Leverage, influence, and the jackknife in clustered regression models: Reliable inference using summclust

James G. MacKinnon
Queen's University
Kingston, Canada
mackinno@queensu.ca

Morten Ørregaard Nielsen
Aarhus University
Aarhus, Denmark
mon@econ.au.dk

Matthew D. Webb
Carleton University
Ottawa, Canada
matt.webb@carleton.ca

Abstract. We introduce a new command, `summclust`, that summarizes the cluster structure of the dataset for linear regression models with clustered disturbances. The key unit of observation for such a model is the cluster. We therefore propose cluster-level measures of leverage, partial leverage, and influence and show how to compute them quickly in most cases. The measures of leverage and partial leverage can be used as diagnostic tools to identify datasets and regression designs in which cluster–robust inference is likely to be challenging. The measures of influence can provide valuable information about how the results depend on the data in the various clusters. We also show how to calculate two jackknife variance matrix estimators efficiently as a by-product of our other computations. These estimators, which are already available in Stata, are generally more conservative than conventional variance matrix estimators. The `summclust` command computes all the quantities that we discuss.

Keywords: st0733, summclust, clustered data, cluster–robust variance estimator, CRVE, grouped data, high-leverage clusters, influential clusters, jackknife, partial leverage, robust inference

1 Introduction

It is now standard in many fields of economics and other disciplines to use cluster–robust inference for the parameters of linear regression models. In the most common case, each of the N observations is assigned to one of G disjoint clusters, which might correspond to, for example, families, schools, villages, hospitals, firms, industries, years, cities, counties, or states. The assignment of observations to clusters is assumed to be known, and observations in different clusters are assumed to be independent, but any pattern of heteroskedasticity or dependence is allowed within each cluster. Under these assumptions, a cluster–robust variance matrix (CRVE) yields asymptotically valid t tests, Wald tests, and confidence intervals. However, even when N is very large, the resulting inferences may be unreliable when G is not large or the clusters are not sufficiently homogeneous.

The literature on cluster–robust inference has grown rapidly recently. Cameron and Miller's (2015) article is a classic survey article. Conley, Gonçalves, and Hansen (2018) survey a broader class of methods for dependent data. MacKinnon, Nielsen, and Webb

(2023a) offer a comprehensive guide to empirical practice. As it discusses, there are two situations in which cluster–robust t tests and Wald tests are at risk of overrejecting to an extreme extent, even when G is not small. The first is when one or a few clusters are much larger than the rest, and the second is when the only "treated" observations belong to just a few clusters; Djogbenou, MacKinnon, and Nielsen (2019) discuss the first case, and MacKinnon and Webb (2017a,b, 2018) discuss the second. In both of these cases, one cluster (or a few of them) has high leverage, in that omitting this cluster has the potential to change the ordinary least-squares (OLS) estimates substantially. When that actually happens, a cluster is said to be influential.

The concepts of leverage and influence are normally applied at the observation level (Belsley, Kuh, and Welsch 1980), but they are equally applicable at the cluster level. Just as high-leverage observations can make heteroskedasticity-robust inference unreliable (Chesher 1989), so, too, can high-leverage clusters make cluster–robust inference unreliable. Just as highly influential observations may lead us to suspect that there is something wrong with the model or the data, so, too, may highly influential clusters. Any situation in which a few clusters have high leverage or high influence should be worrying.

There are at least two different concepts of leverage. The usual one focuses on fitted values or, equivalently, residuals. A cluster is said to have high leverage if removing it has the potential to greatly change the fitted values for that cluster. The second concept is partial leverage (Cook and Weisberg 1980). A cluster is said to have high partial leverage for the jth coefficient if removing that cluster has the potential to greatly change the estimate of the jth coefficient. We discuss both concepts in section 2.1.

Whether a cluster has high leverage, has high partial leverage, or is influential can depend on the sample in rather complicated ways. We provide a new command, `summclust`, that implements computationally efficient ways to identify high-leverage and influential clusters and provides several statistics that collectively summarize the cluster structure of the dataset. These can be useful for detecting cases in which cluster–robust inference may not be reliable. Our leverage and influence calculations also allow us to compute two cluster jackknife variance matrix estimators, which we refer to as CV$_3$ and CV$_{3J}$, at little additional cost. These estimators are already available in Stata by using either the `vce(jackknife)` option or the `jackknife` prefix. Recent work (Hansen 2022; MacKinnon, Nielsen, and Webb 2023c) suggests that CV$_3$ and CV$_{3J}$ generally perform better in finite samples than more widely used CRVEs; see section 7.

The remainder of the article is organized as follows. The next section begins with a brief review of cluster–robust inference for linear regression models. Then section 2.1 introduces our new measures of leverage, partial leverage, and influence at the cluster level. Section 2.2 shows how our results can be used to compute the CV$_3$ and CV$_{3J}$ jackknife variance matrix estimators. Section 2.3 discusses what quantities are reported by `summclust` and should, at least in some cases, be reported by the investigator.

Section 3 provides a detailed description of the `summclust` command, which computes these variance estimators and diagnostic measures. The command uses the following syntax:

`summclust` *varlist*`, cluster(`*varname*`)` $\big[$ *options* $\big]$

`summclust` has many options and can even be used by itself to fit a linear regression model with clustered disturbances. The last few sections of the article illustrate the use of `summclust` and provide evidence on the value of the measures that it calculates. Section 4 presents an empirical illustration in which measures of leverage, partial leverage, and influence are highly informative. Section 4 discusses several special cases in which some or all of these measures can be determined analytically. Section 6 briefly discusses two-way clustering, where `summclust` can be valuable even though it is not explicitly designed to handle this case. Section 7 describes some simulation experiments that suggest that it may be desirable to report many of the quantities calculated by `summclust`, and section 8 concludes.

2   Clustering, leverage, influence, and the jackknife

We focus on the linear regression model

$$\boldsymbol{y}_g = \boldsymbol{X}_g \boldsymbol{\beta} + \boldsymbol{u}_g, \quad g = 1, \dots, G \tag{1}$$

where the data have been divided into G disjoint clusters. The gth cluster has N_g observations, so the sample size is $N = \sum_{g=1}^{G} N_g$. In (1), $\boldsymbol{X}_g$ is an $N_g \times k$ matrix of regressors, $\boldsymbol{\beta}$ is a k-vector of coefficients, $\boldsymbol{y}_g$ is an N_g-vector of observations on the regressand, and $\boldsymbol{u}_g$ is an N_g-vector of disturbances (or error terms). Of course, the $\boldsymbol{X}_g$ may be stacked into an $N \times k$ matrix $\boldsymbol{X}$, and likewise the $\boldsymbol{y}_g$ and $\boldsymbol{u}_g$ may be stacked into N-vectors $\boldsymbol{y}$ and $\boldsymbol{u}$ so that (1) can be rewritten as $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{u}$.

Dividing the sample into clusters becomes meaningful only if we make assumptions about the disturbance vectors $\boldsymbol{u}_g$ and, consequently, the score vectors $\boldsymbol{s}_g = \boldsymbol{X}_g^\top \boldsymbol{u}_g$. For a correctly specified model, $E(\boldsymbol{s}_g) = \boldsymbol{0}$ for all g. We further assume that

$$E(\boldsymbol{s}_g \boldsymbol{s}_g^\top) = \boldsymbol{\Sigma}_g \quad \text{and} \quad E(\boldsymbol{s}_g \boldsymbol{s}_{g'}^\top) = \boldsymbol{0}, \quad g, g' = 1, \dots, G, \quad g' \neq g \tag{2}$$

where $\boldsymbol{\Sigma}_g$ is the symmetric, positive semidefinite variance matrix of the scores for the gth cluster. The second assumption in (2) is crucial. It says that the scores for every cluster are uncorrelated with the scores for every other cluster. We take the number of clusters G and the allocation of observations to clusters as given. The important issue of how to choose the clustering structure, perhaps by testing for the correct level of clustering, is discussed in detail in MacKinnon, Nielsen, and Webb (2023b).

The OLS estimator of $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{\beta}_0 + \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{u}$$

where the second equality depends on the assumption that the data are actually generated by (1) with true value $\boldsymbol{\beta}_0$. It follows that

$$\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \sum_{g=1}^{G} \boldsymbol{X}_g^\top \boldsymbol{u}_g = \left(\sum_{g=1}^{G} \boldsymbol{X}_g^\top \boldsymbol{X}_g\right)^{-1} \sum_{g=1}^{G} \boldsymbol{s}_g \tag{3}$$

From the rightmost expression in (3), we see that the distribution of $\widehat{\boldsymbol{\beta}}$ depends on the disturbance subvectors $\boldsymbol{u}_g$ only through the distribution of the score vectors $\boldsymbol{s}_g$. Asymptotic inference commonly uses the empirical score vectors $\widehat{\boldsymbol{s}}_g = \boldsymbol{X}_g^\top \widehat{\boldsymbol{u}}_g$, in which the $\boldsymbol{u}_g$ are replaced by the residual subvectors $\widehat{\boldsymbol{u}}_g$, to estimate the variance matrix of the $\boldsymbol{s}_g$. This should work well if the sum of the $\boldsymbol{s}_g$, suitably normalized, is well approximated by a multivariate normal distribution with mean zero and if the $\boldsymbol{s}_g$ are well approximated by the $\widehat{\boldsymbol{s}}_g$. However, asymptotic inference can be misleading when either of these approximations is poor.

It follows immediately from (3) that an estimator of the variance of $\widehat{\boldsymbol{\beta}}$ may be based on the usual sandwich formula,

$$\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \left(\sum_{g=1}^{G} \boldsymbol{\Sigma}_g\right) \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \tag{4}$$

The natural way to estimate (4) is to replace the $\boldsymbol{\Sigma}_g$ matrices by their empirical counterparts, that is, the $\widehat{\boldsymbol{s}}_g \widehat{\boldsymbol{s}}_g^\top$. If, in addition, we multiply by a correction for degrees of freedom, we obtain the cluster–robust variance estimator, or CRVE,

$$\text{CV}_1: \qquad \frac{G(N-1)}{(G-1)(N-k)} (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \left(\sum_{g=1}^{G} \widehat{\boldsymbol{s}}_g \widehat{\boldsymbol{s}}_g^\top\right) \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \tag{5}$$

This is by far the most widely used CRVE in practice, and it is the default one implemented in Stata; alternatives to this estimator will be discussed in section 2.2. When $G = N$, the CV_1 estimator reduces to the familiar HC$_1$ estimator (MacKinnon and White 1985), which is robust only to heteroskedasticity of unknown form.

The fundamental unit of inference for clustered observations is not the observation but the cluster; this is evident from (3), (4), and (5). The asymptotic theory for cluster–robust inference has been analyzed by Djogbenou, MacKinnon, and Nielsen (2019) and Hansen and Lee (2019) under the assumption that $G \to \infty$. The quality of the asymptotic approximation depends on the number of clusters G and the heterogeneity of the score vectors (MacKinnon, Nielsen, and Webb 2023a). The more the distributions of the scores vary across clusters, the worse the asymptotic approximation will likely be. Heterogeneity can arise from variation in cluster sizes or from variation in the distributions of the disturbances, the regressors, or both. As we discuss in sections 2.1, 2.3,

and 7, leverage, partial leverage, and summary statistics based on them provide useful measures of heterogeneity across clusters.

Inference about $\boldsymbol{\beta}$ is typically based on cluster–robust t statistics and Wald statistics. If β_j denotes the jth element of $\boldsymbol{\beta}$ and β_{0j} is its value under the null hypothesis, then the appropriate t statistic is

$$t_j = \frac{\widehat{\beta}_j - \beta_{0j}}{\text{s.e.}(\widehat{\beta}_j)}$$

where $\widehat{\beta}_j$ is the OLS estimate and s.e.$(\widehat{\beta}_j)$ is the square root of the jth diagonal element of (5). Under extremely strong assumptions (Bester, Conley, and Hansen 2011), it can be shown that t_j asymptotically follows the $t(G-1)$ distribution. Conventional inference in Stata and other programs is based on this distribution.

As the articles cited in the second paragraph of section 1 discuss, inference based on t_j and the $t(G-1)$ distribution can be unreliable when G is small or the clusters are severely heterogeneous. This is true to an even greater extent for Wald tests of two or more restrictions (Pustejovsky and Tipton 2018). The measures of leverage and partial leverage at the cluster level that we introduce in the next section may help to identify the sort of heterogeneity that is likely to make inference unreliable.

Instead of using the $t(G-1)$ distribution, we can obtain both p-values for t_j and confidence intervals for β_j by using the wild cluster restricted (WCR) bootstrap (Cameron, Gelbach, and Miller 2008). It can sometimes provide much more reliable inferences than the conventional approach; see section 7. Roodman et al. (2019) describe a computationally efficient implementation of this method in the package `boottest`. MacKinnon, Nielsen, and Webb (2023c) propose new versions of the wild cluster bootstrap that involve transforming the empirical scores. When G is reasonably large and the clusters are not very heterogeneous, inferences based on the WCR bootstrap and inferences based on CV$_1$ t statistics combined with the $t(G-1)$ distribution will often be very similar. When they differ noticeably, neither should be relied upon without further investigation.

Section 2.2 discusses two CRVEs, which we refer to as CV$_3$ and CV$_{3J}$, that are both based on the cluster jackknife. In practice, these estimators are often extremely similar. CV$_3$ and CV$_{3J}$ tend to yield more reliable inferences in finite samples than does CV$_1$, especially when the clusters are quite heterogeneous; see section 7 and MacKinnon, Nielsen, and Webb (2023c). Based on this simulation evidence, we recommend computing either CV$_3$ or CV$_{3J}$ essentially always. This is easy to do using `summclust`.

2.1 Identifying high-leverage and influential clusters

At the observation level, there are three classic measures of heterogeneity, namely, leverage, partial leverage, and influence (Belsley, Kuh, and Welsch 1980; Chatterjee and Hadi 1986). In this section, we propose analogous measures at the cluster level.

Measures of leverage at the observation level are based on how much the residual for observation i changes when that observation is omitted from the regression. If h_i denotes the ith diagonal element of the "hat matrix" $\boldsymbol{H} = \boldsymbol{P_X} = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$, then omitting the ith observation changes the ith residual from $\widehat{u}_i$ to $\widehat{u}_i/(1 - h_i)$. Because $0 < h_i < 1$, this delete-one residual is always larger in absolute value than $\widehat{u}_i$. The factor by which the delete-one residual exceeds $\widehat{u}_i$ increases with h_i. Because the average of the h_i is k/N, observations with values of h_i substantially larger than k/N may reasonably be said to have high leverage.

Dropping the gth cluster when we estimate $\boldsymbol{\beta}$ yields the delete-one-cluster estimate $\widehat{\boldsymbol{\beta}}^{(g)}$. Using $\widehat{\boldsymbol{\beta}}^{(g)}$ in place of $\widehat{\boldsymbol{\beta}}$ changes the residual vector for the gth cluster from $\widehat{\boldsymbol{u}}_g$ to $\widehat{\boldsymbol{u}}_g^{(g)}$. These delete-one-cluster residual vectors can be written in two ways:

$$\widehat{\boldsymbol{u}}_g^{(g)} = \boldsymbol{y}_g - \boldsymbol{X}_g\widehat{\boldsymbol{\beta}}^{(g)} = (\mathbf{I} - \boldsymbol{H}_g)^{-1}\widehat{\boldsymbol{u}}_g$$

In the rightmost expression above,

$$\boldsymbol{H}_g = \boldsymbol{X}_g(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}_g^\top$$

is the $N_g \times N_g$ diagonal block of $\boldsymbol{H}$ that corresponds to cluster g. The matrix $\boldsymbol{H}_g$ is the cluster analog of the scalar h_i. Of course, it is not feasible to report the $\boldsymbol{H}_g$. In fact, when any of the clusters are sufficiently large, even computing and storing these matrices may be challenging. As a measure of leverage, we therefore suggest using a matrix norm of the $\boldsymbol{H}_g$. Specifically, we suggest the scalar

$$L_g = \mathrm{Tr}\left(\boldsymbol{H}_g\right) = \mathrm{Tr}\left(\boldsymbol{X}_g^\top\boldsymbol{X}_g(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\right) \tag{6}$$

When the gth cluster contains just one observation, say, the ith, then $L_g = h_i$. Thus, in this special case, the leverage measure that we are proposing reduces to the usual measure of leverage at the observation level.

The trace in (6) is the nuclear norm of the matrix $\boldsymbol{H}_g$. In general, the nuclear norm of a matrix $\boldsymbol{A}$ is the sum of the singular values of $\boldsymbol{A}$. When $\boldsymbol{A}$ is symmetric and positive semidefinite, the singular values are equal to the eigenvalues, which are nonnegative. Because the trace of any square matrix is equal to the sum of the eigenvalues, the trace of a symmetric and positive semidefinite matrix is also its nuclear norm. In principle, we could report any norm of the $\boldsymbol{H}_g$ matrices, but the nuclear norm is particularly easy to compute. Also, because it is linear, we can sum over g and take the sum inside the norm just as if the $\boldsymbol{H}_g$ were scalars. Because $\sum_{g=1}^{G} \boldsymbol{X}_g^\top\boldsymbol{X}_g = \boldsymbol{X}^\top\boldsymbol{X}$, this result means that $G^{-1}\sum_{g=1}^{G} \mathrm{Tr}(\boldsymbol{H}_g) = k/G$, which is analogous to the result that the average of the h_i over all observations is k/N.

High-leverage clusters can be identified by comparing the L_g with k/G, their average. If L_h is substantially larger than k/G for some cluster h, then cluster h may be said to have high leverage. Just how much larger L_h must be is a matter of judgment. A cluster with $L_h = 2k/G$ probably does not qualify, but a cluster with $L_h = 5k/G$ probably does. Cluster h can have high leverage because N_h is considerably larger than G/N, the matrix $\boldsymbol{X}_h$ is somehow extreme relative to the other $\boldsymbol{X}_g$ matrices, or both. We can

compare the leverage of any two clusters by forming ratios. For example, if $L_1 = 3$ and $L_2 = 1$, then we can say that the first cluster has three times the leverage of the second cluster.

The leverage measure we suggest in (6) shows the potential impact of a specified cluster on residuals and fitted values but not on any particular regression coefficient. When interest focuses on just one such coefficient, say, the jth, it may be more interesting to calculate the partial leverage of each cluster. The concept of partial leverage was introduced, for individual observations, in Cook and Weisberg (1980). Let

$$\acute{\boldsymbol{x}}_j = \left\{ \mathbf{I} - \boldsymbol{X}_{[j]} \left(\boldsymbol{X}_{[j]}^\top \boldsymbol{X}_{[j]} \right)^{-1} \boldsymbol{X}_{[j]}^\top \right\} \boldsymbol{x}_j$$

where $\boldsymbol{x}_j$ is the vector of observations on the jth regressor and $\boldsymbol{X}_{[j]}$ is the matrix of observations on all the other regressors. Thus, $\acute{\boldsymbol{x}}_j$ denotes $\boldsymbol{x}_j$ after all the other regressors have been partialed out. The partial leverage of observation i is simply the ith diagonal element of the matrix $\acute{\boldsymbol{x}}_j (\acute{\boldsymbol{x}}_j^\top \acute{\boldsymbol{x}}_j)^{-1} \acute{\boldsymbol{x}}_j^\top$, which is just $\acute{x}_{ji}^2 / (\acute{\boldsymbol{x}}_j^\top \acute{\boldsymbol{x}}_j)$, where $\acute{x}_{ji}^2$ is the ith element of $\acute{\boldsymbol{x}}_j$.

The analogous measure of partial leverage for cluster g is

$$L_{gj} = \frac{\acute{\boldsymbol{x}}_{gj}^\top \acute{\boldsymbol{x}}_{gj}}{\acute{\boldsymbol{x}}_j^\top \acute{\boldsymbol{x}}_j} \tag{7}$$

where $\acute{\boldsymbol{x}}_{gj}$ is the subvector of $\acute{\boldsymbol{x}}_j$ corresponding to the gth cluster. This is what (6) reduces to if we replace $\boldsymbol{X}$ and $\boldsymbol{X}_g$ by $\acute{\boldsymbol{x}}_j$ and $\acute{\boldsymbol{x}}_{gj}$, respectively. It is easy to calculate the partial leverage for every cluster for any coefficient of interest. The average of the L_{gj} is evidently $1/G$, so if cluster h has $L_{hj} \gg 1/G$, it has high partial leverage for the jth coefficient. Moreover, as we will see in section 7, the empirical distribution of the L_{gj} across clusters seems to provide useful diagnostic information.

Young (2022) derives a measure of cluster-level leverage for the first-stage regression used to obtain a linear instrumental-variables estimator. That article calls L_{gj} the group g "share of coefficient leverage" for instrument j and then uses the maximum of the L_{gj} over all the instruments excluded from the structural equation as a measure of the leverage of cluster g. Using simulations based on 1,309 instrumental-variables regressions from 30 published articles, Young finds that inference is much less reliable for models where one or two clusters have high leverage in the first-stage regression than for models where no clusters do so.

One possible consequence of heterogeneity is that the estimates may change a lot when certain clusters are deleted. It can therefore be illuminating to delete one cluster at a time to see how influential each cluster is. To do this computationally efficiently, `summclust` first computes the cluster-level matrices and vectors

$$\boldsymbol{X}_g^\top \boldsymbol{X}_g \quad \text{and} \quad \boldsymbol{X}_g^\top \boldsymbol{y}_g, \quad g = 1, \dots, G \tag{8}$$

These are then used to construct $\boldsymbol{X}^\top \boldsymbol{X}$ and $\boldsymbol{X}^\top \boldsymbol{y}$, and the vector of least-squares estimates when cluster g is deleted is computed as

$$\widehat{\boldsymbol{\beta}}^{(g)} = \left(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{X}_g^\top \boldsymbol{X}_g \right)^{-1} \left(\boldsymbol{X}^\top \boldsymbol{y} - \boldsymbol{X}_g^\top \boldsymbol{y}_g \right) \tag{9}$$

Unless k is extremely large, it should generally not be expensive to compute $\widehat{\boldsymbol{\beta}}^{(g)}$ for every cluster using (9). `summclust` simply has to invert G matrices, each of them $k \times k$, and then multiply each of those matrices by a k-vector.

Especially when they vary a lot, the $\widehat{\boldsymbol{\beta}}^{(g)}$ can reveal a great deal about the sample. In addition, as we shall see in section 2.2, they may be used to calculate jackknife variance matrices. When there is a parameter of particular interest, say, β_j, it may be a good idea to report the $\widehat{\beta}_j^{(g)}$ for $g = 1, \ldots, G$ in either a histogram or a table. By default, `summclust` creates several figures with these and other cluster-level statistics. If $\widehat{\beta}_j^{(h)}$ differs greatly from $\widehat{\beta}_j$ for some cluster h, then cluster h is evidently influential.

In a few extreme cases, there may be a cluster h for which it is impossible to compute $\widehat{\beta}_j^{(h)}$. This will happen, for example, when the regressor corresponding to β_j is a treatment dummy and cluster h is the only treated one. This is an extreme example of the problem of few treated clusters, and inferences based on either the $t(G-1)$ distribution or the WCR bootstrap are likely to be completely unreliable in this case (MacKinnon and Webb 2017a, 2018, 2020).

Identifying influential clusters by comparing the $\widehat{\boldsymbol{\beta}}^{(g)}$ with $\widehat{\boldsymbol{\beta}}$ is very similar to identifying influential observations using the classic methods discussed in Belsley, Kuh, and Welsch (1980) and Chatterjee and Hadi (1986); for an interesting recent extension, see Broderick, Giordano, and Meager (2023). Unlike the leverage measures, the $\widehat{\beta}_j^{(g)}$ may be either positive or negative, must depend on the $\boldsymbol{y}_g$, and necessarily vary across clusters. They may sometimes reveal features of the model or dataset that require further investigation. Perhaps the model does not seem to apply to some clusters, or perhaps there are measurement errors or observations that have been miscoded.

Regression models often include cluster fixed effects. When one of the regressors is a fixed-effects dummy for cluster g, the matrices $\boldsymbol{X}_g^\top \boldsymbol{X}_g$ and $\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{X}_g^\top \boldsymbol{X}_g$ are singular. This will cause the calculation in (9) to fail unless a generalized inverse routine, such as the `invsym()` function in Mata, is used. Although `summclust` uses this function, it also provides options to avoid the problem and save some computer time by partialing out the fixed-effects dummies prior to computing the cluster-level matrices and vectors in (8); see section 3.

Partialing out cluster fixed effects means replacing $\boldsymbol{X}$ and $\boldsymbol{y}$ by $\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{y}}$, the deviations from their cluster means. For example, the element of $\widetilde{\boldsymbol{y}}$ corresponding to the jth observation in the gth cluster is $y_{g,j} - N_g^{-1} \sum_{i=1}^{N_g} y_{g,i}$. The gth subvector of $\widetilde{\boldsymbol{y}}$ is $\widetilde{\boldsymbol{y}}_g$, and the gth submatrix of $\widetilde{\boldsymbol{X}}$ is $\widetilde{\boldsymbol{X}}_g$. Because there is just one fixed effect per cluster, $\widetilde{\boldsymbol{y}}_g$ depends solely on $\boldsymbol{y}_g$ and $\widetilde{\boldsymbol{X}}_g$ depends solely on $\boldsymbol{X}_g$. The calculations in (6) and (9) are now based on $\widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}}$, $\widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{y}}$, the $\widetilde{\boldsymbol{X}}_g^\top \widetilde{\boldsymbol{X}}_g$, and the $\widetilde{\boldsymbol{X}}_g^\top \widetilde{\boldsymbol{y}}_g$. Importantly, the sum of the L_g is now equal to the number of columns in $\widetilde{\boldsymbol{X}}$ instead of the number of columns in $\boldsymbol{X}$.

2.2   Two jackknife variance matrix estimators

Although the CV_1 variance estimator defined in (5) is very widely used, it often does not have good finite-sample properties. Two alternative CRVEs, which are usually known as CV_2 and CV_3, were proposed in Bell and McCaffrey (2002). They are the cluster analogs of the heteroskedasticity-consistent estimators HC_2 and HC_3, which are appropriate when the u_i are independent. These names were coined in MacKinnon and White (1985), who proposed HC_3 as a jackknife variance estimator. In the remainder of this section, we focus on CV_3 because CV_2 is not a jackknife estimator and is not amenable to the computational methods that we propose; see Imbens and Kolesár (2016), Pustejovsky and Tipton (2018), and Niccodemi et al. (2020). Stata 18 added the ability to rapidly calculate CV_2 standard errors, using the option `vce(hc2 clustvar)`. Simulations in MacKinnon, Nielsen, and Webb (2023c) suggest that CV_2 is preferred to CV_1 but that CV_3 is almost always preferred to CV_2.

CV_3 can be written in several ways. One of them is

$$\text{CV}_3: \qquad \frac{G-1}{G} \left(\boldsymbol{X}^\top \boldsymbol{X} \right)^{-1} \left(\sum_{g=1}^{G} \ddot{\boldsymbol{s}}_g \ddot{\boldsymbol{s}}_g^\top \right) \left(\boldsymbol{X}^\top \boldsymbol{X} \right)^{-1} \tag{10}$$

where the modified score vectors $\ddot{\boldsymbol{s}}_g$ are defined as

$$\ddot{\boldsymbol{s}}_g = \boldsymbol{X}_g^\top \boldsymbol{M}_{gg}^{-1} \widehat{\boldsymbol{u}}_g$$

Here $\boldsymbol{M}_{gg} = \boldsymbol{I}_{N_g} - \boldsymbol{X}_g (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}_g^\top$ is the diagonal block corresponding to the gth cluster of the projection matrix $\boldsymbol{M_X}$, which satisfies $\widehat{\boldsymbol{u}} = \boldsymbol{M_X} \boldsymbol{u}$. Although computing CV_3 using (10) works well when all the N_g are very small, it becomes expensive, or perhaps computationally infeasible, when one or more of the N_g is large. The problem is that an $N_g \times N_g$ matrix needs to be stored and inverted for every cluster. Niccodemi et al. (2020) propose a method that is much faster for large clusters, versions of which apply to both CV_2 and CV_3. However, recognizing that CV_3 is a jackknife estimator makes a method available that is even simpler and usually faster.

There are actually two cluster jackknife estimators of $\text{Var}(\widehat{\boldsymbol{\beta}})$. The simplest is probably

$$\text{CV}_{3\text{J}}: \qquad \frac{G-1}{G} \sum_{g=1}^{G} \left(\widehat{\boldsymbol{\beta}}^{(g)} - \overline{\boldsymbol{\beta}} \right) \left(\widehat{\boldsymbol{\beta}}^{(g)} - \overline{\boldsymbol{\beta}} \right)^\top \tag{11}$$

where $\overline{\boldsymbol{\beta}}$ is the sample mean of the $\widehat{\boldsymbol{\beta}}^{(g)}$, which were defined in (9). The expression in (11) is the cluster analog of the usual jackknife variance matrix estimator given in MacKinnon and White [1985, (11)]. Each of the $\widehat{\boldsymbol{\beta}}^{(g)}$ is obtained by deleting a cluster instead of an observation, and the summation is over clusters instead of observations. If $\overline{\boldsymbol{\beta}}$ in (11) is replaced by $\widehat{\boldsymbol{\beta}}$, we instead obtain

$$\text{CV}_3: \qquad \frac{G-1}{G} \sum_{g=1}^{G} \left(\widehat{\boldsymbol{\beta}}^{(g)} - \widehat{\boldsymbol{\beta}} \right) \left(\widehat{\boldsymbol{\beta}}^{(g)} - \widehat{\boldsymbol{\beta}} \right)^\top \tag{12}$$

This version of CV_3 is numerically identical to the one in (10) (MacKinnon, Nielsen, and Webb 2023c, sec. 3). Unless all the clusters are very small, computing CV_3 using (12) is much faster than using (10); timings are reported in MacKinnon, Nielsen, and Webb (2023c).

Many discussions of jackknife variance estimation follow Efron (1979) and use $\overline{\boldsymbol{\beta}}$ as in (11), but others, including Bell and McCaffrey (2002), use $\widehat{\boldsymbol{\beta}}$ as in (12). Although these two jackknife variance estimators are asymptotically the same, they are rarely equal because CV_3 minus $\text{CV}_{3\text{J}}$ is a positive semidefinite matrix. In practice, however, they tend to be very similar (MacKinnon, Nielsen, and Webb 2023c), and there seems to be no good reason to expect either CV_3 or $\text{CV}_{3\text{J}}$ to perform better in general. Interestingly, the original HC_3 estimator proposed in MacKinnon and White (1985) is actually the analog of $\text{CV}_{3\text{J}}$. The modern version of HC_3, which is the analog of CV_3, seems to be due to Davidson and MacKinnon (1993, chap. 16). This version of HC_3 is normally computed by dividing each residual by the corresponding diagonal element of $\boldsymbol{M_X}$, and the factor of $(N-1)/N$ is usually (but incorrectly) omitted.

The factor of $(G-1)/G$ in both (11) and (12) is designed to compensate for the tendency of the $\widehat{\boldsymbol{\beta}}^{(g)}$ to be too spread out. This factor is the analog of the usual factor of $(N-1)/N$ for a jackknife variance matrix at the individual level. It implicitly assumes that all clusters are the same size and perfectly balanced, with disturbances that are independent and homoskedastic. In this special case, the estimators CV_3 and $\text{CV}_{3\text{J}}$ would be identical and unbiased (Bell and McCaffrey 2002). These estimators are already available in Stata. When used with the `cluster()` option, the `vce(jackknife)` option computes $\text{CV}_{3\text{J}}$ standard errors, and the `vce(jackknife, mse)` option computes CV_3 standard errors. Because it is specialized for linear regression models, the implementation in `summclust` is much faster.

Both jackknife estimators may readily be used to compute cluster–robust t statistics. Because there are G terms in the summation, it seems natural to compare these with the $t(G-1)$ distribution, as usual. These procedures should almost always be more conservative than t tests based on the widely used CV_1 estimator. In an important recent article, Hansen (2022) shows that CV_3 has much better worst-case theoretical properties than CV_1. This strongly suggests that t statistics based on CV_3 are likely to yield lower rejection frequencies than ones based on CV_1. The simulation results in section 7 and in MacKinnon, Nielsen, and Webb (2023c) are consistent with this conjecture.

When a model includes fixed effects, some care must be taken when computing CV_3 and $\text{CV}_{3\text{J}}$. As noted in section 2.1, it is computationally attractive to partial out fixed effects prior to calculating $\widehat{\boldsymbol{\beta}}$. However, if we were to partial out any arbitrary regressors prior to computing the delete-one-cluster estimates in (9), then the computed $\widehat{\boldsymbol{\beta}}^{(g)}$ would depend on the values of the partialed-out regressors for the full sample, including those in the gth cluster. Consequently, the values of CV_3 and $\text{CV}_{3\text{J}}$ will be incorrect if we partial out any regressor that affects more than one cluster (such as industry-level fixed effects with firm-level clustering). The regressors that are partialed out must be cluster fixed effects or fixed effects at a finer level (such as firm-level fixed effects with industry-

level clustering), because each of them affects only one cluster. See the discussion of the `absorb()` and `fevar()` options in section 3.

The vector $\boldsymbol{\beta}$ can be identified for the full sample but not when one cluster is deleted. For example, consider the coefficient on a dummy variable that takes on nonzero values only for observations in the gth cluster. This coefficient cannot be identified when cluster g is omitted. In such a case, the matrix $\boldsymbol{X}^{\top}\boldsymbol{X} - \boldsymbol{X}_g^{\top}\boldsymbol{X}_g$ in (9) is singular, and CV_3 and $\text{CV}_{3\text{J}}$ cannot be computed using an ordinary matrix inverse. However, because `summclust` uses the `invsym()` function in Stata, which implements a generalized inverse, the offending element of $\widehat{\boldsymbol{\beta}}^{(g)}$ is simply replaced by 0. The command therefore checks whether any of the $\widehat{\beta}^{(g)}$ coefficients of interest are equal to 0 and issues a warning when they are; see section 3.

There may be more than one set of fixed effects that are invariant at the cluster level. For example, imagine an analysis of students' test scores where the researcher wants to control for both school and neighborhood fixed effects and cluster the standard errors at the state level. In this case, neither of Stata's built-in `regress` and `areg` commands can produce an estimate of CV_3 because the fixed effects for schools and neighborhoods in state g cannot be identified when state g is omitted. However, `summclust` can produce such an estimate.

2.3 What should be reported

We believe that investigators should routinely compute the L_g. They should also compute the L_{gj} for any coefficients of particular interest. In some cases, the L_g and the L_{gj} will be roughly proportional to the N_g (the cluster sizes). That in itself would be informative. It may be even more interesting, however, to find that the relative size of L_g or L_{gj} for some clusters g is much larger or much smaller than the relative size of N_g.

When there are few clusters, it is easy enough to look at all the N_g, $\widehat{\beta}_j^{(g)}$, L_g, and L_{gj} to see whether any clusters are unusually large, are unusually influential, or have unusually high leverage or partial leverage. Once G exceeds 10 or 12, however, it may be more informative to report summary statistics or to plot these quantities. The `summclust` command always reports the minimum, first quartile, median, mean, third quartile, and maximum of the N_g and the L_g. It also reports these quantities for the L_{gj} and the $\widehat{\beta}_j^{(g)}$ for the specified regressor j, and by default it provides a figure containing four scatterplots of the L_g and the L_{gj} against the N_g and the $\widehat{\beta}^{(j)}$; see sections 3 and 4.

Another possibility is to report a few summary statistics, as `summclust` also does. Consider a generic (positive) quantity a_g, which might denote any of N_g, L_g, or L_{gj} for $g = 1, \ldots, G$. It seems plausible that inference may be unreliable when any of the a_g vary substantially across clusters, and we provide some evidence to support this conjecture in section 7.

There are many measures of how much the distribution of the a_g differs from what it would be in the perfectly balanced case. One of these is the scaled variance

$$V_s(a_\bullet) = \frac{1}{(G-1)\bar{a}^2} \sum_{g=1}^{G} (a_g - \bar{a})^2 \tag{13}$$

where the argument $a_\bullet$ denotes the entire set of a_g for $g = 1, \ldots, G$ and $\bar{a}$ denotes the arithmetic mean, which is N/G for the N_g, k/G for the L_g, and $1/G$ for the L_{gj}. These are all positive numbers, so it is reasonable to scale by their squares. Larger values of V_s imply that the a_g are more variable across clusters, relative to their mean. We could report either V_s or its square root, which is often called the coefficient of variation. In the perfectly balanced case, $V_s = 0$. By default, `summclust` reports the coefficient of variation for the cluster sizes, the leverages, the partial leverages, and the $\widehat{\beta}_j^{(g)}$.

Another possibility, although valid only for positive quantities, is to report one or more alternative sample means. The more these differ from the arithmetic mean, the more heterogeneous the clusters must be. Three common alternatives to the arithmetic mean are the harmonic, geometric, and quadratic means:

$$\bar{a}_{\text{harm}} = \left(\frac{1}{G}\sum_{g=1}^{G} 1/a_g\right)^{-1}, \quad \bar{a}_{\text{geo}} = \left(\prod_{g=1}^{G} a_g\right)^{1/G}, \quad \text{and} \quad \bar{a}_{\text{quad}} = \left(\frac{1}{G}\sum_{g=1}^{G} a_g^2\right)^{1/2}$$

Unless all the a_g are the same, the harmonic and geometric means will be less than the arithmetic mean $\bar{a}$, and the quadratic mean (which has the same form as the root mean squared error of an estimator) will be greater than $\bar{a}$. `summclust` optionally reports all three of these alternative means, along with the ratio of each of them to $\bar{a}$. The three ratios provide scale-free measures of cluster heterogeneity; the closer they are to 1, the more homogeneous the clusters are.

Another way to quantify the heterogeneity of the cluster sizes and the regressors is to calculate G^*, the "effective number of clusters," as proposed in Carter, Schnepel, and Steigerwald (2017). The value of G^* depends on the coefficient j for which it is being computed and on a parameter ρ to be discussed below, so we denote it $G_j^*(\rho)$. It is defined as

$$G_j^*(\rho) = \frac{G}{1 + \Gamma_j(\rho)}, \quad \Gamma_j(\rho) = \frac{1}{G}\sum_{g=1}^{G} \left\{ \frac{\gamma_{gj}(\rho) - \bar{\gamma}_j(\rho)}{\bar{\gamma}_j(\rho)} \right\}^2, \quad \bar{\gamma}_j(\rho) = \frac{1}{G}\sum_{g=1}^{G} \gamma_{gj}(\rho) \tag{14}$$

where $0 \leq \rho \leq 1$ and the $\gamma_{gj}(\rho)$ are given by

$$\gamma_{gj}(\rho) = \boldsymbol{e}_j^\top (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}_g^\top \boldsymbol{\Omega}_g(\rho) \boldsymbol{X}_g (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{e}_j, \quad g = 1, \ldots, G \tag{15}$$

Here $\boldsymbol{e}_j$ is a k-vector with 1 in the jth position and 0 everywhere else, so $\boldsymbol{e}_j^\top (\boldsymbol{X}^\top \boldsymbol{X})^{-1}$ is the jth row of $(\boldsymbol{X}^\top \boldsymbol{X})^{-1}$, and $\boldsymbol{\Omega}_g(\rho)$ is an $N_g \times N_g$ matrix with 1 on the principal diagonal and ρ everywhere else. It is easy to see that

$$\boldsymbol{\Omega}_g(\rho) = \rho \boldsymbol{\iota}\boldsymbol{\iota}^\top + (1 - \rho)\boldsymbol{I} \tag{16}$$

where $\boldsymbol{\iota}$ is an N_g-vector of 1s and $\mathbf{I}$ is an $N_g \times N_g$ identity matrix. Note that $\Gamma_j(\rho)$ is just the scaled variance of the $\gamma_{gj}(\rho)$; compare (13).

The parameter ρ may be interpreted as the intracluster correlation coefficient for a model with cluster-level random effects. Because ρ is unknown, Carter, Schnepel, and Steigerwald (2017) suggest calculating $G_j^*(1)$ as a sort of worst case. However, when there are cluster-level fixed effects or fixed effects at a finer level nested within clusters, they will absorb all the intracluster correlation. Thus, it does not make sense to specify $\rho > 0$ in either of these cases. It does seem natural to use $G_j^*(0)$, however, because the amount of intracluster correlation that remains in models with cluster fixed effects is often quite small.

From (15) and (16), we see that

$$\boldsymbol{X}_g^\top \boldsymbol{\Omega}_g(\rho) \boldsymbol{X}_g = \rho(\boldsymbol{\iota}^\top \boldsymbol{X}_g)^\top (\boldsymbol{\iota}^\top \boldsymbol{X}_g) + (1 - \rho)\boldsymbol{X}_g^\top \boldsymbol{X}_g \tag{17}$$

This result makes it inexpensive to compute the $\gamma_{gj}(\rho)$ for any value of ρ by first computing them for $\rho = 0$ and $\rho = 1$. The needed equations are

$$\begin{aligned}
\gamma_{gj}(0) &= \boldsymbol{w}_j^\top \boldsymbol{X}_g^\top \boldsymbol{X}_g \boldsymbol{w}_j \\
\gamma_{gj}(1) &= (\boldsymbol{\iota}^\top \boldsymbol{X}_g \boldsymbol{w}_j)^\top (\boldsymbol{\iota}^\top \boldsymbol{X}_g \boldsymbol{w}_j) \\
\gamma_{gj}(\rho) &= \rho \gamma_{gj}(1) + (1 - \rho)\gamma_{gj}(0)
\end{aligned} \tag{18}$$

where $\boldsymbol{w}_j$ is the jth column of $(\boldsymbol{X}^\top \boldsymbol{X})^{-1}$. After we obtain the $\gamma_{gj}(\rho)$ from (18), it is trivial to compute $G_j^*(\rho)$ using (14). Evidently, $G_j^*(\rho)$ is always less than G. When it is much smaller than G, it can provide a useful warning.

Suppose that we have partialed out cluster fixed effects prior to computing $G_j^*(\rho)$. Then the first term on the right-hand side of (17) should theoretically be a zero matrix because every column of $\boldsymbol{X}_g$ should add to zero. In practice, however, the limitations of floating-point arithmetic mean that this matrix will actually contain extremely small positive numbers. This will cause the computation of $G_j^*(\rho)$ to be numerically unstable. When the fixed effects are not partialed out, similar but more complicated numerical issues arise.

The command `clusteff`, discussed in Lee and Steigerwald (2018), is designed to calculate $G_j^*(\rho)$, with $\rho = 0.9999$ rather than $\rho = 1$ by default to avoid numerical instabilities. However, the only version of this command that we have used does so in a computationally inefficient way that does not use (18). When any of the N_g is large, it can take a very long time or even fail because Stata runs out of memory. For example, it failed with some of the samples in MacKinnon, Nielsen, and Webb (2023a).

Like $V_s(a_\bullet)$ and the alternative sample means for measures of leverage and partial leverage discussed above, $G_j^*(\rho)$ is sensitive not only to variation in cluster sizes but also to other features of the $\boldsymbol{X}_g$ matrices. But it is not sensitive to heteroskedasticity or to any other features of the disturbances. `summclust` computes $G_j^*(0)$, $G_j^*(1)$, and (optionally) $G_j^*(\rho)$ for a specified covariate. However, when there are cluster fixed effects, or fixed effects nested within clusters, it computes only $G_j^*(0)$. For example, it will not

compute $G_j^*(\rho)$ for $\rho \neq 0$ whenever there are state-level fixed effects and clustering at the region level.

The quantity $G_j^*(0)$ is very closely related to $V_s(L_{\bullet j})$, where $L_{\bullet j}$ denotes the entire set of L_{gj}, for $g = 1, \ldots, G$. The $\gamma_g(0)$ defined in (15) and (18) are equal to the L_{gj} defined in (7) divided by $\hat{\boldsymbol{x}}_j^\top \hat{\boldsymbol{x}}_j$. Because this makes the $\gamma_g(0)$ proportional to the L_{gj}, $V_s(L_{\bullet j})$ is numerically identical to $\Gamma(0)$; compare (13) and the middle equation in (14). Thus, we see from the first equation in (14) that $G_j^*(0)$ is simply a monotonically decreasing function of the scaled variance of our measures of partial leverage at the cluster level. When $V_s(L_{\bullet j})$ is large, $G_j^*(0)$ is necessarily much smaller than G.

3 The summclust command

In this section, we describe the summclust command, which calculates many statistics to help assess cluster heterogeneity and also provides CV$_3$ and CV$_{3J}$ standard errors. The command does not rely on any other Stata commands, but it does require a version of Stata that provides Mata's panelsum() function (version 13 or later).

We first present an overview of the summclust command, followed by a simple illustration using nlswork.dta.

3.1 Syntax and options

3.1.1 Syntax

summclust *varlist*, cluster(*varname*) [*options*]

varlist specifies the dependent variable, the independent variable of interest, and other (binary or continuous) independent variables. At least one additional regressor must be specified. Time-series operators and factor variables are not permitted.

	options	Description
*	cluster(*varname*)	clustering variable, for which the number of unique values equals G
	fevar(*varlist*)	create fixed effects for each of the specified variables, using i.*varname*
	absorb(*varname*)	partial out the variable *varname* before computing other statistics; this option should be used only for variables that are nested within the specified clusters; it can often be computationally faster than using fevar() and should be used when there are cluster-level fixed effects to avoid singular omit-one-cluster samples caused by those fixed effects; in cases with an extremely large number of fixed effects, summclust may run into memory issues; if so, one can use the Stata prefix jackknife with the community-contributed command reghdfe (Correia 2014)
	jackknife	calculate the jackknife variance estimator CV_{3J} in addition to CV_3
	<u>add</u>means	display the alternative sample means of the N_g, L_g, L_{gj}, and $\widehat{\beta}_j^{(g)}$, as described in section 2.3; for the N_g, L_g, and L_{gj}, it reports the harmonic, geometric, and quadratic means, as well as the ratio of each of them to the arithmetic mean; for the $\widehat{\beta}_j^{(g)}$, which can be negative, only the quadratic mean and its ratio are reported because the harmonic and geometric means are not defined for negative numbers
	gstar	calculate the effective number of clusters $G^*(0)$ and, when there are no cluster (or subcluster) fixed effects, $G^*(1)$ as well
	<u>rho</u>(*scalar*)	calculate the effective number of clusters, $G^*(\text{rho})$, in addition to $G^*(0)$ and $G^*(1)$; this option can be used with or without the gstar option; the value of rho must be between 0 and 1; the program ends with an error message when an invalid value for rho is entered; if it is not valid to display $G^*(\text{rho})$ because of variables that are invariant within clusters, it reports that $G^*(\text{rho})$ cannot be computed and displays only $G^*(0)$; there is no reason to use the gstar option when this option is used
	<u>table</u>	display the cluster-by-cluster values of cluster size, leverage, partial leverage, and the delete-one-cluster coefficient estimate; if $G > 52$, then the unformatted matrix is displayed instead of a table

Continued on next page

options	Description
<u>sample</u>(*string*)	allow for sample restrictions; the arguments for this option are whatever would follow the `if` in a standard `regress` command; for instance, to restrict the analysis to individuals 25 years of age or older based on a variable `age`, `sample(age>=25)` should be added to the list of options
<u>nog</u>raph	suppress creation of the figure, which is otherwise shown by default
<u>reg</u>table	display a full table of regression output, similar to Stata's `regress` table, but with jackknife standard errors; it reports CV_3 standard errors by default, but it instead reports CV_{3J} standard errors when the `jackknife` option is also specified; if $k > 52$, then the unformatted matrix is displayed instead of a table

  * `cluster()` is required.

3.1.2 Description

`summclust` is a stand-alone command for summarizing cluster variability in several ways. It always calculates measures of cluster-level influence and leverage, and it optionally calculates the effective number of clusters. It also always reports CV_1 and CV_3 standard errors for one coefficient, and it optionally reports a CV_{3J} standard error as well. If requested, it can calculate additional measures of cluster-level heterogeneity. Unless it is told not to, it produces a figure that can help identify the source of cluster-level heterogeneity. Finally, it can optionally produce a full table of regression results with CV_3 standard errors.

By default, `summclust` calculates the CV_3 standard error based on (10). With well-behaved samples, this should match the standard error calculated using Stata's native `jackknife: regress y x, cluster(group)` or `regress y x, cluster(group) vce(jackknife)` commands. However, many samples are not well behaved, in that the regressor matrices for some of the omit-one-cluster subsamples may not have full rank. We will refer to such subsamples, rather informally, as "singular subsamples".

Whenever there are singular subsamples, `summclust` calculates two standard errors. One of these drops the singular subsamples as the native Stata commands do. The other uses a generalized inverse. `summclust` provides guidance as to which standard error is likely to be more reliable. When `regtable` is specified and singular subsamples are present, two versions of the regression table are displayed. Similarly, if `jackknife` is specified and there are singular subsamples, four different standard errors are shown, either CV_3 or CV_{3J}, combined with either the generalized inverse or one that drops the singular subsamples.

`nograph` suppresses creation of the figure, which is otherwise shown by default. The figure shows four scatterplots: leverage against observations per cluster, partial leverage

against observations per cluster, leverage against omit-one-cluster coefficients, and partial leverage against omit-one-cluster coefficients. This figure can be quite informative, but it is computationally costly to produce. We recommend invoking this option after the figure has been inspected.

When `jackknife` is specified, `regtable` uses the CV_{3J} estimates to produce the regression table. Otherwise, CV_3 estimates are used.

3.2   Illustration with nlswork

To illustrate `summclust`'s functionality and syntax, we consider a simple example using `nlswork.dta`, which contains a sample of women who were 14–26 years of age in 1968 from the National Longitudinal Survey of Young Working Women. For this exercise, we restrict the sample to individuals who are 20 to 40 years old.

We estimate a simple Mincer regression using `nlswork.dta` to see whether there is a marriage premium for wages. The variable `msp` is equal to 1 if the person is married and cohabits with their spouse and equal to 0 otherwise. For this example, we cluster by industry. The following code opens the dataset and estimates the regression using Stata's `regress` command:

```
webuse nlswork
keep if inrange(age,20,40)
regress ln_wage i.grade i.age i.birth_yr union race msp, cluster(ind)
```

The output from the command above provides CV_1 standard errors. Alternatively, we can estimate CV_3 and CV_{3J} standard errors using this code:

```
regress ln_wage i.grade i.age i.birth_yr union race msp, cluster(ind) ///
    vce(jackknife, mse)
regress ln_wage i.grade i.age i.birth_yr union race msp, cluster(ind) ///
    vce(jackknife)
```

When either of these commands is run, Stata displays the warning `Note: One or more parameters could not be estimated in 2 jackknife replicates; standard-error estimates include only complete replications.`

The coefficient on `msp` and two or three standard errors can also be obtained using `summclust`. The basic command is

```
summclust ln_wage msp union race, fevar(grade age birth_yr) cluster(ind)
```

This code results in the default output from `summclust`, which is mostly contained in two tables. The first one includes the coefficient on the second variable in the *varlist* (in this case `msp`), the CV_1 and CV_3 standard errors for this coefficient, and the associated t statistics, p-values, and confidence intervals. In this case, `summclust` also displays a warning about singular subsamples and thus produces two `Regression Output` tables. The standard errors in the table that drops singular subsamples match those produced natively in Stata.

(*output omitted*)

```
Cluster summary statistics for msp when clustered by ind_code.
There are 17395 observations within 12 ind_code clusters.
```

(*output omitted*)

```
Regression Output
```

s.e.	Coeff	Sd. Err.	t-stat	P value	CI-lower	CI-upper
CV1	-0.026940	0.008248	-3.2663	0.0075	-0.045093	-0.008787
CV3	-0.026940	0.011150	-2.4161	0.0342	-0.051481	-0.002399

```
Regression Output -- Dropping Singular Omit-One-Cluster Subsamples
```

s.e.	Coeff	Sd. Err.	t-stat	P value	CI-lower	CI-upper
CV3	-0.026940	0.006701	-4.0200	0.0030	-0.042099	-0.011780

In the first table for this example, the CV_1 and CV_3 standard errors are noticeably different, with the latter being considerably larger. However, in the second table, where the two singular subsamples are dropped, the CV_3 standard error becomes much smaller.

The `Cluster Variability` table from `summclust` (below) provides insight into what is happening. It reports summary statistics for N_g, L_g, L_{gj}, and $\widehat{\beta}_j^{(g)}$. Whenever singular subsamples are dropped, two sets of statistics are shown for $\widehat{\beta}_j^{(g)}$. The first (second-last column) uses all the jackknife subsamples with a generalized inverse standard error. The second (final column) uses only the nonsingular subsamples. We can see that the largest value of $\widehat{\beta}_j^{(g)}$ is considerably smaller (and therefore more different from the other values) when none of the subsamples is dropped. This explains why the CV_3 standard error is larger in the first table above than in the second one.

```
Cluster Variability
```

Statistic	Ng	Leverage	Partial L.	all bet_g	kept be_g
min	35.00	0.085945	0.000700	-0.032772	-0.032772
q1	144.50	0.633594	0.004399	-0.027655	-0.027917
median	905.00	2.794231	0.038554	-0.026891	-0.027082
mean	1449.58	4.583333	0.083333	-0.026398	-0.027571
q3	2112.50	6.190322	0.105043	-0.025268	-0.026587
max	5736.00	17.008305	0.353148	-0.019198	-0.024202
coefvar	1.19	1.166238	1.320154	0.131277	0.074100

It is evident from this table that the clusters are extremely heterogeneous. The largest cluster contains almost one-third of the sample and is 167 times the size of the smallest. There are also extreme differences in both leverage and partial leverage across clusters. The ratio of the largest to the smallest value is 198 for leverage and 504.5 for partial leverage. The sum of the leverages is $12 \times 4.583333 = 55$, which is the number of estimated coefficients. Although both sets of $\widehat{\beta}_j^{(g)}$ vary quite a bit, dropping one cluster never changes the sign of the coefficient.

The option `fevar()` is used when there are factor variables, which would be specified as `i.`*varname* in conventional Stata syntax. In the above example, the arguments to

`fevar()` are `grade`, `age`, and `birth_yr`. For each argument, a set of temporary dummy variables is created. These dummy variables are included in the regression, and there is no constant term if they are present.

The sample code above does not illustrate several additional options. The most important of these is the `absorb()` option, which operates like `fevar()`. It treats its argument, a single variable, as an additional factor variable to include in the set of regressors. `absorb(`*varname*`)` can be used when including `i.`*varname* in a regression would result in many fixed effects. Speed can often be increased, perhaps substantially, by partialing out the absorbed fixed effects from the dependent and all the independent variables. It is advisable to use `absorb()` rather than `fevar()` whenever their argument corresponds to a set of cluster fixed effects because the elements of $\widehat{\boldsymbol{\beta}}^{(g)}$ that correspond to the fixed effects cannot be identified in that case; see section 2.1.

The `absorb()` option should be used with care. Partialing out fixed effects is valid for the measures of leverage and influence and for the jackknife variance matrices only when the absorbed variable yields fixed effects that can be partialed out on a cluster-by-cluster basis. That is, `absorb()` should be used only for straight cluster fixed effects or for fixed effects at a finer level, such as state $\times$ year fixed effects for a panel with clustering at the state level. It is not valid to partial out fixed effects that are not limited to one cluster. In that case, the $\widehat{\boldsymbol{\beta}}^{(g)}$ and quantities based on them would be different for the original data and the data after partialing out because the partialed-out observations for the gth cluster would depend on other clusters as well. Accordingly, `summclust` checks to ensure that the clustering variable is invariant within each value of the absorbed variable. When it is not invariant, a warning is displayed, and the values of L_g, L_{gj}, $\widehat{\beta}_j^{(g)}$, CV_3, and $\mathrm{CV}_{3\mathrm{J}}$ are not available.

To see the difference between `fevar()` and `absorb()`, we can estimate an expanded regression that includes industry fixed effects. Consider the following two commands:

```
summclust ln_wage msp union race, cluster(ind) fevar(grade age birth_yr ind)
summclust ln_wage msp union race, cluster(ind) fevar(grade age birth_yr) ///
    absorb(ind)
```

For the command that uses `fevar()` for all the categorical variables, some of the output is

```
Regression Output
   s.e. |      Coeff   Sd. Err.   t-stat  P value    CI-lower    CI-upper
    CV1 |  -0.018955   0.007014  -2.7025   0.0206   -0.034392   -0.003517
    CV3 |  -0.018955   0.007586  -2.4987   0.0296   -0.035651   -0.002258
```

Because every one of the jackknife subsamples is singular, only the results based on the generalized inverse are reported. In contrast, when `absorb()` is used for the industry fixed effects, the corresponding output is instead

```
Regression Output
  s.e. │      Coeff   Sd. Err.   t-stat  P value   CI-lower   CI-upper

   CV1 │  -0.018955   0.007014  -2.7025   0.0206  -0.034392  -0.003517
   CV3 │  -0.018955   0.007586  -2.4987   0.0296  -0.035651  -0.002258


Regression Output -- Dropping Singular Omit-One-Cluster Subsamples

  s.e. │      Coeff   Sd. Err.   t-stat  P value   CI-lower   CI-upper
   CV3 │  -0.018955   0.004173  -4.5418   0.0014  -0.028396  -0.009514
```

These two tables highlight a key reason for using `absorb()`. Because only two of the jackknife subsamples are singular, `summclust` can report both standard errors. Observe that when all 12 jackknife samples are used, the standard errors are the same regardless of whether industry fixed effects are specified using `fevar()` or `absorb()`.

Using the `fevar()` option leads to the output below for the measures of cluster variability:

```
Cluster Variability
 Statistic │        Ng     Leverage   Partial L.  beta no g

       min │     35.00     1.079703     0.000276  -0.021394
        q1 │    144.50     1.617131     0.003970  -0.020316
    median │    905.00     3.752372     0.033630  -0.019050
      mean │   1449.58     5.500000     0.083333  -0.018880
        q3 │   2112.50     7.066207     0.092329  -0.018852
       max │   5736.00    17.728424     0.382133  -0.012367

   coefvar │      1.19     0.957329     1.422090   0.126464
```

Using the `absorb()` option leads instead to the output below:

```
Cluster Variability
 Statistic │       Ng    Leverage   Partial L.  all bet_g   kept be_g

       min │    35.00    0.079703     0.000700  -0.021394   -0.021394
        q1 │   144.50    0.617131     0.004399  -0.020316   -0.020601
    median │   905.00    2.752372     0.038554  -0.019050   -0.019281
      mean │  1449.58    4.500000     0.083333  -0.018880   -0.019538
        q3 │  2112.50    6.066207     0.105044  -0.018852   -0.019028
       max │  5736.00   16.728424     0.353143  -0.012367   -0.016767

   coefvar │     1.19    1.170068     1.320148   0.126464    0.061639
```

The $\widehat{\beta}_j^{(g)}$ when all clusters are retained are identical for both options. But because there are two singular subclusters, there are two versions of the $\widehat{\beta}_j^{(g)}$ for the `fevar()` results.

The leverage estimates are also smaller when we use the `absorb()` option. Recall that, for the original model with no industry fixed effects, the leverages summed to 55. In the first case just above, where the industry fixed effects are included as regressors in `fevar()`, the regression has 66 coefficients, and so the leverages sum to $12 \times 5.5 = 66$. In the second case, where the industry fixed effects are partialed out using `absorb()`,

the regression has 54 coefficients, and so the leverages sum to $12 \times 4.5 = 54$. Thus, for the first case each of the leverages is greater than the corresponding one for the second case by precisely 1.

3.2.1 Examples

In the examples that follow, we include the `nograph` option to reduce computational time.

This example illustrates the `jackknife` and `table` options:

```
summclust ln_wage msp union race, cluster(ind) fevar(grade age birth_yr) ///
    nograph jackknife table
```

Regression Output

s.e.	Coeff	Sd. Err.	t-stat	P value	CI-lower	CI-upper
CV1	-0.026940	0.008248	-3.2663	0.0075	-0.045093	-0.008787
CV3	-0.026940	0.011150	-2.4161	0.0342	-0.051481	-0.002399
CV3J	-0.026940	0.011004	-2.4482	0.0324	-0.051160	-0.002720

In addition to the two standard tables, there is the following table:

Cluster by Cluster Statistics

ind_code	Ng	Leverage	Partial L.	beta no g
1	119	0.581881	0.002825	-0.026959
2	35	0.085945	0.000700	-0.027206
3	170	0.685307	0.005341	-0.026823
4	3451	12.753229	0.241651	-0.021861
5	974	2.448713	0.114532	-0.024202
6	2626	7.815303	0.095555	-0.027393
7	1599	4.565341	0.048163	-0.026587
8	513	2.494440	0.018808	-0.029519
9	836	3.131195	0.028945	-0.032772
10	114	0.336320	0.003457	-0.027917
11	5736	17.008305	0.353148	-0.019198
12	1222	3.094021	0.086874	-0.026333

This table makes it easy to see whether the high leverage clusters are also the largest clusters. That is clearly the case here. After the program runs, this table is stored as the Mata matrix `scall`.

To obtain summary statistics on the four (or five) measures of cluster variability, we can use the `addmeans` option:

```
summclust ln_wage msp union race, cluster(ind) fevar(grade age birth_yr) ///
    nograph addmeans
```

This command produces the following table:

```
Alternative Sample Means and Ratios to Arithmetic Mean
                       Ng     Leverage  Partial L.  all bet_g   kept be_g

  Harmonic Mean    206.576    0.608440   0.004988        .           .
  Harmonic Ratio     0.143    0.132751   0.059853        .           .
  Geometric Mean   623.091    2.042731   0.025557        .           .
 Geometric Ratio     0.430    0.445687   0.306684        .           .
  Quadratic Mean  2193.268    6.870062   0.134308   0.026605    0.027654
 Quadratic Ratio     1.513    1.498923   1.611699  -1.007868   -1.003015
```

Once again, we see that there is extreme variability across the clusters. This is particularly noticeable for the ratio of the harmonic mean to the arithmetic mean, which is between 0.060 and 0.143 for the cluster size, leverage, and partial leverage measures. Recall that these ratios would be close to 1 if the clusters were relatively homogeneous. This table is stored in Mata's memory as `bonus`.

To obtain estimates of the effective number of clusters, we can use either the `gstar` option or the `rho()` option. The former displays $G_j^*(0)$ and $G_j^*(1)$. The latter requires a specified value of ρ and displays $G_j^*(0)$ and $G_j^*(1)$ along with $G_j^*(\rho)$. When there are fixed effects at the cluster or subcluster level, only $G_j^*(0)$ is reported.

For the `nlswork` example, the first option may be called as

```
summclust ln_wage msp union race, cluster(ind) fevar(grade age birth_yr) ///
    nograph gstar
```

This yields

```
Effective Number of Clusters
----------------------------
G*(0)  =  5.495
G*(1)  =  1.376
----------------------------
```

The second option, using $\rho = 0.5$ as an illustration, may be called as

```
summclust ln_wage msp union race, cluster(ind) fevar(grade age birth_yr) ///
    nograph rho(0.5)
```

This yields

```
Effective Number of Clusters
----------------------------
G*(0)  =  5.495
G*(.5) =  1.433
G*(1)  =  1.376
----------------------------
```

In this example, the effective number of clusters is clearly substantially less than the actual number of clusters. This provides more evidence that inference using the CV_1 standard error together with the $t(G-1)$ distribution is likely to be unreliable. These three

quantities can be accessed in Mata's memory as `gstarzero`, `gstarrho`, and `gstarone`, respectively.

By using the `regtable` option, one can display a modified version of the regression table that is similar to the default output from Stata's `regress` command. The command is

```
summclust ln_wage msp union race, cluster(ind) fevar(grade age birth_yr) ///
    nograph regtable
```

When there are singular subsamples, two versions of this table will be displayed. In this example, the table is quite long, so we do not reproduce it here.

3.3   List of stored results

All the results that are displayed as output can also be found in Mata's memory. To access one of these after running `summclust`, simply add the following line:

```
mata: object_name
```

The *object_name* can take one of the following values:

`cvstuff`: This matrix stores the table with the title `Regression Output`. It is 2×6 when the `jackknife` option is not used (the default) and 3×6 when `jackknife` is used.

`clustsum`: The matrix with the measures of cluster variability.

`scall`: This matrix stores the $G \times 4$ table created by the `table` option with the title `Cluster by Cluster Statistics`.

`bonus`: This 6×4 matrix contains the alternative sample means and their ratios to the arithmetic mean created by the `addmeans` option.

`cnames`: The string matrix with the cluster names, to match with elements in `scall`. This matrix is calculated only when the option `table` is specified.

`gstarzero`: This scalar contains $G^*(0)$, created by the `gstar` or `rho()` option.

`gstarone`: This scalar contains $G^*(1)$, created by the `gstar` or `rho()` option.

`gstarrho`: This scalar contains $G^*(\rho)$, created by the `rho()` option.

`regresstab`: This matrix contains the table shown when the `regtable` option is specified.

Scalars within matrices can be referenced on a cell-by-cell basis. For example, the CV$_3$ standard error is stored in the second row and second column of `cvstuff`, and to display it one can enter the following command:

```
mata: cvstuff[2,2]
```

Additionally, several results are available as scalars or matrices in return memory using `r()`. The available scalars are

Macros

`r(gstarzero)`	effective number of clusters for the coefficient of interest using $\rho = 0$
`r(gstarrho)`	effective number of clusters for the coefficient of interest using the value of ρ specified in `rho`(ρ)
`r(gstarone)`	effective number of clusters for the coefficient of interest using $\rho = 1$
`r(beta)`	estimate $\widehat{\beta}$ for the coefficient of interest
`r(cv1se)`	CV_1 standard error for the coefficient of interest
`r(cv1t)`	CV_1 t statistic for the coefficient of interest
`r(cv1p)`	p-value for the null hypothesis that $\beta = 0$ for the coefficient of interest using the CV_1 standard error
`r(cv1lci)`	lower bound of the 95% confidence interval for β using the CV_1 standard error
`r(cv1uci)`	upper bound of the 95% confidence interval for β using the CV_1 standard error

The standard error, t statistic, p-value, and confidence interval bounds are also available for the CV_3 and CV_{3J} standard errors. To access these, replace "1" in the above with either "3" or "3J"; for example, the p-value using CV_{3J} is available in `cv3Jp`. In the event of singular subsamples, there are two versions of the CV_3 or CV_{3J} results. The ones where singular subsamples have been dropped have a suffix of `drop`. For instance, `cv3sedrop` is used instead of `cv3se`.

The available matrices are

`r(ng)`	$G \times 1$ matrix contains the number of observations, N_g, for each cluster
`r(leverage)`	$G \times 1$ matrix contains the leverage, L_g, for each cluster
`r(partlev)`	$G \times 1$ matrix contains the partial leverage, L_{gj}, for each cluster
`r(betanog)`	$G \times 1$ matrix contains the $\widehat{\beta}_j^{(g)}$ for each cluster

4 Empirical example

We consider an empirical example from Busso and Galiani (2019) that studies an experiment where retail firms were randomly assigned to enter one of 72 different geographic markets (*mercados* in Spanish) within the Dominican Republic. After randomization, 21 markets had no entrants and so were in the control group, 18 had one entrant, another 18 had two, and the remaining 15 had three. The primary analysis distinguishes only between the 51 treated markets and the 21 control markets. The number of observations (stores) per market varies from 20 to 55.

This example is interesting because conventional wisdom (for example, MacKinnon, Nielsen, and Webb [2023a]) suggests that, with 72 clusters that do not vary much in size, and with neither few treated nor few control clusters, inference based on CV_1 standard errors and the $t(71)$ distribution should work well. However, our leverage measures suggest otherwise, and alternative inference methods yield noticeably different results.

The model we fit is

$$Y_{sd} = \alpha + \gamma Z_d + \boldsymbol{X}_{sd}\boldsymbol{\beta} + \epsilon_{sd} \qquad (19)$$

Here s indexes stores and d indexes markets. The treatment variable Z_d equals 1 if market d is treated (there was entry) and 0 if it was a control (there was no entry). The coefficient of interest is γ, which measures the causal effect of increased competition on an outcome Y. We focus on just one of several outcomes, namely, the log of demeaned prices after treatment. The results from this regression are found in table 5, panel B, column 4, row 1 of Busso and Galiani (2019). The table states that there are 72 clusters and 2,025 observations; however, the replication dataset that we use contains just 1,926 observations.

Regression (19) includes 17 control variables in the row vector $\boldsymbol{X}_{sd}$. These are the first lag of the outcome variable, the number of retailers in each district pretreatment, a lagged quality index, eight province fixed effects, the total district beneficiaries of a conditional cash transfer program, the percent beneficiaries of that program, the average income in the market, two market education measures, and a binary indicator for the urban status of the market. Thus, the total number of regressors is 19.

The OLS estimate of γ, its CV$_1$ standard error, the p-value for a test that $\gamma = 0$, and a 0.95 confidence interval are shown in the first row of table 1. Allowing for different numbers of reported digits, these estimates accord with the ones in Busso and Galiani (2019). The estimate of -0.01469 has the expected sign (average prices declined). However, the p-value is just slightly less than 0.05, and the confidence interval barely excludes 0.

Table 1. Estimates of the treatment effect

Method	$\widehat{\gamma}$	Standard error	p-value	Confidence interval
CV$_1$	-0.01469	0.007243	0.0461	$[-0.02913, -0.00025]$
CV$_2$	-0.01469	0.008078	0.0730	$[-0.03080, 0.00142]$
CV$_3$	-0.01469	0.009090	0.1105	$[-0.03281, 0.00343]$
CV$_{3J}$	-0.01469	0.009087	0.1104	$[-0.03281, 0.00343]$
WRC-C bootstrap	-0.01469		0.0891	$[-0.03121, 0.00243]$
WCR-S bootstrap	-0.01469		0.0913	$[-0.03121, 0.00254]$

NOTES: There are $N = 1926$ observations and $G = 72$ clusters. The two WCR bootstraps use $B = 999{,}999$ and a seed of 56,829,046. WRC-C is the classic WCR bootstrap of Cameron, Gelbach, and Miller (2008), and WCR-S is the "score" variant proposed in MacKinnon, Nielsen, and Webb (2023c). It involves transforming the restricted empirical scores in a way based on the jackknife, but it still uses CV$_1$. The bootstrap results were obtained using version 4.2.0 of `boottest`.

We next use the `summclust` command to calculate the cluster-level characteristics of the model and dataset. Some key ones are reported in table 2. It is evident that cluster sizes are well balanced, varying from 20 to 55, with the first and third quartiles equal to 24 and 27. However, both the leverages L_g and the partial leverages L_{g1} vary considerably. The former range from 0.1308 to 0.7378, and the latter from 0.0001 to 0.0642. The coefficients of variation are 0.3887 and 1.0598, respectively. The latter is moderately large, although not enormous. The two values of G^* are slightly smaller than $G/2$, which also suggests that the sample is not well balanced.

Table 2. Leverage and partial leverage for $\widehat{\gamma}$

Statistic	N_g	Leverage	Partial leverage	$\widehat{\gamma}^{(g)}$
Minimum	20	0.130842	0.000099	−0.017550
First quartile	24	0.204104	0.003166	−0.015089
Median	26	0.235813	0.009001	−0.014791
Mean	26.75	0.263889	0.013889	−0.014663
Third quartile	27	0.292042	0.020926	−0.014070
Maximum	55	0.737797	0.064242	−0.010723
Coef. of variation	0.21	0.388686	1.059813	0.074061

NOTES: There are $N = 1926$ observations and $G = 72$ clusters. The effective numbers of clusters are $G^*_\gamma(0) = 34.16$ and $G^*_\gamma(1) = 33.33$.

The coefficient of variation of the $\widehat{\gamma}^{(g)}$ is small because most of them do not vary much. However, the most extreme values are notable. The estimate of γ, which is -0.01469, could be as small as -0.01755 or as large as -0.01072 if just 1 out of the 72 clusters was dropped.

These results suggest that CV_1, the default CRVE, may not be particularly reliable in this case. We therefore consider five alternative procedures. The second, third, and fourth rows of table 1 report the CV_2, CV_3, and CV_{3J} standard errors, along with the p-values and confidence intervals associated with them. The CV_2 p-value is noticeably larger than the CV_1 one and suggests that the estimate is not significant at the 0.05 level. The CV_3 and CV_{3J} rows are almost identical. At 0.1105, the CV_3 p-value does not even allow us to reject the null at the 0.10 level. The fifth and six rows of table 1 report two WCR bootstrap p-values and the associated 0.95 confidence intervals. At 0.0891 and 0.0913, these are a bit smaller than the jackknife ones, but they clearly do not allow us to reject the null hypothesis at the 0.05 level.

In view of the reasonably large number of clusters and the fact that cluster sizes do not vary much, the large discrepancy between the results for CV_1 and the other procedures may seem surprising. However, it is not all that surprising when we note how much the leverages and, especially, the partial leverages vary.

By default, `summclust` produces a figure like figure 1, with its title created by the program using the name of the clustering variable, in this case `mercado`. This figure plots both leverage and partial leverage against the number of observations per cluster and

also against the omit-one-cluster coefficients. These four subfigures may help to reveal the source of cluster-level heterogeneity. For this example, neither the large leverages nor the large partial leverages come exclusively from clusters with many observations or extreme omit-one-cluster coefficients.
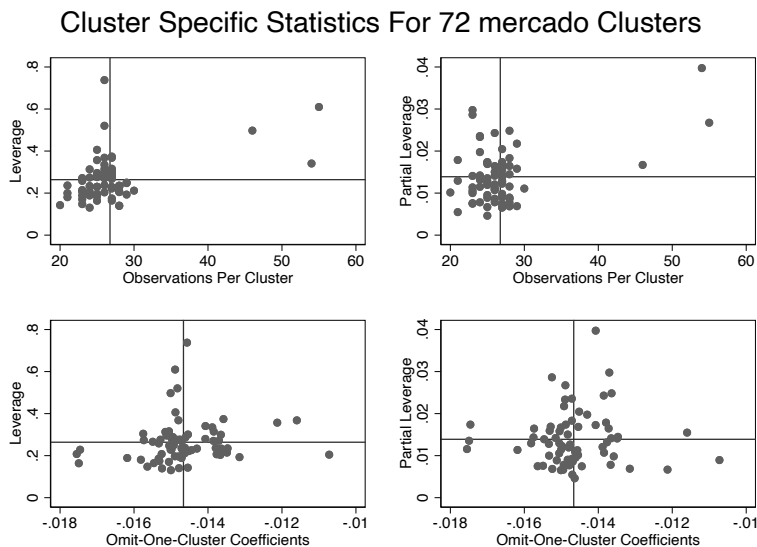

Figure 1. Example `summclust` figure. NOTES: A figure like this is always produced unless the `nograph` option is specified. It plots both leverage and partial leverage against cluster size and against the omit-one-cluster coefficients for, in this case, 72 clusters specified by a variable called `mercado`.

To explore what is driving the differences in partial leverage, we create an additional scatterplot. Figure 2 plots partial leverage against the number of observations per cluster, with different symbols depending on whether a given market (cluster) was treated. The figure has two interesting features. The first is that the three rather large clusters have fairly small partial leverage. The second is that the 12 clusters with the highest partial leverage are all control markets. The first result is quite surprising because large clusters often have high leverage. But figure 2 makes it clear that there is, in general, no simple relationship between cluster sizes and partial leverage. The second result is not so surprising, because only 21 out of the 72 clusters are controls. Many of the control clusters presumably have high partial leverage because control clusters are relatively rare. See (32) in section 5.4 for an explanation.
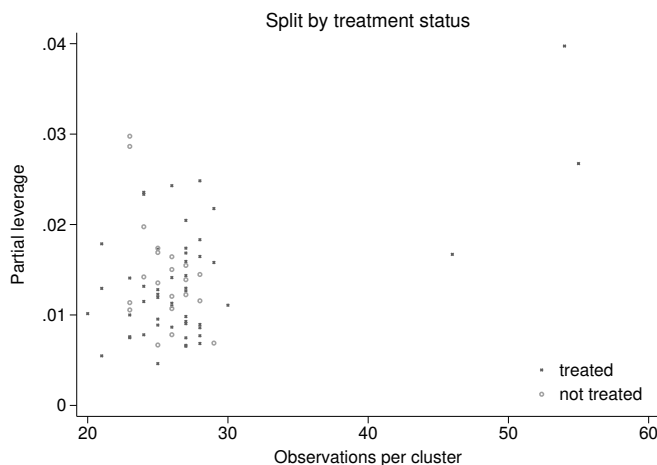
Figure 2. Partial leverage versus cluster size. NOTES: The figure plots partial leverage against cluster size for 72 clusters. An X marks a treated cluster, and a circle marks a control cluster.

5   Simple analytical examples

In this section, we discuss several simple examples in which we can calculate our measures of leverage and influence analytically. These examples are quite revealing.

5.1   Estimation of the mean

Finding the sample mean is equivalent to performing a least-squares regression in which the only regressor is $x_i = 1$ for all $i = 1, \ldots, N$. In this case, it is easy to see that $\boldsymbol{X}_g^\top \boldsymbol{X}_g = N_g$ and $\boldsymbol{X}^\top \boldsymbol{X} = N$. Therefore,

$$L_g = \mathrm{Tr}\left(\boldsymbol{H}_g\right) = \frac{N_g}{N} = \frac{N_g}{\sum_{h=1}^G N_h} \tag{20}$$

In this simple case, cluster leverage is exactly proportional to cluster size. In other cases, we can interpret leverage as a generalization of cluster size that also accounts for other types of heterogeneity.

Evidently, $\widehat{\beta} = \overline{y} = N^{-1} \sum_{g=1}^{G} N_g \overline{y}_g$, where $\overline{y}$ and $\overline{y}_g$ denote the sample average for the full sample and for cluster g, respectively. This expression can be rewritten as

$$\widehat{\beta} = \sum_{g=1}^{G} \frac{N_g}{N} \overline{y}_g = \sum_{g=1}^{G} L_g \widehat{\beta}_g \qquad (21)$$

so that $\widehat{\beta}$ is seen to be a weighted average of the G estimates $\widehat{\beta}_g = \overline{y}_g$, with the weight for each cluster equal to its leverage. Similarly, we find that

$$\widehat{\beta}^{(g)} = \frac{N}{N - N_g} \sum_{h \neq g} L_h \widehat{\beta}_h \qquad (22)$$

where the first factor simply makes up for the fact that we are summing over $G - 1$ clusters instead of G as in (21). Subtracting (21) from (22), we conclude that

$$\widehat{\beta}^{(g)} - \widehat{\beta} = \frac{N_g}{N} \left(\widehat{\beta}^{(g)} - \widehat{\beta}_g \right) = L_g \left(\widehat{\beta}^{(g)} - \widehat{\beta}_g \right) \qquad (23)$$

Therefore, cluster g will be influential whenever omitting it yields an estimate $\widehat{\beta}^{(g)}$ that differs substantially from the estimate $\widehat{\beta}_g$ for cluster g itself, especially when cluster g also has high leverage.

5.2   Single regressor plus constant

Consider a regression design with one regressor, x_i, and a constant term. Then

$$\boldsymbol{X}_g^\top \boldsymbol{X}_g = \begin{bmatrix} N_g & \sum_{i=1}^{N_g} x_{g,i} \\ \sum_{i=1}^{N_g} x_{g,i} & \sum_{i=1}^{N_g} x_{g,i}^2 \end{bmatrix}, \quad (\boldsymbol{X}^\top \boldsymbol{X})^{-1} = \frac{1}{N^2 \widehat{\sigma}_x^2} \begin{bmatrix} \sum_{i=1}^{N} x_i^2 & -\sum_{i=1}^{N} x_i \\ -\sum_{i=1}^{N} x_i & N \end{bmatrix}$$

where $\widehat{\sigma}_x^2$ denotes the sample variance of the x_i. After some algebra, we find that

$$L_g = \frac{N_g}{N \widehat{\sigma}_x^2} \left\{ \widehat{\sigma}_x^2 + \widehat{\sigma}_{x,g}^2 + (\overline{x}_g - \overline{x})^2 \right\} \qquad (24)$$

where $\overline{x}_g$ and $\widehat{\sigma}_{x,g}^2$ denote the sample mean and sample variance of the x_i within cluster g. Expression (24) is a straightforward generalization of (20). The last two terms within the large braces are the sample variance of the $x_{g,i}$ within cluster g and the square of the difference between $\overline{x}_g$ and $\overline{x}$. The sum of these terms is the sample variance of the $x_{g,i}$ around $\overline{x}$ within cluster g. Thus, cluster g will have high leverage when the variance of the $x_{g,i}$ around $\overline{x}$ within that cluster is large relative to the variance $\widehat{\sigma}_x^2$ for the full sample. If everything except cluster sizes were perfectly balanced, L_g would evidently reduce to $2N_g/N$.

The partial leverage for x is just

$$L_{g2} = \frac{N_g \left\{ \widehat{\sigma}_{x,g}^2 + (\overline{x}_g - \overline{x})^2 \right\}}{N \widehat{\sigma}_x^2} \qquad (25)$$

the total variation around $\overline{x}$ within cluster g divided by the total variation within the sample. If everything except cluster sizes were perfectly balanced, it would reduce to N_g/N.

5.3 One regressor plus fixed effects

Suppose there is one regressor, x_i, and there are cluster-level fixed effects, which have been partialed out. In this case, we can write all quantities as deviations from their cluster averages, and there is no distinction between leverage and partial leverage. Then $\widetilde{\boldsymbol{X}}_g^\top \widetilde{\boldsymbol{X}}_g = \sum_{i=1}^{N_g} (x_{g,i} - \overline{x}_g)^2 = N_g \widehat{\sigma}_{x,g}^2$. Similarly, $\widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{X}} = \sum_{g=1}^{G} N_g \widehat{\sigma}_{x,g}^2$ is the average variance of the x_i across all clusters. We find that

$$L_g = \frac{N_g \widehat{\sigma}_{x,g}^2}{\sum_{h=1}^{G} N_h \widehat{\sigma}_{x,h}^2} \tag{26}$$

which is again a straightforward generalization of (20). The leverage of cluster g is proportional to N_g times the variance of the $x_{g,i}$ around $\overline{x}_g$. Thus, for example, doubling the variance of the $x_{g,i}$ has the same effect on leverage as doubling N_g.

In this case, using (26), we easily see that

$$\widehat{\beta} = \frac{\sum_{g=1}^{G} N_g \widehat{\sigma}_{xy,g}}{\sum_{g=1}^{G} N_g \widehat{\sigma}_{x,g}^2} = \sum_{g=1}^{G} L_g \frac{\widehat{\sigma}_{xy,g}}{\widehat{\sigma}_{x,g}^2} = \sum_{g=1}^{G} L_g \widehat{\beta}_g \tag{27}$$

where $\widehat{\sigma}_{xy,g} = (1/N_g) \sum_{i=1}^{N_g} (x_{g,i} - \overline{x}_g)(y_{g,i} - \overline{y}_g)$ is the sample covariance of x_i and y_i within cluster g. The rightmost expressions in (21) and (27) are identical. In both cases, $\widehat{\beta}$ is seen to be a weighted average of the G cluster estimates, with the weight for each cluster equal to its leverage.

When cluster g is omitted, we obtain

$$\widehat{\beta}^{(g)} = \frac{\sum_{h \neq g} N_h \widehat{\sigma}_{xy,h}}{\sum_{h \neq g} N_h \widehat{\sigma}_{x,h}^2} = \frac{\sum_{h \neq g} L_h \widehat{\beta}_h}{\sum_{h \neq g} L_h} \tag{28}$$

which would specialize to (22) if (20) were true. As before, $\widehat{\beta}^{(g)}$ is a weighted average of the $\widehat{\beta}_h$, with weights proportional to the L_g, which in this case are also the partial leverages. Subtracting (27) from (28), we find that

$$\widehat{\beta}^{(g)} - \widehat{\beta} = L_g \left(\widehat{\beta}^{(g)} - \widehat{\beta}_g \right) \tag{29}$$

which is formally identical to the rightmost expression in (23), although of course L_g is defined in (26), not (20). Cluster g will be influential whenever $\widehat{\beta}^{(g)}$ differs substantially from the estimate $\widehat{\beta}_g$ for cluster g itself, especially when cluster g also has high leverage.

5.4   Treatment model with a constant term

Now we specialize section 5.2 to the case in which the single regressor is a treatment dummy denoted by d_i. Let $\bar{d}_g$ and $\bar{d}$ denote the proportion of treated observations in cluster g and in the sample, respectively. Then (24) becomes

$$L_g = \frac{N_g}{N}\left(\frac{\bar{d}_g}{\bar{d}} + \frac{1 - \bar{d}_g}{1 - \bar{d}}\right) \tag{30}$$

The first factor here is the relative size of the gth cluster. The second factor depends on how much $\bar{d}_g$ differs from $\bar{d}$. When $\bar{d}_g = \bar{d}$, we see that $L_g = 2N_g/N$. Otherwise, the first term inside the parentheses causes leverage to be high whenever $\bar{d}_g$ is large relative to $\bar{d}$, and the second term causes leverage to be high whenever $\bar{d}_g$ is small relative to $\bar{d}$. As $\bar{d}$ increases for a given $\bar{d}_g$, the first term becomes smaller relative to the second term. Thus, the gth cluster will tend to be influential when it has either a large proportion of treated observations and the overall proportion is small or a small proportion of treated observations and the overall proportion is large.

We can also obtain the partial leverage of the treatment dummy for this case. Expression (25) simply becomes

$$L_{g2} = \frac{N_g}{N}\left(\frac{\bar{d}_g}{\bar{d}} + \frac{\bar{d} - \bar{d}_g}{1 - \bar{d}}\right) \tag{31}$$

Once again, the first factor is the relative size of the gth cluster. The second factor reduces to 1 when $\bar{d}_g = \bar{d}$, so that $L_{g2} = N_g/N$ in that special case.

We can further specialize (30) and (31) to models in which the treatment is applied at the cluster level. Suppose that all observations in clusters $g = 1, \ldots, G_1$ are treated and no observations in the $G_0 = G - G_1$ control clusters from $G_1 + 1$ to G are treated. Then we find that $\bar{d}_g = 1$ for $g = 1, \ldots, G_1$ and $\bar{d}_g = 0$ for $g = G_1 + 1, \ldots, G$. Inserting these into (30) shows that

$$L_g = \begin{cases} \frac{N_g}{N}\frac{1}{\bar{d}} & \text{for } g = 1, \ldots, G_1, \\[2mm] \frac{N_g}{N}\frac{1}{1-\bar{d}} & \text{for } g = G_1 + 1, \ldots, G \end{cases} \tag{32}$$

Inserting them into (31) shows that

$$L_{g2} = \begin{cases} \frac{N_g}{N}\frac{\bar{d}+1}{\bar{d}} & \text{for } g = 1, \ldots, G_1, \\[2mm] \frac{N_g}{N}\frac{\bar{d}}{1-\bar{d}} & \text{for } g = G_1 + 1, \ldots, G \end{cases}$$

Thus, any cluster tends to have high leverage if N_g/N is large. A treated cluster has high leverage and partial leverage if $\bar{d}$ is small. Conversely, a control cluster has high leverage and partial leverage if $\bar{d}$ is large.

5.5 Treatment with fixed effects

Finally, we consider the case of cluster-level fixed effects, where treatment is randomly applied at the individual level. This is a special case of section 5.3. We cannot consider cluster fixed effects with cluster-level treatment, because the treatment dummy would be invariant within clusters. We specialize (26) and find that

$$
L_g = \frac{N_g \overline{d}_g \left(1 - \overline{d}_g \right)}{\sum_{h=1}^{G} N_h \overline{d}_h \left(1 - \overline{d}_h \right)} \tag{33}
$$

Thus, as before, the leverage of cluster g, relative to the average for the other clusters, is proportional to its size, N_g. It also depends on the proportion of treated observations in the cluster. The maximum (relative) leverage for cluster g occurs at $\overline{d}_g = 1/2$ and is symmetric around $1/2$. The result (29) continues to hold. It tells us that cluster g will be influential when its leverage (33) is large and $\widehat{\beta}^{(g)}$ differs greatly from $\widehat{\beta}_g$.

6 Two-way clustering

Up to this point, we have focused on one-way clustering. However, it is also important to compute measures of leverage, partial leverage, and influence when there is clustering in two or more dimensions (Cameron, Gelbach, and Miller 2011). In the simplest and most commonly encountered case, where there is two-way clustering, we recommend computing the usual one-way measures of leverage, partial leverage, and influence for each of the two clustering dimensions. This requires calling `summclust` twice.

When the number of clusters in either dimension is small or when the data are seriously unbalanced in either dimension, conventional inference based on a two-way version of CV$_1$, together with the $t\big(\min(G-1, H-1)\big)$ distribution, can be seriously unreliable. MacKinnon, Nielsen, and Webb (2021) therefore suggest using the usual two-way CV$_1$ estimator and applying the original WCR bootstrap to the dimension with the fewest clusters or the most unbalanced clusters. Simulation evidence suggests that this often provides more reliable inferences than the t distribution, but these inferences may still be problematic.

It may also be interesting to calculate measures of leverage, partial leverage, and influence for the intersection of the two clustering dimensions, especially when the number of nonempty intersections is not large. This means calling `summclust` a third time. Suppose there are two clustering dimensions, with G clusters in the first dimension and H clusters in the second. Then the number of intersection clusters is at most GH, but it can be smaller if some of the intersection clusters are empty. To use `summclust` for the intersections, we must create a new variable that uniquely identifies each nonempty intersection cluster. Running `summclust` for this case may be expensive when the number of nonempty intersections is large, especially if k is also large.

Note that, when `summclust` is invoked three times for each of two clustering dimensions and their intersection, the CV$_3$ standard error that it reports for each of the three cases is based on a different pattern of one-way clustering. When two-way clustering is

appropriate, none of these standard errors is valid. However, what `summclust` reports can be used to compute an asymptotically valid variance as

$$\widehat{\text{Var}}_{2W}(\widehat{\beta}_j) = \widehat{\text{Var}}_G(\widehat{\beta}_j) + \widehat{\text{Var}}_H(\widehat{\beta}_j) - \widehat{\text{Var}}_{GH}(\widehat{\beta}_j) \tag{34}$$

Here $\widehat{\beta}_j$ is the OLS estimate of a coefficient of interest, and the three estimated variances on the right-hand side of (34) are the squares of the CV$_3$ or CV$_{3J}$ standard errors reported by `summclust` for clustering dimension G, clustering dimension H, and the intersection of the two clustering dimensions, respectively.

Asymptotically, the two-way variance $\text{Var}_{2W}(\widehat{\beta}_j)$ should not be less than either of the one-way variances. Therefore, if $\widehat{\text{Var}}_{2W}(\widehat{\beta}_j)$ is less than either $\widehat{\text{Var}}_G(\widehat{\beta}_j)$ or $\widehat{\text{Var}}_H(\widehat{\beta}_j)$, it makes sense to replace it by the larger of those two variance estimates. Doing this also eliminates the risk of having to take the square root of a negative number. The appropriate t distribution has $\min(G-1, H-1)$ degrees of freedom if $\widehat{\text{Var}}_{2W}(\widehat{\beta}_j)$ is used and $G-1$ or $H-1$ degrees of freedom if it is replaced by either $\widehat{\text{Var}}_G(\widehat{\beta}_j)$ or $\widehat{\text{Var}}_H(\widehat{\beta}_j)$, respectively. We conjecture that, especially when this is done, the two-way standard error based on either jackknife estimator will yield more conservative, and generally more reliable, inferences than the usual two-way standard error based on CV$_1$.

As we discuss in section 3, it is often invalid to partial out fixed effects when computing a jackknife CRVE. This can be particularly tricky in the case of two-way clustering. For example, suppose there are G states and H years. Then it may be desirable to partial out the state fixed effects when computing $\widehat{\text{Var}}_G(\widehat{\beta}_j)$ but invalid to partial out the year fixed effects. Similarly, it may be desirable to partial out the year fixed effects when computing $\widehat{\text{Var}}_H(\widehat{\beta}_j)$ but invalid to partial out the state fixed effects. Finally, it is invalid to partial out either set of fixed effects when computing $\widehat{\text{Var}}_{GH}(\widehat{\beta}_j)$. The `absorb()` option of `summclust` normally detects cases where partialing out is invalid and refuses to display jackknife standard errors and several other quantities.

7   Simulation experiments

One of the reasons for calculating leverages and partial leverages is to identify cases in which inference may be problematic. The objective of the simulation experiments in this section is to see whether the rejection frequencies for cluster–robust t tests can be predicted from the features of the X matrix reported by `summclust`. There are 3,000 cases, each corresponding to a particular X matrix. For each case, we generate 10,000 values of y and use them to estimate rejection frequencies for t tests or bootstrap tests at the 0.05 level.

In the experiments, there are either 20 clusters and 2,000 observations or 30 clusters and 3,000 observations. The cluster sizes N_g are determined by a parameter $\gamma \geq 0$, as follows:

$$N_g = \left[N \frac{\exp(\gamma g/G)}{\sum_{j=1}^{G} \exp(\gamma j/G)} \right], \qquad g = 1, \dots, G-1$$

Here $[\cdot]$ denotes the integer part of its argument, and $N_G = N - \sum_{j=1}^{G-1} N_g$. As γ increases, the cluster sizes become increasingly unbalanced. The value of γ is chosen randomly from the $U[2, 4]$ distribution, so the cluster sizes tend to greatly vary. When $G = 20$, the smallest cluster has between 8 and 32 observations, and the largest has between 229 and 378. When $G = 30$, the smallest cluster has between 7 and 32 observations, and the largest has between 237 and 396.

There are five regressors, one of which is the test regressor, plus a constant term. The regressors equal either 0 or 1. With probability $1 - p_c$, all the observations in a cluster are 0. With probability p_c, they randomly equal either 0 or 1, both with probability 0.5. Thus, when $p_c = 1$, all variation is at the individual level, and leverage tends to be proportional to cluster sizes. As p_c declines, the samples become more unbalanced. In the experiments, the values of p_c are chosen to be 0.25, 0.30, 0.35, 0.40, 0.50, and 0.60, each for one-sixth of the cases. Smaller values of p_c tend to be associated with larger discrepancies between actual rejection frequencies and 0.05, the nominal level of the tests.

For each experiment, we obtain 12,000 estimated rejection frequencies. One-quarter of these are based on CV$_1$ and the $t(G - 1)$ distribution, one-quarter on CV$_3$ and the $t(G - 1)$ distribution, and one-quarter on each of the WRC-C and WCR-S bootstraps. To predict these rejection frequencies, we use a generalized additive model based on smoothing splines; see James et al. (2021, sec. 7.7). The base model can be written as

$$r_i = \beta_0 + f_1(V_{si}) + f_2\left(V_{si}^{1/2}\right) + \beta_1 G_{i0}^* + u_i \tag{35}$$

where r_i is the rejection frequency for case i. Here V_{si} denotes $V_s(L_{\bullet j})$, the scaled variance of the partial leverages L_{gj} for the test regressor for case i, G_{i0}^* denotes $G_j^*(0)$ for the test regressor for case i (recall from section 2.3 that it is a monotonically decreasing function of the L_{gj}), and $f_1(\cdot)$ and $f_2(\cdot)$ are smoothing splines with five degrees of freedom. Because everything on the right-hand side of (35) is a function of V_{si}, this model is simply using the V_{si} to predict the r_i in a potentially nonlinear way.

Figure 3 shows the fitted values from (35), which are predicted rejection frequencies, plotted against the scaled variance of the partial leverages L_{gj} for four methods of inference and two sample sizes. Panel (a) shows them for t tests based on both CV$_1$ (solid lines) and CV$_3$ (dashed lines) for $G = 20$ and $G = 30$, and panel (b) shows them for WRC-C and WCR-S bootstrap tests for the same two cases. The model seems to fit quite well, at least for the asymptotic tests, as can be seen from the values of R^2 reported for each of the curves. It also fits well for the bootstrap tests, and in fact it has smaller residuals for them than for the asymptotic tests. The lower R^2 values for the bootstrap tests simply reflect the fact that there is much less variation to explain.
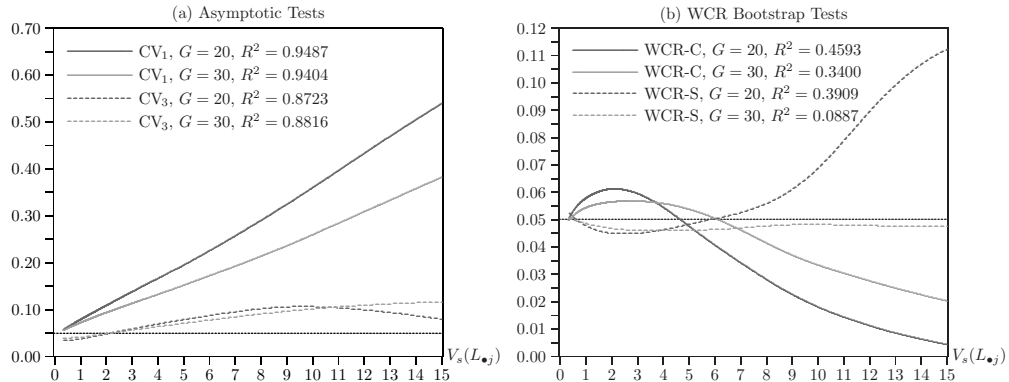
Figure 3. Predicted rejection frequencies for asymptotic and bootstrap tests at 0.05 level. NOTES: Each of the curves shows fitted values from the generalized additive model (35) that predicts observed rejection frequencies, based on 10,000 replications, using nonlinear functions of the $V_s(L_{\bullet j})$; see the text for details. Bootstrap rejection frequencies are based on $B = 399$. WRC-C is the classic restricted wild cluster bootstrap, and WCR-S is the score variant proposed in MacKinnon, Nielsen, and Webb (2023c).

We can see from figure 3 that t tests based on CV$_1$ often overreject to an extreme degree. For the very smallest values of $V_s(L_{\bullet j})$, the tests tend to overreject modestly, with predicted rejection frequencies of 0.058 for $G = 20$ and 0.055 for $G = 30$. However, these then rise quite rapidly and almost linearly. For $G = 30$, there are four cases (out of 3,000) for which $V_s(L_{\bullet j}) > 15$. These are not shown in the figure, but the approximately linear relationship continues to hold, and the fit for these extreme cases is reasonably good.

In contrast, the t tests based on CV$_3$ tend to underreject for small values of $V_s(L_{\bullet j})$. For the very smallest values, the predicted rejection frequencies are 0.033 for $G = 20$ and 0.039 for $G = 30$. Although it is not obvious from the figure, the CV$_3$ tests are predicted to underreject somewhat more than half the time, because, in our experiments, most values of $V_s(L_{\bullet j})$ are quite small. As $V_s(L_{\bullet j})$ increases, rejection frequencies increase, although for $G = 20$ they start to decline again once $V_s(L_{\bullet j})$ exceeds about 9.6. The predicted rejection frequencies never exceed 0.105 for $G = 20$ and 0.118 for $G = 30$. In a few cases (74 for $G = 20$ and 5 for $G = 30$), the matrix that is inverted in (9) was singular for at least one omit-one-cluster subsample. This happened whenever one of the regressors took the same value for all observations in $G - 1$ of the clusters. These cases were dropped.

Panel (b) of figure 3 shows the fitted values from (35) for WRC-C and WCR-S bootstrap t tests plotted against the scaled variance of the L_{gj}. Notice that the scale of the vertical axis differs greatly from the one in Panel (a). All tests, especially the WCR-S ones, perform quite well for smaller values of $V_s(L_{\bullet j})$. Except for WCR-S with $G = 30$, however, the rejection-frequency curves are not even close to being linear. This is also the only case for which the fitted values do not deviate greatly from 0.05 for large values

of $V_s(L_{\bullet j})$. In every other case, a large value of $V_s(L_{\bullet j})$ tends to be associated with substantial levels of overrejection or underrejection.

It is natural to ask whether we can improve the fit of (35) by adding additional explanatory variables that are not simply functions of the $V_s(L_{\bullet j})$. The answer is that we can. In particular, the variables $\bar{a}_{\mathrm{geo}}(L_{\bullet j})$ and $G_j^*(1)$ are often significant when they are added. However, the spline $f_1(V_{si})$ always remains highly significant, even when many other regressors are included. Thus, at least in these experiments, the scaled variance of the partial leverages, which is the square of their coefficient of variation, seems to be particularly revealing.

Based on these results, which are of course extremely dependent on the way in which the regressors are generated, it seems sensible for investigators to look at several different summary measures for both leverage and partial leverage. That is why `summclust` reports several of them. In this case, the most informative summary measure appears to be the scaled variance, defined in (13), of the partial leverage measures L_{gj}, defined in (7), for the regressor of interest. `summclust` reports the square root of this in the `coefvar` line of the `Cluster Variability` table. In general, cluster–robust inference seems to be most reliable when the partial leverages do not vary greatly across clusters.

8   Conclusions

We have discussed a new command, `summclust`, that is designed to summarize the cluster structure of the dataset for a linear regression model with clustered disturbances. Because the key unit of observation is the cluster, it makes sense to examine measures of influence, leverage, and partial leverage at the cluster level. These are easy to compute and are conceptually very similar to the corresponding classic measures at the observation level (Belsley, Kuh, and Welsch 1980; Chatterjee and Hadi 1986). The `summclust` command calculates all of them and also reports several summary statistics.

Our measure of influence at the cluster level can provide valuable information about how empirical results depend on the data in the various clusters. Investigators should be wary if dropping one or two clusters changes the results dramatically. However, apart from such cases, the most interesting quantities that `summclust` calculates generally seem to be the partial leverages and measures that summarize their distribution.

It has long been known that cluster–robust inference can be unreliable when the number of clusters is small. More recent work, including MacKinnon and Webb (2017b, 2018) and Djogbenou, MacKinnon, and Nielsen (2019), has shown that it can also be severely unreliable when cluster sizes vary a lot or when few clusters are treated in the context of difference-in-differences and other treatment models. In both of these cases, leverage and partial leverage tend to vary greatly across clusters. It therefore seems natural to use our measures of leverage and partial leverage as diagnostic tools to identify datasets and regression designs in which cluster–robust inference is likely to be challenging. Simulation results in section 7 suggest that the extent to which partial leverage varies across clusters can be particularly informative. We believe that

investigators should always look at the summary statistics reported by `summclust` and exercise caution whenever they indicate substantial variation across clusters.

As we discussed in section 2.2, the computations needed for leverage and influence are very similar to the ones needed to compute cluster jackknife variance matrix estimators. The `summclust` command therefore computes two very similar jackknife estimators, which we refer to as CV_3 and CV_{3J}, almost as a by-product of other computations. These are the same estimators that Stata can produce using the `vce(jackknife, mse)` and `vce(jackknife)` options. However, because `summclust` is designed explicitly for linear regression models fit by OLS, it is faster than using these `vce` options. Moreover, when `summclust` is already being used to obtain cluster-level measures of influence and leverage for diagnostic purposes, the additional cost of computing the jackknife variance estimators is minimal.

When the number of clusters is reasonably large and the variation of leverage and partial leverage across clusters is small, we would expect conventional inference based on CV_1 standard errors to perform well. If so, the CV_3 standard errors reported by `summclust` should be very similar to the CV_1 standard errors reported by one of Stata's regression commands. When this is the case, there is probably no need for investigators to worry further about the reliability of their inferences. In many cases, however, the CV_3 and CV_1 standard errors will differ noticeably. This happens for the empirical example in section 4, where there are 72 clusters but partial leverage varies a lot. In such cases, the CV_3 standard errors are almost certain to be more conservative and are very likely to be more reliable than the CV_1 ones.

p-values and confidence intervals that are even more reliable can often be obtained by using the restricted wild cluster bootstrap, which is implemented natively with `wildbootstrap` in Stata 18 and in the package `boottest` (Roodman et al. 2019). Recent versions of that package implement the WCR-S bootstrap (MacKinnon, Nielsen, and Webb 2023c) in addition to the classic WRC-C bootstrap. We strongly recommend that both variants be calculated whenever the CV_3 and CV_1 standard errors disagree. When the two bootstrap p-values agree, as they do for the empirical example in section 4, then it is probably safe to rely on either of them. When they disagree, then neither of them may be entirely reliable, but we would be inclined to use the one given by the WCR-S bootstrap.

Up to this point, everything in this section has been based on the assumption that there is one-way clustering with a known clustering structure. When more than one level of clustering is plausible, investigators need to choose among them, and this can be challenging; see the discussions in MacKinnon, Nielsen, and Webb (2023a,b). The measures of leverage and influence produced by `summclust` may be helpful in deciding at what level to cluster.

The current version of `summclust` is not explicitly designed to handle two-way clustering. However, as we discussed in section 6, it can be called for each clustering dimension to produce two sets of diagnostic statistics. If it is called three times, once for each dimension and once for their intersection, then it can also be used to compute two-way

cluster jackknife variance matrix estimators. At present, however, little is known about the properties of these estimators.

9   Acknowledgments

10   Programs and supplemental material

To install the software files as they exist at the time of publication of this article, type

```
. net sj 23-4
. net install st0733      (to install program files, if available)
. net get st0733          (to install ancillary files, if available)
```

The command `summclust` can be installed from the Statistical Software Components archive by typing

```
. ssc install summclust
```

or from GitHub by typing

```
. net install summclust, ///
from("https://raw.githubusercontent.com/mattdwebb/summclust/main/")
```

11   References

Bell, R. M., and D. F. McCaffrey. 2002. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology* 28: 169–181.

Belsley, D. A., E. Kuh, and R. E. Welsch. 1980. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. New York: Wiley. https://doi.org/10.1002/0471725153.

Bester, C. A., T. G. Conley, and C. B. Hansen. 2011. Inference with dependent data using cluster covariance estimators. *Journal of Econometrics* 165: 137–151. https://doi.org/10.1016/j.jeconom.2011.01.007.

Broderick, T., R. Giordano, and R. Meager. 2023. An automatic finite-sample robustness metric: When can dropping a little data change conclusions? arXiv:2011.14999 [stat.ME]. https://doi.org/10.48550/arXiv.2011.14999.

Busso, M., and S. Galiani. 2019. The causal effect of competition on prices and quality: Evidence from a field experiment. *American Economic Journal: Applied Economics* 11: 33–56. https://doi.org/10.1257/app.20150310.

Cameron, A. C., J. B. Gelbach, and D. L. Miller. 2008. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics* 90: 414–427. https://doi.org/10.1162/rest.90.3.414.

———. 2011. Robust inference with multiway clustering. *Journal of Business and Economic Statistics* 29: 238–249. https://doi.org/10.1198/jbes.2010.07136.

Cameron, A. C., and D. L. Miller. 2015. A practitioner's guide to cluster–robust inference. *Journal of Human Resources* 50: 317–372. https://doi.org/10.3368/jhr.50.2.317.

Carter, A. V., K. T. Schnepel, and D. G. Steigerwald. 2017. Asymptotic behavior of a *t*-test robust to cluster heterogeneity. *Review of Economics and Statistics* 99: 698–709. https://doi.org/10.1162/REST_a_00639.

Chatterjee, S., and A. S. Hadi. 1986. Influential observations, high-leverage points, and outliers in linear regression. *Statistical Science* 1: 379–393. https://doi.org/10.1214/ss/1177013622.

Chesher, A. 1989. Hájek inequalities, measures of leverage and the size of heteroskedasticity robust Wald tests. *Econometrica* 57: 971–977. https://doi.org/10.2307/1913779.

Conley, T. G., S. Gonçalves, and C. B. Hansen. 2018. Inference with dependent data in accounting and finance applications. *Journal of Accounting Research* 56: 1139–1203. https://doi.org/10.1111/1475-679X.12219.

Cook, R. D., and S. Weisberg. 1980. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics* 22: 495–508. https://doi.org/10.2307/1268187.

Correia, S. 2014. reghdfe: Stata module to perform linear or instrumental-variable regression absorbing any number of high-dimensional fixed effects. Statistical Software Components S457874, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457874.html.

Davidson, R., and J. G. MacKinnon. 1993. *Estimation and Inference in Econometrics.* New York: Oxford University Press.

Djogbenou, A. A., J. G. MacKinnon, and M. Ø. Nielsen. 2019. Asymptotic theory and wild bootstrap inference with clustered errors. *Journal of Econometrics* 212: 393–412. https://doi.org/10.1016/j.jeconom.2019.04.035.

Efron, B. 1979. Bootstrapping methods: Another look at the jackknife. *Annals of Statistics* 7: 1–26. https://doi.org/10.1214/aos/1176344552.

Hansen, B. E. 2022. Jackknife standard errors for clustered regression. Working paper, University of Wisconsin.

Hansen, B. E., and S. Lee. 2019. Asymptotic theory for clustered samples. *Journal of Econometrics* 210: 268–290. https://doi.org/10.1016/j.jeconom.2019.02.001.

Imbens, G. W., and M. Kolesár. 2016. Robust standard errors in small samples: Some practical advice. *Review of Economics and Statistics* 98: 701–712. https://doi.org/10.1162/REST_a_00552.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2021. *An Introduction to Statistical Learning: With Applications in R*. 2nd ed. New York: Springer. https://doi.org/10.1007/978-1-0716-1418-1.

Lee, C. H., and D. G. Steigerwald. 2018. Inference for clustered data. *Stata Journal* 18: 447–460. https://doi.org/10.1177/1536867X1801800210.

MacKinnon, J. G., M. Ø. Nielsen, and M. D. Webb. 2021. Wild bootstrap and asymptotic inference with multiway clustering. *Journal of Business and Economic Statistics* 39: 505–519. https://doi.org/10.1080/07350015.2019.1677473.

———. 2023a. Cluster-robust inference: A guide to empirical practice. *Journal of Econometrics* 232: 272–299. https://doi.org/10.1016/j.jeconom.2022.04.001.

———. 2023b. Testing for the appropriate level of clustering in linear regression models. *Journal of Econometrics* 235: 2027–2056. https://doi.org/10.1016/j.jeconom.2023.03.005.

———. 2023c. Fast and reliable jackknife and bootstrap methods for cluster-robust inference. *Journal of Applied Econometrics* 38: 671–694. https://doi.org/10.1002/jae.2969.

MacKinnon, J. G., and M. D. Webb. 2017a. Pitfalls when estimating treatment effects using clustered data. *The Political Methodologist*. https://polmeth.org/blog/pitfalls-when-estimating-treatment-effects-using-clustered-data.

———. 2017b. Wild bootstrap inference for wildly different cluster sizes. *Journal of Applied Econometrics* 32: 233–254. https://doi.org/10.1002/jae.2508.

———. 2018. The wild bootstrap for few (treated) clusters. *Econometrics Journal* 21: 114–135. https://doi.org/10.1111/ectj.12107.

———. 2020. Randomization inference for difference-in-differences with few treated clusters. *Journal of Econometrics* 218: 435–450. https://doi.org/10.1016/j.jeconom.2020.04.024.

MacKinnon, J. G., and H. White. 1985. Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics* 29: 305–325. https://doi.org/10.1016/0304-4076(85)90158-7.

Niccodemi, G., R. Alessie, V. Angelini, J. Mierau, and T. Wansbeek. 2020. Refining clustered standard errors with few clusters. Working Paper 2020002-EEF, University of Groningen.

Pustejovsky, J. E., and E. Tipton. 2018. Small sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models. *Journal of Business and Economic Statistics* 36: 672–683. https://doi.org/10.1080/07350015.2016.1247004.

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. https://doi.org/10.1177/1536867X19830877.

Young, A. 2022. Consistency without inference: Instrumental variables in practical application. *European Economic Review* 147: 104112. https://doi.org/10.1016/j.euroecorev.2022.104112.

About the authors

James G. MacKinnon is the Sir Edward Peacock Professor of Econometrics at Queen's University in Kingston, Ontario, Canada.

Morten Ørregaard Nielsen holds a DNRF Chair in Econometrics and is a professor of economics at Aarhus University in Aarhus, Denmark.

Matthew D. Webb is an associate professor at Carleton University and the Ottawa-Carleton Graduate School of Economics in Ottawa, Ontario, Canada.