The CSDpanel function package for gretl*

by Sven Schreiber† and Jörg Breitung‡

v0.2, July 2020

1 Summary

This function package provides some tests and estimates related to the phenomenon of cross-sectional dependence in panel data. You can think of panels with macroeconomic data where the cross-sectional dimension is typically filled with countries or states or regions and is not too large. In contrast, the time dimension should typically not be too short. Therefore, altogether $N$ and $T$ should be of roughly similar magnitude.

So far the package is limited to balanced datasets. Depending on popular demand, this restriction might be lifted in the future, but if you currently have unbalanced data you would have to remove those groups or time periods where some missing values occur.

Currently the package contains the following functionality:

• Test of no cross-section dependence: The two tests offered are that proposed by Baltagi, Feng, and Kao (2012) and the $\lambda_{CD}$ statistic by Pesaran (2012).1

• Provide two-way clustered standard errors that are also (but not exclusively) robust to cross-section dependence, suggested by Driscoll and Kraay (1998).

• Apply the common correlated effects (CCE) estimator by Pesaran (2006). The implementation here assumes homogeneous coefficients, i.e. it is not the mean-group variant.2

2 Using the package in a script

After installing (e.g. with pkg install CSDpanel.zip if you are online3) you use the standard activation as with every function package:

```gretl
include CSDpanel.gfn
```

*Please report problems and ask questions on the gretl mailing list, see https://gretl.ml.univpm.it/postorius/lists/gretl-users.gretl.ml.univpm.it/.

†Macroeconomic Policy Institute Düsseldorf [IMK at Hans Boeckler Foundation] and Free University Berlin

‡University of Cologne


2The mean group variant may be added in the future. However, it would be interesting to hear a reason why the average of a heterogeneous coefficient has any particular meaning...

3Or if you are using a version before gretl 2018d, just install CSDpanel.zip.
Afterwards, proceed to load your dataset, run some standard panel specifications, and then use the \texttt{testCSD()} or \texttt{robustCSD()} functions from section 4. Also see the example script.

If you want to employ the CCE estimator, you do not necessarily have to run a standard model before that, you could directly proceed to \texttt{panelCCE()}. However, a comparison is probably the usual thing to do.

3 The menu-driven (GUI) interface

\textbf{Installation} If the package does not yet appear in the list of locally installed functions, you can navigate to the menu File / Function packages / On server (or the translation in your respective local language). Find \texttt{CSDpanel} and click on the installation button (disk icon).

\textbf{Menu activation} This is only necessary once after the installation. In the window of local packages (File / Function packages / On local machine) select \texttt{CSDpanel} and then either click the add-to-menu button (\textasciitilde sign icon) or right-click on the name and then click on add to menu.

\textit{“Semi-manual" usage} One way to use the functions in the package is to open again the window of local packages, select \texttt{CSDpanel} and click the execute button (cogwheel icon). You will get a choice of the three functions described in section 4, and after choosing you will get a graphical interface to enter the respective parameters.

\textbf{Recommended usage} The GUI integration of this package happens in the window of a previously estimated panel model. Estimate your (static) panel specification as you would normally do. Then in the output window there will be a new menu entry “Cross-section dependence” in the “Analysis” menu. The advantage is that your model specification is already fully known at this point, and therefore the following dialog window just contains some clickable checkboxes (plus one optional parameter field). You may run several of the options (test, estimate) in one go.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{example.png}
\end{figure}
4 List of public functions

- testCSD

**Returns** 2x2 matrix, holding the $\lambda_{CD}$ statistic and its two-sided p-value in the first row, and the BFK test statistic and p-value in the second row.

**Takes** a series of panel residuals to be tested. Next an integer verbosity switch where 0 means silent, 1 gives some printout including a visual “heatmap” of the cross-sectional correlation matrix (up to 20 groups only), and 2 prints out the full matrix even if it is very large (default: 1).

Finally a matrix pointer (optional, i.e. default: null); pass this as a means to capture the estimated correlation matrix if you need it.

**Example** invocation:

```plaintext```
matrix stat_pv = testCSD($uhat, 0, &rescorrels)
```

If the verbosity is not set to silent then the function also prints out the test result and some descriptive information about the largest correlations.

- robustCSD

**Returns** matrix with the whole robust variance-covariance (VCV) matrix of the regressors (not just the variances or standard errors from the diagonal)

**Takes** as first input a bundle: a lookalike to the $model$ bundle from a panel regression.

The second (optional) input is another bundle to pass options. Currently the only supported option is the scalar/input lag truncation parameter for the underlying long-run variance estimation (default: integer part of $4 \times (T/100)^{2/9}$, as gretl’s own option “nw2”).

**Example** invocation:

```plaintext```
matrix myDK = robustCSD($model, defbundle(lagtrunc, 10))
```

The function also prints out the model results with the new robust standard errors and associated p-values. You do not need to capture the return value if all you want is this printout.

- panelCCE

**Returns** a bundle which is the $model$ bundle generated by gretl in the panel estimation with the transformed regressors that is performed internally.

**Takes** a series as the dependent variable, then a list of regressors, then a boolean switch whether or not robust (in the Arellano sense) standard errors should be used (default 0/no), and finally a boolean switch whether fixed unit effects should be used (default 1/yes; as opposed to a fully pooled OLS estimation).

**Example** invocation:

```plaintext```
bundle bCCE = panelCCE(y, X, 1)
```

Prints out the model estimates.
References


A Changelog

- July 2020: require 2018b (instead of 2018a) due to internal use of the lrcovar() function; update the documentation a little
- May 2018: initial release

B Example script

This script is reproduced here for convenience and is shipped with the package.

```bash
## Sample script for the CSDpanel package
include CSDpanel.gfn --force
set verbose off
open countrypan_nomiss.gdt --frompkg=CSDpanel
# Initial regression
panel inflation const urate # FE
# execute the test/CSD calculations
testCSD($uhat)
# compare the implementation of the Pesaran test with
# the native version
modtest --xdepend
print "----------------------"
# estimate the DK standard error(s)
matrix DriscKraay = robustCSD($model)
print "------ Estimate ------"
## estimate Pesaran's approach
# standard (usual, non-robust SEs)
panelCCE(inflation, deflist(const, urate))
# and robust (Arellano)
panelCCE(inflation, deflist(const, urate), 1)
```

4