the likelihood function; this could be achieved either by using impulse dummies for the corresponding observations, or in the often interesting case of the end of the sample, by simply shortening the sample. However, here we use an approach which makes it unnecessary to re-estimate the model: We rely on the fact that the conditioning data are negligible relative to the rest of the large sample. Hence, given that in any case our calculated confidence bands are only valid asymptotically, we estimate the model once over the entire sample including the data observations on which we will later condition. This can be regarded as a computational shortcut.

#### 1.2.1 Definition and representation of the Gonzalo-Granger decomposition

As shown by [1], when the permanent and transitory components are assumed to be linear combinations of the contemporaneous values \( y_t \) only, the PT decomposition is uniquely given as follows:

\[
y_t = \beta_\perp (a_\perp' \beta_\perp)^{-1} a_\perp' y_t + a (\beta' a)^{-1} \beta' y_t,
\]

where the first part is the non-stationary permanent component, and the second part is the transitory component given by a linear combination of the co-integrating relationships.

We will use the alternative formulation by [2] (based in turn on [3]) with only slightly different notation, which proves especially useful with the Stock-Watson decomposition below. An important projection matrix is given by

\[
P = Q^{-1} a \left[ \beta' Q^{-1} a \right]^{-1} \beta'
\]

Since \( \psi_t = Py_t \) is a linear combination of the co-integrating relations \( \beta' y_t \), it is obviously stationary, and it is actually shown by [3] that this is just the GG transitory component:

\[
y_t^{\text{transGG}} = \psi_t = Py_t
\]

This transitory component will in general have a non-zero mean, however. For an economic interpretation it is especially useful to consider a transformation of the transitory component which will have an unconditional expectation of zero, because the sign of that transformed component automatically tells us whether the observed level of a variable is below or above its permanent component. For example the sign of an output gap estimate is important for identifying a recessionary or overheating economy.
To this end we use the expression (again adapted from [3]) for the mean of the co-integrating relationships:

$$E(\beta'y_t) = -\left(\beta'Q^{-1}\alpha\right)^{-1}\beta'Q^{-1}\mu,$$

which enables us to calculate the de-meaned transitory component:

$$\hat{\psi}_t = \psi_t - E(\psi_t)$$
$$= Q^{-1}\alpha[\beta'Q^{-1}\alpha]^{-1}\left(\beta'y_t + [\beta'Q^{-1}\alpha]^{-1}\beta'Q^{-1}\mu\right)$$
$$= Q^{-1}\alpha[\beta'Q^{-1}\alpha]^{-1}\left(\beta', [\beta'Q^{-1}\alpha]^{-1}\beta'Q^{-1}\mu\right)(y'_t, 1)'$$
$$\equiv G(y'_t, 1)'$$

Of course it is well known how to test the hypothesis that the GG transitory component of a certain variable vanishes completely. From the definitions of the GG decomposition it is clear that this involves a test that the $i$-th row of $\alpha$ is zero, which is a standard test problem given the co-integration rank and the estimated co-integration coefficients. (Namely the test for weak exogeneity of the $i$-th variable.) Here we are instead concerned with the uncertainty of the transitory component at a certain period, assuming that it exists at all.

1.2.2 The Delta method for the GG decomposition

We can express the de-meaned transitory GG component $\hat{\psi}_t$ in period $\tau \in \{p, ..., T\}$ as a function of the underlying short-run coefficient vector $k$ (whose estimates are $\sqrt{T}$-consistent), of the super-consistent co-integration coefficients $\hat{\beta}$, and of the data; since the Gonzalo-Granger transitory component $\hat{\psi}_t$ only depends on the contemporaneous observations, we only need to condition on $y_t$:

$$\hat{\psi}_t = f_{GG}(k, \hat{\beta}, y_t)$$

By (7) we have $f_{GG} = G(y'_t, 1)'$. Let $J_{GG} = \partial \hat{\psi}_t / \partial k'$ be the Jacobian matrix of that function with respect to $k$, treating the co-integration coefficients $\hat{\beta}$ as (asymptotically) fixed and conditioning on the data in period $\tau$.

Nevertheless, it is important to keep in mind that the derived confidence intervals are only valid for the chosen period $\tau$ and not as confidence bands for the entire sample, since we cannot condition on the entire sample and still have random estimates. When we display our calculations in a form that resembles confidence bands for the time series, it is just done for convenience, since different readers may be interested in different periods.

\[1\] In the paper we present the analytical form of this Jacobian. In earlier versions we used gretl’s fdjac() function as a numerical approximation. It turned out that the approximation is almost perfect for the Gonzalo-Granger case, and still good in the Stock-Watson case below, where it yielded results that were somewhat more volatile.
1.2.3 Definition and representation of the Stock-Watson decomposition

In a standard formulation, and assuming a fixed initial value, the permanent SW components are given by

$$y_t^{\text{permaSW}} = y_0 + C \mu t + C \sum_{s=1}^{t} \epsilon_s,$$ (9)

where $C$ is the long-run moving-average impact matrix of reduced rank (which however is not directly of interest here). For the co-integrated VAR model the SW decomposition essentially yields the multivariate Beveridge-Nelson decomposition, i.e. the permanent component is a multivariate random walk. In contrast, the permanent component of the GG decomposition is autocorrelated in differences. This property of the SW decomposition implies an appealing interpretation: Given our knowledge at time $t$, only the SW transitory component of the time series is expected to change in the future (because it is expected to converge to its unconditional expectation, or in the demeaned case, to zero), so it is especially important for forecasting. Of course, the GG and SW permanent components only differ by stationary terms and are co-integrated, therefore they share the same long-run features.

Again following [3] and [2] the transitory SW component can be written as the sum of two terms,

$$y_t^{\text{transSW}} = \psi_t = \psi_{1t} + \psi_{2t},$$ (10)

where the part $\psi_{1t}$ represents the error-correcting movements of the system and is identical to the GG transitory component above, while the part $\psi_{2t}$ are the remaining transitory movements of the system which do not contribute to the long-run equilibrium. For this latter part we need to define another lag polynomial if $p > 1$: $B^*(L) = B^*_0 + B^*_1 L + ... + B^*_{p-2} L^{p-2}$, where $B^*_j = \sum_{i=j+1}^{p} B_i$. Then it can be represented as a distributed lag of the observable variables:

$$\psi_{2t} = -(I - P)Q^{-1}B^*(L)\Delta y_t.$$ (11)

This second part remains to be demeaned as well, which can be achieved by using the known unconditional expectation of the differences:

$$E(\Delta y_t) = (I - P)Q^{-1}\mu$$ (12)

Using the abbreviation $\mu^* \equiv (I - P)Q^{-1}\mu$ we can now write:

$$\tilde{\psi}_{2t} = \psi_{2t} - E(\psi_{2t})$$
$$= -(I - P)Q^{-1}B^*(L) (\Delta y_t - \mu^*)$$
$$= \left( -[I - P] Q^{-1} \right) \left( B^*_{0}, -B^*_0 \mu^*, B^*_1, -B^*_1 \mu^*, ..., B^*_{p-2}, -B^*_{p-2} \mu^* \right) \times (\Delta y_t, \Delta y_{t-1}, 1, ..., \Delta y_{t-p+2}, 1)^\prime$$
$$= \left( -[I - P] Q^{-1} \right) \left( B^*_{0}, B^*_1, ..., B^*_{p-2}, -B^*(1) \mu^* \right) \times (\Delta y_t, \Delta y_{t-1}, ..., \Delta y_{t-p+2}, 1)^\prime$$

4
Then combining the two parts we have for the SW transitory component:

\[ \tilde{\psi}_t = \tilde{\psi}_{1t} + \tilde{\psi}_{2t} \]

\[ = \left( P \cdot -[I - P] Q^{-1} \begin{bmatrix} B_0^*, B_1^*, ..., B_{p-2}^* \end{bmatrix}, s_\mu \right) \times \]

\[ (y_t', \Delta y_t', \Delta y_{t-1}', ..., \Delta y_{t-p+2}', 1)' \]

\[ = S(y_t', \Delta y_t', \Delta y_{t-1}', ..., \Delta y_{t-p+2}', 1)' , \]

where the last element relating to the constant term is given by

\[ s_\mu = \left( Q^{-1} \alpha [\beta' Q^{-1} \alpha]^{-1} [\beta' Q^{-1} \alpha]^{-1} \beta' + [I - P] Q^{-1} B^* (1) [I - P] \right) Q^{-1} \mu. \] (15)

Note that also for the SW transitory component it is known how to test the hypothesis that it vanishes for a certain variable. In addition to the zero row of \( \alpha \) that was needed for the vanishing GG component, here the \( i \)-th rows of the various short-run coefficient matrices would also have to be zero. These restrictions essentially mean that the variable would be a strongly exogenous random walk. Again, for a given co-integration rank and super-consistently estimated co-integration coefficients, that would be a standard test problem.

### 1.2.4 The Delta method for the SW decomposition

The calculation of the uncertainty for the SW transitory component is analogous to the procedure for the GG component above. Again we can express the \( \tilde{\psi}_\tau \) (the demeaned overall transitory components) in period \( \tau \in \{p, ..., T\} \) as a function of \( k \), of the co-integration coefficients \( \beta \), and of the data; the only difference now is that we have to condition on the lagged values as well, \( y_{\tau}, ..., y_{\tau-p+1} \):

\[ \tilde{\psi}_\tau = f_{SW}(k; \beta, y_{\tau}, ..., y_{\tau-p+1}) \] (16)

The function \( f_{SW} \) is given by (14), \( f_{SW} = S(y_t', \Delta y_t', \Delta y_{t-1}', ..., \Delta y_{t-p+2}', 1)' \). Let \( J_{SW} = \partial \tilde{\psi}_\tau / \partial k \) be the Jacobian matrix of that function with respect to \( k \), where the details are again provided in the paper.

Of course, the interpretation remains only valid for a single chosen period.

### 1.3 The bootstrap method

Of course we hope that the bootstrap may yield some small-sample refinements over the asymptotic approximation by the delta method, for example by taking into account explicitly the variation of the co-integration coefficients estimates.

To be concrete, the distribution of the GG transitory component for the period of interest \( \tau \) can be simulated with the following algorithm. As a starting point we can use the standard estimates of (2).

1. Using the point estimates as the auxiliary data-generating process, simulate artificial data for the periods \( t = p, ..., T \) by drawing from a suitable distribution describing the innovation process \( \epsilon_t \). This could either be a random draw from a fitted parametric distribution like a multivariate normal distribution with covariance matrix \( \tilde{\Theta} \) (and
mean zero, of course), or re-sampling from the estimated residuals. We will use the observed values of \( y_t \) as the initial values of the artificial data in periods \( t = 0..p - 1 \). Note that even though the resulting artificial data may be very different from the original data because it will have different underlying realizations of the stochastic trends, this does not affect the distribution of the transitory components.

2. Re-estimate the VECM using the same specification that was applied to the original data, but with the artificial data created in the previous step. Then record the estimates of \( \hat{\psi}_{1\tau} \) as defined in equation (7), which means using the new estimated \( G \) coefficients of the current simulation run, but always employing the originally observed data \((y'_t; 1)\). Denote that estimate by \( \hat{\psi}_{1\tau,w} \), where \( w \) is a simulation index running from 1 to some sufficiently large integer \( W \).

3. Repeat the previous two steps \( W \) times to get simulated distributions of (the estimate of) \( \hat{\psi}_{1\tau} \).

4. For the \( i \)-th variable calculate variants of the confidence intervals for the estimate of \( \hat{\psi}_{1\tau} \) in the following two ways:

   (a) First we base the intervals directly on the distributions of \( \hat{\psi}_{1\tau,w} \) over all \( w \) and construct a confidence interval using the empirical quantiles of the simulated distributions: with \( \gamma \) as the nominal coverage of the error band (1 minus the type-1 error) and the quantiles of \( \hat{\psi}_{1\tau,w} \) given by \( \hat{\psi}_{1\tau,(1-\gamma)/2} \) and \( \hat{\psi}_{1\tau,(1+\gamma)/2} \), the intervals are constructed as

   \[
   [\hat{\psi}_{1\tau,(1-\gamma)/2}, \hat{\psi}_{1\tau,(1+\gamma)/2}].
   \]

   This construction is analogous to what [4] have called “other-percentile” bands in the slightly different context of impulse-response analysis, and they criticized their use as “clearly [amplifying] any bias present in the estimation procedure” (p. 1125).

   (b) Because of this criticism we also consider a Hall-type bootstrap, where the relevant distributions are given by \( \hat{\psi}_{1\tau,w} - \hat{\psi}_{1\tau} \), i.e., for each variable and period the bootstrap realizations are corrected by the original point estimate. Denoting the quantiles of these corrected distributions by \( \hat{\psi}_{1\tau,(1-\gamma)/2} \) and \( \hat{\psi}_{1\tau,(1+\gamma)/2} \), the Hall-type error bands are given by

   \[
   [\hat{\psi}_{1\tau} - (\hat{\psi}_{1\tau,w} - \hat{\psi}_{1\tau})_{(1-\gamma)/2}, \hat{\psi}_{1\tau} - (\hat{\psi}_{1\tau,w} - \hat{\psi}_{1\tau})_{(1+\gamma)/2}].
   \]

   Note that the upper quantiles of the corrected distributions are used for the calculation of the lower error band margins, and vice versa. This “counter-acting swapping” serves to cancel out any bias of the estimation procedure.

Of course the bootstrap procedure can be simultaneously applied to all periods in the sample. However, we still do not get confidence “bands” because we cannot condition on the

\[\text{In order not to overload the notation, we do not formally distinguish here between the true transitory component (true of course conditional on period-}\tau\text{ data) and its original point estimate, because we hope it is clear from the context that only the estimate can be used here.}\]
entire sample and do valid inference. As with the delta method, we can only derive valid confidence intervals for certain periods of interest.

For the SW transitory component the bootstrap method in this case is completely analogous to the GG case and to save space we will not repeat the details of the algorithm here. Essentially, the distribution of the $G$ coefficients is replaced by that of the $S$ coefficients, and of course the transitory component must be constructed using the extended data vector which includes lags, according to the formulas in Section 1.2.3.

2 Possible usages

The main usages are demonstrated in the example script that comes with the package (but in a different order).

2.1 Defaults only, no bootstrap (GUI or script)

The first, easiest, and fastest possibility is to use the `PTdefaults()` function. This is also the function that you see (without knowing it) when you execute the `PTconf` package in gretl’s graphical interface. This function takes the same arguments as gretl’s `vecm` command, namely the lag order $p$, the cointegration rank $r$, and an existing list of variables `endo`. It then retrieves both the GG (Gonzalo-Granger) and SWP (Stock-Watson-Proietti) transitory components of all variables (the point estimates in each period) under the names `<nameofvar>GG` and `<nameofvar>SWP`, contained in the gretl list that is returned. Furthermore, the limits of asymptotically justified confidence intervals with nominal 95% coverage are also retrieved, under the following names: `<nameofvar>GGlow`, `<nameofvar>GGup`, `<nameofvar>SWPlow`, `<nameofvar>SWPup`. Bear in mind that in smaller samples (and with an estimated cointegration matrix as it is done here) the true 95% confidence intervals may be much wider than these nominal-asymptotic ones.

The next two arguments are optional and only needed when you want to produce the default plots. The switch `plottrans` activates it and at the same time determines whether the GG (1) or SWP (2) components are plotted. The following list `which` may specify a subset of the full variable list. As before, only 95% asymptotic intervals are used in this quick-and-easy usage.

2.2 Full control (script only)

A more advanced but still fairly easy (hopefully!) usage is to specify more options, as follows.

1. You first call the `setupPT()` function, where apart from the (first three) parameters of the `PTdefaults()` function you can enter the number of bootstrap draws (or choose 0 to skip the bootstrap) and the significance level (i.e., 1 minus the desired coverage). You get a gretl bundle back which you have to pass on (in pointer form, e.g. `myPTb` if you choose the name `myPTb`) as the first argument to each of the following functions.

---

3Enabling GUI access to the more advanced features described below is on the to-do list for version 1.0.
(a) Note that only a single significance level (a scalar) can be given to `setupPT()`. If you want to get results for several significance levels at once, you need to change the `sigs` member of the bundle that `setupPT()` returns to a suitable vector, see the example script for a demo.4

(b) Another bundle member that is interesting at this stage is `restrBeta`. If you do nothing, then the cointegration matrix $\beta$ (dimension $n \times r$) is unrestricted and estimated by the Johansen procedure. If you know $\beta$ a priori you can specify it by creating/defining `restrBeta`. Again, see the example script for a demo.

2. Then optionally you can call `printPToptions()` to get a summary of what options are stored in the bundle so far.

3. The next step would be to call the `putPTconf()` function which computes the transitory components along with the confidence intervals, including running a bootstrap if that was specified. The result is that the main bundle then contains several arrays of matrices, e.g. when the bundle is called $b$: 
   
   b.amDeltaGG b.amDeltaSWP b.amBootnaiveGG b.amBootnaiveSWP b.amBootHallGG b.amBootHallSWP

   Each of these arrays contains $n$ matrices, one for each variable. Each matrix has $T + p$ rows (the sample size with initial values) and has twice the number of columns as the number of chosen significance levels. For example, for a single significance level 5% the columns would correspond to the 2.5% and 97.5% quantiles; for two significance levels 5% and 10% the columns would refer to: 2.5%, 5%, 95%, 97.5%, always ordered ascending.

   If you call `putPTconf()` with the optional integer argument you can re-set the number of bootstrap draws that was previously specified. That probably makes sense only when you want to run the same setup several times with a different number of bootstrap replications.5

4. To transform the results into series in your current workfile you can use the function `getPTtranslist()` which returns a list. In the `vars` list argument specify which variables you are interested in, and in the string argument `kind` you can include either “GG” or “SWP” or both. The `details` argument ranges from 0 to 2 and controls how much of the settings is included in the names of the produced series. For example, apart from the point estimate `detail==0` will give just `<nameofvar>_GGup` and `<nameofvar>_GGlow`. What this contains is determined by the `conftype` parameter; 0 for the asymptotic, 1 for the direct bootstrap, or 2 for the Hall bootstrap series. Only the first of the specified significance levels is picked here.

   The choice `detail == 1` would change this to: 
   `<nameofvar>_r<rank>p<lagorder>GG<quantile>`, for example `y_r1p4GG025` for a variable “y” with a single cointegration relation, 4 lags, and the 2.5% quantile of the dis-

---
4Do not run different setups with different scalar significance levels instead, because that would trigger separate bootstraps.
5However, a more efficient way would then probably be to run only one bootstrap with the maximum number of draws, and to analyze a lower number by using only a subset of the draws. Anyway, it’s only an optional parameter.
tribution. Only one of the Hall bootstrap or the asymptotic delta-method series pairs
are retrieved, but here for all specified significance levels.

Finally, detail == 2 also adds to the variable name one of the suffixes _asymp, _bDir<bootreps>, or _bHall<bootreps>, as it retrieves all those variants. If you want
to compare the asymptotic method and the bootstrap (without manually going into
the arrays of matrices described above, that is), this is the only way.

Now that the series are in your workfile, you can plot them or do anything you want. If
you want the permanent component, make use of the additive definition \( x_t = x_t^{\text{perma}} + x_t^{\text{trans}} \) to construct it as the difference of the observed variable and the calculated transitory
component (and similarly for the uncertainty of the permanent component).

Finally, if you want to plot some of the confidence intervals of the transitory com-
ponents, there is the \texttt{plotPTtrans()} function. The optional arguments are: The list argument
which may specify a subset of the variables, with transtype you choose between GG (1) or
SWP (2), and conftype works as in the \texttt{getPTtranslist()} function.

## 3 Public functions

(To learn about the further private functions, simply check out the source code and the
comments therein.)

- \texttt{setupPT}, returns bundle
  
  int p[1::2] "lag order (levels)",
  int r[1::1] "cointegration rank",
  const list endo "endog. variables list",
  int bootreps[0::0] "bootstrap draws (0 to skip)",
  scalar sig[0:1:0.05] "significance level",
  int verbosity[0::0] "how much echo and feed-
  back to give"

- \texttt{printPToptions}, returns nothing
  arguments: bundle *b

- \texttt{putPTconf}, returns nothing (adds stuff to the bundle)
  arguments: bundle *b, int newbreps[-1::-1]

- \texttt{getPTtranslist}, returns list
  arguments: bundle *b, const list vars, string kind, int details[0:2:0], int
  conftype[0:2:0]

- \texttt{PTdefaults}, returns list
  arguments: int p[1::2] "lag order (levels)",
  int r[1::1] "cointegration rank",
  const list endo "endog. variables list",
  int plottrans[0::2:0] "plot trans. components?"
  ["none", "GG", "SWP"],
  const list which[null] "plot which vars (empty=all)",
  int bootreps[0::0] "bootstrap draws (0 to skip)",
  int bconftype[1:2:1] "which boot-
  strap bands" ["direct", "Hall"],
  scalar sig[0:1:0.05] "significance level",
  bool verbosity[1] "print output?"

- \texttt{plotPTtrans}, returns nothing
  arguments: bundle *b, const list which[null],
  int transtype[1:2:2] "GG or SWP?" ["GG", "SWP"],
  int conftype[0:2:0] ["delta", "direct bootstrap", "Hall boot-
  strap"]
4 Sample script

This hasl sample script is of course included in the package and directly executable. It is

```hasl
# Example script for the Perma-Trans-decomposition estimation
# with confidence intervals
include PTconf.gfn

open greene5_1.gdt # US macro data

# specify the system
list endo = log(realcons realinvs realgdp)
smpl 1960:1 2000:4
lagorder = 4
cirank = 1

# prelim: check if the rank makes some sense
coint2 lagorder endo

### the actual PT package use
bundle b = setupPT(lagorder, cirank, ends) # lag order / CIrank / ...

# override default 5% sig
b.sigs = {0.1, 0.01}
# comment out to estimate beta unrestrictedly
# (name "restrBeta" is mandatory!)
matrix b.restrBeta = {1, 0, -1} # could be totally wrong...

printPToptions(&b)

# do the main calculations
putPTconf(&b)

# convert results to list of series
level_of_detail = 2
list results = getPTtranslist(&b, l_realgdp, "GG SWP", level_of_detail)

# Demonstrate a bootstrap:
# (only 500 draws in this example script to keep runtimes short)
bootsreps = 500
delete b.restrBeta # revert to unrestricted estimate

set stopwatch # timing of the bootstrap
putPTconf(&b, bootsreps) # update the estimates
print "Bootstrap timing:"
eval $stopwatch

list bootstrap = getPTtranslist(&b, l_realgdp, "GG SWP", level_of_detail)

# or alternatively a one-stop-shopping experience
# (e.g., unrestricted beta)
list defaultrresults = PTdefaults(lagorder, cirank, ends, 1)
```

10
5 Limitations, further plans, and changelog

- No exogenous variables yet except the unrestricted constant. This means neither seasonal dummies, nor a linear trend (apart from the induced drift term in the nonstationary directions), nor changing the constant to be restricted.

- Partial restrictions of $\beta$ are not allowed yet.

- Speeding up the bootstrap through parallel execution.... (leveraging a common VAR bootstrap framework that could also be used in other packages and addons....)

If you need these features please make yourselves heard; preferably on the gretl mailing list, but a private mail to the address given in the package will also be fine. The more demand there is, the more motivation for additional features...

Changelog v0.99 (March 2020): update this document, reflecting final publication of the underlying paper; internal code and file rearrangement to make packaging easier; add menu attachment; add more choices (with defaults) to the GUI function \texttt{PTdefaults}(); make it work if the initial sample contains missings (beginning or end); require gretl 2018a because of internal syntax

References


