

vcmway: A one-stop solution for robust inference with multi-way clustering

Ariel Gu

Newcastle Business School
Northumbria University
Newcastle, UK
ariel.gu@northumbria.ac.uk

Hong Il Yoo

Durham University Business School
Durham University
Durham, UK
h.i.yoo@durham.ac.uk

Abstract. Most Stata commands allow `cluster(varname)` as an option, popularizing the use of standard errors that are robust to one-way clustering. But when it comes to adjusting standard errors for multi-way clustering, there is no solution that is as widely applicable. While several user-written packages support multi-way clustering, each package is compatible with only a subset of models that Stata’s ever-expanding library of commands allows the researcher to estimate. We introduce a command `vcmway` that provides a one-stop solution for multi-way clustering. `vcmway` works with any estimation command that allows `cluster(varname)` as an option, and adjusts standard errors, individual significance statistics and confidence intervals in output tables for multi-way clustering in specified dimensions. The covariance matrix used in making this adjustment is stored in `e(V)`, meaning that any subsequent call to postestimation commands that use `e(V)` as input (e.g. `test` and `margins`) will also produce results that are robust to multi-way clustering.

Keywords: `st0001`, `vcmway`, `ivreg2`, `cmgreg`, `reghdfe`, `boottest`, two-way clustering, multi-way clustering

1 Introduction

The use of one-way clustered standard errors in empirical research is now commonplace. In comparison with usual heteroskedasticity-robust standard errors which assume the independence of regression errors across all observations, clustered standard errors offer an extra layer of robustness by allowing for correlations across observations which belong to the same cluster. In the analysis of a labor force panel survey, for example, repeated observations on an individual may form a cluster, and standard errors may be clustered at the individual level to attain robustness to within-individual correlations over time. Cameron and Miller (2015) provide a masterful review of the methods and literature.

In recent years, the use of multi-way clustered standard errors has also received growing attention. This extension robustifies one-way clustered standard errors further, by allowing for other clusters within which regression errors may be correlated. For example, in the analysis of panel data on bilateral trade flows, two-way clustering may make standard errors robust to within-country correlation as well as within-country pair correlation over time (Cameron et al., 2011). In labor economics, Dube et al. (2010, 2016) analyze earnings and employment data on pairs of US counties that share state

border segments, and apply two-way clustering to account for within-state correlations and within-border segment correlations. In the analysis of firm-level panel data in finance, Thompson (2011) has pioneered the use of two-way clustering that accounts for within-firm correlations over time and within-time period correlations across firms. Gow et al. (2010) show that many of well-known findings in the accounting literature may be sensitive to the use of two-way clustering that similarly accounts for within-firm and within-time period correlations.

Most Stata commands allow `cluster(varname)` as an option, facilitating and popularizing the use of one-way clustered standard errors. No similar one-stop solution is available for multi-way clustering. Popular user-written commands, such as `cmgreg` (Gelbach and Miller, 2009), `ivreg2` (Baum et al., 2010), and `reghdfe` (Correia, 2017), support multi-way clustering but only in the context of specific models (e.g. `cmgreg` in the context of `regress`; `ivreg2` in that of `ivregress`; and `reghdfe` in that of `areg`, `xtreg`, `fe` and `xtivreg`, `fe`). The versatile `boottest` package (Roodman et al., 2019) supports multi-way clustering with all aforementioned models as well as non-linear models utilizing `ml maximize`, but it still leaves out many commands that allow `cluster(varname)` as an option, such as `xtreg`, `re` and ever-expanding library of user-written commands. Moreover, `boottest` does not store multi-way clustered covariance matrices in `e(V)`, meaning that it does not make adjustment for multi-way clustering to postestimation commands such as `margins` and `predictnl`.¹ In general, to apply robust inference with multi-way clustering in a new context, Stata users may need to conduct a fresh search for a relevant command which may not always exist.

This article describes a simple approach to obtain almost any Stata command's output with multi-way clustered standard errors, and a new command `vcemway` which automates the process. The main idea is to use `ereturn repost` to replace an active output's covariance matrix, stored in `e(V)`, with its multi-way clustered counterpart that has been computed by the researcher. Computing a multi-way clustered covariance matrix is relatively straightforward, when the researcher can use `cluster(varname)` to adjust for one-way clustering in each dimension of interest separately.

1. The `boottest` package does not return an asymptotic covariance matrix in `e(V)` because it is primarily intended as a tool to implement a type of percentile-t bootstrapping. Percentile-t methods compute p-values and confidence sets based on the simulated distributions of relevant test statistics across bootstrapped samples. A large body of Monte Carlo studies suggest that in finite samples, percentile-t methods may produce p-values and confidence sets which are more accurate than their counterparts derived using asymptotic distributions. The researcher, however, needs to trade off the ease of use for this potential benefit. For example, `boottest` is incompatible with widely used postestimation commands as mentioned earlier, and it also displays results for different hypotheses in separate blocks of output in a similar manner to Stata's native `test`. The latter feature may make it difficult to obtain an overview of the results, when the model includes a large number of coefficients and the researcher would like to test the individual significance of each coefficient. In addition, high computational costs may make the use of percentile-t bootstrapping unattractive when the researcher is working with a large number of hypotheses, a large sample size, and a non-linear model which requires numerical methods for estimation. While `boottest` comes with a parallel command `waldtest` that computes asymptotic p-values and confidence intervals using analytic formulas without bootstrapping, `waldtest` also displays results for each hypothesis in a separate block of output, and returns no `e(V)` that postestimation commands can access.

The command `vcemway` provides a one-stop solution for multi-way clustering in Stata, comparable to what option `cluster(varname)` already provides for one-way clustering. Specifically, `vcemway` works with any estimation command that allows `cluster(varname)` as an option. It has an intuitive syntax diagram, and adjusts standard errors, individual significance statistics and confidence intervals in output tables for multi-way clustering in specified dimensions. The covariance matrix used in making this adjustment overwrites the estimation command's $\mathbf{e}(V)$, meaning that any subsequent call to postestimation commands that use $\mathbf{e}(V)$ as input (e.g. `test`, `margins` and `predictnl`) will also produce results that are robust to multi-way clustering.

2 Adjusting for Multi-way Clustering

To facilitate discussion, suppose that the researcher is interested in running a linear regression of outcome y on two regressors X and Z , by executing `regress y X Z`. She is also interested in adjusting standard errors for clustering in m non-nested dimensions, identified by variables `id1`, `id2`, \dots , `id m` respectively. Let \mathbb{M} denote a set of all $2^m - 1$ distinct combinations of the m cluster variable names, including singletons. For example, when $m = 2$, there are $2^2 - 1 = 3$ elements in \mathbb{M} : `id1`, `id2`, and `(id1, id2)`. When $m = 3$, there are $2^3 - 1 = 7$ elements in \mathbb{M} : `id1`, `id2`, `id3`, `(id1, id2)`, `(id1, id3)`, `(id2, id3)`, and `(id1, id2, id3)`. Extensions to $m \geq 4$ are straightforward.

Cameron et al. (2011) show that an asymptotically valid m -way clustered covariance matrix, V_{mway} , may be conveniently computed in Stata and other software packages that allow for one-way clustering. Specifically, V_{mway} can be obtained by combining $2^m - 1$ one-way clustered covariance matrices as follows

$$V_{mway} = \sum_{g \in \mathbb{M}} (-1)^{1+|g|} V_g \quad (1)$$

where V_g is a covariance matrix adjusted for one-way clustering in “groups” formed by variable(s) in $g \in \mathbb{M}$, and $|g|$ is the number of variable(s) listed in g . The “groups” in this context are defined in the sense of Stata's `egen` function `group(varlist)`. For example, if g involves only one variable, say $g = \text{id1}$, the groups coincide with the clusters identified by `id1`; executing `regress y X Z, cluster(id1)` stores V_g in $\mathbf{e}(V)$ of Stata's `ereturn` results. If g involves two or more variables, say $g = (\text{id1}, \text{id2})$, the groups can be identified by a new variable `id1_id2` generated using `egen id1_id2 = group(id1 id2)`; executing `regress y X Z, cluster(id1_id2)` stores V_g in $\mathbf{e}(V)$.

To see the algebraic structure of formula (1) more clearly, it is useful to examine two-way ($m = 2$) and three-way ($m = 3$) clustering in detail. A two-way clustered covariance matrix is given by

$$V_{mway} = V_{\text{id1}} + V_{\text{id2}} - V_{(\text{id1}, \text{id2})} \quad (2)$$

and a three-way clustered covariance matrix is given by

$$V_{mway} = V_{\text{id1}} + V_{\text{id2}} + V_{\text{id3}} - V_{(\text{id1}, \text{id2})} - V_{(\text{id1}, \text{id3})} - V_{(\text{id2}, \text{id3})} + V_{(\text{id1}, \text{id2}, \text{id3})}. \quad (3)$$

As these examples illustrate, the multiplication factor $(-1)^{1+|g|}$ in (1) means that V_g is added to the sum when g involves an odd number of variable(s), and subtracted from the sum when g involves an even number of variables.

V_{mway} can be used to construct Wald statistics in the usual manner to draw inferences robust to m -way clustering. To construct t statistics, the researcher can obtain m -way clustered standard errors by taking the square roots of the diagonal entries in V_{mway} . Let n_c denote the number of clusters identified by variable $c \in \{\text{id1}, \text{id2}, \dots, \text{idm}\}$, i.e. the number of distinct values in variable c , and G denote the minimum of these m cluster sizes, $G = \min(n_{\text{id1}}, n_{\text{id2}}, \dots, n_{\text{idm}})$. As G grows arbitrarily large, the usual asymptotic distributions apply: a Wald statistic testing q restrictions is approximately $\chi^2(q)$ -distributed, and a t statistic is approximately standard normal. When G is small, there is no guaranteed method to improve finite sample inference. One useful starting point would be to conduct F tests using $F_{q,G-1}$ critical values in lieu of the asymptotic Wald tests using $\chi^2(q)$ critical values, and t tests using t_{G-1} critical values in lieu of the standard normal critical values (Cameron et al., 2011; Cameron and Miller, 2015).

The results of Cameron et al. (2011) allow the researcher to write a relatively simple Stata program to robustify test statistics to m -way clustering. As a minimal example, consider the following program, `mymway`, that adjusts for two-way clustering on `id1` and `id2`.

```

.           program define mymway, eclass
1.                 quietly regress y X Z, cluster(id1)
2.                 matrix V_id1 = e(V)
3.                 quietly regress y X Z, cluster(id2)
4.                 matrix V_id2 = e(V)
5.                 quietly regress y X Z, cluster(id1_id2)
6.                 matrix V_id1_id2 = e(V)
7.                 matrix V_mway = V_id1 + V_id2 - V_id1_id2
8.                 ereturn repost V = V_mway
9.                 ereturn scalar df_r = .
10.                regress
11.                end

```

Lines 1-6 estimate the linear regression model of interest accounting for one-way clustering on each $g \in \mathbb{M}$, and create Stata matrices to store each V_g . Line 7 combines the three stored matrices to compute a m -way clustered covariance matrix `V_mway` using formula (1), or more directly (2). Line 8 is an important step that replaces the active covariance matrix `e(V)` in Stata's memory with `V_mway`; this allows the researcher to robustify test statistics produced by all existing postestimation commands such as `test`, `margins` and `predictnl` to m -way clustering, without having to modify or rewrite such programs. Line 9 effectively sets the residual degrees of freedom to ∞ , thereby requesting the use of asymptotic t tests and Wald tests; replacing `df_r = .` with `df_r = #` would request the use of t and F tests based on $t_{\#}$ and $F_{q,\#}$ critical values, where $G - 1$ mentioned in the preceding paragraph is one possible choice for integer $\#$. Finally, line 10 reports the output of `regress` that has been modified to include m -way clustered standard errors as well as the corresponding asymptotic t statistics (or "z" statistics in Stata's parlance) and confidence intervals. Once defined, the program can be executed

by entering `mymway` at Stata's command prompt.

While generalizing the minimal example to accommodate other estimation commands and clustering in $m \geq 3$ dimensions is seemingly straightforward, there are several implementation issues that deserve close attention. First, repeatedly estimating the same model $2^m - 1$ times may require a substantial amount of computer time, especially when the sample size is large or the estimation command of interest solves a complicated numerical optimization task. In those cases where postestimation command `predict` can be used to compute observation-level residuals (e.g. after `regress`) or scores (e.g. after `ml`, `maximize`), Stata's programming command `_robust` can provide a timesaver.² Using the observation-level residuals or scores, `_robust` can compute a covariance matrix adjusted for one-way clustering on `varname`, even when the preceding estimation run did not specify `cluster(varname)` as an option. This allows the researcher to execute the time-consuming step of estimating the model only once, and compute each V_g subsequently via a simple algebraic task.

Second, the researcher should be on alert for the consequences of missing values in cluster identifiers. Continuing with the minimal example above, suppose that `id1` has missing values for one set of observations, and `id2` has missing values for another set of observations, meaning their group variable `id1_id2` has missing values for the union of the two sets. Then, each instance of `regress` in lines 1, 3 and 5 will use a different estimation sample, making the covariance matrix `V_mway` in line 7 invalid. A similar problem arises for `_robust`, as each one-way clustered covariance matrix will be computed over a different set of observations. Stata command `markout` makes it easy to define an estimation sample that excludes observations with missing values in any cluster identifier.

Third, using a one-way clustered covariance matrix produced by Stata's built-in `cluster(varname)` as V_g is equivalent to adopting a particular small-cluster correction factor. As Cameron et al. (2011) point out, when computing V_g , Stata multiplies the underlying large sample covariance formula by $[n_g/(n_g - 1)] \times [(N - 1)/(N - \#)]$ to mitigate small sample bias, where n_g is the number of clusters or groups identified by $g \in \mathbb{M}$, N is the number of observations in the estimation sample, and $\#$ is the number of estimated coefficients for some commands (e.g. `regress`) and 1 for others (e.g. `ml`, `maximize`). Our minimal example and its possible extensions therefore apply a potentially different correction factor to each component matrix V_g of V_{mway} , in the same manner as user-written command `cmgreg` (Gelbach and Miller, 2009). Of course, the researcher may choose to apply the same correction factor to all component matrices. For example, user-written command `reghdfe` applies a more conservative correction factor, specifically $[G/(G - 1)] \times [(N - 1)/(N - \#)]$, where G is the minimum cluster size defined above. User-written command `ivreg2` also applies this conservative factor when option `small` is specified; otherwise, it uses a large sample formula directly without any small sample correction.

2. A user-written command `reghdfe` (Correia, 2017) exploits this advantage offered by `_robust` to allow for multi-way clustering in the context of `areg`, `xtreg`, `fe` and `xtivreg`, `fe`. We thank an anonymous referee for alerting us to `reghdfe` and its use of `_robust`.

Fourth, Cameron et al. (2011) point out that in some applications, V_{mway} based on equation (1) may not be positive semi-definite. As a solution, they suggest that the researcher may replace negative eigenvalues of V_{mway} with 0s, and reconstruct the covariance matrix using the updated eigenvalues and the original eigenvectors. By taking advantage of Stata’s Mata environment, this approach can be implemented by inserting the following three command lines between lines 7 and 8 in the minimal example.

```
.      mata: symeigensystem(st_matrix("V_mway"), EVEC = ., eval = .)
.      mata: eval = eval :* (eval :> 0)
.      mata: st_matrix("V_mway", EVEC*diag(eval)*EVEC')
```

In Section 3, we will introduce command `vcemway` that generalizes the minimal example above, taking into consideration the major implementation issues. Before proceeding, readers are reminded that like one-way clustering, multi-way clustering should be applied with careful attention to the model of interest. For example, in non-linear models such as `probit`, the presence of cluster-specific random effects is a form of model misspecification that can render parametric estimators inconsistent; it would be inappropriate consider `probit` with multi-way clustered standard errors as a substitute for a more fundamental modeling solution such as the crossed random effects probit model that `meprobit` supports. §VII.C of Cameron and Miller (2015) and references therein provide further information on cluster-robust inferences in non-linear models.

Even in the context of linear models, the potential limitations of multi-way clustering should be recognized. A prominent example arises in the analysis of paired or “dyadic” data, such as bilateral trade flow data where each observation is on a distinct country pair, {A, B}. While two-way clustering on variables identifying A and B adjusts for error correlation in the likes of {Australia, Canada} and {Australia, USA}, and in the likes of {USA, UK} and {Canada, UK}, it fails to account for correlation in the likes of {Australia, USA} and {USA, UK}; the last two country pairs share neither A nor B, since USA appears in alternate positions. More general clustering methods for dyadic data have been developed by Aronow et al. (2015) and Cameron and Miller (2014).

In the context of linear models with multi-way fixed effects, Correia (2015) advises that the researcher should drop singleton groups in each dimension of fixed effects iteratively until no singleton group remains, before clustering standard errors in those dimensions or upper dimensions that nest them (e.g. county-level fixed effects and state-level clustering). Including singleton groups that comprise one observation may have undue effects on statistical inference by making small-sample correction factors smaller than otherwise, even though they have no effect on coefficient estimates and large sample covariance formulas. His user-written command `reghdfe` allows for multi-way clustering in the context of linear models with multi-way fixed effects, and automates this advice.

3 Command `vcmway`: a one-stop solution for multi-way clustering

`vcmway` is a new user-written command which automates the multi-way clustering approach described in Section 2. The researcher can apply `vcmway` to any existing estimation command that allows one-way clustering via `cluster(varname)` as an option, and obtain standard errors and a covariance matrix that have been adjusted for clustering in $m \geq 2$ dimensions.

`vcmway` expands on our minimal example in Section 2, `mymway`, to provide a convenient tool that addresses the implementation issues that we had subsequently discussed. When `predict` can be applied to compute observation-level residuals or scores for the estimation command of interest, `vcmway` speeds up execution time by avoiding repeated estimation of the same model with iterative one-way clustering; instead, it uses `_robust` to obtain the component matrices V_g of formula (1).³ `vcmway` builds in a sample marker which ensures that every component matrix is computed using the same estimation sample, specifically a set of observations in which none of the m cluster identifiers is missing. When the resulting m -way clustered covariance matrix V_{mway} is not positive semi-definite, `vcmway` reconstructs the matrix after replacing its negative eigenvalues with zeros and displays a telltale warning message. Finally, `vcmway` offers options that allow the researcher to choose their preferred small-sample adjustment factor and customize the residual degrees of freedom for t and F tests.

3.1 Syntax

The syntax diagram for `vcmway` is

```
vcmway cmdline_main, cluster(varlist) [vmcfactor(type) vmdfr(#)
      cmdline_options]
```

Required option `cluster()` accepts *varlist* that lists the names of $m \geq 2$ variables which identify the clustering dimensions of interest. As we will explain shortly, optional options `vmcfactor()` and `vmdfr()` allow the researcher to supply their preferred small-sample correction factor and residual degrees of freedom. In the remaining syntax diagram, *cmdline_main* refers to the main component of an estimation command line that the researcher would like to execute, such as `xtreg y X Z`; and *cmdline_options* refers to required and/or optional options in that command line, such as `re` and `nonest`. To complete this example, executing `vcmway xtreg y X Z, cluster(id1 id2 id3) re nonest` will report a linear random effects regression, with standard errors that have been adjusted for clustering in the three variables and computed using the default settings (see below) for options `vmcfactor()` and `vmdfr()`.

3. For some estimation commands (e.g. `xtreg`, `re`), `predict` does not support an option that computes the observation-level residuals or scores. In such cases, `vcmway` still applies the repeated estimation approach as outlined in `mymway`.

Optional options `vmcfactor(type) vmdfr(#)` can be applied as follows.

- `vmcfactor(type)` specifies the type of small-cluster correction factor that `vcmway` applies to the component covariance matrices V_g of formula (1). Input *type* may be `default`, `minimum` or `none`. Recall that each component matrix V_g of formula (1) is a one-way clustered covariance matrix. When computing V_g , Stata incorporates a small-sample correction factor of $[n_g / (n_g - 1)] \times [(N - 1) / (N - \#)]$, where n_g is the number of clusters or groups identified by $g \in \mathbb{M}$, N is the number of observations in the estimation sample, and $\#$ is the number of estimated coefficients for some commands (e.g. `regress`) and 1 for others (e.g. `ml`, `maximize`). The small-cluster correction factor henceforth refers to the first term in the product, $n_g / (n_g - 1)$.

Unless specified otherwise, `vcmway` assumes the `default` type that uses Stata's one-way clustered covariance matrices without further modification. That is, each V_g incorporates its own small-cluster correction factor of $n_g / (n_g - 1)$.

The `minimum` type requests the use of a conservative correction factor that is identical across all component matrices. Specifically, every V_g is recalculated by replacing $n_g / (n_g - 1)$ with $G / (G - 1)$, where G is the size of the smallest clustering dimension. For instance, suppose that `cluster(id1 id2 id3)` has been specified, and there are 180, 30, and 78 clusters in `id1`, `id2` and `id3`, respectively. G will be 30 in this case.

Finally, the `none` type requests the use of no small-cluster correction. In this case, every V_g is recalculated by replacing $n_g / (n_g - 1)$ with 1.

- `vmcfactor(#)` specifies the residual degrees of freedom for t and F tests. The default setting varies from command to command. In case the estimation command in question reports large sample test statistics (e.g. `ivregress` and `ml`, `maximize`), $\#$ is set to missing so that the researcher can carry out the large sample tests instead of t and F tests. In case the estimation command reports small sample statistics (e.g. `regress`), it is set to $(G - 1)$ where G is the size of the smallest clustering dimension.

4 Examples

Consider `nlswork.dta` used by examples in Stata's help menu for `xtreg`. This file provides a panel data set wherein variable `idcode` identifies individuals and variable `year` identifies survey years. We will estimate a linear regression model that specifies the logarithm of an individual's wage (`ln_wage`) as a function of her years of schooling (`grade`), work experience (`ttl_exp`) and squared work experience. The following example applies `vcmway` with `regress` to compute the OLS estimates of this model with standard errors adjusted for two-way clustering in `idcode` and `year`.

```
. webuse nlswork
(National Longitudinal Survey. Young Women 14-26 years of age in 1968)
. vcmway reg ln_w grade ttl_exp c.ttl_exp#c.ttl_exp, cluster(idcode year)
```

```

Linear regression                Number of obs   =   28,532
                                F(3, 4708)      =   1302.74
                                Prob > F             =   0.0000
                                R-squared            =   0.3003
                                Root MSE         =   .39996

                                (Std. Err. adjusted for clustering on idcode year)

```

ln_wage	Coef.	Robust Std. Err.	t	P> t	[95% Conf. Interval]	
grade	.0733695	.0029983	24.47	0.000	.0669388	.0798002
t1l_exp	.0450585	.0075979	5.93	0.000	.0287627	.0613543
c.t1l_exp#c.t1l_exp	-.0006412	.0004239	-1.51	0.153	-.0015505	.0002681
_cons	.5140105	.0294174	17.47	0.000	.4509165	.5771046

Notes:

Std. Err. adjusted for 2-way clustering on idcode year

Number of clusters in idcode = 4709

Number of clusters in year = 15

Stata's default small-cluster correction factors have been applied. See `vce` for detail.

Residual degrees of freedom for t tests and F tests = 14

F(,) and Prob > F above only account for one-way clustering on idcode.

Use `test` to compute F(,) and Prob > F that account for 2-way clustering.

At the end of the usual output table, `vce` includes notes to convey key information concerning its implementation.⁴ While one cannot tell from the output log in black and white, the notes display `vce` and `test` as blue-colored hyperlinks to the relevant help files. Options `vmcfactor()` and/or `vmdfr()` will change these notes in an expected manner. For example, `vmcfactor(minimum)` will induce `vce` to report that “[t]he small-cluster correction factor is $G/(G-1)$ where $G = 15$, the minimum of 2 cluster sizes.” The null hypothesis for the joint test statistic (e.g. F test above) that Stata reports as part of an output table is command-specific, and automatically identifying the null is a difficult task; `vce` therefore does not attempt to update that part of Stata output. Nevertheless, since post-estimation commands after `vce` will make use of a multi-way clustered covariance matrix, the researcher can use `test` to carry out a multi-way clustered version of the relevant joint hypothesis test.

For comparison, consider the following two command lines that produce t statistics which are robust to clustering in only one of the two dimensions at a time.

```

. reg ln_w grade t1l_exp c.t1l_exp#c.t1l_exp, cluster(idcode)
. reg ln_w grade t1l_exp c.t1l_exp#c.t1l_exp, cluster(year)

```

With adjustment for one-way clustering on `idcode`, the t statistics are 33.83, 18.71, -4.32 and 19.31, from `grade` to `_cons`. With adjustment for one-way clustering on `year`, the t statistics are 31.07, 6.07, -1.56 and 27.25, in the same order. Thus, each two-

4. After execution of `vce`, typing `estimates replay` or equivalent command lines will display the output table without the notes. Typing `vce` without any other component will re-display the output table with the notes.

way clustered t statistic, shown in the output table above, is smaller than either of its one-way clustered counterparts. In general, whether multi-way clustering leads to smaller t statistics (or equivalently, larger standard errors) than one-way clustering is an empirical question that cannot be answered ex ante. Cameron et al. (2011, §4) provide several empirical applications comparing two-way clustering to one-way clustering. In those applications, two-way clustered standard errors turn out to be larger than the average of two one-way clustered standard errors, and sometimes they also turn out to be larger than both one-way clustered standard errors as in the present example.

Applying `vcemway` alongside estimation commands that have more complex syntax diagrams is equally straightforward. For illustration, we will extend the OLS regression above to incorporate random effects at the individual-level (i.e `idcode`-level). The following example estimates the resulting random effects model and adjusts its standard errors for two-way clustering in `idcode` and `year`. Since the model accounts for `idcode`-level random effects and `year` is not nested in `idcode`, the researcher must specify `xtreg`'s native `nonest` option to allow Stata to produce a covariance matrix adjusted for clustering in `year`.

```
.      xtset idcode
      panel variable:  idcode (unbalanced)
.      vcemway xtreg ln_w grade ttl_exp c.ttl_exp#c.ttl_exp, cluster(idcode year) re nonest
Random-effects GLS regression           Number of obs   =    28,532
Group variable: idcode                  Number of groups =     4,709
R-sq:                                   Obs per group:
      within = 0.1410                    min =           1
      between = 0.4042                    avg  =          6.1
      overall = 0.2996                    max  =          15
                                         Wald chi2(3)    =    3792.57
corr(u_i, X) = 0 (assumed)               Prob > chi2    =     0.0000
                                         (Std. Err. adjusted for clustering on idcode year)
```

	ln_wage	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
	grade	.0726953	.0045887	15.84	0.000	.0637015	.081689
	ttl_exp	.0475898	.0062452	7.62	0.000	.0353494	.0598302
	c.ttl_exp#c.ttl_exp	-.0009575	.0003564	-2.69	0.007	-.0016559	-.000259
	_cons	.5130608	.0542205	9.46	0.000	.4067905	.6193311
	sigma_u	.2833309					
	sigma_e	.29686061					
	rho	.4766933	(fraction of variance due to u_i)				

Notes:

Std. Err. adjusted for 2-way clustering on idcode year
 Number of clusters in idcode = 4709
 Number of clusters in year = 15

Stata's default small-cluster correction factors have been applied. See `vcemway` for detail.
`chi2()` and `Prob > chi2` above only account for one-way clustering on `idcode`.
 Use `test` to compute `chi2()` and `Prob > chi2` that account for 2-way clustering.

For comparison, consider one-way clustered z statistics obtained by executing the following command lines.

```
. xtreg ln_w grade ttl_exp c.ttl_exp#c.ttl_exp, cluster(idcode) re
. xtreg ln_w grade ttl_exp c.ttl_exp#c.ttl_exp, cluster(year) re nonest
```

With adjustment for one-way clustering on `idcode`, the z statistics are 35.06, 24.24, -8.94 and 19.89, from `grade` to `_cons`. With adjustment for one-way clustering on `year`, the z statistics are 15.92, 7.79, -2.73 and 9.47, respectively. As in the OLS example above, two-way clustering leads to more conservative inferences (i.e. smaller z statistics) than either of one-way clustering approaches.

5 Acknowledgments

We would like to thank an anonymous referee for many useful references and constructive suggestions. We would also like to thank Kenju Kamei for helpful discussion.

6 References

- Aronow, P.M., C. Samii, and V.A. Assenova. 2015. Cluster-Robust Variance Estimation for Dyadic Data. *Political Analysis* 23: 564-577.
- Baum, C.F., M.E. Schaffer, and S. Stillman. 2010. ivreg2: Stata module for extended instrumental variables/2SLS, GMM and AC/HAC, LIML and k-class regression. <http://ideas.repec.org/c/boc/bocode/s425401.html>.
- Cameron, A.C., J.B. Gelbach, and D.L. Miller. 2011. Robust Inference With Multiway Clustering. *Journal of Business and Economic Statistics* 29: 238-249.
- Cameron, A.C. and D.L. Miller. 2014. Robust Inference for Dyadic Data. http://faculty.econ.ucdavis.edu/faculty/cameron/research/dyadic_cameron_miller_december2014_with_tables.pdf.
- Cameron, A.C. and D.L. Miller. 2015. A Practitioner's Guide to Cluster-Robust Inference. *Journal of Human Resources* 50: 317-372.
- Correia, S. 2015. Singletons, Cluster-Robust Standard Errors and Fixed Effects: A Bad Mix. <http://scorreia.com/research/singletons.pdf>
- Correia, S. 2017. reghdfe: Stata module for linear and instrumental-variable/gmm regression absorbing multiple levels of fixed effects. <https://ideas.repec.org/c/boc/bocode/s457874.html>
- Dube, A., T. Lester, and M. Reich. 2010. Minimum wage effects across state borders: estimates using contiguous counties. *Review of Economics and Statistics*

92: 945-964.

Dube, A., T. Lester, and M. Reich. 2016. Minimum Wage Shocks, Employment Flows, and Labor Market Frictions. *Journal of Labor Economics* 34: 663-704.

Gelbach, J.B., and D.L. Miller. 2009. Multi-way clustering with OLS. <http://faculty.econ.ucdavis.edu/faculty/dlmiller/statafiles/>.

Gow, I.D., G. Ormazabal, and D.J. Taylor. 2010. Correcting for Cross-Sectional and Time-Series Dependence in Accounting Research. *Accounting Review* 85: 483-512.

Roodman, D., M.Ø. Nielsen, J.G. MacKinnon, and M.D. Webb. 2019. Fast and wild: Bootstrap inference in Stata using boottest. *Stata Journal* 19: 4-60.

Thompson, A.B. 2011. Simple formulas for standard errors that cluster by both firm and time. *Journal of Financial Economics* 99: 1-10.

About the authors

Ariel Gu is a Lecturer in Finance at Newcastle Business School, Northumbria University, UK. Hong Il Yoo is an Associate Professor in Economics at Durham University Business School, Durham University, UK.