Journal of Statistical Software

November 2017, Volume 82, Issue 3. *doi: 10.18637/jss.v082.i03*

Robust Standard Error Estimators for Panel Models: A Unifying Approach

Giovanni Millo
Group Insurance Research, Generali SpA
DEAMS, University of Trieste

Abstract

The different robust estimators for the standard errors of panel models used in applied econometric practice can all be written and computed as combinations of the same simple building blocks. A framework based on high-level wrapper functions for most common usage and basic computational elements to be combined at will, coupling user-friendliness with flexibility, is integrated in the **plm** package for panel data econometrics in R. Statistical motivation and computational approach are reviewed, and applied examples are provided.

Keywords: panel data, covariance matrix estimators, R.

1. Introduction

This paper is about computing estimators for the covariance matrix of parameters in a linear panel model, of the kind commonly used in applied practice to produce "robust" standard errors. Different estimators are usually preferred in one or the other branch of applied econometrics, from large microeconometric panels (Arellano 1987) to moderately-sized panel time series in macroeconomics (Driscoll and Kraay 1998) and large panels in finance (Petersen 2009; Cameron, Gelbach, and Miller 2011; Thompson 2011), up to pooled time series in political science (Beck and Katz 1995). Software implementations are in most cases to be found in one or the other commercial package, often as user-programmed additional routines; or sometimes GAUSS (Aptech Systems, Inc. 2006) or MATLAB (The MathWorks Inc. 2017) code is available. My work aims at bringing them all together under the common umbrella of the R environment (R Core Team 2017), once again the all-purpose statistical software.

From a software design viewpoint, I translate some results from the recent literature (Petersen 2009; Thompson 2011; Cameron *et al.* 2011) into a comprehensive computational framework, in turn integrated into the **plm** package for panel data econometrics (Croissant and Millo

2008). I describe a general expression for "clustering" estimators; then I review two-level clustering as a combination of simple clustering estimators and the extension to persistent effects by summation of lagged terms; lastly, I show how applying a weighting scheme to lagged covariance terms yields Driscoll and Kraay (1998)'s spatial correlation consistent (SCC) estimator (and, as a special case, that of Newey and West 1987).

From an application perspective, I extend the treatment of Petersen (2009) to double-clustering estimators plus time-persistent shocks as in Thompson (2011): a structure which, based on simulations in Petersen (2009), can be conjectured to successfully account for both individual effects and persistent idiosyncratic shocks. My approach also allows easy extension to a combination of effects which has not, to my knowledge, received attention in the literature yet: double-clustering as in Cameron *et al.* (2011) plus time-decaying correlation as in Driscoll and Kraay (1998). A practical example is given in Section 6.

One not-so-minor aim of this paper is to overcome sectoral barriers between different, if contiguous, disciplines: it is striking, for example, how few citations Driscoll and Kraay (1998) on the panel generalization of the Newey and West (1987) estimator gets in the finance literature, especially in those papers that advocate what is essentially an unweighted version of their SCC covariance. Also, Arellano (1987) and Froot (1989), in the different contexts of fixed effects panels with serial correlation and of industry-clustered financial data, independently developed what is computationally the same estimator (referred in the following as one-way clustering) first described by Liang and Zeger (1986). Cross-referencing between the different branches of statistical and econometric research is still uncommon in this subject, to the point that raising awareness might be useful.[1] From the point of view of political science, where panel – or time-series cross-section (TSCS) – data are an important methodological field, the functionality outlined here allows researchers to progress beyond the now-ubiquitous application of panel-corrected standard errors (PCSE, Beck and Katz 1995) to pooled specifications, along the lines of Wilson and Butler (2007): both comparing it with alternative strategies and possibly combining it with individual effects, in order to tackle the all-important, and often overlooked, issue of individual heterogeneity (Wilson and Butler 2007, Section 2.2).

In this sense, my work is meant to provide R users with a comprehensive set of modular tools: lower level components, conceptually corresponding to the statistical "objects" involved (see Zeileis 2006), and a higher-level set of "wrapper functions" corresponding to standard covariance estimators as they would be used in statistical packages: White heteroskedasticity-consistent, clustering, SCC and so on. Wrappers work by combining the same, few lower-level components in multiple ways in the spirit of the *Lego principle* of Hothorn, Hornik, Van De Wiel, and Zeileis (2006), with obvious benefits for both flexibility and code maintenance. This toolset should enable users to follow the work of Petersen (2009); Cameron *et al.* (2011); Thompson (2011) in detail, experimenting with settings and comparing estimates' magnitudes (see Petersen 2009) for specification and diagnostic purposes, at least as far as linear models in two panel dimensions are concerned.

Clustered standard errors for non-panel models are another field of application. For some time, there has been R code available for one- or two-way clustering in a linear model (see Arai 2009). This last has recently evolved into a package for multi-way clustering, **multiwayvcov**

[1] Also note that Fama and MacBeth (1973)'s covariance estimator popular in finance, actually first and foremost an estimator for the averages of the coefficients, is known in the econometrics literature as the Mean Group estimator of Pesaran and Smith (1995). See Ibragimov and Müller (2010) for a formal justification of the Fama-MacBeth method.

(Graham, Arai, and Hagströmer 2016); in turn, many of the features of the latter have been incorporated into the **sandwich** package by Berger, Graham, and Zeileis (2017). The **sandwich** package was the original, object-oriented implementation of sandwich estimators in R (Zeileis 2006) and provides the generic function `vcovHC`, panel methods for which are presented here. Nevertheless, up to two clustering dimensions all this functionality is effectively encompassed by that presented here, provided the data are treated like a *faux* panel specifying one or two indices. Moreover, integration within the **plm** package means that the estimators presented here can seamlessly interact with panel features like individual or time effects. By contrast, extending clustering to more than two dimensions in a panel context does not fit into the panel data infrastructure of package **plm** and is out of the scope of this paper.

When estimating regression models, R creates a model object which, besides estimation results, carries on a wealth of useful information, including the original data. Robust testing in R is done retrieving the necessary elements from the model object, using them to calculate a robust covariance matrix for coefficient estimates and then feeding the latter to the actual test function, which can be a t-test for significance, a Wald restriction test and so on. Therefore the approach to diagnostic testing is more flexible than with procedural languages, where diagnostics usually come with standard output. In our case, for example, one can obtain different estimates of the standard errors under various kinds of dependence without re-estimating the model, and present them compactly.

When appropriate, I will highlight some features of R that make it easy and effective to combine statistical objects; in particular, functions as arguments; modularity and components reusing; function application over arrays of arbitrary dimension; and in general object orientation, which ensures application of the right method with the appropriate defaults for the object at hand through standard syntax.

The paper is organized into three main bodies. The next two sections (Sections 2 and 3) review the statistical foundations of the methods and set the notation in terms of a few low-level components according to the Lego principle. Section 4 on the computational framework, arguably the heart of the paper, describes the statistical building blocks in terms of computational objects characterized by a few standard "switches", and their combinations in terms of user-friendly "wrapper" functions; then, in an object-oriented fashion, it discusses how and when it is (statistically) appropriate to apply the resulting user-level methods to '`plm`' objects estimated in different ways: by either (pooled) ordinary least squares (OLS), fixed effects (FE), random effects (RE), or by first-differencing methods (FD). The remainder of the paper (Sections 5 and 6) sets the new estimators in the context of the **plm** package and provides some examples of application.

The functionality described here is available in package **plm** since version 1.5-1 and the package is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=plm`.

2. Robust covariance estimators

In this section I briefly review the ideas behind robust covariance estimators of the *sandwich* type, in order to provide a basis for the subsequent treatment of their panel extension. The reader is referred to any econometrics textbook, e.g., Greene (2003) – on which this paragraph is based – for a formal treatment.

Consider a linear model $y = X\beta + \epsilon$ and the OLS estimator $\hat{\beta}_{OLS} = (X^\top X)^{-1} X^\top y$. If one is interested in making inference on β, then an estimate of $\mathsf{VAR}(\hat{\beta})$ is needed. If the error terms ϵ are independent and identically distributed, then the covariance matrix takes the familiar textbook form: $\mathsf{VAR}(\hat{\beta}) = \hat{\sigma}^2 (X^\top X)^{-1}$, where $\hat{\sigma}^2$ is an estimate of the error variance. This case is synthetically dubbed *spherical errors*, and the relative formulation of $V(\hat{\beta}_{OLS})$ is often referred to, somewhat inappropriately, as "OLS covariance"[2].

Let us consider robust estimation in the context of the simple linear model outlined above. The problem at hand is to estimate the covariance matrix of the OLS estimator relaxing the assumptions of serial correlation and/or homoskedasticity without imposing any particular structure to the errors' variance or interdependence.

As the estimator of the OLS parameters' covariance matrix is

$$\hat{V} = \frac{1}{n} \left(\frac{X^\top X}{n} \right)^{-1} \left(\frac{1}{n} X^\top [\sigma^2 \Omega] X \right) \left(\frac{X^\top X}{n} \right)^{-1}$$

in order to consistently estimate V it is not necessary to estimate all the $n(n+1)/2$ unknown elements in the Ω matrix but only the $K(K+1)/2$ ones in

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} \mathbf{x}_i \mathbf{x}_j^\top,$$

which may be called the *meat* of the sandwich, the two $\left(\frac{X^\top X}{n} \right)^{-1}$ being the *bread*. (From now on, we will concentrate exclusively on the meat, and we will dispose of the $1/n$ terms throughout.) All that is required are *pointwise consistent* estimates of the errors, which is satisfied by consistency of the estimator for β (see Greene 2003). In the heteroskedasticity case, correlation between different observations is ruled out and the *meat* reduces to

$$S_0 = \frac{1}{n} \sum_{i=1}^{n} \sigma_i^2 \mathbf{x}_i \mathbf{x}_i^\top,$$

where the n unknown σ_i^2s can be substituted by e_i^2 (see White 1980). In the serial correlation case, the natural estimation counterpart would be

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} e_i e_j \mathbf{x}_i \mathbf{x}_j^\top,$$

but this structure proves too general to achieve convergence. Newey and West (1987) devise a (heteroskedasticity and) autocorrelation consistent estimator that works based on the assumption of correlation dying out as the distance between observations increases. The Newey-West HAC estimator for the *meat* takes that of White and adds a sum of covariances between the different residuals, smoothed out by a *kernel function* giving weights decreasing with distance:

[2]The reason is that OLS is "best linear unbiased" (BLUE) under sphericity; yet this is confusing because other covariance estimators can be more appropriate for $\hat{\beta}_{OLS}$ under different conditions. Notice that Thompson (2011) uses the same name referring to the case of heteroskedasticity but no dependence (here: *White*).

$$S_0 + \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} w_l e_t e_{t-l} \left(\mathbf{x}_t \mathbf{x}_{t-l}^\top + \mathbf{x}_{t-l} \mathbf{x}_t^\top \right)$$

with w_l the weight from the kernel smoother, e.g., the Bartlett kernel function: $w_l = 1 - \frac{l}{L+1}$ (for a discussion of alternative kernels see Zeileis 2006). The lag l is usually truncated well below sample size: one popular rule of thumb is $L = n^{1/4}$ (see Greene 2003; Driscoll and Kraay 1998).

In the following I will consider the extensions of this framework for a panel data setting where, thanks to added dimensionality, various combinations of the two above structures will turn out to be able to accommodate very general types of dependence.

3. Clustering estimators in a panel setting

Let us now consider a *panel* (or *longitudinal*) setting where data are collected on two different dimensions: to fix ideas, let us think of n entities (individuals, firms, countries, ...) which we here label *groups* and index by $i = 1, \ldots, n$ sampled at T points in *time*. The two dimensions are not fully symmetric as for the sake of practical relevance I have considered one dimension (time) having a natural ordering and the other having none. This setting is sufficiently general to describe the vast majority of applications; a symmetric extension would nevertheless be straightforward. Different choices of dimensions are possible and have been explored in the literature: e.g., Froot (1989), in the context of financial data, discusses sampling from "independent" industries in order to increase sample size, clustering within industry to account for dependence. Thus the two dimensions would be *industry* and a generic counter: clustering would be done according to industry, while meaningless across the "other" dimension. The model is then

$$y_{it} = X_{it}\beta + \epsilon_{it}.$$

For now I consider again the familiar OLS estimator $\hat{\beta}_{OLS}$, which in this setting is referred to as *pooled OLS* because it *pools* all observations together irrespective of their belonging to a given group (but see Section 4.4 for an extension to three other common panel estimators).

Clustering estimators work by extending the "sandwich" principle to panel data. Besides heteroskedasticity, the added dimensionality allows to obtain robustness against totally unrestricted timewise or cross-sectional correlation, provided this is along the "smaller" dimension. In the case of "large-N" (*wide*) panels, the big cross-sectional dimension allows robustness against serial correlation[3]; in "large-T" (*long*) panels, on the converse, robustness to cross-sectional correlation can be attained drawing on the large number of time periods observed. As a general rule, the estimator is asymptotic in the number of clusters: see Cameron *et al.* (2011, Section 2).

Imposing cross-sectional (serial) independence in fact restricts all covariances between observations belonging to different individuals (time periods) to zero, yielding an error covariance

[3]This is the case of the seminal contribution by Arellano (1987).

matrix with a block-diagonal structure: in the former case, $V(\epsilon) = I_n \otimes \Omega_i$, where

$$\Omega_i = \begin{bmatrix} \sigma^2_{i1} & \sigma_{i1,i2} & \dots & \dots & \sigma_{i1,iT} \\ \sigma_{i2,i1} & \sigma^2_{i2} & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \sigma^2_{iT-1} & \sigma_{iT-1,iT} \\ \sigma_{iT,i1} & \dots & \dots & \sigma_{iT,iT-1} & \sigma^2_{iT} \end{bmatrix} \quad (1)$$

and the consistency relies on the cross-sectional dimension being "large enough" with respect to the number of free covariance parameters in the diagonal blocks. The other case is symmetric.

3.1. White-Arellano, or one-way clustering

White's heteroskedasticity-consistent covariance matrix[4] has been extended to clustered data by Liang and Zeger (1986) and to econometric panel data by Arellano (1987), who applied it in a fixed effects setting. Observations can be clustered by the cross-sectional (group) index which is arguably the most popular use of this estimator, and is appropriate in *large, short* panels because it is based on n-asymptotics; or by the time index, which is based on T-asymptotics and therefore appropriate for *long* panels. In the first case, the covariance estimator is robust against cross-sectional heteroskedasticity and also against serial correlation of arbitrary form. In the second case, symmetrically, against timewise heteroskedasticity and cross-sectional correlation. Arellano's original estimator, an instance of the first case, has the form:

$$V_{\text{White-Arellano}} = (X^\top X)^{-1} \sum_{i=1}^{n} X_i^\top u_i u_i^\top X_i (X^\top X)^{-1}. \quad (2)$$

It is of course still feasible to rule out serial correlation and compute an estimator that is robust to heteroskedasticity only, based on the following error structure:

$$\Omega_i = \begin{bmatrix} \sigma^2_{i1} & \dots & \dots & 0 \\ 0 & \sigma^2_{i2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & \dots & \sigma^2_{iT} \end{bmatrix} \quad (3)$$

in which case the original White estimator applies:

$$V_{\text{White}} = (X^\top X)^{-1} \sum_{i=1}^{n} \sum_{t=1}^{T} u^2_{it} \mathbf{x}_{it} \mathbf{x}^\top_{it} (X^\top X)^{-1}. \quad (4)$$

Some notation

Before discussing bidirectional extensions of this estimator, for the sake of clarity I will now define the "meat" of the two versions of the Arellano estimator, henceforth $V_{C\cdot}$, with respect to

[4] See White (1980, 1984).

the clustering dimension: the original, group-clustered version, robust vs. heteroskedasticity and *serial* dependence, will be

$$V_{CX} = \sum_{i=1}^{n} X_i^\top u_i u_i^\top X_i, \tag{5}$$

while the time-clustered version, robust vs. heteroskedasticity and *cross-sectional* dependence, will be:

$$V_{CT} = \sum_{t=1}^{T} X_t^\top u_t u_t^\top X_t. \tag{6}$$

The standard White estimator on pooled data, which is symmetric w.r.t. clustering,

$$V_W = \sum_{t=1}^{T} \sum_{i=1}^{n} u_{it}^2 \mathbf{x}_{it} \mathbf{x}_{it}^\top \tag{7}$$

will be conveniently written as

$$V_W = \sum_{i=1}^{n} X_i^\top \operatorname{diag}(u_i^2) X_i = \sum_{t=1}^{T} X_t^\top \operatorname{diag}(u_t^2) X_t, \tag{8}$$

where $\operatorname{diag}(u^2)$ is a matrix with squares of elements of u on the diagonal and zeros elsewhere, so that all of these expressions share the common structure

$$V_{C\cdot} = \sum_{\cdot} X_\cdot^\top \mathsf{E}(u) X_\cdot \tag{9}$$

with $\mathsf{E}(u)$ an appropriate function of the residuals.

This symmetric representation will provide a good starting point for the extension to double clustering.

3.2. Double clustering

Some recent research in finance (Petersen 2009; Cameron *et al.* 2011; Thompson 2011) advocates double clustering, motivating it by the need to account for *persistent shocks* and at the same time for cross-sectional or spatial correlation.

This estimator combining both individual and time clustering relies on a combination of the asymptotics of each: the minimum number of clusters along the two dimensions must go to infinity: see, again, Cameron *et al.* (2011, Section 2). Apart from this, any dependence structure is allowed within each group *or* within each time period, while cross-serial correlations between observations belonging to different groups *and* time periods are ruled out.

The double-clustered estimator is easily calculated by summing up the group-clustering and the time-clustering ones, then subtracting the standard White estimator (referred to in Cameron *et al.* 2011 as *time-group-clustering*, in Thompson 2011 as *white0*) in order to avoid double-counting the error variances along the diagonal:

$$V_{CXT} = V_{CX} + V_{CT} - V_W. \tag{10}$$

In order to control for the effect of common shocks, Thompson (2011) proposes to add to the sum of covariances one more term, related to the covariances between observations from any group at different points in time. Given a maximum lag L, this will be the sum over $l = 1, \ldots, L$ of the following generic term:

$$V_{CT,l} = \sum_{t=1}^{T} X_t^\top u_t u_{t-l}^\top X_{t-l} \tag{11}$$

representing the covariance between pairs of observations from any group distanced l periods in time, summed with its transpose. As the correlation between observations belonging to the *same* group at different points in time has already been captured by the group-clustering term, to avoid double counting one must subtract the within-groups part:

$$V_{W,l} = \sum_{t=1}^{T} \sum_{i=1}^{n} [x_{it} u_{it} u_{i,t-l}^\top x_{i,t-l}^\top] \tag{12}$$

again summed with its transpose, for each l. The resulting estimator

$$V_{CXT,L} = V_{CX} + V_{CT} - V_W + \sum_{l=1}^{L} [V_{CT,l} + V_{CT,l}^\top] - \sum_{l=1}^{L} [V_{W,l} + V_{W,l}^\top] \tag{13}$$

is robust to cross-sectional and timewise correlation inside, respectively, time periods and groups *and* to the cross-serial correlation between observations belonging to different groups, up to the Lth lag. It also resembles another well-known estimator from the econometric literature: the Newey-West nonparametric estimator, the difference being that instead of adding a (possibly truncated) sum of unweighted lag terms, the latter downweighs the correlation between "distant" terms through a kernel smoother function. Kernel-smoothed estimators will be the subject of the next section.

3.3. Kernel-based smoothing

As cited above, in a time series context Newey and West (1987) proposed an estimator that is robust to serial correlation as well as to heteroskedasticity. This estimator, based on the hypothesis of the serial correlation dying out "quickly enough", takes into account the covariance between units by: weighting it through a kernel smoother function giving less weight as they get more distant; and adding it to the standard White estimator.

Driscoll and Kraay's "SCC"

Driscoll and Kraay (1998) adapted the Newey-West estimator to a panel time series context, where not only serial correlation between residuals from the same individual in different time periods is taken into account, but also cross-serial correlation between different individuals in different times and, within the same period, cross-sectional correlation (see also Arellano 2003, p. 19).

The Driscoll and Kraay estimator, labeled SCC (as in "spatial correlation consistent"), is defined as the time-clustering version of Arellano plus a sum of lagged covariance terms, weighted by a distance-decreasing kernel function w_l:

$$\begin{aligned} V_{SCC,L} &= V_{CT} + \textstyle\sum_{l=1}^{L} w_l [\sum_{t=1}^{T} X_t^\top u_t u_{t-l}^\top X_{t-l} + \sum_{t=1}^{T} [X_t^\top u_t u_{t-l}^\top X_{t-l}]^\top] \\ &= V_{CT} + \textstyle\sum_{l=1}^{L} w_l [V_{CT,l} + V_{CT,l}^\top]. \end{aligned} \tag{14}$$

The "scc" covariance estimator requires the data to be a mixing sequence, i.e., roughly speaking, to have serial and cross-serial dependence dying out quickly enough with the T dimension, which is therefore supposed to be fairly large: Driscoll and Kraay (1998), based on Monte Carlo simulation, put the practical minimum at $T > 20-25$; the n dimension is irrelevant in this respect and is allowed to grow at any rate relative to T.

Panel Newey-West

By restricting the cross-sectional and cross-serial correlation to zero one gets a "pure" panel version of the original Newey-West estimator, as discussed, e.g., in Petersen (2009):

$$\begin{aligned} V_{NW,L} &= V_W + \textstyle\sum_{l=1}^{L} w_l [\sum_{t=1}^{T} \sum_{i=1}^{n} [\mathbf{x}_{it} u_{it} u_{i,t-l}^\top \mathbf{x}_{i,t-l}^\top] + \sum_{t=1}^{T} [\sum_{i=1}^{n} [\mathbf{x}_{it} u_{it} u_{i,t-l}^\top \mathbf{x}_{i,t-l}^\top]^\top] \\ &= V_W + \textstyle\sum_{l=1}^{L} w_l [V_{W,l} + V_{W,l}^\top]. \end{aligned} \tag{15}$$

As is apparent from Equation 14, if the maximum lag order is set to 0 (no serial or cross-serial dependence is allowed) the SCC estimator reverts to the cross-section version (time-clustering) of the Arellano estimator V_{CT}. On the other hand, if the cross-serial terms are all unweighted (i.e., if $w_l = 1\, \forall l$) then $V_{SCC,L|w=1} = V_{CT,L}$.

3.4. Unconditional estimators

Unconditional covariance estimators are based on the assumption of no error correlation in time (cross-section) and of an unrestricted but invariant correlation structure inside every cross-section (time period). They are popular in contexts characterized by relatively small samples, with prevalence of the time dimension.

Beck and Katz PCSE

Beck and Katz (1995), in the context of political science models with moderate time and cross-sectional dimensions, introduced the so-called panel corrected standard errors (PCSE), an estimator with good small-sample properties which, in the original time-clustering setting, is robust against cross-sectional heteroskedasticity and correlation.

In this framework and with reference to Equation 9, the "pcse" covariance is defined in terms of the $E_i = E \;\; \forall i$ function of the residuals as:

$$E = \frac{\sum_n \hat{e}_n \hat{e}_n^\top}{N}.$$

A sufficient, although not necessary condition for consistency of the "pcse" estimator (Beck and Katz 1996) is that the covariance matrix of the errors in every group be the same: $\Omega = \Sigma \otimes I_T$, with

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{1,2} & \dots & \dots & \sigma_{1,T} \\ \sigma_{2,1} & \sigma_2^2 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \sigma_{T-1}^2 & \sigma_{T-1,T} \\ \sigma_{T,1} & \dots & \dots & \sigma_{T,T-1} & \sigma_T^2. \end{bmatrix} \tag{16}$$

A possible further restriction is to assume correlation away imposing that Σ be diagonal, thus restricting the estimator to unconditional intragroup heteroskedasticity.

4. Computational framework

Generalizing the computational problem at hand and dividing it into modules is necessary for writing software that be both full-featured and easy to read and to maintain. In this section I show a generic formulation capable of generating all the estimators considered up to now; in the following I will consider a small-sample correction module. These building blocks will allow to construct a very general covariance estimating function whose usage in various testing environments will then be discussed in the light of object-oriented econometric computing.

Two defining features of R as a programming language are object-orientation and functional nature. In this sense, according to the object-oriented nature of R, in the next paragraph I will formulate a general computing module, the *software counterpart* of the *statistical objects* V_W, V_{CX}, V_{CT}, $V_{W,l}$, $V_{CX,l}$ and $V_{CT,l}$ which are in turn the building blocks for any of the estimators considered here. In turn, according to the functional nature of R, the computing module will be formulated as a function of: a (panel-indexed) vector of errors; an integer lag order; and lastly of a function to be applied to the error vector, taking the lag order as an argument. The ability of R to treat functions as a data type will make the translation of this formalization into software seamless.

4.1. A general, computing-oriented formulation

The most general formulation of the comprehensive estimator can be written as a kernel-weighted version of Formula 3 in Thompson (2011):

$$V_{CXT,L|w} = V_{CT} + \sum_{l=1}^{L} w_l[V_{CT,l} + V_{CT,l}^\top] + V_{CX} - V_W - \sum_{l=1}^{L} w_l[V_{W,l} + V_{W,l}^\top]. \quad (17)$$

In turn, all building blocks for Equation 17 can be generated by combining a clustering dimension (n or t), a lag order l and a function of the errors f. Starting from the general formulation:

$$V_g(t, l, f) = \sum_{t=1}^{T} X_t^\top f(u_t, u_{t-l}) X_{t-l} \quad (18)$$

inserting the outer product function and setting the lag to zero (so that $f(u_t, u_{t-l}) = u_t u_t^\top$) we get the time- (group-)clustering estimator

$$V_{CT} = V_g(t, 0, f = u_t u_t^\top) \quad (19)$$

and for the "White" terms on the diagonal, with the dot denoting indifferently n or t as clustering dimension,

$$V_W = V_g(\cdot, 0, f = \mathrm{diag}(u_\cdot^2)), \quad (20)$$

while for the cross-serial terms

$$V_{CT,l} = V_g(t, l, f = u_t u_{t-l}^\top) \quad (21)$$

Label	Notation
White heteroskedastic	V_W
Group clustering	V_{CX}
Time clustering	V_{CT}
Double clustering	$V_{CXT} = V_{CX} + V_{CT} - V_W$
Time clustering + shocks	$V_{CT,L} = V_{CT} + \sum_{l=1}^{L}[V_{CT,l} + V_{CT,l}^\top]$
Panel Newey-West	$V_{NW,L} = V_W + \sum_{l=1}^{L} w_l[V_{W,l} + V_{W,l}^\top]$
Driscoll and Kraay's SCC	$V_{SCC,L} = V_{CT} + \sum_{l=1}^{L} w_l[V_{CT,l} + V_{CT,l}^\top]$
Double-clustering + shocks	$V_{CXT,L} = V_{CT} + \sum_{l=1}^{L}[V_{CT,l} + V_{CT,l}^\top] + V_{CX} - V_W - \sum_{l=1}^{L}[V_{W,l} + V_{W,l}^\top] = V_{CT,L} + V_{CX} - V_{NW,L\|w=1}$

Table 1: Covariance structures as combinations of the basic building blocks.

and for the purely autoregressive ones:

$$V_{W,l} = V_g(t, l, f = \mathrm{diag}(u_t \times u_{t-l})) \tag{22}$$

(where $\times$ is the element-by-element product) so that by a (possibly weighted) combination of the above we can get all relevant estimators: see Table 1.

As observed, the SCC estimator differs from the (one-way) time-shocks-robust version of the double-clustering à la CGM only by the distance-decaying weighting of the covariances between different periods, so that $V_{CT,L} = V_{SCC,L|w=1}$.

Obviously, as there is no natural univariate ordering between individuals, a full generalization encompassing both the double-clustering and a two-way SCC estimator does not seem sensible. For the same reason, while the software components allow fully symmetric operation, i.e., it would be practically feasible to compute a group-clustering version of $V_{SCC,L}$ or $V_{CX,L}$, this is devoid of sense from a statistical viewpoint because the notion of a linear, unidimensional spatial lag is generally meaningless[5].

4.2. Unconditional estimation in the general framework

Unconditional estimators can also be computed from the general formulation by precalculating the unconditional error covariance $E = \frac{\sum_{t=1}^{T} u_t u_t^\top}{T}$ and substituting it inside the generic Equation 17 as a constant matrix:

$$V_{UT} = V_g(t, 0, f = E). \tag{23}$$

One noteworthy feature of R is the ability to treat functions as first-class objects (R Core Team 2017), which means they are just another, although very special, data type and can be fed to another function as argument. So a function might indifferently take as argument a function or a precalculated matrix, which is the case here[6].

[5] Spatial lags, where applicable, are defined in a completely different way based on two-dimensional proximity matrices (see Anselin 1988). One very special case of linear spatial arrangement allowing for a simple definition of lag is Chudik, Pesaran, and Tosetti (2011)'s circular world, where each observation has one neighbor to the left and one to the right. Yet in that setting dependence would have to consider both directions, while serial dependence only originates from the past.

[6] Another example of use of this powerful feature for passing a covariance matrix or the function calculating it is in Zeileis (2006).

4.3. Unbalanced panels

Unbalancedness is one of the major computational nuisances in panel data econometrics. In the case at hand, the problem is to compute the generic formula in Equation 17 taking heed that unbalanced samples will have incomplete groups (time periods) for some t (i). As the ultimate goal of estimation of the *meat* is to average the $k \times k$ matrix products $X_t^\top f(u_t, u_{t-l})X$ over time periods (or, symmetrically, groups), missing data will give rise to empty positions in X_t and, correspondingly, in $f(u_t, u_{t-l})$.

Fortunately, R has two particularly useful features for treating data in a "generic" way, as independent as possible from dimensions: `list` objects and the `apply` family of functions.

In general R makes it relatively easy to deal with unbalanced data through the use of structures like `lists`, very flexible containers which can hold e.g., matrices of different dimensions and on which operators (and, more generally, *functions* of any kind) can be *applied*. The `apply` family has members for working member-by-member on `lists` (`lapply`), subgroup by subgroup on *ragged arrays* (`tapply`) where (possibly heterogeneous) subgroups are defined by a grouping variable, or dimension by dimension on arrays, which is the original use of `apply`. One notable advantage of this operator is that it can work on arbitrary subsets of the array's dimensions, provided the function to be applied is compatible.[7]

If a function is *applied* that allows discarding of `NA` values, one can easily get consistent averaging over multidimensional arrays: in this case, an average of t $k \times k$ bidimensional matrices of uneven dimensions.

An example will clarify things. Let us take an array of three 3×3 matrices with a missing value, and average over the third dimension. By default, missing values propagate throughout operations:

```
R> a <- array(1, dim = c(3, 3, 3))
R> a[1, 1, 1] <- NA
R> apply(a, 1:2, mean)

     [,1] [,2] [,3]
[1,]   NA    1    1
[2,]    1    1    1
[3,]    1    1    1
```

but the default behavior can be overridden forcing discarding of `NA`s:

```
R> apply(a, 1:2, mean, na.rm = TRUE)

     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    1    1    1
[3,]    1    1    1
```

[7] Notable examples of the usefulness of `lists` and *ragged arrays* for unbalanced panel data econometrics are, respectively, one-by-one inversion of lists of submatrices in general GLS calculations and time- (group-) demeaning of data based on grouping indices; both in package **plm**.

In the latter case, the resulting 3×3 matrix will contain all averages computed on the correct number of items (i.e., for the $[1,1]$ position, $(1+1)/2$).

Analogously, in our case it will be convenient to make use of standard tridimensional arrays making a $k \times k \times t$ matrix – basically a "pile" of $X_t^\top f(u_t, u_{t-l})X$ terms – and then *applying* the `mean` function over the third dimension, obtaining an appropriate calculation of Equation 17 as a result. In fact, for every value of t every product involving a missing element will produce an `NA` in the relative $k \times k$ matrix; but then averaging over the T dimension will discard `NA`s and apply the correct denominator.[8]

The same goes for the estimation of the unconditional covariance in Beck and Katz type estimators. This feature, which has been unavailable for unbalanced panels for a while and then has been twice mentioned in the literature as a potentially complicated computational problem (Franzese 1996; Bailey and Katz 2011) is solved nicely and without effort in R by applying (sic!) the above method, which acts as advocated by Franzese (1996), averaging elements in the unconditional covariance matrix on the correct number of observations[9].

4.4. Application to FE, RE and FD models: The demeaning framework

From a software viewpoint, the methods provided here can be transparently applied either to pooled OLS or to any other model represented by a '`plm`' object. As usual, what is computationally feasible is not necessarily sound from a statistical viewpoint.

The application of the above estimators to pooled data is always warranted, subject to the relevant assumptions mentioned before. In some, but not all cases, these can also be applied to random or fixed effects panel models, or models estimated on first-differenced data. The general idea is to use both the covariates and residuals from the transformed (partially or totally demeaned, first differenced) model used in estimation.

In all of these cases the estimator is computed as OLS on transformed data, e.g., in the fixed effects case $\hat{\beta}_{FE} = (\tilde{X}^\top \tilde{X})^{-1}\tilde{X}^\top \tilde{y}$ with $\tilde{y}_{it} = y_{it} - y_{i.}$ and $\tilde{x}_{jit} = x_{jit} - x_{ji.}$ for each $\mathbf{x}_j$ in X; while in the random effects case this time-demeaning is partial and $\tilde{y}_{it} = y_{it} - \theta y_{i.}$ with $0 < \theta < 1$. Sandwich estimators can then be computed by applying the usual formula to the transformed data and residuals $\tilde{u} = \tilde{y} - \tilde{X}\hat{\beta}$: see Arellano (1987) and Wooldridge (2002, Section 10.59) for the fixed effects case, Wooldridge (2002, Chapter 10) in general.

In the following I discuss when it is appropriate to apply clustering estimators to the residuals of demeaned or first-differenced models.

Fixed effects

The fixed effects estimator requires particular caution. In fact, under the hypothesis of spherical errors in the original model, the time-demeaning of data induces a serial correlation $\mathsf{COR}(u_{it}, u_{i,t-1}) = -1/(T-1)$ in the demeaned residuals (see Wooldridge 2002, p. 275).

The White-Arellano estimator has originally been devised for this case. By way of symmetry, it can be used for time-clustered data with time fixed effects. The combination of group-

[8] Of course the most delicate programming issue becomes correct handling of the positions of incomplete u_t subvectors and X_t submatrices in the relevant tth "layer" of the tridimensional array.

[9] This estimation method, based on all available covariances between two given observations, corresponds to the `pairwise` option in the **pcse** function and package (Bailey and Katz 2011); it must be noted, though, that the default option there (`casewise`) is to use a balanced subset of the data.

clustering with time fixed effects and the reverse seems inappropriate because of the serial (cross-sectional) correlation induced by the time- (cross-sectional-) demeaning.

By analogy, the Newey-West type estimators can be safely applied to models with individual fixed effects (for an application, see Golden and Picci 2008), while the time and two-way cases require caution.

Random effects

In the random effects case, as Wooldridge (2002) notes, the quasi-time demeaning procedure removes the random effects reducing the model on transformed data to a pooled regression, thus preserving the properties of the White-type estimators.

By extension of this line of reasoning, all above estimators seem to be applicable to the demeaned data of a random effects model, provided *the transformed errors* meet the relevant assumptions.

First-differences

First-differencing, like fixed effects estimation, removes time-invariant effects. Roughly speaking, the choice between the two rests on the properties of the error term: if it is assumed to be well-behaved in the original data, then FE is the most efficient estimator and is to be preferred; if on the contrary the original errors are believed to behave as a random walk, then first-differencing the data will yield stationary and uncorrelated errors, and is therefore advisable (see Wooldridge 2002, p. 281). Given this, FD estimation is nothing else than OLS on differenced data, and the usual clustering formula applies (see Wooldridge 2002, p. 282). As in the RE case, the statistical properties of the various covariance estimators will depend on whether *the transformed errors* meet the relevant assumptions.

From the viewpoint of software implementation, the application to fixed or random effects and to first-difference models is greatly simplified by the availability in **plm** of a comprehensive data transformation infrastructure, allowing to easily extract the original data from the model object and apply the relevant transformation (see Croissant and Millo 2008).

4.5. Small-sample corrections

Two kinds of small-sample corrections are implemented: corrections for a small number of observations, derived from the work of MacKinnon and White (1985) and summarized in Zeileis (2006), and corrections for a small number of clusters, described in Cameron *et al.* (2011, p. 8).

All work by multiplying each residual by the square root of the appropriate correction factor $\sqrt{c}$, so that the various squares and cross-products of residuals are correctly multiplied by c while the correction can work at vector level, separating the small-sample-correction module from the other logical steps of computation. The cluster-level correction in turn works at single-clustering level, according to the relevant numerosity parameters, as suggested in Cameron *et al.* (2011): therefore small-sample cluster-level corrections for different clustering dimensions are seamlessly combined.

In all these cases $c > 0$, and $c \to 1$ as the total number of observations or, in the latter case, the number of clusters diverge.

5. R implementation

In this section I will first put the covariance estimators in the context of the **plm** package for panel data econometrics and provide a minimal background on robust restriction testing through interoperability between testing functions and covariance estimators. Then I will describe how the new approach detailed in this paper has been implemented, substituting the existing procedures in the simpler cases while extending functionality to the more complex ones. Lastly I will provide some applied examples to illustrate usage.

5.1. plm and generic sandwich estimators

Robust covariance estimators à la White or à la Newey and West for different kinds of regression models are available in package **sandwich** (Zeileis and Lumley 2017) under form of appropriate methods for the generic functions `vcovHC` and `vcovHAC` (Zeileis 2004, 2006). These are designed for data sampled along one dimension, therefore they cannot generally be used for panel data; yet they provide a uniform and flexible software approach which has become standard in the R environment. The procedures described in this paper have therefore been designed to be syntactically compliant with them.

plm (Croissant and Millo 2008) is an R package for panel data econometrics in which an S3 method for '`plm`' objects for the generic function `vcovHC` has long been available, allowing to apply sandwich estimators to panel models in a way that is natural for users of the **sandwich** package. In fact, despite the different structure "under the hood", the user will, e.g., specify a robust covariance for the diagnostics table of a panel model in the same way she would for a linear or a generalized linear model, the object-orientation features of R taking care that the right statistical procedure be applied to the model object at hand. What will change, though, are the defaults: the `vcovHC` method for '`lm`' objects defaults to the original White estimator, while the `vcovHC` method for '`plm`' objects to clustering by groups, both the most obvious choices for the object at hand.

As an example, Munnell (1990) specifies a Cobb-Douglas production function that relates the gross social product (gsp) of a given US state to the input of public capital (pcap), private capital (pc), labor (emp) and state unemployment rate (unemp) added to capture business cycle effects. Considering this model, whose dataset is a built-in example in **plm**,

```
R> library("plm")
R> data("Produc", package = "plm")
R> fm <- log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp
```

and the function `coeftest` from package **lmtest** (Zeileis and Hothorn 2002) which produces a compact coefficients table allowing for a flexible choice of the covariance matrix, I calculate the "robust" diagnostic table for two statistically equivalent models: OLS by `lm`

```
R> lmmod <- lm(fm, Produc)
R> library("lmtest")
R> library("sandwich")
R> coeftest(lmmod, vcov = vcovHC)

t test of coefficients:
```

```
               Estimate Std. Error t value  Pr(>|t|)
(Intercept)  1.6433023  0.0716070 22.9489 < 2.2e-16 ***
log(pcap)    0.1550070  0.0186973  8.2903 4.668e-16 ***
log(pc)      0.3091902  0.0126283 24.4839 < 2.2e-16 ***
log(emp)     0.5939349  0.0197887 30.0139 < 2.2e-16 ***
unemp       -0.0067330  0.0013501 -4.9872 7.497e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

and pooled OLS by `plm`

```
R> plmmod <- plm(fm, Produc, model = "pooling")
R> coeftest(plmmod, vcov = vcovHC)

t test of coefficients:

               Estimate Std. Error t value  Pr(>|t|)
(Intercept)  1.6433023  0.2441821  6.7298 3.211e-11 ***
log(pcap)    0.1550070  0.0601195  2.5783   0.01010 *
log(pc)      0.3091902  0.0462297  6.6881 4.209e-11 ***
log(emp)     0.5939349  0.0686061  8.6572 < 2.2e-16 ***
unemp       -0.0067330  0.0030904 -2.1787   0.02964 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As can be seen, the estimated SEs will turn out different as the types of the model objects to be tested are different, unless one overrides the defaults: here specifying the `method` as `"white1"` and the small sample correction as `"HC3"` will replicate the `lm` results:

```
R> coeftest(plmmod,
+     vcov = function(x) vcovHC(x, method = "white1", type = "HC3"))

t test of coefficients:

               Estimate Std. Error t value  Pr(>|t|)
(Intercept)  1.6433023  0.0716070 22.9489 < 2.2e-16 ***
log(pcap)    0.1550070  0.0186973  8.2903 4.668e-16 ***
log(pc)      0.3091902  0.0126283 24.4839 < 2.2e-16 ***
log(emp)     0.5939349  0.0197887 30.0139 < 2.2e-16 ***
unemp       -0.0067330  0.0013501 -4.9872 7.497e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As observed, these features have long been present in **plm**, but limited to one-way clustering (see Croissant and Millo 2008, Section 6.7); one-way SCC and the unconditional Beck and Katz (BK) estimator have also been added at a later stage, each one with its own infrastructure. With the exception of BK, this functionality has now been replaced by a combination of a general parameter covariance estimator as in Equation 17 and specific wrappers, replicating its different particularizations for the most common forms of usage.

5.2. The new modular framework

In this section I show how to use the basic "building block": the general estimator in Equation 17. This is unlikely to be much used in practice but it is left available at user level both for educational use and to possibly allow combinations not implemented in the higher-level wrappers. Then I show what is probably going to be the preferred option for practicing econometricians, that is the higher-level wrappers combining different particularizations of the general estimator to obtain one- or two-way clustering or kernel-weighted estimators à la White, Arellano, CGM, NW or DK. Lastly I show how to easily define custom combinations of the above to estimate more complicated covariance structures.

The general parameter covariance estimator has been implemented in R in a function `vcovG` which is the software counterpart to Equation 18 and can be used for calculating $V_{W,l}$, $V_{CT,l}$ or $V_{CX,l}$. This is visible at the user level and can be used as such, leaving the default lag at 0, to calculate V_W, V_{CT} or V_{CX}. According to the formalization in Equation 18, besides a '`plm`' object and a small-sample correction, it takes as arguments a clustering dimension (`cluster`), a function of the errors corresponding to $\mathsf{E}(u)$ in Equation 9 (`inner`) and a lag order. The `inner` argument accepts either one of two strings `"cluster"` or `"white"`, specifying respectively $\mathsf{E}(u) = uu^\top$ and $\mathsf{E}(u) = \text{diag}(u^\top u)$, or a user-supplied function.

Next, I calculate the Arellano estimator V_{CX} by specifying `"group"` as the clustering dimension, `"cluster"` as the inner function and 0 as the lag order:

```
R> vcovG(plmmod, cluster = "group", inner = "cluster", l = 0)

              (Intercept)     log(pcap)       log(pc)      log(emp)
(Intercept)  0.0596248904 -9.637916e-03 -0.0068911857  0.0148866870
log(pcap)   -0.0096379163  3.614354e-03 -0.0002956929 -0.0031157168
log(pc)     -0.0068911857 -2.956929e-04  0.0021371841 -0.0017597732
log(emp)     0.0148866870 -3.115717e-03 -0.0017597732  0.0047067982
unemp        0.0003700792 -8.058266e-05 -0.0000586966  0.0001366349
                    unemp
(Intercept)  3.700792e-04
log(pcap)   -8.058266e-05
log(pc)     -5.869660e-05
log(emp)     1.366349e-04
unemp        9.550671e-06
attr(,"cluster")
[1] "group"
```

For the convenience of the user, a wrapper function `vcovHC` is provided which reproduces the syntax and results of the stand-alone version already available in **plm**, thus ensuring both retrocompatibility with **plm** and naming consistency with the **sandwich** package. Thus, the following statement reproduces the same output as above (suppressed) in a more intuitive way:

```
R> vcovHC(plmmod)
```

Higher-level functions are needed, and provided, in order to produce the double-clustering and kernel-smoothing estimators by (possibly weighted) sums of the former terms. The general tool in this respect, in turn based on `vcovG`, is `vcovSCC`, which computes weighted sums of $V_{\cdot,l}$ according to a weighting function which is by default the Bartlett kernel. Again, this function is available at user level and the default values will yield the Driscoll and Kraay estimator, $V_{SCC,L}$:

```
R> vcovSCC(plmmod)

              (Intercept)     log(pcap)       log(pc)      log(emp)
(Intercept)  0.0226046609 -5.514511e-03 -6.334497e-04  5.759358e-03
log(pcap)   -0.0055145106  1.367029e-03  1.319429e-04 -1.402905e-03
log(pc)     -0.0006334497  1.319429e-04  5.843328e-05 -1.862888e-04
log(emp)     0.0057593584 -1.402905e-03 -1.862888e-04  1.497875e-03
unemp       -0.0003377024  8.428261e-05  3.257782e-06 -8.034358e-05
                    unemp
(Intercept) -3.377024e-04
log(pcap)    8.428261e-05
log(pc)      3.257782e-06
log(emp)    -8.034358e-05
unemp        6.445790e-06
attr(,"cluster")
[1] "time"
```

No weighting (equivalent to passing the constant 1 as the weighting function: `wj = 1`) will produce the building blocks for double-clustering, according to Equation 13, so that V_{CXT} could be easily obtained defining it at user level as:[10]

```
R> myvcovDC <- function(x, ...) {
+    Vcx <- vcovHC(x, cluster = "group", method = "arellano", ...)
+    Vct <- vcovHC(x, cluster = "time", method = "arellano", ...)
+    Vw <- vcovHC(x, method = "white1", ...)
+    return(Vcx + Vct - Vw)
+ }
R> myvcovDC(plmmod)

              (Intercept)     log(pcap)       log(pc)      log(emp)
(Intercept)  0.0635274416 -1.087953e-02 -0.0067108330  0.0159466020
log(pcap)   -0.0108795286  3.809110e-03 -0.0002102193 -0.0033786244
log(pc)     -0.0067108330 -2.102193e-04  0.0020211433 -0.0017355810
log(emp)     0.0159466020 -3.378624e-03 -0.0017355810  0.0049283961
```

[10] Notice the use of the prefix "my" to indicate that this function has been defined by the user in this session, as opposed to built-in functions. This is done only for the sake of clarity, as R leaves complete naming freedom to the user; yet adhering to naming conventions of some sort is advisable in order to avoid inadvertently replacing built-in functions.

```
unemp        0.0002236813 -4.386756e-05 -0.0000544364  0.0000986291
                    unemp
(Intercept)  2.236813e-04
log(pcap)   -4.386756e-05
log(pc)     -5.443640e-05
log(emp)     9.862910e-05
unemp        1.108906e-05
attr(,"cluster")
[1] "group"
```

Again, convenience wrappers are provided to make usage more intuitive: `vcovNW` computes the panel Newey-West estimator $V_{NW,L}$ (output omitted); `vcovDC` the double-clustering one V_{CXT}, which is constructed not unlike `myvcovDC` from the example above, and gives exactly the same output (suppressed):

```
R> vcovDC(plmmod)
```

More complicated structures allowing for two-way clustering and error persistence in the sense of Thompson (2011) are easily obtained by combination, the same way as illustrated above, following the lines of Section 4.1. Below the case of double-clustering plus four periods of persistent (unweighted) shocks à la Thompson (2011) (notice that the weighting function `wj` has been defined as the constant 1 but must still be a function of two arguments):

```
R> myvcovDCS <- function(x, maxlag = NULL, ...) {
+    w1 <- function(j, maxlag) 1
+    VsccL.1 <- vcovSCC(x, maxlag = maxlag, wj = w1, ...)
+    Vcx <- vcovHC(x, cluster = "group", method = "arellano", ...)
+    VnwL.1 <- vcovSCC(x, maxlag = maxlag, inner = "white", wj = w1, ...)
+    return(VsccL.1 + Vcx - VnwL.1)
+  }
R> myvcovDCS(plmmod, maxlag = 4)

              (Intercept)     log(pcap)       log(pc)      log(emp)
(Intercept)  0.0766973526 -0.0160969792 -4.713237e-03  0.0191602519
log(pcap)   -0.0160969792  0.0043713347  2.332514e-04 -0.0042963693
log(pc)     -0.0047132370  0.0002332514  1.066283e-03 -0.0012435555
log(emp)     0.0191602519 -0.0042963693 -1.243556e-03  0.0052481667
unemp       -0.0006069241  0.0001587212 -9.439635e-06 -0.0001351121
                    unemp
(Intercept) -6.069241e-04
log(pcap)    1.587212e-04
log(pc)     -9.439635e-06
log(emp)    -1.351121e-04
unemp        1.403075e-05
attr(,"cluster")
[1] "time"
```

6. Applied examples

In the following applied examples, I will present the complete array of standard error estimates for each estimator in Table 1. A complete array of methods is presented for the sake of illustration; nevertheless one must keep in mind that the sample size and the number of clusters in either cross-section or time might prove inadequate for some estimators, as reported in the reference papers (see in particular Thompson 2011). The examples below must therefore be seen as examples of computational feasibility, not of statistical soundness of each method.

In fact, even limiting to those methods that are not at odds with the given sample size, the strategy of computing standard errors in all potentially sensible ways and taking the most conservative ones does indeed reduce type I error but at the same time decreases the power of the significance test.[11]

Another purpose of this section is to illustrate some ways to efficiently perform such multiple comparisons through some features of R. Looping on a vector of functions is a useful consequence of R treating functions as a data type. For the sake of clarity, let us predefine some functions for calculating the different covariance estimators in Section 4.1 according to the names reported there and with the appropriate parameters (leaving the maximum lag calculation at its default value of $L = T^{\frac{1}{4}}$):

```
R> Vw <- function(x) vcovHC(x, method = "white1")
R> Vcx <- function(x) vcovHC(x, cluster = "group", method = "arellano")
R> Vct <- function(x) vcovHC(x, cluster = "time", method = "arellano")
R> Vcxt <- function(x) Vcx(x) + Vct(x) - Vw(x)
R> Vct.L <- function(x) vcovSCC(x, wj = function(j, maxlag) 1)
R> Vnw.L <- function(x) vcovNW(x)
R> Vscc.L <- function(x) vcovSCC(x)
R> Vcxt.L <- function(x)
+    Vct.L(x) + Vcx(x) - vcovNW(x, wj = function(j, maxlag) 1)
```

then build up a vector of functions on which to loop:

```
R> vcovs <- c(vcov, Vw, Vcx, Vct, Vcxt, Vct.L, Vnw.L, Vscc.L, Vcxt.L)
R> names(vcovs) <- c("OLS", "Vw", "Vcx", "Vct", "Vcxt", "Vct.L", "Vnw.L",
+    "Vscc.L", "Vcxt.L")
```

in order to calculate a comprehensive table of p values from robust estimators:

```
R> cfrtab <- matrix(nrow = length(coef(plmmod)), ncol = 1 + length(vcovs))
R> dimnames(cfrtab) <- list(names(coef(plmmod)), c("Coefficient",
+    paste("s.e.", names(vcovs))))
R> cfrtab[, 1] <- coef(plmmod)
R> for (i in 1:length(vcovs)) {
+    cfrtab[, 1 + i] <- coeftest(plmmod, vcov = vcovs[[i]])[, 2]
+ }
R> print(t(round(cfrtab, 4)))
```

[11] We are grateful to an anonymous reviewer for this observation.

```
             (Intercept) log(pcap) log(pc) log(emp)   unemp
Coefficient       1.6433    0.1550  0.3092   0.5939 -0.0067
s.e. OLS          0.0576    0.0172  0.0103   0.0137  0.0014
s.e. Vw           0.0708    0.0185  0.0125   0.0195  0.0013
s.e. Vcx          0.2442    0.0601  0.0462   0.0686  0.0031
s.e. Vct          0.0944    0.0232  0.0063   0.0246  0.0018
s.e. Vcxt         0.2520    0.0617  0.0450   0.0702  0.0033
s.e. Vct.L        0.1875    0.0461  0.0079   0.0480  0.0031
s.e. Vnw.L        0.1144    0.0299  0.0206   0.0316  0.0020
s.e. Vscc.L       0.1503    0.0370  0.0076   0.0387  0.0025
s.e. Vcxt.L       0.2722    0.0657  0.0389   0.0736  0.0036
```

6.1. PPP regression

This example applies the new combination `Vcxt.L`, which as observed is undocumented in the literature, in the appropriate context of a "long" panel. Its main purpose is to show how to apply the methodology discussed in the paper to linear hypothesis testing.

Coakley, Fuertes, and Smith (2006) present a purchasing power parity (PPP) regression on quarterly data 1973Q1 to 1998Q4 for 17 developed countries, so that $N = 17$ and $T = 104$ which is fairly typical of a "long" panel.[12] The estimated model is

$$\Delta s_{it} = \alpha + \beta(\Delta p - \Delta p^*)_{it} + \nu_{it},$$

where s_{it} is the relative exchange rate against USD and $(\Delta p - \Delta p^*)_{it}$ is the inflation differential between the country and the US.

```
R> data("Parity", package = "plm")
R> fm <- ls ~ ld
R> pppmod <- plm(fm, data = Parity, effect = "twoways")
```

The hypothesis of interest is $\beta = 1$, therefore instead of significance diagnostics we report the corresponding robust Wald test from `linearHypothesis` in package **car** (Fox and Weisberg 2011):

```
R> library("car")
R> linearHypothesis(pppmod, "ld = 1", vcov = Vcxt.L)

Linear hypothesis test

Hypothesis:
ld = 1

Model 1: restricted model
Model 2: ls ~ ld
```

[12] The first of three examples in the original SCC paper (Driscoll and Kraay 1998) is also a purchasing power parity regression, on annual data 1973–1993 for a sample of 107 countries.

```
Note: Coefficient covariance matrix supplied.

  Res.Df Df  Chisq Pr(>Chisq)
1   1648
2   1647  1 2.2942     0.1299
```

6.2. Petersen's artificial data

The last example draws on a well-known simulated dataset, replicating the original results. To complement his paper, Petersen (2009) produced a simple artificial dataset, which has become an informal benchmark for practitioners. The data can be retrieved from http://www.kellogg.northwestern.edu/faculty/petersen/htm/papers/se/test_data.txt; a copy is provided in the accompanying materials to this article. He provides the following estimates of standard errors: classical, White heteroskedastic, clustered by firm or year, double-clustered by firm and year; and coefficients and standard errors estimated according to the Fama-MacBeth procedure. In the following, I replicate his results in R with **plm**.

```
R> petersen <- read.table(file = "test_data.txt")
R> colnames(petersen) <- c("firmid", "year", "x", "y")
R> ptrmod <- plm(y ~ x, data = petersen, index = c("firmid", "year"),
+    model = "pooling")
R> vcovs <- c(vcov, Vw, Vcx, Vct, Vcxt)
R> names(vcovs) <- c("OLS", "Vw", "Vcx", "Vct", "Vcxt")
R> cfrtab <- matrix(nrow = length(coef(ptrmod)), ncol = 1 + length(vcovs))
R> dimnames(cfrtab) <- list(names(coef(ptrmod)), c("Coefficient",
+    paste("s.e.", names(vcovs))))
R> cfrtab[, 1] <- coef(ptrmod)
R> for(i in 1:length(vcovs)) {
+    cfrtab[, 1 + i] <- coeftest(ptrmod, vcov = vcovs[[i]])[, 2]
+  }
R> print(t(round(cfrtab, 4)))

            (Intercept)      x
Coefficient      0.0297 1.0348
s.e. OLS         0.0284 0.0286
s.e. Vw          0.0284 0.0284
s.e. Vcx         0.0669 0.0505
s.e. Vct         0.0222 0.0317
s.e. Vcxt        0.0646 0.0525
```

One should notice a small difference w.r.t. the results of Petersen: in fact, to replicate them exactly one shall specify to use the same small sample correction Stata (StataCorp. 2015) uses by default: e.g., in the double-clustering case,

```
R> coeftest(ptrmod, vcov = function(x) vcovDC(x, type = "sss"))[, 2]
```

```
(Intercept)           x
 0.06506392  0.05355802
```

which yields the same results as double-clustering in Petersen's example.[13]

7. Conclusions

I have reviewed the different robust estimators for the standard errors of panel models used in applied econometric practice, representing them as combinations of atomic building blocks, which can be thought of as the computational counterparts of statistical objects. In turn, these have been defined, according to the functional orientation of R, as variations of the same general element obtained by choosing a clustering dimension (group or time), a lag order and a function of the residuals (either the element-by-element or the outer product).

While it is feasible to combine these constituents *ad hoc* at user level, the standard estimators used in applied practice (White, Arellano, Newey-West, Driscoll and Kraay SCC, double clustering) are provided under form of predefined combinations ("wrapper" functions) for the sake of user-friendliness. Nevertheless, the user enjoys the freedom to combine elements at will, possibly experimenting with non-standard solutions.

The software framework described is integrated in the R package **plm**, so that composite covariance methods can be applied to objects representing panel models of different kinds (FE, RE, FD and, obviously, OLS). The estimate of the parameters' covariance thus obtained can in turn be plugged into diagnostic testing functions, producing either significance tables or hypothesis tests. A function is a regular object type in R, hence compact comparisons of standard errors from different (statistical) methods can be produced simply by looping on covariance types, as shown in the examples.

An extension to multiple clustering dimensions as in Cameron *et al.* (2011) is ill-suited to bidimensional econometric panels, and hence out of the scope of this paper; it has recently been implemented in package **multiwayvcov** for linear models (Graham *et al.* 2016), and can foreseeably be adapted to panel settings by combining the latter with demeaning functionality in **plm** (i.e., treating the transformed data as a classical linear model, see Section 4.4) in ways that look fairly straightforward but are out of the scope of the present work.

Lastly, one caveat applies. This paper is concerned with design-efficient computing of a quite general class of estimators. Generality will mean that many different estimators can be fitted to the data obtaining numerical estimates. Advances in computing power have made most of these computationally very cheap, hence a conservative "fit-them-all" strategy is feasible (although conservativeness will come at the expense of test power: see the observations at the beginning of Section 6). It must nevertheless be borne in mind that computability does not by any means guarantee statistical soundness, and that the hypotheses under which a covariance estimator is consistent and has desirable properties in finite samples are usually a subset of those under which it is actually computable.

[13]Petersen also reports Fama-MacBeth estimates. As observed in Section 1, these are nothing else but a mean groups estimator where means are taken over time instead of, as is customary in panel time series econometrics, over individual observations. Therefore this last part of Petersen's results can be replicated by swapping indices in the **plm** function `pmg`, as in the following statement: `coeftest(pmg(y ~ x, data = petersen, index = c("year", "firmid")))`.

Acknowledgments

My work has benefited greatly, at different stages, from discussions with: Liviu Andronic; Yves Croissant and Kevin Tappe, my coauthors in **plm**; Achim Zeileis, who originally devised the object-oriented framework for robust inference (Zeileis 2004, 2006); and Mahmood Arai, who provided the original implementation of panel (double-)clustered standard errors in R (Arai 2009). Despite many differences in the software approach, this implementation has been inspired by his work. Useful comments and suggestions from John Driscoll, Harry Kelejian and Douglas Miller are also gratefully acknowledged. Prof. Ron Smith is thanked for the `Parity` data.

References

Anselin L (1988). *Spatial Econometrics: Methods and Models.* Kluwer Academic Publisher, Dordrecht. doi:10.1007/978-94-015-7799-1.

Aptech Systems, Inc (2006). *GAUSS Mathematical and Statistical System 8.0.* Aptech Systems, Inc., Black Diamond, Washington. URL http://www.Aptech.com/.

Arai M (2009). "Cluster-Robust Standard Errors Using R." *Technical report*, Department of Economics and SULCIS, Stockholm University. URL http://people.su.se/ma/clustering.pdf.

Arellano M (1987). "Computing Robust Standard Errors for within Group Estimators." *Oxford Bulletin of Economics and Statistics*, **49**(4), 431–434. doi:10.1111/j.1468-0084.1987.mp49004006.x.

Arellano M (2003). *Panel Data Econometrics.* Oxford University Press.

Bailey D, Katz JN (2011). "Implementing Panel-Corrected Standard Errors in R: The **pcse** Package." *Journal of Statistical Software*, **42**(1), 1–11. doi:10.18637/jss.v042.c01.

Beck N, Katz JN (1995). "What to Do (and Not to Do) with Time-Series Cross-Section Data." *The American Political Science Review*, **89**(3), 634–647. doi:10.2307/2082979.

Beck N, Katz JN (1996). "Nuisance vs. Substance: Specifying and Estimating Time-Series-Cross-Section Models." *Political Analysis*, **6**(1), 1–36. doi:10.1093/pan/6.1.1.

Berger S, Graham N, Zeileis A (2017). "Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R." *Working Paper 2017-12*, Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universität Innsbruck. URL http://EconPapers.RePEc.org/RePEc:inn:wpaper:2017-12.

Cameron AC, Gelbach JB, Miller DL (2011). "Robust Inference with Multiway Clustering." *Journal of Business & Economic Statistics*, **29**(2), 238–249. doi:10.1198/jbes.2010.07136.

Chudik A, Pesaran MH, Tosetti E (2011). "Weak and Strong Cross-Section Dependence and Estimation of Large Panels." *The Econometrics Journal*, **14**(1), c45–c90. doi:10.1111/j.1368-423x.2010.00330.x.

Coakley J, Fuertes AM, Smith R (2006). "Unobserved Heterogeneity in Panel Time Series Models." *Computational Statistics & Data Analysis*, **50**(9), 2361–2380. doi:10.1016/j.csda.2004.12.015.

Croissant Y, Millo G (2008). "Panel Data Econometrics in R: The **plm** Package." *Journal of Statistical Software*, **27**(2), 1–43. doi:10.18637/jss.v027.i02.

Driscoll JC, Kraay AC (1998). "Consistent Covariance Matrix Estimation with Spatially Dependent Panel Data." *The Review of Economics and Statistics*, **80**(4), 549–560. doi:10.1162/003465398557825.

Fama EF, MacBeth JD (1973). "Risk, Return, and Equilibrium: Empirical Tests." *The Journal of Political Economy*, **81**(3), 607–636. doi:10.1086/260061.

Fox J, Weisberg S (2011). *An R Companion to Applied Regression*. 2nd edition. Sage, Thousand Oaks.

Franzese R (1996). "A GAUSS Procedure to Estimate Panel-Corrected Standard-Errors with Non-Rectangular and/or Missing Data." *The Political Methodologist*, **7**(2), 2–3.

Froot KA (1989). "Consistent Covariance Matrix Estimation with Cross–Sectional Dependence and Heteroskedasticity in Financial Data." *Journal of Financial and Quantitative Analysis*, **24**(3), 333–355. doi:10.2307/2330815.

Golden MA, Picci L (2008). "Pork-Barrel Politics in Postwar Italy, 1953–94." *American Journal of Political Science*, **52**(2), 268–289. doi:10.1111/j.1540-5907.2007.00312.x.

Graham N, Arai M, Hagströmer B (2016). "**multiwayvcov**: Multi-Way Standard Error Clustering." R package version 1.2.3, URL https://CRAN.R-project.org/package=multiwayvcov.

Greene WH (2003). *Econometric Analysis*. 5th edition. Prentice Hall.

Hothorn T, Hornik K, Van De Wiel MA, Zeileis A (2006). "A Lego System for Conditional Inference." *The American Statistician*, **60**(3), 257–263. doi:10.1198/000313006x118430.

Ibragimov R, Müller UK (2010). "*t*-Statistic Based Correlation and Heterogeneity Robust Inference." *Journal of Business & Economic Statistics*, **28**(4), 453–468. doi:10.1198/jbes.2009.08046.

Liang KY, Zeger SL (1986). "Longitudinal Data Analysis Using Generalized Linear Models." *Biometrika*, **73**(1), 13–22. doi:10.1093/biomet/73.1.13.

MacKinnon JG, White H (1985). "Some Heteroskedasticity-Consistent Covariance Matrix Estimators with Improved Finite Sample Properties." *Journal of Econometrics*, **29**(3), 305–325. doi:10.1016/0304-4076(85)90158-7.

Munnell A (1990). "Why Has Productivity Growth Declined? Productivity and Public Investment." *New England Economic Review*, **January/February**, 3–22.

Newey WK, West KD (1987). "A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix." *Econometrica*, **55**(3), 703–708. doi:10.2307/1913610.

Pesaran MH, Smith R (1995). "Estimating Long-Run Relationships from Dynamic Heterogeneous Panels." *Journal of Econometrics*, **68**(1), 79–113. doi:10.1016/0304-4076(94)01644-f.

Petersen MA (2009). "Estimating Standard Errors in Finance Panel Data Sets: Comparing Approaches." *The Review of Financial Studies*, **22**(1), 435–480. doi:10.1093/rfs/hhn053.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

StataCorp (2015). *Stata Data Analysis Statistical Software: Release 14*. StataCorp LP, College Station. URL http://www.stata.com/.

The MathWorks Inc (2017). *MATLAB – The Language of Technical Computing, Version R2017a*. Natick. URL http://www.mathworks.com/products/matlab/.

Thompson SB (2011). "Simple Formulas for Standard Errors That Cluster by Both Firm and Time." *Journal of Financial Economics*, **99**(1), 1–10. doi:10.1016/j.jfineco.2010.08.016.

White H (1980). *Asymptotic Theory for Econometricians*. Academic Press, Orlando.

White H (1984). "A Heteroskedasticity-Consistent Covariance Matrix and a Direct Test for Heteroskedasticity." *Econometrica*, **48**(4), 817–838. doi:10.2307/1912934.

Wilson SE, Butler DM (2007). "A Lot More to Do: The Sensitivity of Time-Series Cross-Section Analyses to Simple Alternative Specifications." *Political Analysis*, **15**(2), 101–123. doi:10.1093/pan/mpl012.

Wooldridge JM (2002). *Econometric Analysis of Cross-Section and Panel Data*. MIT Press.

Zeileis A (2004). "Econometric Computing with HC and HAC Covariance Matrix Estimators." *Journal of Statistical Software*, **11**(10), 1–17. doi:10.18637/jss.v011.i10.

Zeileis A (2006). "Object-Oriented Computation of Sandwich Estimators." *Journal of Statistical Software*, **16**(9), 1–16. doi:10.18637/jss.v016.i09.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." *R News*, **2**(3), 7–10.

Zeileis A, Lumley T (2017). ***sandwich**: Robust Covariance Matrix Estimators*. R package version 2.4-0, URL https://CRAN.R-project.org/package=sandwich.

Affiliation:

Giovanni Millo
Group Insurance Research, Generali SpA
and
DEAMS, University of Trieste
Via Machiavelli 3
34132 Trieste, Italy
Telephone: +39/040/671184
Fax: +39/040/671160
E-mail: Giovanni.Millo@Generali.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
November 2017, Volume 82, Issue 3 *Submitted:* 2015-12-15
doi:10.18637/jss.v082.i03 *Accepted:* 2016-10-12