The Stata Journal (2017)
17, Number 2, pp. 314–329

Regression clustering for panel-data models with fixed effects

Demetris Christodoulou
University of Sydney
Sydney, Australia
demetris.christodoulou@sydney.edu.au

Vasilis Sarafidis
Monash University
Melbourne, Australia
Vasilis.Sarafidis@monash.edu

Abstract. In this article, we describe the `xtregcluster` command, which implements the panel regression clustering approach developed by Sarafidis and Weber (2015, *Oxford Bulletin of Economics and Statistics* 77: 274–296). The method classifies individuals into clusters, so that within each cluster, the slope parameters are homogeneous and all intracluster heterogeneity is due to the standard two-way error-components structure. Because the clusters are heterogeneous, they do not share common parameters. The number of clusters and the optimal partition are determined by the clustering solution, which minimizes the total residual sum of squares of the model subject to a penalty function that strictly increases in the number of clusters. The method is available for linear short panel-data models and useful for exploring heterogeneity in the slope parameters when there is no a priori knowledge about parameter structures. It is also useful for empirically evaluating whether any normative classifications are justifiable from a statistical point of view.

Keywords: st0475, xtregcluster, panel data, parameter heterogeneity

1 Introduction

Standard panel-data analysis imposes the restriction that all individuals share the same slope coefficients, and any unobserved heterogeneity across individuals is attributed solely to the presence of individual-specific, time-invariant effects (that is, differential intercepts). This restriction can be difficult to justify, both theoretically and empirically (for example, Burnside [1996], Baltagi and Griffin [1997], Pesaran, Shin, and Smith [1999]).

The `xtregcluster` command implements the regression clustering approach developed by Sarafidis and Weber (2015), which groups individuals into distinct clusters. Within each cluster, the slope coefficients are homogeneous, and all intracluster heterogeneity is attributed to the two-way error-components structure. The clusters themselves are heterogeneous; that is, the slope coefficients are different across clusters.

Both the number of clusters and the optimal partition are treated as unknown and are determined from the data based on minimizing a model information criterion that is strongly consistent. That is, it estimates the true number of clusters with probability one as N grows for any T fixed. Therefore, the method is valid for panel datasets characterized by a large number of individuals and a short time-series length.

`xtregcluster` is useful for exploring slope parameter heterogeneity in the absence of a priori information regarding parameter structures. The algorithm can also confirm whether the restriction of slope parameter homogeneity is supported by the data. Moreover, it is useful for examining whether a priori classification of individual entities is optimal from a statistical point of view, such as industrial classifications (for example, North American Industry Classification System codes), risk classifications (for example, credit ratings), or arbitrarily imposed classification schemes (for example, univariate quantile classes).

2 A partially heterogeneous panel-data model

Consider the linear panel-data model

$$y_{i_\omega t} = \boldsymbol{\beta}'_\omega \mathbf{x}_{i_\omega t} + \epsilon_{i_\omega t} \tag{1}$$

where $y_{i_\omega t}$ denotes the observation of the dependent variable for the ith individual in cluster ω at time period t, $\mathbf{x}_{i_\omega t}$ is a $K \times 1$ vector of covariates, and $\boldsymbol{\beta}_\omega$ is a $K \times 1$ vector of fixed parameters that are common within clusters but vary across clusters. The default error term of the model is composite and subject to a one-way error-component structure:

$$\epsilon_{i_\omega t} = \eta_{i_\omega} + e_{i_\omega t} \tag{2}$$

The regressors are assumed to be exogenous with respect to the idiosyncratic error component, $e_{i_\omega t}$, but they can be endogenous with respect to η_{i_ω}. The vector, $\mathbf{x}_{i_\omega t}$, can include dynamic terms if they are exogenous. A two-way error-components structure can be easily implemented by specifying the factorial notation `i.`*timevar*. Hence, the error term structure becomes $\epsilon_{i_\omega t} = \eta_{i_\omega} + \tau_t + e_{i_\omega t}$.

In Stata, the estimation of the model described by (1) and (2) corresponds to the `xtreg, fe` command, except each cluster has its own regression structure such that $\omega = 1, 2, \ldots, \Omega_0$ with $i_\omega = 1, 2, \ldots, N_{\omega_0}$ and $t = 1, 2, \ldots, T$. The total number of individuals across clusters is $N = \sum_{\omega=1}^{\Omega_0} N_\omega$, and the total sample size is $N \times T$.

Note that the whole time series of a given individual entity belongs to a cluster, not a subset of observations of individuals. That is, an individual can be classified only in one cluster. The focus is the analysis of "short panels", where $N \gg T$ with $N \to \infty$ and T fixed (for unbalanced panels, the average $\overline{T}$ is fixed).

The key estimation problem is how to obtain estimates of the model's slope coefficients when the value of the true number of clusters, Ω_0, and membership of individuals to clusters are both unknown. Sarafidis and Weber (2015) develop a partitional clustering approach for estimating the true number of clusters, as well as the corresponding partition (see also Kaufman and Rousseeuw [2005] and Everitt et al. [2011]). Sarafidis and Weber (2015) show through analytical work and simulations that their proposed solution yields a strongly consistent estimate of Ω_0; that is, $\widehat{\Omega} \to \Omega_0$ with prob $\to 1$ as $N \to \infty$, for any fixed T.

3 Estimation algorithm

The optimal value of Ω—denoted as $\widehat{\Omega}$—and the corresponding partition are determined based on the following algorithm:

1. Specify some value for Ω, where $\Omega \leq \xi \leq \Omega_0$, with $\xi > 1$ and $\xi \in \mathbb{Z}^+$.
2. Randomize the panel identifiers.
3. Obtain an initial partition using one of the following three ways: i) a random classification based on the standard uniform distribution; ii) an a priori classification; and iii) a classification based on certain observed variables (characteristics) and obtained using the official Stata command `cluster kmeans`.
4. Reallocate the first individual to all remaining clusters, each time saving the value of the residual sum of squares (RSS) that arises for each cluster, RSS_ω. Assign the individual into the cluster that corresponds to the smallest value of the total RSS across all ω; that is, $\text{RSS} = \sum_{\omega=1}^{\Omega} \text{RSS}_\omega$.
5. Repeat step 4 for every other individual in the sample.
6. Repeat steps 3 and 4 until RSS cannot be reduced any further with some tolerance criterion.
7. Repeat steps 1 to 6 for different values of Ω. $\widehat{\Omega}$ is the value of Ω that minimizes the following model information criterion (MIC),

$$\text{MIC} = N \ln\left(\frac{\text{RSS}}{N\overline{T}}\right) + \Omega\theta_N \tag{3}$$

   where $\overline{T} = 1/N \sum_{i=1}^{N} T_i$ denotes the average time-series length for unbalanced panels. For panels with equal-length time series, labeled by Stata as strongly or weakly balanced panels, $\overline{T} = T$. The term $\Omega\theta_N$ is a required penalty because the minimum RSS is monotone decreasing in the number of clusters and will tend to overparameterize the model by allowing for more clusters than may actually exist. Essentially, the penalty provides a filter to ensure that the preferred clustering outcome partitions between clusters rather than within clusters. θ_N can take any value, provided $\lim_{N\to\infty} N^{-1}\theta_N = 0$ and $\lim_{N\to\infty} [\log\{\log(N)\}]^{-1}\theta_N = \infty$. The theoretical properties of the algorithm are discussed in Sarafidis and Weber (2015, sec. 4).

The rationale behind step 2 is to make sure the results are not dependent on the original ordering of the individuals. This is important because, in real data, numerical panel identifiers with a smaller value are often associated with early "entrants", whereas larger values can be associated with late "entrants" in the dataset (for example, panel identifiers may be associated with the age of a company). To the extent that this feature implies dependencies among individuals, one should randomize the order within our algorithm according to which individuals are reallocated to different clusters.

The total number of possible partitions is exponential in N. Therefore, it becomes infeasible to search over all possible partitions, even for relatively small values of N and Ω. Thus one should use different initial partitions (that is, iterate steps 3–7) before $\widehat{\Omega}$ is determined to avoid local minimums; we provide examples in section 5.

4 The xtregcluster command

As with all `xt` commands, `xtregcluster` requires that the data be `xtset`. It begins by obtaining an initial partition through a random uniform classification, a predetermined classification, or a classification based on certain observed variables. Then, it reclassifies individuals to clusters so that total RSS is minimized.

4.1 Syntax

`xtregcluster` *depvar indepvars* [*if*] [*in*] [*weight*] `,`
{`random` | `preclass(`*varname*`)` | `prevars(`*varlist* | `X` | `b)`} `omega(`*numlist*`)`
[`prevarsopt(`*kmeansopt*`)` `theta(#)` `seed(#)` `name(`*varname*`)` `iterate(#)`
`tolerance(#)` `nolog` `graph` `table`]

4.2 Options

`random` obtains the initial partition using random selection from the standard uniform distribution. `random`, `preclass()`, or `prevars()` is required.

`preclass(`*varname*`)` obtains the initial partition based on a predetermined classification, using a categorical variable. `omega()` is not allowed with `preclass(`*varname*`)`, because the preclassification determines the size of Ω. `random`, `preclass()`, or `prevars()` is required.

`prevars(`*varlist* | `X` | `b)` obtains the initial partition based on observed variables, using the official Stata command `cluster kmeans`. The observed variables can be part of the explanatory variables but not necessarily so. `prevars(X)` includes the whole set of regressors and is equivalent to specifying `prevars(`*indepvars*`)`. You may also combine `prevars(X` *varlist*`)`. `prevars(b)` uses the individual-specific estimated slopes of all *indepvars*. `random`, `preclass()`, or `prevars()` is required.

`omega(`*numlist*`)` specifies the numerical range of $\Omega > 1$. `omega()` is required with `random` and `prevars()` but is not allowed with `preclass()`. `omega()` takes integer values.

`prevarsopt(`*kmeansopt*`)` can be specified only with `prevars(`*varlist*`)`. *kmeansopt* takes all options from `cluster kmeans` with the exceptions of `name()`, `generate()`, and `k()`, which are already specified by the `omega()` option above.

`theta(#)` specifies the value of θ_N in the penalty function of (3) for overfitting Ω. The default is `theta(`$1/3\ln(N)+2/3\sqrt{N}$`)`, which is found to perform well by Sarafidis and Weber (2015) in their simulation study. `theta()` can take any other real argument. Other common values for θ_N are $\ln(N)$ and $\sqrt{N}$.

`seed(#)` sets the random-number seed for the entire program. The seed is relevant for randomizing the numerical panel identifiers. The seed is also relevant for the `random` method of obtaining the initial partition. The default is `seed(123)`.

`name(`*varname*`)` specifies the name prefix for the newly generated variables that identify the levels in the optimized partitions for each Ω. The default is `name(omega#)`, where # is a positive integer as specified in `omega()`.

`iterate(#)` specifies the maximum number of iterations for minimizing the total RSS, given `omega()`. The default is `iterate(100)`.

`tolerance(#)` specifies the tolerance for the convergence of total RSS, given `omega()`. The default is `tolerance(1e-6)`.

`nolog` suppresses the RSS iteration log.

`graph` gives a visual diagnostic with cluster-specific scatterplots of the observed dependent variables against the linear predictor together with superimposed linear fits for each ω corresponding to $\widehat{\Omega}$.

`table` prints a table of estimates of the slope coefficients for each ω corresponding to $\widehat{\Omega}$. That is, it reports results using cluster-specific fixed-effects (FE) regressions based on $\widehat{\Omega}$, for example, `xtreg if omega4==1, fe` for $\omega = 1$ and similarly for the remaining clusters, that is, $\omega = 2, \ldots, \widehat{\Omega}$.

4.3 Output and stored results

`xtregcluster` prints the method used to obtain the initial partition and its associated total RSS, labeled as `Iteration 0 Total RSS`. This RSS varies according to the size of Ω and how the initial partition was obtained. Next, the output prints the RSS at the end of every iteration up to convergence, followed by a report on the MIC for the specified value of Ω, contrasted with the MIC corresponding to $\Omega = 1$. The total RSS reported for $\Omega = 1$ is obtained from the pooled FE regression, and it is the same regardless of the choice of the initial partition or the value of Ω. The report ends with a recommendation of whether to proceed either with the pooled FE regression, `xtreg, fe`, or with cluster-specific FE regressions.

`xtregcluster` stores the following in `e()`:

Scalars

`e(N)`	N panels in estimation
`e(T)`	balanced T or average $\overline{T}_i$
`e(NT)`	NT or $N\overline{T}_i$
`e(rss_pool)`	pooled RSS
`e(mic_pool)`	pooled MIC
`e(omega_opt)`	optimal $\widehat{\Omega}$ given specified range
`e(theta)`	specified θ_N
`e(rss_tot`Ω`)`	total RSS_Ω for every Ω
`e(mic_tot`Ω`)`	MIC_Ω for every Ω

Macros

`e(cmdline)`	estimator
`e(name_opt)`	optimal partition variable

Matrices

`e(rss`Ω`)`	total RSS_Ω at every iteration for every Ω

`xtregcluster` generates indicator variables with common name prefixes, as provided in `name()`, for every Ω specified in `omega()`. For instance, the `name(om)` and `omega(2/3)` options will generate three indicator variables with names `om2` and `om3`. The stored result, `e(name_opt)`, takes the name of one of these variables corresponding to the smallest MIC_Ω, with the optimal value of $\widehat{\Omega}$ stored in `e(omega_opt)`. The command also returns all total RSS_Ω and MIC_Ω for every Ω specified in `omega()`, plus the iteration logs in matrix form, for example, `e(rss_tot2)`, `e(mic_tot2)`, `e(rss2)` and `e(rss_tot3)`, `e(mic_tot3)`, `e(rss3)`.

Stored results can be used for subsequent analysis. For example, alternative penalties θ_N for the calculation of the MIC may be specified either through the `theta()` option or manually calculated using stored results. The latter approach is recommended, given the computational cost in running `xtregcluster` again. For example, one may wish to check the sensitivity of the suggested optimal partition using the less severe penalty, $\ln(N)$, or the stricter penalty, $\sqrt{N}$:

```
. display e(N) * ln(e(rss_tot)/e(NT)) + e(omega_opt) * ln(e(N))
. display e(N) * ln(e(rss_tot)/e(NT)) + e(omega_opt) * sqrt(e(N))
```

Asymptotically, that is, for large N, the choice of θ_N is immaterial, although in small samples it can possibly give substantially different results.

The `table` option contrasts estimation results of the heterogeneous slopes for each $\omega = 1, 2, \ldots, \widehat{\Omega}$. This `table` does not report standard errors, because clustering of individuals based on minimizing RSS implies that the usual formula for obtaining standard errors is no longer valid. However, the next section explains how to reproduce this table and obtain bootstrapped standard errors that are valid for inference. We refrain from making bootstrapping a default treatment, because it is computationally intensive and requires a large number of repetitions to produce reliable estimates for standard errors.

The `graph` option produces a single graph with overlaid scatterplots and linear fits of all heterogeneous slopes across the clustered individuals for every $\omega = 1, 2, \ldots, \widehat{\Omega}$. The `graph` applies a colored scheme to readily assist the visual distinction among the clusters.

Section 5 demonstrates how to reproduce this graph in monochrome for publication purposes and also how to plot each cluster in a separate graph against the pooled slope, which is useful for large $\widehat{\Omega}$.

4.4 Practical considerations

`xtregcluster` is computationally intensive for large N and large Ω. Therefore, one should reduce the dimension of the data to the required minimum before executing `xtregcluster`, both in terms of columns (variables) and rows (observations). For example, if the model contains the variables y, x_1, and x_2, and the dataset in memory contains many more variables, then you should reduce as follows:

```
. preserve
. keep id date y x1 x2
. keep if !missing(panelvar,timevar,y,x1,x2)
. xtregcluster y x1 x2, random
. restore
```

`xtregcluster` is relevant for short panel data where individuals have at least two observations, that is, $T_i > 1$. If an individual has only $T_i = 1$, it cannot be classified meaningfully into a cluster using an FE regression. If your dataset contains such cases, `xtregcluster` will issue a warning that individuals with $T_i = 1$ are excluded from estimation.

The initial partition can be obtained using estimates of the individual-specific slope coefficients using `prevars(b)`. This method is feasible only for panel datasets where all individuals have a sufficient number of observations for estimating the individual-specific slopes; that is, $T_i \geq k + 1$. If there is even a single individual with $T_i < k + 1$, `xtregcluster` will issue the following error:

```
. xtregcluster y x1 x2, prevars(b)
Some panels have Ti < k+1. Choose an alternative initial partition,
or qualify the sample to panels with enough observations.
insufficient observations
r(2001);
```

If the user insists on using `prevars(b)` for obtaining the initial partition, the entire analysis must be restricted to individuals with enough observations:

```
. bysort id: generate Ti = _N
. xtregcluster y x1 x2 x3 x4 if Ti>=5, prevars(b)
```

This approach is not recommended for models with a large number of explanatory variables, because many individual units may be dropped—resulting in a great loss of degrees of freedom.

Because the clustering algorithm used by `xtregcluster` aims to minimize the within-cluster RSS, the properties of the estimated standard errors obtained using standard formulas are no longer known. Therefore, once $\widehat{\Omega}$ and the corresponding partition have been determined, we recommend computing standard errors using the method

of bootstrapping, which provides estimates of the distribution one would get if one were able to draw repeated samples of N points from the unknown true distribution (Sarafidis and Weber 2015). Following the execution of `xtregcluster`, one can produce bootstrapped standard errors as follows:

```
. local optomega = e(omega_opt)
. local optname  = e(name_opt)
. quietly forvalues i = 1/`optomega´ {
2. xtreg y x1 x2 if `optname´==`i´, vce(bootstrap, reps(1000) nodots)
3. estimates store omega`i´
4. }
. estimates table omega*, se stats(N_g Tbar N r2_w rho corr)
```

We recommend the minimum of 1,000 repetitions, preferably even more, to obtain reliably precise estimates. The statistics in `stats()` report key diagnostics for `xtreg, fe` and are described in [XT] **xtreg**.

5 Application

To demonstrate the application of `xtregcluster`, we estimate a translog production function for Spanish dairy farms. `help xtregcluster` provides another application using Stata's `productivity.dta` on U.S. public capital productivity.

5.1 Dairy farm production

Consider a translog functional form for modeling Spanish dairy farm production output. `dairy.csv` is obtained from William Greene's webpage on panel-data econometrics and contains observations on output (cow milk) and several inputs, such as the number of cows used, size of land, labor, and feed.[1] The panel structure is balanced with all $N = 247$ farms observed over the same period of 1993–1998:

```
. import delimited dairy.csv
(28 vars, 1,482 obs)
. xtset farm year
       panel variable:  farm (strongly balanced)
        time variable:  year, 93 to 98
                delta:  1 unit
```

Note that the `xtregcluster` command is also useful for hierarchical datasets with repeated observations over higher-level cross-sectional units, absent of a time variable. For example, if this dataset contained repeated observations for many farms within counties, then it would suffice to `xtset` using only the lower-level panel identifier:

```
. xtset farm
       panel variable:  farm (balanced)
```

1. We thank William Greene for giving us permission to use the data, which are available online at http://people.stern.nyu.edu/wgreene/Econometrics/PanelDataSets.htm.

The variable `yit` denotes the log of the demeaned farm output, while variables $\texttt{x}_k$ and $\texttt{x}_{k\ell}$ are the regressors used in the translog function, where x_k for $k = 1, \ldots, K$ denotes the log of the kth input, that is, the demeaned number of cows, land size, labor, and feed, whereas $\texttt{x}_{k\ell} = \texttt{x}_k\texttt{x}_\ell$.

We start by obtaining the initial partition based on a uniform random classification for $\Omega = 2, 3$:

```
. xtregcluster yit x1-x34, random omega(2/3)
Initial partition via randomized classification and seed 123
Omega = 2
Iteration 0:   Total RSS =           7.654268
Iteration 1:   Total RSS =           6.492979
Iteration 2:   Total RSS =           6.353171
Iteration 3:   Total RSS =           6.316846
Iteration 4:   Total RSS =           6.316779
Iteration 5:   Total RSS =           6.316779
Omega = 3
Iteration 0:   Total RSS =           7.579115
Iteration 1:   Total RSS =           5.759485
Iteration 2:   Total RSS =           5.603187
Iteration 3:   Total RSS =           5.543064
Iteration 4:   Total RSS =           5.502915
Iteration 5:   Total RSS =           5.469064
Iteration 6:   Total RSS =           5.461379
Iteration 7:   Total RSS =           5.461363
Iteration 8:   Total RSS =           5.461215
Iteration 9:   Total RSS =           5.461215
```

Omega	Total RSS	MIC
1	7.887	-1280.962
2	6.317	-1323.483
3	5.461	-1347.117

```
Proceed with xtreg if omega3==`i´,fe  where `i´=1,2,3
```

The output suggests that there exist at least three distinct clusters for these data, given the model. Notice how two new variables have been created to indicate cluster membership of all individuals for the two values of Ω, with names `omega2` and `omega3`. The name `omega` is the default prefix indicating the size of Ω as specified in `omega()`.

Because the value of Ω corresponding to the minimum value of MIC equals the maximum value specified in `omega()`, it is necessary to explore larger values, for example, `omega(2/10)`. To conserve space, we suppress the iteration logs by specifying the `nolog` option. We may also specify a new name prefix:[2]

2. Because this may take awhile, the user may actually wish to see the iteration log working in action.

```
. xtregcluster yit x1-x34, random omega(2/10) name(om) nolog
Initial partition via randomized classification and seed 123
```

Omega	Total RSS	MIC
1	7.887	-1280.962
2	6.317	-1323.483
3	5.461	-1347.117
4	4.999	-1356.638
5	4.626	-1363.494
6	4.483	-1358.955
7	4.001	-1374.719
8	3.820	-1373.825
9	3.781	-1364.030
10	3.580	-1365.253

```
Proceed with xtreg if om7==`i´,fe  where `i´=1,2,3,4,5,6,7
```

The output suggests the optimal value of Ω is 7. However, because the results are based on a particular initial partition—uniform random selection with seed 123—we recommend trying alternative initial partitions. As an example, the initial partition can be determined based on the regressors, $\mathbf{X}$, using the `prevars(X)` option:

```
. xtregcluster yit x1-x34, prevars(X) omega(2/10) name(omX) nolog
Initial partition via the variation in x1 x2 x3 x4 x11 x22 x33 x44 x12
> x13 x14 x23 x24 x34 and seed 123
```

Omega	Total RSS	MIC
1	7.887	-1280.962
2	6.155	-1329.891
3	5.658	-1338.374
4	4.976	-1357.779
5	4.684	-1360.402
6	4.374	-1364.992
7	4.132	-1366.781
8	3.863	-1371.059
9	3.805	-1362.498
10	3.574	-1365.671

```
Proceed with xtreg if omX8==`i´,fe  where `i´=1,2,3,4,5,6,7,8
```

The results now indicate that the optimal value of Ω is 8. Because the value of MIC_8 under `prevars(X)` is larger (-1371.059) than the value of MIC_7 under `random` initial partition (-1374.719), we set $\widehat{\Omega} = 7$. This example shows how important it is to experiment between different initial partitions. It is also wise to try different random seed numbers, using the `seed()` option.

To see how many individuals are assigned in each cluster, `tabulate` the variable that holds the optimal partition, as estimated above using the random initial partition with $\widehat{\Omega} = 7$. Remember to qualify the sample only to one observation per individual:

```
. egen tag = tag(farm)
. tabulate `e(name_opt)´ if tag
```

omX8	Freq.	Percent	Cum.
1	33	13.36	13.36
2	29	11.74	25.10
3	18	7.29	32.39
4	28	11.34	43.72
5	28	11.34	55.06
6	24	9.72	64.78
7	35	14.17	78.95
8	52	21.05	100.00
Total	247	100.00	

The `table` and `graph` options provide additional information about the final model. These options can be entered from the outset when specifying the numerical list in `omega()`. Then, a `table` of estimates and a `graph` of scatterplots with linear fits are displayed for $\omega = 1, 2, \ldots, \widehat{\Omega}$. If the user forgets to enter these options, one can repeat `xtregcluster` only for $\widehat{\Omega}$:

```
. drop om7
. xtregcluster yit x1-x34, random omega(7) name(om) nolog table graph
Initial partition via randomized classification and seed 123
```

Omega	Total RSS	MIC
1	7.887	-1280.962
7	4.001	-1374.719

```
Proceed with xtreg if om7==`i´,fe  where `i´=1,2,3,4,5,6,7
Table: Panel data fixed effects estimates by omega
```

Variable	om7_1	om7_2	om7_3	om7_4	om7_5
x1	0.681	0.652	1.452	0.835	0.666
x2	-0.130	0.117	-0.173	0.251	0.413
x3	-0.416	-0.057	0.071	0.025	0.745
x4	0.359	0.219	0.197	0.096	0.397
x11	-0.406	-0.082	2.267	-2.360	-1.960
x22	-0.909	0.052	1.190	-0.328	0.050
x33	4.581	-0.924	-0.680	0.386	1.531
x44	0.038	0.014	-0.050	-1.233	0.319
x12	1.153	-0.220	-1.059	0.083	0.549
x13	-1.369	0.951	-0.575	-0.901	-0.476
x14	0.197	0.181	-0.159	1.450	0.104
x23	-0.590	0.571	0.707	-0.281	-1.388
x24	-0.093	0.193	-0.040	-0.222	-0.102
x34	-0.081	-0.761	0.247	0.727	0.304
_cons	11.493	11.515	11.500	11.642	11.497
N_g	31.00	50.00	24.00	24.00	32.00
Tbar	6.00	6.00	6.00	6.00	6.00
N	186	300	144	144	192
r2_w	0.92	0.89	0.90	0.90	0.91
rho	0.97	0.94	0.90	0.92	0.98
corr	0.07	0.22	-0.35	0.09	-0.81

Variable	om7_6	om7_7	Pooled
x1	0.550	0.507	0.669
x2	-0.103	0.228	0.035
x3	-0.186	0.090	0.013
x4	0.559	0.478	0.378
x11	0.912	-0.403	0.220
x22	0.320	0.311	-0.054
x33	2.722	-0.257	-0.213
x44	-0.130	0.309	0.105
x12	-0.279	1.069	0.008
x13	-0.487	0.377	0.023
x14	-0.325	-0.037	-0.093
x23	-0.327	-0.014	0.031
x24	0.119	-0.644	-0.018
x34	0.853	-0.193	0.021
_cons	11.398	11.468	11.565
N_g	43.00	43.00	247.00
Tbar	6.00	6.00	6.00
N	258	258	1482
r2_w	0.94	0.94	0.84
rho	0.94	0.94	0.70
corr	-0.29	-0.40	0.15

Note: For a description of model diagnostics see stored results in xtreg,fe.

The `graph` option produces a graph of the heterogeneous slopes across the clustered individuals, but applies a colored scheme to readily assist the visual distinction among the clusters, and also overlays all plots into one graph. Given $\widehat{\Omega} = 7$, `graph` will overlay

seven scatterplots plus seven linear fits in a multicolor visual. Hence, the clusters may not be entirely discernible. To manually reproduce a similar graph, but in monochrome scheme for printing, while keeping each plot in a separate graph, execute the following routine:

```
. * Pooled fitted values
. quietly xtreg yit x1-x34, fe
. predict xb, xb
. * Fitted values by cluster
. forvalues i = 1/7 {
  2.    quietly xtreg yit x1-x34 if om7==`i´, fe
  3.    estimates store om`i´
  4.    quietly predict xb`i´ if om7==`i´, xb
  5.    twoway (lfit yit xb, lwidth(*3) lpattern(solid) lcolor(gs8))
>        (scatter yit xb`i´ if om7==`i´, msymbol(oh) mlwidth(*.3) mcolor(gs0))
>        (lfit yit xb`i´ if om7==`i´, lwidth(*2.25) lpattern(dash) lcolor(gs0)),
>        aspect(1) ysize(1) xsize(1) scheme(sj) legend(off)
>        ytitle("Log of milk production (output)")
>        title("{&omega} = `i´", ring(0) pos(11) margin(medium))
>        name(g`i´, replace)
  6. }
. graph combine g1 g2 g3 g4 g5 g6 g7, row(2)
>       ysize(2) xsize(4) imargin(small) scheme(sj)
```

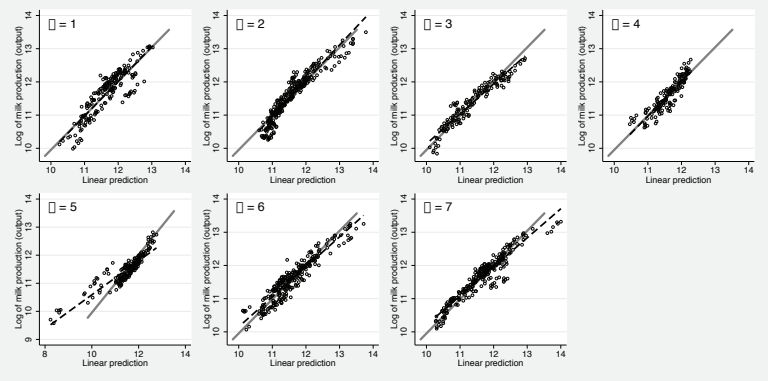

Figure 1. Heterogeneous dairy production functions

The gray solid line in figure 1 gives the pooled FE linear fit, which is the same across all plots. The dashed lines provide the linear fit for each ω. The linear fit for individuals classified in $\omega = 2$ has a slightly steeper slope, whereas the linear fit for individuals classified in $\omega = 3, 5, 6, 7$ has less steep slopes than the pooled FE model. The individuals in $\omega = 1, 4$ have virtually identical slopes as the pooled FE, but this does not necessarily mean that the estimated slope coefficients for $\omega = 1, 4$ are close to those of the pooled FE model. The fitted value, $\widehat{y}_{it}$, is a linear combination of all explanatory variables times their estimated slopes (that is, the linear predictor). Hence, it is still possible to have similar values of $\widehat{y}_{it}$ but weighted differently by the cluster-

specific slope coefficients. Indeed, as shown in the output from the `table` option above, the cluster-specific coefficients for $\omega = 1, 4$ are considerably different from each other and, by comparison, from the pooled FE model.

To enhance interpretation in such cases, one may want to obtain cluster-specific plots for a given explanatory variable, rather than the linear combination of all of them. To achieve this, for every cluster, we can project the residuals obtained from a regression of the dependent variable on all other remaining independent variables onto the residuals obtained from a regression of the independent variable of interest to all other independent variables. A regression of the two predicted residuals gives the same slope coefficient as reported by the `table` option.[3] As a demonstration, we examine the slope heterogeneity in `x2` (the production input of land). From the `table` output above, the pooled FE slope coefficient is shown to be close to zero, yet there seems to be substantial variation in the slopes for $\omega = 1, 2, \ldots, 7$. We can visualize the differential slopes as follows:

```
. * First project x2 for the pooled sample
. xtreg yit x1 x3-x34, fe
  (output omitted)
. predict e1_x2_pool, e
. xtreg x2  x1 x3-x34, fe
  (output omitted)
. predict e2_x2_pool, e
. * Then project x2 for each omega
. quietly forvalues i = 1/7 {
  2.    xtreg yit x1 x3-x34 if om7==`i´, fe
  3.    predict e1_x2_`i´ if om7==`i´, e
  4.    xtreg x2 x1 x3-x34 if om7==`i´, fe
  5.    predict e2_x2_`i´ if om7==`i´, e
  6.    twoway (lfit e1_x2_pool e2_x2_pool, range(-.4 .4) lw(*2) lp(solid) lc(gs8))
>           (scatter e1_x2_`i´ e2_x2_`i´ if om7==`i´, ms(oh) mlw(*.3) mc(gs2))
>           (lfit e1_x2_`i´ e2_x2_`i´ if om7==`i´, lw(*1.5) lp(dash) lc(gs0)),
>           title("{&omega} = `i´", ring(0) pos(11) margin(medium) size(*1.25))
>           ytitle("yit projection") xtitle("x2 projection")
>           legend(off) scheme(sj) name(x2_`i´,replace)
  7. }
  (output omitted)
. graph combine x2_1 x2_2 x2_3 x2_4 x2_5 x2_6 x2_7, row(2) imargin(small)
>              ysize(2) xsize(5) scheme(sj)
```

It is clear from figure 2 that there is considerable heterogeneity in the cluster-specific coefficients of `x2`, even though the pooled FE slope is quite flat. The user may repeat the same process for every other variable of interest.

3. This result follows from the Frisch–Waugh–Lovell theorem.

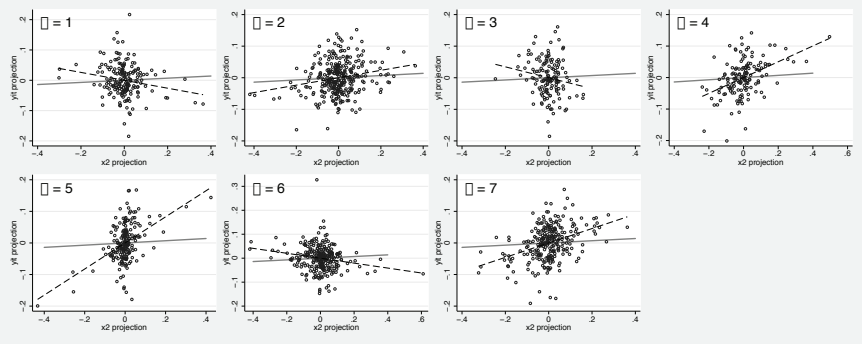

Figure 2. Heterogeneous slope coefficient for `x2` (production input of land)

Note that for `dairy.csv`, we cannot obtain the initial partition based on estimates of the individual-specific slopes using the `prevars(b)` option, because the number of degrees of freedom required to estimate an individual-specific regression exceeds the available observations; that is, $k + 1 = 16 > T = 6$. This is a balanced dataset and all individuals have $T = 6$. If one tries to estimate this model, an error message prompts the user to take a different course of action:

```
. xtregcluster y x1-x34, prevars(b) omega(2/10) name(omb)
Some panels have Ti < k+1. Choose an alternative initial partition,
or qualify the sample to panels with enough observations
insufficient observations
r(2001);
```

Lastly, one may also wish to explore a two-way error structure including time-specific FE. This can be easily implemented by specifying `i.year` as part of *indepvars* and repeating the entire analysis above.

6 Conclusions

`xtregcluster` is useful for discovering potential heterogeneous clusters in linear short panel-data models with FE, similar to an exploratory data analysis approach. It can also be used to assess the validity of the assumption of slope homogeneity or of normatively imposed preclassifications.

7 Acknowledgments

We acknowledge Karl Keesman's input during the development of the first draft of this program. We also acknowledge useful comments from participants at the 2015 Oceania Stata Users Group meeting at Canberra.

8 References

Baltagi, B. H., and J. M. Griffin. 1997. Pooled estimators vs. their heterogeneous counterparts in the context of dynamic demand for gasoline. *Journal of Econometrics* 77: 303–327.

Burnside, C. 1996. Production function regressions, returns to scale, and externalities. *Journal of Monetary Economics* 37: 177–201.

Everitt, B. S., S. Landau, M. Leese, and D. Stahl. 2011. *Cluster Analysis*. 5th ed. Chichester, UK: Wiley.

Kaufman, L., and P. J. Rousseeuw. 2005. *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ: Wiley.

Pesaran, M. H., Y. Shin, and R. P. Smith. 1999. Pooled mean group estimation of dynamic heterogeneous panels. *Journal of the American Statistical Association* 94: 621–634.

Sarafidis, V., and N. Weber. 2015. A partially heterogeneous framework for analyzing panel data. *Oxford Bulletin of Economics and Statistics* 77: 274–296.

About the authors

Demetris Christodoulou is at the University of Sydney Business School and General Convenor of the research network Methodological and Empirical Advances in Financial Analysis.

Vasilis Sarafidis is at the Department of Econometrics and Business Statistics at Monash University, and a founding member of Methodological and Empirical Advances in Financial Analysis.