Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in **R**

Achim Zeileis
Universität Innsbruck

Susanne Köll
Universität Innsbruck

Nathaniel Graham
Texas A&M
International University

Abstract

This introduction to the object-oriented implementation of clustered covariances in the R package **sandwich** is a (slightly) modified version of Zeileis, Köll, and Graham (2020), published in the *Journal of Statistical Software.*

Clustered covariances or clustered standard errors are very widely used to account for correlated or clustered data, especially in economics, political sciences, and other social sciences. They are employed to adjust the inference following estimation of a standard least-squares regression or generalized linear model estimated by maximum likelihood. Although many publications just refer to "the" clustered standard errors, there is a surprisingly wide variety of clustered covariances, particularly due to different flavors of bias corrections. Furthermore, while the linear regression model is certainly the most important application case, the same strategies can be employed in more general models (e.g., for zero-inflated, censored, or limited responses).

In R, functions for covariances in clustered or panel models have been somewhat scattered or available only for certain modeling functions, notably the (generalized) linear regression model. In contrast, an object-oriented approach to "robust" covariance matrix estimation – applicable beyond `lm()` and `glm()` – is available in the **sandwich** package but has been limited to the case of cross-section or time series data. Starting with **sandwich** 2.4.0, this shortcoming has been corrected: Based on methods for two generic functions (`estfun()` and `bread()`), clustered and panel covariances are provided in `vcovCL()`, `vcovPL()`, and `vcovPC()`. Moreover, clustered bootstrap covariances are provided in `vcovBS()`, using model `update()` on bootstrap samples. These are directly applicable to models from packages including **MASS**, **pscl**, **countreg**, and **betareg**, among many others. Some empirical illustrations are provided as well as an assessment of the methods' performance in a simulation study.

Keywords: clustered data, covariance matrix estimator, object orientation, simulation, R.

1. Introduction

Observations with correlations between objects of the same group/cluster are often referred to as "cluster-correlated" observations. Each cluster comprises multiple objects that are correlated within, but not across, clusters, leading to a nested or hierarchical structure (Galbraith, Daniel, and Vissel 2010). Ignoring this dependency and pretending observations are independent not only across but also within the clusters, still leads to parameter estimates that are consistent (albeit not efficient) in many situations. However, the observations' information

will typically be overestimated and hence lead to overstated precision of the parameter estimates and inflated type I errors in the corresponding tests (Moulton 1986, 1990). Therefore, clustered covariances are widely used to account for clustered correlations in the data.

Such clustering effects can emerge both in cross-section and in panel (or longitudinal) data. Typical examples for clustered cross-section data include firms within the same industry or students within the same school or class. In panel data, a common source of clustering is that observations for the same individual at different time points are correlated while the individuals may be independent (Cameron and Miller 2015).

This paper contributes to the literature particularly in two respects: (1) Most importantly, we discuss a set of computational tools for the R system for statistical computing (R Core Team 2018), providing an object-oriented implementation of clustered covariances/standard errors in the R package **sandwich** (Zeileis 2004, 2006b). Using this infrastructure, sandwich covariances for cross-section or time series data have been available for models beyond `lm()` or `glm()`, e.g., for packages **MASS** (Venables and Ripley 2002), **pscl**/**countreg** (Zeileis, Kleiber, and Jackman 2008), and **betareg** (Cribari-Neto and Zeileis 2010; Grün, Kosmidis, and Zeileis 2012), among many others. However, corresponding functions for clustered or panel data had not been available in **sandwich** but have been somewhat scattered or available only for certain modeling functions.

(2) Moreover, we perform a Monte Carlo simulation study for various response distributions with the aim to assess the performance of clustered standard errors beyond `lm()` and `glm()`. This also includes special cases for which such a finite-sample assessment has not yet been carried out in the literature (to the best of our knowledge).

The rest of this manuscript is structured as follows: Section 2 discusses the idea of clustered covariances and reviews existing R packages for sandwich as well as clustered covariances. Section 3 deals with the theory behind sandwich covariances, especially with respect to clustered covariances for cross-sectional and longitudinal data, clustered data, as well as panel data. Section 4 then takes a look behind the scenes of the new object-oriented R implementation for clustered covariances, Section 5 gives an empirical illustration based on data provided from Petersen (2009) and Aghion, Van Reenen, and Zingales (2013). The simulation setup and results are discussed in Section 6.

2. Overview

In the statistics as well as in the econometrics literature a variety of strategies are popular for dealing with clustered dependencies in regression models. Here we give a a brief overview of clustered covariances methods, some background information, as well as a discussion of corresponding software implementations in R and Stata (StataCorp 2019).

2.1. Clustered dependencies in regression models

In the statistics literature, random effects (especially random intercepts) are often introduced to capture unobserved cluster correlations (e.g., using the **lme4** package in R, Bates, Mächler, Bolker, and Walker 2015). Alternatively, generalized estimating equations (GEE) can account for such correlations by adjusting the model's scores in the estimation (Liang and Zeger 1986), also leading naturally to a clustered covariance (e.g., available in the **geepack** package for R, Halekoh, Højsgaard, and Yan 2005). Feasible generalized least squares can be applied in

a similar way and is frequently used in panel data econometrics (e.g., available in the **plm** package, Croissant and Millo 2008).

Another approach, widely used in econometrics and the social sciences, is to assume that the model's score function was correctly specified but that only the remaining likelihood was potentially misspecified, e.g., due to a lack of independence as in the case of clustered correlations (see White 1994, for a classic textbook, and Freedman 2006, for a criticial review). This approach leaves the parameter estimator unchanged and is known by different labels in different parts of the literature, e.g., quasi-maximum likelihood (QML), independence working model, or pooling model in ML, GEE, or panel econometrics jargon, respectively. In all of these approaches, only the covariance matrix is adjusted in the subsequent inference by using a sandwich estimator, especially in Wald tests and corresponding confidence intervals.

Important special cases of this QML approach combined with sandwich covariances include: (1) independent but heteroscedastic observations necessitating heteroscedasticity-consistent (HC) covariances (see e.g., Long and Ervin 2000), (2) autocorrelated time series of observations requiring heteroscedasticity- and autocorrelation-consistent (HAC) covariances (such as Newey and West 1987; Andrews 1991), and (3) clustered sandwich covariances for clustered or panel data (see e.g., Cameron and Miller 2015).

2.2. Clustered covariance methods

In the statistics literature, the basic sandwich estimator has been introduced first for cross-section data with independent but heteroscedastic observations by Eicker (1963) and Huber (1967) and was later popularized in the econometrics literature by White (1980). Early references for sandwich estimators accounting for correlated but homoscedastic observations are Kloek (1981) and Moulton (1990), assuming a correlation structure equivalent to exchangeable working correlation introduced by Liang and Zeger (1986) in the more general framework of generalized estimating equations. Kiefer (1980) has implemented yet another clustered covariance estimator for panel data, which is robust to clustering but assumes homoscedasticity (see also Baum, Nichols, and Schaffer 2011). A further generalization allowing for cluster correlations and heteroscedasticity is provided by the independence working model in GEE (Liang and Zeger 1986), naturally leading to clustered covariances. In econometrics, Arellano (1987) first proposed clustered covariances for the fixed effects estimator in panel models.

Inference with clustered covariances with one or more cluster dimension(s) is examined in Cameron, Gelbach, and Miller (2011). In a wider context, Cameron and Miller (2015) include methods in the discussion where not only inference is adjusted but also estimation altered. This discussion focuses on the "large G" framework where not only the number of observations n but also the number of clusters G is assumed to approach infinity. An alternative but less frequently-used asymptotic framework is the "fixed G" setup, where the number of clusters is assumed to be fixed (see Bester, Conley, and Hansen 2011; Hansen and Lee 2017). However, as this also requires non-standard (non-normal) inference it cannot be combined with standard tests in the same modular way as implemented in **sandwich** and is not pursued further here.

In a recent paper, Abadie, Athey, Imbens, and Wooldridge (2017) survey state-of-the-art methods for clustered covariances with special focus on treatment effects in economic experiments with possibly multiple cluster dimensions. They emphasize that, in terms of effect on the covariance estimation, clustering in the treatment assignment is more important than

merely clustering in the sampling. However, if a fixed cluster effect is included in the model, treatment heterogeneity across clusters is a requirement for clustered covariances to be necessary.

2.3. R packages for sandwich covariances

Various kinds of sandwich covariances have already been implemented in several R packages, with the linear regression case receiving most attention. But some packages also cover more general models.

The standard R package for sandwich covariance estimators is the **sandwich** package (Zeileis 2004, 2006b), which provides an object-oriented implementation for the building blocks of the sandwich that rely only on a small set of extractor functions (`estfun()` and `bread()`) for fitted model objects. The function `sandwich()` computes a plain sandwich estimate (Eicker 1963; Huber 1967; White 1980) from a fitted model object, defaulting to what is known as HC0 or HC1 in linear regression models. `vcovHC()` is a wrapper to `sandwich()` combined with `meatHC()` and `bread()` to compute general HC covariances ranging from HC0 to HC5 (see Zeileis 2004, and the references therein). `vcovHAC()`, based on `sandwich()` with `meatHAC()` and `bread()`, computes HAC covariance matrix estimates. Further convenience interfaces `kernHAC()` for Andrews' kernel HAC (Andrews 1991) and `NeweyWest()` for Newey-West-style HAC (Newey and West 1987, 1994) are available. However, in versions prior to 2.4.0 of **sandwich** no similarly object-oriented approach to clustered sandwich covariances was available.

Another R package that includes heteroscedasticity-consistent covariance estimators (HC0–HC4), for models produced by `lm()` only, is the **car** package (Fox and Weisberg 2019) in function `hccm()`. Like `vcovHC()` from **sandwich** this is limited to the cross-section case without clustering, though.

Covariance estimators in **car** as well as in **sandwich** are constructed to insert the resulting covariance matrix estimate in different Wald-type inference functions, as `confint()` from **stats** (R Core Team 2018), `coeftest()` and `waldtest()` from **lmtest** (Zeileis and Hothorn 2002), `Anova()`, `Confint()`, `S()`, `linearHypothesis()`, and `deltaMethod()` from **car** (Fox and Weisberg 2019).

2.4. R packages for clustered covariances

The lack of support for clustered sandwich covariances in standard packages like **sandwich** or **car** has led to a number of different implementations scattered over various packages. Typically, these are tied to either objects from `lm()` or dedicated model objects fitting certain (generalized) linear models for clustered or panel data. The list of packages includes: **multiwayvcov** (Graham, Arai, and Hagströmer 2016), **plm** (Croissant and Millo 2008), **geepack** (Halekoh *et al.* 2005), **lfe** (Gaure 2013), **clubSandwich** (Pustejovsky and Tipton 2017), and **clusterSEs** (Esarey and Menger 2019), among others.

In **multiwayvcov**, the implementation was object-oriented, in many aspects building on **sandwich** infrastructure. However, certain details assumed 'lm' or 'glm'-like objects. In **plm** and **lfe** several types of sandwich covariances are available for the packages' `plm` (panel linear models) and `felm` (fixed-effect linear models), respectively. The **geepack** package can estimate independence working models for 'glm'-type models, also supporting clustered covariances for

the resulting 'geeglm' objects. Finally, **clusterSEs** and **clubSandwich** focus on the case of ordinary or weighted least squares regression models.

In a nutshell, there is good coverage of clustered covariances for (generalized) linear regression objects albeit potentially necessitating reestimating a certain model using a different model-fitting function/packages. However, there was no object-oriented implementation for clustered covariances in R, that enabled plugging in different model objects from in principle any class. Therefore, starting from the implementation in **multiwayvcov**, a new and object-oriented implementation was established and integrated in **sandwich**, allowing application to more general models, including zero-inflated, censored, or limited responses.

2.5. **Stata** software for clustered covariances

Base Stata as well as contributed extension modules provide implementations for several types of clustered covariances. One-way clustered covariances are available in base Stata for a wide range of estimators (e.g., `reg`, `ivreg2`, `xtreg`, among others) by adding the `cluster` option. Furthermore, `suest` provides clustered covariances for seemingly unrelated regression models, `xtpcse` supports linear regression with panel-corrected standard errors, and the `svy` suite of commands can be employed in the presence of nested multi-level clustering (Nichols and Schaffer 2007).

The contributed Stata module **avar** (Baum and Schaffer 2013) allows to construct the "meat" matrix of various sandwich covariance estimators, including HAC, one- and two-way clustered, and panel covariances (à la Kiefer 1980 or Driscoll and Kraay 1998). An alternative implementation for the latter is available in the contributed Stata module **xtscc** (Hoechle 2007). While there is a wide variety of clustered covariances for many regression models in Stata, it is not always possible to fine-tune the flavor of covariance, e.g., with respect to the finite sample adjustment (a common source of confusion and differences across implementations, see e.g., Stack Overflow 2014).

3. Methods

To establish the theoretical background of sandwich covariances for clustered as well as panel data the notation of Zeileis (2006b) is adopted. Here, the conceptual building blocks from Zeileis (2006b) are briefly repeated and then carried further for clustered covariances.

3.1. Sandwich covariances

Let (y_i, x_i) for $i = 1, \ldots, n$ be data with some distribution controlled by a parameter vector θ with k dimensions. For a wide range of models the (quasi-)maximum likelihood estimator $\hat{\theta}$ is governed by a central limit theorem (White 1994) so that $\hat{\theta}$ is approximately normally distributed with $\mathcal{N}(\theta, n^{-1}S(\theta))$. Moreover, the covariance matrix is of sandwich type with a meat matrix $M(\theta)$ between two slices of bread $B(\theta)$:

$$S(\theta) = B(\theta) \cdot M(\theta) \cdot B(\theta) \tag{1}$$

$$B(\theta) = \left(\mathsf{E}\left[-\frac{\partial \psi(y, x, \theta)}{\partial \theta} \right] \right)^{-1} \tag{2}$$

$$M(\theta) = \mathsf{VAR}[\psi(y, x, \theta)]. \tag{3}$$

An estimating function

$$\psi(y, x, \theta) \quad = \quad \frac{\partial \Psi(y, x, \theta)}{\partial \theta} \tag{4}$$

is defined as the derivative of an objective function $\Psi(y, x, \theta)$, typically the log-likelihood, with respect to the parameter vector θ. Thus, an empirical estimating (or score) function evaluates an estimating function at the observed data and the estimated parameters such that an $n \times k$ matrix is obtained (Zeileis 2006b):

$$\begin{pmatrix} \psi(y_1, x_1, \hat{\theta})^{\top} \\ \vdots \\ \psi(y_n, x_n, \hat{\theta})^{\top} \end{pmatrix}. \tag{5}$$

The estimate $\hat{B}$ for the bread is based on second derivatives, i.e., the empirical version of the inverse Hessian

$$\hat{B} \quad = \quad \left(\frac{1}{n} \sum_{i=1}^{n} - \left. \frac{\partial \psi(y_i, x_i, \theta)}{\partial \theta} \right|_{\theta=\hat{\theta}} \right)^{-1}, \tag{6}$$

whereas $\hat{M}, \hat{M}_{\mathrm{HAC}}, \hat{M}_{\mathrm{HC}}$ compute outer product, HAC and HC estimators for the meat, respectively,

$$\hat{M} \quad = \quad \frac{1}{n} \sum_{i=1}^{n} \psi(y_i, x_i, \hat{\theta}) \psi(y_i, x_i, \hat{\theta})^{\top} \tag{7}$$

$$\hat{M}_{\mathrm{HAC}} \quad = \quad \frac{1}{n} \sum_{i,j=1}^{n} w_{|i-j|} \, \psi(y_i, x_i, \hat{\theta}) \psi(y_j, x_j, \hat{\theta})^{\top} \tag{8}$$

$$\hat{M}_{\mathrm{HC}} \quad = \quad \frac{1}{n} X^{\top} \begin{pmatrix} \omega(r(y_1, x_1^{\top} \hat{\theta})) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \omega(r(y_n, x_n^{\top} \hat{\theta})) \end{pmatrix} X. \tag{9}$$

The outer product estimator in Equation 7 corresponds to the basic sandwich estimator (Eicker 1963; Huber 1967; White 1980). $w_{|i-j|}$ in Equation 8 is a vector of weights (Zeileis 2004). In Equation 9, functions $\omega(\cdot)$ derive estimates of the variance of the empirical working residuals $r(y_1, x_1^{\top} \hat{\theta}), \ldots, r(y_n, x_n^{\top} \hat{\theta})$ and may also depend on hat values as well as degrees of freedom (Zeileis 2006b). To obtain the HC type estimator in Equation 9 it is necessary that the score can be factorized into empirical working residuals times the regressor vector.

$$\psi(y_i, x_i, \hat{\theta}) = r(y_i, x_i^{\top} \hat{\theta}) \cdot x_i \tag{10}$$

This is, however, only possible in situations where the parameter of the response distribution depends on a single linear predictor (possibly through a link function).

The building blocks for the calculation of the sandwich are provided by the **sandwich** package, where the `sandwich()` function calculates an estimator of the sandwich $S(\theta)$ (see Equation 1) by multiplying estimators for the meat (from Equation 3) between two slices of bread (from Equation 2). A natural idea for an object-oriented implementation of these estimators is to provide common building blocks, namely a simple `bread()` extractor that computes $\hat{B}$ from Equation 6 and an `estfun()` extractor that returns the empirical estimating functions from

Equation 5. On top of these extractors a number of meat estimators can be defined: `meat()` for $\hat{M}$ from Equation 7, `meatHAC()` for $\hat{M}_{\mathrm{HAC}}$ from Equation 8, and `meatHC()` for $\hat{M}_{\mathrm{HC}}$ from Equation 9, respectively. In addition to the `estfun()` method a `model.matrix()` method is needed in `meatHC()` for the decomposition of the scores into empirical working residuals and regressor matrix.

3.2. Clustered covariances

For clustered observations, similar ideas as above can be employed but the data has more structure that needs to be incorporated into the meat estimators. Specifically, for one-way clustering there is not simply an observation i from $1, \ldots, n$ observations but an observation (i, g) from $1, \ldots, n_g$ observations within cluster/group g (with $g = 1, \ldots, G$ and $n = n_1 + \cdots + n_G$). As only the G groups can be assumed to be independent while there might be correlation withing the cluster/group, the empirical estimation function is summed up within each group prior to computing meat estimators. Thus, the core idea of many clustered covariances is to replace Equation 5 with the following equation and then proceeding "as usual" in the computation of meat estimators afterwards:

$$
\begin{pmatrix} \sum_{i=1}^{n_1} \psi(y_{i,1}, x_{i,1}, \hat{\theta})^\top \\ \vdots \\ \sum_{i=1}^{n_G} \psi(y_{i,G}, x_{i,G}, \hat{\theta})^\top \end{pmatrix} = \begin{pmatrix} \psi(y_{1,1}, x_{1,1}, \hat{\theta})^\top + \cdots + \psi(y_{n_1,1}, x_{n_1,1}, \hat{\theta})^\top \\ \vdots \\ \psi(y_{1,G}, x_{1,G}, \hat{\theta})^\top + \cdots + \psi(y_{n_G,G}, x_{n_G,G}, \hat{\theta})^\top \end{pmatrix}. \tag{11}
$$

The basic meat estimator based on the outer product from Equation 7 then becomes:

$$
\hat{M}_{\mathrm{CL}} = \frac{1}{n} \sum_{g=1}^{G} \left(\sum_{i=1}^{n_g} \psi(y_{i,g}, x_{i,g}, \hat{\theta}) \right) \left(\sum_{i=1}^{n_g} \psi(y_{i,g}, x_{i,g}, \hat{\theta}) \right)^\top. \tag{12}
$$

In the case where observation is its own cluster, the clustered $\hat{M}_{\mathrm{CL}}$ corresponds to the basic $\hat{M}$.

The new function `meatCL()` in the **sandwich** package implements this basic trick along with several types of bias correction and the possibility for multi-way instead of one-way clustering.

Types of bias correction

The clustered covariance estimator controls for both heteroscedasticity across as well as within clusters, assuming that in addition to the number of observations n the number of clusters G also approaches infinity (Cameron, Gelbach, and Miller 2008; Cameron and Miller 2015). Although many publications just refer to "the" clustered standard errors, there is a surprisingly wide variation in clustered covariances, particularly due to different flavors of bias corrections. The bias correction factor can be split in two parts, a "cluster bias correction" and an "HC bias correction". The cluster bias correction captures the effect of having just a finite number of clusters G and it is defined as

$$
\frac{G}{G-1}. \tag{13}
$$

The HC bias correction can be applied additionally in a manner similar to the corresponding

cross-section data estimators. HC0 to HC3 bias corrections for cluster g are defined as

$$\text{HC0}: \quad 1 \tag{14}$$

$$\text{HC1}: \quad \frac{n-1}{n-k} \tag{15}$$

$$\text{HC2}: \quad \frac{G-1}{G} \cdot (I_{n_g} - H_{gg})^{-1} \tag{16}$$

$$\text{HC3}: \quad \frac{G-1}{G} \cdot (I_{n_g} - H_{gg})^{-2}, \tag{17}$$

where n is the number of observations and k is the number of estimated parameters, I_{n_g} is an identity matrix of size n_g, H_{gg} is the block from the hat matrix H that pertains to cluster g. In case of a linear model, the hat (or projection) matrix is defined as $H = X(X^\top X)^{-1}X^\top$, with model matrix X. A generalized hat matrix is available for generalized linear models (GLMs, see Equation 12.3 in McCullagh and Nelder 1989). To apply these factors to $\hat{M}_{\text{CL}}$ (Equation 12), their square root has to be applied to each of the ψ factors in the outer product (see Cameron and Miller 2015, Section VI.B for the linear regression case).

Thus, it is completely straightforward to incorporate the scalar factors for HC0 and HC1. However, the cluster generalizations of HC2 and HC3 (due to Kauermann and Carroll 2001; Bell and McCaffrey 2002) require some more work. More precisely, the square root of the correction factors from Equations 16 and 17 has to be applied to the (working) residuals prior to computing the clustered meat matrix. Thus, the empirical working residuals $r(y_g, x_g^\top \hat{\theta})$ in group g are adjusted via

$$\tilde{r}(y_g, x_g^\top \hat{\theta}) = \sqrt{\frac{G-1}{G}} \cdot (I_{n_g} - H_{gg})^{-\alpha/2} \cdot r(y_g, x_g^\top \hat{\theta}) \tag{18}$$

with $\alpha = 1$ for HC2 and $\alpha = 2$ for HC3, before obtaining the adjusted empirical estimating functions based on Equation 10 as

$$\tilde{\psi}(y_i, x_i, \hat{\theta}) = \tilde{r}(y_i, x_i^\top \hat{\theta}) \cdot x_i. \tag{19}$$

Then these adjusted estimating functions can be employed "as usual" to obtain the $\hat{M}_{\text{CL}}$. Note that, by default, adding the cluster adjustment factor $G/(G-1)$ from Equation 13 simply cancels out the factor $(G-1)/G$ from the HC2/HC3 adjustment factor. Originally, Bell and McCaffrey (2002) recommended to use the factor $(G-1)/G$ for HC3 (i.e., without cluster adjustment) but not for HC2 (i.e., with cluster adjustment). Here, we handle both cases in the same way so that both collapse to the traditional HC2/HC3 estimators when $G = n$ (i.e., where every observation is its own cluster).

Note that in terms of methods in R, it is not sufficient for the cluster HC2/HC3 estimators to have just `estfun()` and `model.matrix()` extractors but an extractor for (blocks of) the full hat matrix are required as well. Currently, no such extractor method is available in base R (as `hatvalues()` just extracts diag$H$) and hence HC2 and HC3 in `meatCL()` are just available for 'lm' and 'glm' objects.

Two-way and multi-way clustered covariances

Certainly, there can be more than one cluster dimension, as for example observations that are characterized by housholds within states or companies within industries. Therefore, it can

sometimes be helpful that one-way clustered covariances can be extended to so-called multi-way clustering as shown by Miglioretti and Heagerty (2007), Thompson (2011) and Cameron *et al.* (2011).

Multi-way clustered covariances comprise clustering on $2^D - 1$ dimensional combinations. Clustering in two dimensions, for example in *id* and *time*, gives $D = 2$, such that the clustered covariance matrix is composed of $2^2 - 1 = 3$ one-way clustered covariance matrices that have to be added or subtracted, respectively. For two-way clustered covariances with cluster dimensions *id* and *time*, the one-way clustered covariance matrices on *id* and on *time* are added, and the two-way clustered covariance matrix with clusters formed by the intersection of *id* and *time* is subtracted. Additionally, a cluster bias correction $\frac{G(\cdot)}{G(\cdot)-1}$ is added, where $G(id)$ is the number of clusters in cluster dimension *id*, $G(time)$ is the number of clusters in cluster dimension *time*, and $G(id \cap time)$ is the number of clusters formed by the intersection of *id* and *time*.

$$\hat{M}_{\mathrm{CL}} \quad = \quad \frac{G(id)}{G(id)-1}\hat{M}_{id} + \frac{G(time)}{G(time)-1}\hat{M}_{time} - \frac{G(id \cap time)}{G(id \cap time)-1}\hat{M}_{id \cap time}. \tag{20}$$

The same idea is used for obtaining clustered covariances with more than two clustering dimensions: Meat parts with an odd number of cluster dimensions are added, whereas those with an even number are subtracted. Particularly when there is only one observation in each intersection of *id* and *time*, Petersen (2009), Thompson (2011) and Ma (2014) suggest to subtract the standard sandwich estimator $\hat{M}$ from Equation (7) instead of $\frac{G(id \cap time)}{G(id \cap time)-1}\hat{M}_{(id \cap time)}$ in Equation (20).

3.3. Clustered covariances for panel data

The information in panel data sets is often overstated, as cross-sectional as well as temporal dependencies may occur (Hoechle 2007). Cameron and Trivedi (2005, p. 702) noticed that "NT correlated observations have less information than NT independent observations". For panel data, the source of dependence in the data is crucial to find out what kind of covariance is optimal (Petersen 2009). In the following, panel Newey-West standard errors as well as Driscoll and Kraay standard errors are examined (see also Millo 2017, for a unifying view).

To reflect that the data are now panel data with a natural time ordering within each *group g* (as opposed to variables without a natural ordering, such as individuals, countries, or firms, Millo 2017), we change our notation to an index (g, t) for $g = 1, \ldots, G$ *groups* sampled at $t = 1, \ldots, n_g$ points in *time* (with $n = n_1 + \cdots + n_G$).

Clustered covariances account for the average within-cluster covariance, where a cluster or group is any set of observations which can be identified by a variable to "cluster on". In the case of panel data, observations are frequently grouped by time and by one or more cross-sectional variables such as individuals or countries that have been observed repeatedly over time.

Panel Newey-West

Newey and West (1987) proposed a heteroscedasticity and autocorrelation consistent standard error estimator that is traditionally used for time-series data, but can be modified for use in panel data (see for example Petersen 2009). A panel Newey-West estimator can be obtained by

setting the cross-sectional as well as the cross-serial correlation to zero (Millo 2017), effectively assuming that each cross-sectional group is a separate, autocorrelated time-series. The meat is composed of

$$\hat{M}_{\mathrm{PL}}^{NW} \quad = \quad \frac{1}{n} \sum_{\ell=0}^{L} w_\ell \left[\sum_{t=1}^{n_{\max}} \sum_{g=1}^{G} \psi(y_{g,t}, x_{g,t}, \hat{\theta}) \psi(y_{g,t-\ell}, x_{g,t-\ell}, \hat{\theta})^\top \right]. \tag{21}$$

Newey and West (1987) employ a Bartlett kernel for obtaining the weights as $w_\ell = 1 - \frac{\ell}{L+1}$ at lag ℓ up to lag L. As Petersen (2009) noticed, the maximal lag length L in a panel data set is $n_{\max} - 1$, i.e., one less than the maximum number of *time* periods per *group* $n_{\max} = \max_g n_g$.

Driscoll and Kraay

Driscoll and Kraay (1998) have adapted the Newey-West approach by using the aggregated estimating functions at each time point. This can be shown to be robust to spatial and temporal dependence of general form, but with the caveat that a long enough time dimension must be available.

Thus, the idea is again to replace Equation 5 by Equation 11 before computing $\hat{M}_{\mathrm{HAC}}$ from Equation 8. Note, however, that the aggregation is now done across *groups g* within each time period t. This yields a panel sandwich estimator where the meat is computed as

$$\hat{M}_{\mathrm{PL}} \quad = \quad \frac{1}{n} \sum_{\ell=0}^{L} w_\ell \left[\sum_{t=1}^{n_{\max}} \left(\sum_{g=1}^{G} \psi(y_{g,t}, x_{g,t}, \hat{\theta}) \right) \left(\sum_{g=1}^{G} \psi(y_{g,t-\ell}, x_{g,t-\ell}, \hat{\theta}) \right)^\top \right], \tag{22}$$

The weights w_ℓ are usually again the Bartlett weights up to lag L. Note that for $L = 0$, $\hat{M}_{\mathrm{PL}}$ reduced to $\hat{M}_{\mathrm{CL}(time)}$, i.e., the one-way covariance clustered by time. Also, for the special case that there is just one observation at each time point t, this panel covariance by Driscoll and Kraay (1998) simply yields the panel Newey-West covariance.

The new function `meatPL()` in the **sandwich** package implements this approach analogously to `meatCL()`. For the computation of the weights w_ℓ the same function is employed that `meatHAC()` uses.

3.4. Panel-corrected standard errors

Beck and Katz (1995) proposed another form or panel-corrected covariances – typically referred to as panel-corrected standard errors (PCSE). They are intended for panel data (also called time-series-cross-section data in this literature) with moderate dimensions of time and cross-section (Millo 2017). They are robust against panel heteroscedasticity and contemporaneous correlation, with the crucial assumption that contemporaneous correlation across clusters follows a fixed pattern (Millo 2017; Johnson 2004). Autocorrelation within a cluster is assumed to be absent.

Hoechle (2007) argues that for the PCSE estimator the finite sample properties are rather poor if the cross-sectional dimension is large compared to the time dimension. This is in contrast to the panel covariance by Driscoll and Kraay (1998) which relies on large-t asymptotics and is robust to quite general forms of cross-sectional and temporal dependence and is consistent independently of the cross-sectional dimension.

To emphasize again that both cross section *and* and time ordering are considered, index (g, t) is employed for the observation from group $g = 1, \ldots, G$ at time $t = 1, \ldots, n_g$. Here, n_g denotes the last time period observed in cluster g, thus allowing for unbalanced data. In the balanced case (that we focus on below) $n_g = T$ for all groups g so that there are $n = G \cdot T$ observations overall.

The basic idea for PCSE is to employ the outer product of (working) residuals within each cluster g. Thus, the working residuals are split into $T \times 1$ vectors for each cluster g: $r(y_1, x_1^\top \hat{\theta}), \ldots, r(y_G, x_G^\top \hat{\theta})$. For balanced data these can be arranged in a $T \times G$ matrix,

$$R \quad = \quad [r(y_1, x_1^\top \hat{\theta}) \quad r(y_2, x_2^\top \hat{\theta}) \quad \ldots \quad r(y_G, x_G^\top \hat{\theta})], \tag{23}$$

and the meat of the panel-corrected covariance matrix can be computed using the Kronecker product as

$$\hat{M}_{\mathrm{PC}} \quad = \quad \frac{1}{n} X^\top \left[\frac{(R^\top R)}{T} \otimes \boldsymbol{I}_T \right] X. \tag{24}$$

The details for the unbalanced case are omitted here for brevity but are discussed in detail in Bailey and Katz (2011).

The new function `meatPC()` in the **sandwich** package implements both the balanced and unbalanced case. As for `meatHC()` it is necessary to have a `model.matrix()` extractor in addition to the `estfun()` extractor for splitting up the empirical estimating functions into residuals and regressor matrix.

4. Software

As conveyed already in Section 3, the **sandwich** package has been extended along the same lines it was originally established in Zeileis (2006b). The new clustered and panel covariances require a new `meat*()` function that ideally only extracts the `estfun()` from a fitted model object. For the full sandwich covariance an accompanying `vcov*()` function is provided that couples the `meat*()` with the `bread()` estimate extracted from the model object.

The new sandwich covariances `vcovCL()` for clustered data and `vcovPL()` for panel data, as well as `vcovPC()` for panel-corrected covariances all follow this structure and are introduced in more detail below.

Model classes which provide the necessary building blocks include 'lm', 'glm', 'mlm', and 'nls' from **stats** (R Core Team 2018), 'betareg' from **betareg** (Cribari-Neto and Zeileis 2010; Grün *et al.* 2012), 'clm' from **ordinal** (Christensen 2019), 'coxph' and 'survreg' from **survival** (Therneau 2020), 'crch' from **crch** (Messner, Mayr, and Zeileis 2016), 'hurdle' and 'zeroinfl' from **pscl**/**countreg** (Zeileis *et al.* 2008), 'mlogit' from **mlogit** (Croissant 2020), and 'polr' and 'rlm' from **MASS** (Venables and Ripley 2002). For all of these an `estfun()` method is available along with a `bread()` method (or the default method works). In case the models are based on a single linear predictor only, they also provide `model.matrix()` extractors so that the factorization from Equation 10 into working residuals and regressor matrix can be easily computed.

4.1. Clustered covariances

One-, two-, and multi-way clustered covariances with HC0–HC3 bias correction are implemented in

```
vcovCL(x, cluster = NULL, type = NULL, sandwich = TRUE, fix = FALSE, ...)
```

for a fitted-model object x with the underlying meat estimator in

```
meatCL(x, cluster = NULL, type = NULL, cadjust = TRUE, multi0 = FALSE, ...)
```

The essential idea is to aggregate the empirical estimating functions within each cluster and then compute a HC covariance analogous to `vcovHC()`.

The `cluster` argument allows to supply either one cluster vector or a list (or data frame) of several cluster variables. If no cluster variable is supplied, each observation is its own cluster per default. Thus, by default, the clustered covariance estimator collapses to the basic sandwich estimator. The `cluster` specification may also be a `formula` if `expand.model.frame(x, cluster)` works for the model object x.

The bias correction is composed of two parts that can be switched on and off separately: First, the cluster bias correction from Equation 13 is controlled by `cadjust`. Second, the HC bias correction from Equations 14–17 is specified via `type` with the default to use `"HC1"` for 'lm' objects and `"HC0"` otherwise. Moreover, `type = "HC2"` and `"HC3"` are only available for 'lm' and 'glm' objects as they require computation of full blocks of the hat matrix (rather than just the diagonal elements as in `hatvalues()`). Hence, the hat matrices of (generalized) linear models are provided directly in `meatCL()` and are not object-oriented in the current implementation.

The `multi0` argument is relevant only for multi-way clustered covariances with more than one clustering dimension. It specifies whether to subtract the basic cross-section HC0 covariance matrix as the last subtracted matrix in Equation 20 instead of the covariance matrix formed by the intersection of groups (Petersen 2009; Thompson 2011; Ma 2014).

For consistency with Zeileis (2004), the `sandwich` argument specifies whether the full sandwich estimator is computed (default) or only the meat.

Finally, the `fix` argument specifies whether the covariance matrix should be fixed to be positive semi-definite in case it is not. This is achieved by converting any negative eigenvalues from the eigendecomposition to zero. Cameron *et al.* (2011) observe that this is most likely to be necessary in applications with fixed effects, especially when clustering is done over the same groups as the fixed effects.

4.2. Clustered covariances for panel data

For panel data,

```
vcovPL(x, cluster = NULL, order.by = NULL, kernel = "Bartlett",
  sandwich = TRUE, fix = FALSE, ...)
```

based on

```
meatPL(x, cluster = NULL, order.by = NULL, kernel = "Bartlett",
  lag = "NW1987", bw = NULL, adjust = TRUE, ...)
```

computes sandwich covariances for panel data, specifically including panel Newey and West (1987) and Driscoll and Kraay (1998). The essential idea is to aggregate the empirical estimating functions within each time period and then compute a HAC covariance analogous to `vcovHAC()`.

Again, `vcovPL()` returns the full sandwich if the argument `sandwich = TRUE`, and `fix = TRUE` forces a positive semi-definite result if necessary.

The `cluster` argument allows the variable indicating the cluster/group/id variable to be specified, while `order.by` specifies the time variable. If only one of the two variables is provided, then it is assumed that observations are ordered within the other variable. And if neither is provided, only one cluster is used for all observations resulting in the standard (Newey and West 1987) estimator. Finally, `cluster` can also be a list or formula with both variables: the cluster/group/id and the time/ordering variable, respectively. (The formula specification requires again that `expand.model.frame(x, cluster)` works for the model object `x`.)

The weights in the panel sandwich covariance are set up by means of a `kernel` function along with a bandwidth `bw` or the corresponding `lag`. All kernels described in Andrews (1991) and implemented in `vcovHAC()` by Zeileis (2006a) are available, namely truncated, Bartlett, Parzen, Tukey-Hanning, and quadratic spectral. For the default case of the Bartlett kernel, the bandwidth `bw` corresponds to `lag + 1` and only one of the two arguments should be specified. The `lag` argument can either be an integer or one of three character specifications: `"max"`, `"NW1987"`, or `"NW1994"`. `"max"` (or equivalently, `"P2009"` for Petersen 2009) indicates the maximum lag length $T - 1$, i.e., the number of time periods minus one. `"NW1987"` corresponds to Newey and West (1987), who have shown that their estimator is consistent if the number of lags increases with time periods T, but with speed less than $T^{1/4}$ (see also Hoechle 2007). `"NW1994"` sets the lag length to floor$[4 \cdot (\frac{T}{100})^{2/9}]$ (Newey and West 1994).

The `adjust` argument allows to make a finite sample adjustment, which amounts to multiplication by $n/(n-k)$, where n is the number of observations, and k is the number of estimated parameters.

4.3. Panel-corrected covariance

Panel-corrected covariances and panel-corrected standard errors (PCSE) a la Beck and Katz (1995) are implemented in

```
vcovPC(x, cluster = NULL, order.by = NULL, pairwise = FALSE,
  sandwich = TRUE, fix = FALSE, ...)
```

based on

```
meatPC(x,  cluster = NULL, order.by = NULL, pairwise = FALSE,
  kronecker = FALSE, ...)
```

They are usually used for panel data or time-series-cross-section (TSCS) data with a large-enough time dimension. The arguments `sandwich`, `fix`, `cluster`, and `order.by` have the same meaning as in `vcovCL()` and `vcovPL()`.

While estimation in balanced panels is straightforward, there are two alternatives to estimate the meat for unbalanced panels (Bailey and Katz 2011). For `pairwise = TRUE`, a pairwise balanced sample is employed, whereas for `pairwise = FALSE`, the largest balanced subset of the panel is used. For details, see Bailey and Katz (2011).

The argument `kronecker` relates to estimation of the meat and determines whether calculations are executed with the Kronecker product or element-wise. The former is typically

computationally faster in moderately large data sets while the latter is less memory-intensive so that it can be applied to larger numbers of observations.

4.4. Further functionality: Bootstrap covariances

As an alternative to the asymptotic approaches to clustered covariances in `vcovCL()`, `vcovPL()`, and `vcovPC()`, the function `vcovBS()` provides a bootstrap solution. Currently, a default method is available as well as a dedicated method for 'lm' objects with more refined boot-strapping options. Both methods take the arguments

```
vcovBS(x, cluster = NULL, R = 250, ..., fix = FALSE,
  use = "pairwise.complete.obs", applyfun = NULL, cores = NULL)
```

where `cluster` and `fix` work the same as in `vcovCL()`. `R` is the number of bootstrap repli-cations and `use` is passed on to the base `cov()` function. The `applyfun` is an optional `lapply()`-style function with arguments `function(X, FUN, ...)`. It is used for refitting the model to the bootstrap samples. The default is to use the basic `lapply()` function unless the `cores` argument is specified so that `parLapply()` (on Windows) or `mclapply()` (otherwise) from the basic **parallel** package are used with the desired number of `cores`.

The default method samples clusters (where each observation is its own cluster by default), i.e., case-based "xy" (or "pairs") resampling (see e.g., Davison and Hinkley 1997). For ob-taining a covariance matrix estimate it is assumed that an `update(x, subset = ...)` of the model with the resampled subset can be obtained, the `coef()` extracted, and finally the covariance computed with `cov()`. In addition to the arguments listed above, it sends `...` to `update()`, and it provides an argument `start = FALSE`. If set to `TRUE`, the latter leads essentially to `update(x, start = coef(x), subset = ...)`, i.e., necessitates support for a `start` argument in the model's fitting function in addition to the `subset` argument. If available, this may reduce the computational burden of refitting the model.

The `glm` method essentially proceeds in the same way but instead of using `update()` for estimating the coefficients on the "xy" bootstrap sample, `glm.fit()` is used which speeds up computations somewhat.

Because the "xy" method makes the same assumptions as the asymptotic approach above, its results approach the asymptotic results as the number of bootstrap replications `R` becomes large, assuming the same bias adjustments and small sample corrections are applied – see e.g., Efron (1979) or Mammen (1992) for a discussion of asymptotic convergence in bootstraps. Bootstrapping will often converge to slightly different estimates when the sample size is small due to a limited number of distinct iterations – for example, there are 126 distinct ways to resample 5 groups with replacement, but many of them occur only rarely (such as drawing group 1 five times); see Webb (2014) for more details.

This makes "xy" bootstraps unnecessary if the desired asymptotic estimator is available and performs well (`vcovCL` may not be available if a software package has not implemented the `estfun()` function, for example). Bootstrapping also often improves inference when nonlinear models are applied to smaller samples. However, the "xy" bootstrap is not the only technique available, and the literature has found a number of alternative bootstrapping approaches – which make somewhat different assumptions – to be useful in practice, even for linear models applied to large samples. Hence, the `vcovBS()` method for 'lm' objects provides several additional bootstrapping `type`s and computes the bootstrapped coefficient estimates

somewhat more efficiently using `lm.fit()` or `qr.coef()` rather than `update()`. The default `type`, however, is also pair resampling (`type = "xy"`) as in the default method. Alternative `type` specifications are

- `"residual"`. The residual cluster bootstrap resamples the residuals (as above, by cluster) which are subsequently added to the fitted values to obtain the bootstrapped response variable. Coefficients can then be estimated using `qr.coef()`, reusing the QR decomposition from the original fit. As Cameron *et al.* (2008) point out, the residual cluster bootstrap is not well-defined when the clusters are unbalanced as residuals from one cluster cannot be easily assigned to another cluster with different size. Hence a warning is issued in that case.

- `"wild"` (or equivalently `"wild-rademacher"` or `"rademacher"`). The wild cluster bootstrap does not actually resample the residuals but instead reforms the dependent variable by multiplying the residual by a randomly drawn value and adding the result to the fitted value (see Cameron *et al.* 2008). By default, the factors are drawn from Rademacher distribution, i.e., selecting either -1 or 1 with probability 0.5.

- `"mammen"` (or `"wild-mammen"`). This draws the wild bootstrap factors as suggested by Mammen (1993).

- `"webb"` (or `"wild-webb"`). This implements the six-point distribution suggested by Webb (2014), which may improve inference when the number of clusters is small.

- `"norm"` (or `"wild-norm"`). The standard normal/Gaussian distribution is used for drawing the wild bootstrap factors.

- User-defined function. This should take a single argument `n`, the number of random values to produce, and return a vector of random factors of corresponding length.

5. Illustrations

The main motivation for the new object-oriented implementation of clustered covariances in **sandwich** was the applicability to models beyond `lm()` or `glm()`. Specifically when working on Berger, Stocker, and Zeileis (2017) – an extended replication of Aghion *et al.* (2013) – clustered covariances for negative binomial hurdle models were needed to confirm reproducibility. After doing this "by hand" in Berger *et al.* (2017), we show in Section 5.1 how the same results can now be conveniently obtained with the new general `vcovCL()` framework.

Furthermore, to show that the new **sandwich** package can also replicate the classic linear regression results that are currently scattered over various packages, the benchmark data from Petersen (2009) is considered in Section 5.2. This focuses on linear regression with model errors that are correlated within clusters. Section 5.2 replicates a variety of clustered covariances from the R packages **multiwayvcov**, **plm**, **geepack**, and **pcse**. More specifically, one- and two-way clustered standard errors from **multiwayvcov** are replicated with `vcovCL()`. Furthermore, one-way clustered standard errors from **plm** and **geepack** can also be obtained by `vcovCL()`. The Driscoll and Kraay standard errors from **plm**'s `vcovSCC()` can also be computed with the new `vcovPL()`. Finally, panel-corrected standard errors can be estimated by function `vcovPC()` from **pcse** and are benchmarked against the new `vcovPC()` from **sandwich**.

5.1. Aghion *et al.* (2013) and Berger *et al.* (2017)

Aghion *et al.* (2013) investigate the effect of institutional owners (these are, for example, pension funds, insurance companies, etc.) on innovation. The authors use firm-level panel data on innovation and institutional ownership from 1991 to 1999 over 803 firms, with the data clustered at company as well as industry level. To capture the differing value of patents, citation-weighted patent counts are used as a proxy for innovation, whereby the authors weight the patents by the number of future citations. This motivates the use of count data models.

Aghion *et al.* (2013) mostly employ Poisson and negative binomial models in a quasi-maximum likelihood approach and cluster standard errors by either companies or industries. Berger *et al.* (2017) argue that zero responses should be treated separately both for statistical and economic reasons, as there is a difference in determinants of "first innovation" and "continuing innovation". Therefore, they employ two-part hurdle models with a binary part that models the decision to innovate at all, and a count part that models ongoing innovation, respectively.

A basic negative binomial hurdle model is fitted with the `hurdle` function from the **pscl** package (Zeileis *et al.* 2008) using the Aghion *et al.* (2013) data provided in the **sandwich** package.

```
R> data("InstInnovation", package = "sandwich")
R> library("pscl")
R> h_innov <- hurdle(
+    cites ~ institutions + log(capital/employment) + log(sales),
+    data = InstInnovation, dist = "negbin")
```

The partial Wald tests for all coefficients based on clustered standard errors can be obtained by using `coeftest()` from **lmtest** (Zeileis and Hothorn 2002) and setting `vcov = vcovCL` and providing the company-level clustering (with a total of 803 clusters) via `cluster = ~ company`.

```
R> library("sandwich")
R> library("lmtest")
R> coeftest(h_innov, vcov = vcovCL, cluster = ~ company)


t test of coefficients:

                             Estimate Std. Error t value Pr(>|t|)
count_(Intercept)             0.49025    0.89445    0.55   0.5836
count_institutions            0.00397    0.00430    0.92   0.3567
count_log(capital/employment) 0.30431    0.15214    2.00   0.0455 *
count_log(sales)              0.38434    0.04999    7.69  1.7e-14 ***
zero_(Intercept)             -0.01007    0.26908   -0.04   0.9701
zero_institutions             0.00658    0.00209    3.15   0.0016 **
zero_log(capital/employment) -0.17431    0.06481   -2.69   0.0072 **
zero_log(sales)               0.17601    0.02776    6.34  2.5e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that institutional owners are have a small but positive impact in both submodels but that only the coefficient in the zero hurdle is significant.

Below, the need for clustering is brought out through a comparison of clustered standard errors with "standard" standard errors and basic (cross-section) sandwich standard errors. As an additional reference, a simple clustered bootstrap covariance can be computed by `vcovBS()` (by default with `R = 250` bootstrap samples).

```
R> suppressWarnings(RNGversion("3.5.0"))
R> set.seed(0)
R> vc <- list(
+     "standard" = vcov(h_innov),
+     "basic" = sandwich(h_innov),
+     "CL-1" = vcovCL(h_innov, cluster = ~ company),
+     "boot" = vcovBS(h_innov, cluster = ~ company)
+ )
R> se <- function(vcov) sapply(vcov, function(x) sqrt(diag(x)))
R> se(vc)
```

	standard	basic	CL-1	boot
count_(Intercept)	0.2249894	0.6034277	0.8944541	0.8667209
count_institutions	0.0016022	0.0024664	0.0043029	0.0041135
count_log(capital/employment)	0.0546614	0.0817540	0.1521416	0.1494889
count_log(sales)	0.0126165	0.0322071	0.0499917	0.0612687
zero_(Intercept)	0.1460246	0.1530406	0.2690844	0.2791141
zero_institutions	0.0013508	0.0013473	0.0020870	0.0020777
zero_log(capital/employment)	0.0333649	0.0358768	0.0648053	0.0666365
zero_log(sales)	0.0164233	0.0165897	0.0277648	0.0295080

This clearly shows that the usual standard errors greatly overstate the precision of the estimators and the basic sandwich covariances are able to improve but not fully remedy the situation. The clustered standard errors are scaled up by factors between about 1.5 and 2, even compared to the basic sandwich standard errors. Moreover, the clustered and bootstrap covariances agree very well – thus highlighting the need for clustering – with `vcovBS()` being computationally much more demanding than `vcovCL()` due to the resampling.

5.2. Petersen (2009)

Petersen (2009) provides simulated benchmark data (http://www.kellogg.northwestern. edu/faculty/petersen/htm/papers/se/test_data.txt) for assessing clustered standard error estimates in the linear regression model. This data set contains a dependent variable y and regressor x for 500 `firm`s over 10 `year`s. It is frequently used in illustrations of clustered covariances (e.g., in **multiwayvcov**, see Graham *et al.* 2016) and is also available in **sandwich**. The corresponding linear model is fitted with `lm()`.

```
R> data("PetersenCL", package = "sandwich")
R> p_lm <- lm(y ~ x, data = PetersenCL)
```

One-way clustered standard errors

One-way clustered covariances for linear regression models are available in a number of different R packages. The implementations differ mainly in two aspects: (1) Whether a simple 'lm' object can be supplied or (re-)estimation of a dedicated model object is necessary. (2) Which kind of bias adjustment is done by default (HC type and/or cluster adjustment).

The function `cluster.vcov()` from **multiwayvcov** (whose implementation strategy `vcovCL()` follows) can use 'lm' objects directly and then applies both the HC1 and cluster adjustment by default. In contrast, **plm** and **geepack** both require re-estimation of the model and then employ HC0 without cluster adjustment by default. In **plm**, a pooling model needs to be estimated with `plm()` and in **geepack** an independence working model needs to be fitted with `geeglm()`.

The **multiwayvcov** results can be replicated as follows.

```
R> library("multiwayvcov")
R> se(list(
+    "sandwich" = vcovCL(p_lm, cluster = ~ firm),
+    "multiwayvcov" = cluster.vcov(p_lm, cluster = ~ firm)
+ ))

            sandwich multiwayvcov
(Intercept) 0.067013     0.067013
x           0.050596     0.050596
```

And the **plm** and **geepack** results can be replicated with the following code. (Note that **geepack** does not provide a `vcov()` method for 'geeglm' objects, hence the necessary code is included below.)

```
R> library("plm")
R> p_plm <- plm(y ~ x, data = PetersenCL, model = "pooling",
+    indexes = c("firm", "year"))
R> library("geepack")
R> vcov.geeglm <- function(object) {
+    vc <- object$geese$vbeta
+    rownames(vc) <- colnames(vc) <- names(coef(object))
+    return(vc)
+ }
R> p_gee <- geeglm(y ~ x, data = PetersenCL, id = PetersenCL$firm,
+    corstr = "independence", family = gaussian)
R> se(list(
+    "sandwich" = vcovCL(p_lm, cluster = ~ firm,
+      cadjust = FALSE, type = "HC0"),
+    "plm" = vcovHC(p_plm, cluster = "group"),
+    "geepack" = vcov(p_gee)
+ ))

            sandwich      plm  geepack
(Intercept) 0.066939 0.066939 0.066939
x           0.050540 0.050540 0.050540
```

Two-way clustered standard errors

It would also be feasible to cluster the covariances with respect to both dimensions, `firm` and `year`, yielding similar but slightly larger standard errors. Again, vcovCL() from **sandwich** can replicate the results of cluster.vcov() from **multiwayvcov**. Only the default for the correction proposed by Ma (2014) is different.

```
R> se(list(
+    "sandwich" = vcovCL(p_lm, cluster = ~ firm + year, multi0 = TRUE),
+    "multiwayvcov" = cluster.vcov(p_lm, cluster = ~ firm + year)
+  ))


            sandwich multiwayvcov
(Intercept) 0.065066    0.065066
x           0.053561    0.053561
```

However, note that the results should be regarded with caution as cluster dimension `year` has a total of only 10 clusters. Theory requires that each cluster dimension has many clusters (Petersen 2009; Cameron *et al.* 2011; Cameron and Miller 2015).

Driscoll and Kraay standard errors

The Driscoll and Kraay standard errors for panel data are available in `vcovSCC()` from **plm**, defaulting to a HC0-type adjustment. In **sandwich** the `vcovPL()` function can be used for replication, setting `adjust = FALSE` to match the HC0 (rather than HC1) adjustment.

```
R> se(list(
+    "sandwich" = vcovPL(p_lm, cluster = ~ firm + year, adjust = FALSE),
+    "plm" = vcovSCC(p_plm)
+  ))


            sandwich      plm
(Intercept) 0.024357 0.024357
x           0.028163 0.028163
```

Panel-corrected standard errors

Panel-corrected covariances a la Beck and Katz are implemented in the package **pcse** – providing the function that is also named `vcovPC()` – which can handle both balanced and unbalanced panels. For the balanced Petersen data the two `vcovPC()` functions from **sandwich** and **pcse** agree.

```
R> library("pcse")
R> se(list(
+    "sandwich" = sandwich::vcovPC(p_lm, cluster = ~ firm + year),
+    "pcse" = pcse::vcovPC(p_lm, groupN = PetersenCL$firm,
+      groupT = PetersenCL$year)
+  ))
```

```
            sandwich      pcse
(Intercept) 0.022201 0.022201
x           0.025276 0.025276
```

And also when omitting the last year for the first firm to obtain an unbalanced panel, the **pcse** results can be replicated. Both strategies for balancing the panel internally (pairwise vs. casewise) are illustrated in the following.

```
R> PU <- subset(PetersenCL, !(firm == 1 & year == 10))
R> pu_lm <- lm(y ~ x, data = PU)
```

and again, panel-corrected standard errors from **sandwich** are equivalent to those from **pcse**.

```
R> se(list(
+    "sandwichT" = sandwich::vcovPC(pu_lm, cluster = ~ firm + year,
+      pairwise = TRUE),
+    "pcseT" = pcse::vcovPC(pu_lm, PU$firm, PU$year, pairwise = TRUE),
+    "sandwichF" = sandwich::vcovPC(pu_lm, cluster = ~ firm + year,
+      pairwise = FALSE),
+    "pcseF" = pcse::vcovPC(pu_lm, PU$firm, PU$year, pairwise = FALSE)
+  ))
```

```
            sandwichT    pcseT sandwichF    pcseF
(Intercept)  0.022070 0.022070  0.022603 0.022603
x            0.025338 0.025338  0.025241 0.025241
```

6. Simulation

For a more systematic analysis, a Monte Carlo simulation is carried out to assess the performance of clustered covariances beyond linear and generalized linear models. For the linear model, there are a number of simulation studies in the literature (including Cameron *et al.* 2008; Arceneaux and Nickerson 2009; Petersen 2009; Cameron *et al.* 2011; Harden 2011; Thompson 2011; Cameron and Miller 2015; Jin 2015), far fewer for generalized linear models (see for example Miglioretti and Heagerty 2007) and, to our knowledge, no larger systematic comparisons for models beyond. Therefore, we try to fill this gap by starting out with simulations of (generalized) linear models similar to the ones mentioned above and then moving on to other types of maximum likelihood regression models.

6.1. Simulation design

The main focus of the simulation is to assess the performance of clustered covariances (and related methods) at varying degrees of correlation within the clusters. The two most important parameters to control for this are the cluster correlation ρ, obviously, and the number of clusters G as bias decreases with increasing number of clusters (Green and Vavreck 2008; Arceneaux and Nickerson 2009; Harden 2011). More specifically, the cluster correlation varies from 0 to 0.9 and the number of clusters G ranges from 10 to 250.

In a first step, only balanced clusters with a low number of observations per cluster (5) are considered. All models are specified through linear predictors (with up to three regressors) but with different response distributions. A Gaussian copula is employed to introduce the cluster correlation ρ for the different response distributions. The methods considered are the different clustered covariances implemented in **sandwich** as well as competing methods such as basic sandwich covariances (without clustering), mixed-effects models with a random intercept, or generalized estimating equations with an exchangeable correlation structure.

Linear predictor

Following Harden (2011) the linear predictor considered for the simulation is composed of three regressors that are either *correlated* with the clustering, *clustered*, or *uncorrelated*. The correlation of the first type of regressor is controlled by separate parameter ρ_x. Thus, in the terminology of Abadie *et al.* (2017), the parameter ρ above controls the correlation in the sampling process while the parameter ρ_x controls the extent of the clustering in the "treatment" assignment.

More formally, the model equation is given by

$$h(\mu_{i,g}) = \beta_0 + \beta_1 \cdot x_{1,i,g} + \beta_2 \cdot x_{2,g} + \beta_3 \cdot x_{3,i,g}, \tag{25}$$

where $\mu_{i,g}$ is the expectation of the response for observation i within cluster g and the link function $h(\cdot)$ depends on the model type. The regressor variables are all drawn from standard normal distributions but at different levels (cluster vs. individual observation).

$$
\begin{align}
x_{1,i,g} &\sim \rho_x \cdot \mathcal{N}_g(0,1) + (1 - \rho_x) \cdot \mathcal{N}_{i,g}(0,1) \tag{26} \\
x_{2,g} &\sim \mathcal{N}_g(0,1) \tag{27} \\
x_{3,i,g} &\sim \mathcal{N}_{i,g}(0,1) \tag{28}
\end{align}
$$

Regressor $x_{1,i,g}$ is composed of a linear combination of a random draw at cluster level ($\mathcal{N}_g$) and a random draw at individual level ($\mathcal{N}_{i,g}$) while regressors $x_{2,g}$ and $x_{3,i,g}$ are drawn only at cluster and individual level, respectively. Emphasis is given to the investigation of regressor $x_{1,i,g}$ with correlation (default: $\rho_x = 0.25$) which is probably the most common in practice. Furthermore, by considering the extremes $\rho_x = 1$ and $\rho_x = 0$ the properties of $x_{1,i,g}$ coincide with those of $x_{2,g}$ and $x_{3,i,g}$, respectively.

The vector of coefficients $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^\top$ is fixed to either one of

$$
\begin{align}
\beta &= (0, 0.85, 0.5, 0.7)^\top \tag{29} \\
\beta &= (0, 0.85, 0, 0)^\top \tag{30}
\end{align}
$$

which have been selected based on Harden (2011).

Response distributions

The response distributions encompass Gaussian (`gaussian`, with identity link) as the standard classical scenario as well as binary (`binomial`, with a size of one and a logit link) and Poisson (`poisson`, with log link) from the GLM exponential family. To move beyond the GLM, we also consider the beta (`betareg`, with logit link and fixed precision parameter $\phi = 10$), zero-truncated Poisson(`zerotrunc`, with log link), and zero-inflated Poisson (`zeroinfl`, with log link and fixed inflation probability $\pi = 0.3$) distributions.

Label	Model	Object	Variance-covariance matrix
CL-0	(g)lm	m	`vcovCL(m, cluster = id, type = "HC0")`
CL-1	(g)lm	m	`vcovCL(m, cluster = id, type = "HC1")`
CL-2	(g)lm	m	`vcovCL(m, cluster = id, type = "HC2")`
CL-3	(g)lm	m	`vcovCL(m, cluster = id, type = "HC3")`
PL	(g)lm	m	`vcovPL(m, cluster = id, adjust = FALSE)`
PC	(g)lm	m	`vcovPC(m, cluster = id, order.by = round)`
standard	(g)lm	m	`vcov(m)`
basic	(g)lm	m	`sandwich(m)`
random	(g)lmer	m_re	`vcov(m_re)`
gee	geeglm	m_gee	`m_gee$geese$vbeta`

Table 1: Covariance matrices for responses from the exponential family in 'sim-CL.R'.

Sandwich covariances

The types of covariances being compared include "standard" covariances (*standard*, without considering any heteroscedasticity or clustering/correlations), basic sandwich covariances (*basic*, without clustering), Driscoll and Kraay panel covariances (*PL*), Beck and Katz panel-corrected covariances (*PC*), and clustered covariances with HC0–HC3 adjustment (*CL-0–CL-3*). As further references, covariances from a clustered bootstrap (*BS*), a mixed-effects model with random intercept (*random*), and a GEE with exchangeable correlation structure (*gee*) are assessed.

Outcome measure

In order to assess the validity of statistical inference based on clustered covariances, the empirical coverage rate of the 95% Wald confidence intervals (from 10,000 replications) is the outcome measure of interest. If standard errors are estimated accurately, the empirical coverage should match the nominal rate of 0.95. And empirical coverages falling short of 0.95 are typically due to underestimated standard errors and would lead to inflated type I errors in partial Wald tests of the coefficients.

Simulation code

The supplementary R script 'sim-CL.R' comprises the simulation code for the data generating process described above and includes functions `dgp()`, `fit()`, and `sim()`. While `dgp()` specifies the data generating process and generates a data frame with (up to) three regressors `x1`, `x2`, `x3` as well as cluster dimensions `id` and `round`, `fit()` is responsible for the model estimation as well as computation of the covariance matrix estimate and the empirical coverage. The function `sim()` sets up all factorial combinations of the specified scenarios and loops over the fits for each scenario (using multiple cores for parallelization).

Table 1 shows how the different types of covariances are calculated for responses from the exponential family. A pooled or marginal model (`m`), a random effects model (`m_re`, using **lme4**, Bates *et al.* 2015), and a GEE with an exchangeable correlation structure (`m_gee`, using **geepack**, Halekoh *et al.* 2005) are fitted. For the other (non-GLM) responses, the functions `betareg()` (from **betareg**, Cribari-Neto and Zeileis 2010), `zerotrunc()` (from **countreg**, Zeileis and Kleiber 2020), and `zeroinfl()` (from **pscl**/**countreg**, Zeileis *et al.* 2008) are used.
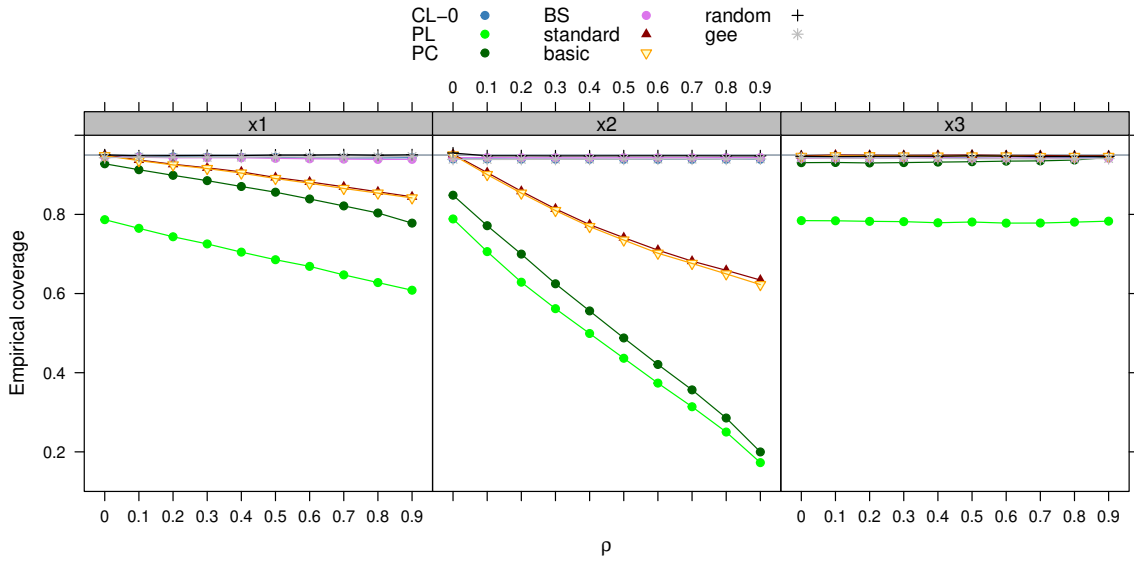
Figure 1: Experiment I. Gaussian response with $G = 100$ (balanced) clusters of 5 observations each. Regressor x1 is correlated ($\rho_x = 0.25$), x2 clustered, and x3 uncorrelated. The coverage (from 10,000 replications) is plotted on the y-axis against the cluster correlation ρ of the response distribution (from a Gaussian copula) on the x-axis. The horizontal reference line indicates the nominal coverage of 0.95.

6.2. Results

Based on the design discussed above, the simulation study investigates the performance of clustered covariances for the following settings.

- Experiment I: Different types of regressors for a Gaussian response distribution.

- Experiment II: Different GLM response distributions.

- Experiment III: Response distributions beyond the GLM.

- Experiment IV: GLMs with HC0–HC3 bias corrections.

Experiment I

Figure 1 shows the results from Experiment I and plots the empirical coverage probabilities (from 10,000 replications) on the y-axis for the coefficients of the correlated regressor x1, the clustered regressor x2, and the uncorrelated regressor x3 against the cluster correlation ρ on the y-axis.

While for the uncorrelated regressor x3 all methods – except the Driscoll and Kraay PL estimator – perform well and yield satisfactory coverage rates, the picture is different for the correlated and clustered regressors x1 and x2. With increasing cluster correlation ρ the performance deteriorates for those methods that either do not account for the clustering at all (i.e., "standard" covariance and basic sandwich covariance) or that treat the data as panel data (i.e., PL and PC).
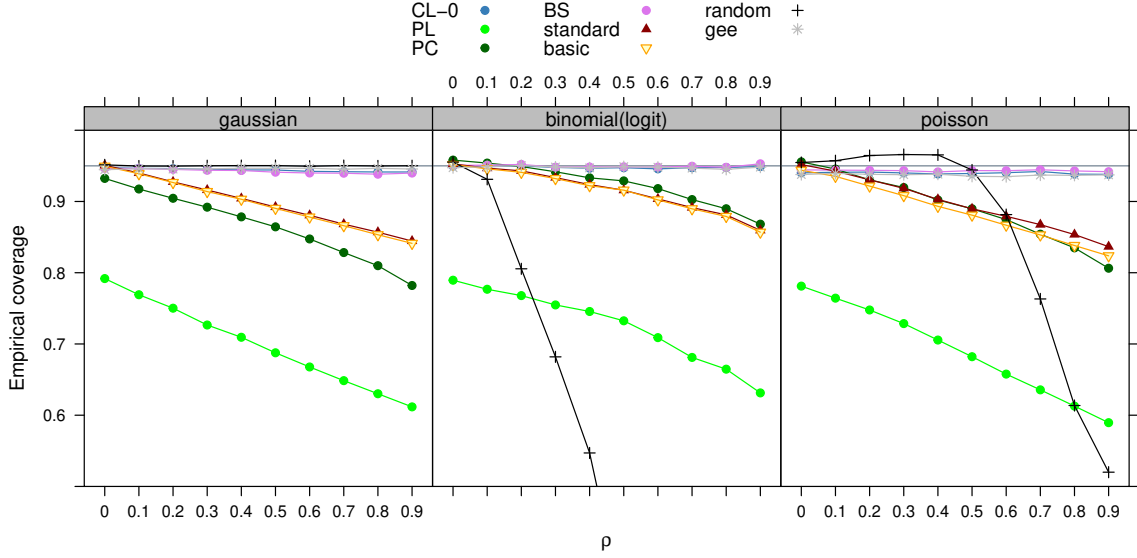
Figure 2: Experiment II. Exponential family response distributions with $G = 100$ (balanced) clusters of 5 observations each. The only regressor x1 is correlated ($\rho_x = 0.25$). The coverage (from 10,000 replications) is plotted on the *y*-axis against the cluster correlation ρ of the response distribution (from a Gaussian copula) on the *x*-axis. The horizontal reference line indicates the nominal coverage of 0.95.

The reason for the poor performance of the panel data covariances is the low number of 5 observations per cluster. This has already been documented in the literature: In a Monte-Carlo study, Driscoll and Kraay (1998) use a minimum of 20–25 observations per cluster and Hoechle (2007) notes that the PC estimator can be quite imprecise if the crosss-sectional dimension is large compared to the time dimension. As shown in Appendix A, the performance improves if an exponentially decaying AR(1) correlation structure is employed instead of the exchangeable structure and if the number of observations per cluster increases.

As the effects of regressor x1 are in between the effects of the clustered regressor x2 and the uncorrelated regressor x3, the following simulation experiments focus on the situation with a single correlated regressor x1.

Experiment II

Figure 2 illustrates the results from Experiment II. The settings are mostly analogous to Experiment I with two important differences: (1) GLMs with Gaussian/binomial/Poisson response distribution are used. (2) There is only one regressor (x1, correlated with $\rho_x = 0.25$).

Overall, the results for binomial and Poisson response are very similar to the Gaussian case in Experiment I. Thus, this confirms that clustered covariances also work well with GLMs.

The only major difference between the linear Gaussian and nonlinear binomial/Poisson cases is the performance of the mixed-effects models with random intercept. While in linear models the marginal (or "population-averaged") approach employed with clustered covariances leads to analogous models compared to mixed-effects models, this is not the case in nonlinear models. With nonlinear links, mixed-effects models correspond to "conditional" rather than
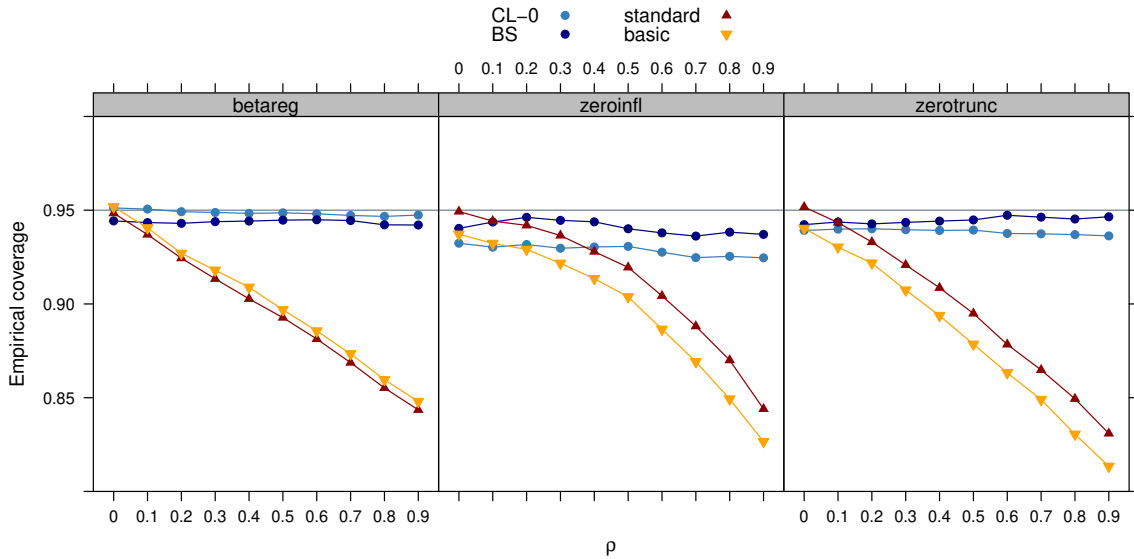
Figure 3: Experiment III. Response distributions beyond the GLM (beta regression, zero-truncated Poisson, and zero-inflated Poisson) with $G = 100$ (balanced) clusters of 5 observations each. The only regressor x1 is correlated ($\rho_x = 0.25$). The coverage (from 10,000 replications for basic/standard/CL-0 and 1,000 replications for BS) is plotted on the y-axis against the cluster correlation ρ of the response distribution (from a Gaussian copula) on the x-axis. The horizontal reference line indicates the nominal coverage of 0.95.

"marginal" models and for obtaining marginal expectations the random effects have to be integrated out (see Molenberghs, Kenward, Verbeke, Iddi, and Efendi 2013; Fitzmaurice 2014). Consequently, fixed effects have to be interpreted differently and their confidence intervals do not contain the population-averaged effects, thus leading to the results in Experiment II.

Experiment III

Figure 3 shows the outcome of Experiment III whose settings are similar to the previous Experiment I. But now more general response distributions beyond the classic GLM framework are employed, revealing that the clustered covariances from vcovCL() indeed also work well in this setup. Again, the performance of the non-clustered covariances deteriorates with increasing cluster correlation. For the zero-truncated and zero-inflated Poisson distribution the empirical coverage rate is slightly lower than 0.95. However, given that this does not depend on the extent of the correlation ρ this is more likely due to the quality of the normal approximation in the Wald confidence interval. The clustered bootstrap covariance (BS) performs similarly to the clustered HC0 covariance but is computationally much more demanding due to the need for resampling and refitting the model (here with R = 250 bootstrap samples).

Experiment IV

Figure 4 depicts the findings of Experiment IV. The y-axis represents again the empirical coverage from 10,000 replications, but in contrast to the other simulation experiments, the number of clusters G is plotted on the x-axis, ranging from 10 to 250 clusters. Gaussian,
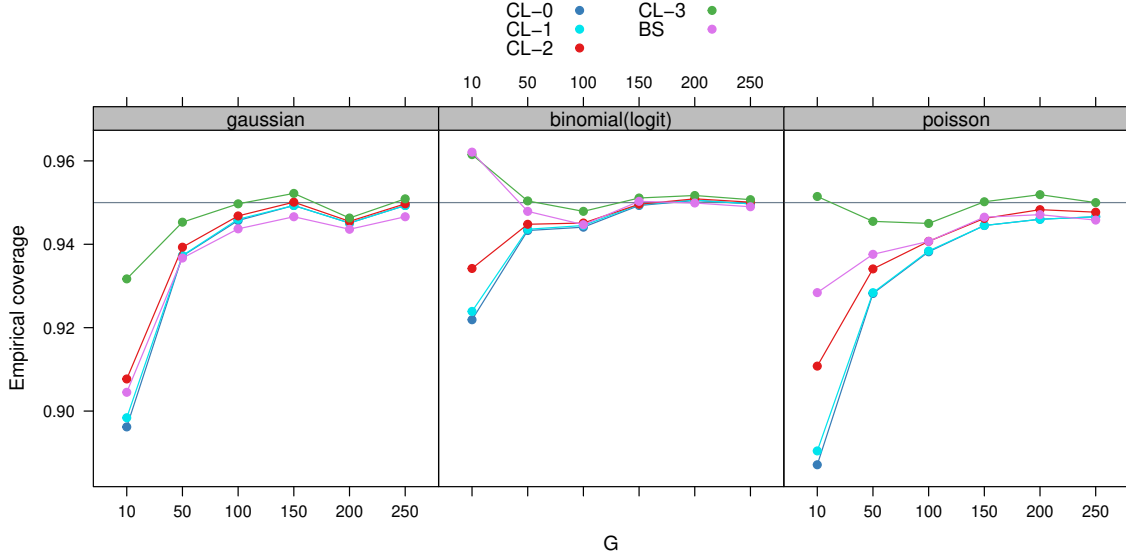
Figure 4: Experiment IV. Exponential family response distributions with cluster correlation $\rho = 0.25$ (from a Gaussian copula). The only regressor x1 is correlated ($\rho_x = 0.25$). The coverage (from 10,000 replications) on the y-axis for different types of bias adjustment (HC0–HC3 and bootstrap) is plotted against the number of clusters G on the x-axis. The number of clusters increases with $G = 10, 50, \ldots, 250$ while the number of observations per cluster is fixed at 5. The horizontal reference line indicates the nominal coverage of 0.95.

binomial and Poisson responses are compared with each other, with the focus on clustered standard errors with HC0–HC3 types of bias correction. (Recall that HC2 and HC3 require block-wise components of the full hat matrix and hence are at the moment only available for lm() and glm() fits, see Section 4.1.) Furthermore, clustered bootstrap standard errors are included for comparison (using R = 250 bootstrap samples).

In most cases, all of the standard errors are underestimated for $G = 10$ clusters (except clustered standard errors with HC3 bias correction for the binomial and Poisson response). As found in previous studies for clustered HC0/HC1 covariances (Arceneaux and Nickerson 2009; Petersen 2009; Harden 2011; Cameron and Miller 2015; Pustejovsky and Tipton 2018, among others), the larger the number of clusters G, the better the coverage and the less standard errors are underestimated. In our study about 50–100 clusters are enough for sufficiently accurate coverage rates using clustered standard errors without a bias correction (CL-0 in the figure), which is consistent with other authors' findings.

Additionally, it can be observed that the higher the number of clusters, the less the different types of HC bias correction differ. However, for a small number of clusters, the HC3 correction works best, followed by HC2, HC1 and HC0. This is also consistent with the findings for cross-sectional data, e.g., Long and Ervin (2000) suggest to use HC3 in the linear model for small samples with fewer than 250 observations. Moreover, bootstrap covariances perform somewhat better than HC0/HC1 for a small number of clusters. However, Pustejovsky and Tipton (2018) have argued that HC3 bias corrections for clustered covariances (CL-3 in the figure, sometimes referred to "CR3" or "CR3VE") tend to over-correct for small G, and

recommend HC2 bias corrections ("CR2", CL-2 in the figure) and t tests using degrees of freedom calculated similar to Satterthwaite (1946). While Experiment IV finds that HC3 adjustments slightly over-correct for $G = 10$ in binomial models, overall, our findings are not consistent with the suggestion that HC3 is overly aggressive in correcting small G bias in clustered standard errors.

7. Summary

While previous versions of the **sandwich** package already provided a flexible object-oriented implementation of covariances for cross-section and time series data, the corresponding functions for clustered and panel data have only been added recently (in version 2.4-0 of the package). Compared to previous implementations in R that were somewhat scattered over several packages, the implementation in **sandwich** offers a wide range of "flavors" of clustered covariances and, most importantly, is applicable to any model object that provides methods to extract the `estfun()` (estimating functions; observed score matrix) and `bread()` (inverse Hessian). Therefore, it is possible to apply the new functions `vcovCL()`, `vcovPL()`, and `vcovPC()` to models beyond linear regression. A thorough Monte Carlo study assesses the performance of these functions in regressions beyond the standard linear Gaussian scenario, e.g., for exponential family distributions and beyond and for the less frequently used HC2 and HC3 adjustments. This shows that clustered covariances work reasonably well in the models investigated but some care is needed when applying panel estimators (`vcovPL()` and `vcovPC()`) in panel data with "short" panels and/or non-decaying autocorrelations.

Computational details

The packages **sandwich**, **boot**, **countreg**, **geepack**, **lattice**, **lme4**, **lmtest**, **multiwayvcov**, **plm** and **pscl** are required for the applications in this paper. For replication of the simulation study, the supplementary R script `sim-CL.R` is provided along with the corresponding results `sim-CL.rda`.

R version 4.0.4 has been used for computations. Package versions that have been employed are **sandwich** 3.0–1, **boot** 1.3–27, **countreg** 0.2-0, **geepack** 1.3–2, **lattice** 0.20–41, **lme4** 1.1–26, **lmtest** 0.9–39, **multiwayvcov** 1.2.3, **plm** 2.4–1, and **pscl** 1.5.5.

R itself and all packages (except **countreg**) used are available from CRAN at https://CRAN.R-project.org/. **countreg** is available from https://R-Forge.R-project.org/projects/countreg/.

Acknowledgments

References

Abadie A, Athey S, Imbens GW, Wooldridge J (2017). "When Should You Adjust Standard Errors for Clustering?" *Technical report*, National Bureau of Economic Research. `doi:10.3386/w24003`.

Aghion P, Van Reenen J, Zingales L (2013). "Innovation and Institutional Ownership." *The American Economic Review*, **103**(1), 277–304. `doi:10.1257/aer.103.1.277`.

Andrews DWK (1991). "Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation." *Econometrica*, **59**, 817–858. `doi:10.2307/2938229`.

Arceneaux K, Nickerson DW (2009). "Modeling Certainty with Clustered Data: A Comparison of Methods." *Political Analysis*, **17**(2), 177–190. `doi:10.1093/pan/mpp004`.

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`.

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

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using **lme4**." *Journal of Statistical Software*, **67**(1), 1–48. `doi:10.18637/jss.v067.i01`.

Baum C, Nichols A, Schaffer M (2011). "Evaluating One-Way and Two-Way Cluster-Robust Covariance Matrix Estimates." In *German Stata Users' Group Meetings, July 2011*. URL `https://www.stata.com/meeting/boston10/boston10_baum.pdf`.

Baum CF, Schaffer ME (2013). "**avar**: Stata Module to Perform Asymptotic Covariance Estimation for IID and Non-IID Data Robust to Heteroskedasticity, Autocorrelation, 1- and 2-Way Clustering, and Common Cross-Panel Autocorrelated Disturbances." Statistical Software Components, Boston College Department of Economics. URL `https://ideas.RePEc.org/c/boc/bocode/s457689.html`.

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

Bell RM, McCaffrey DF (2002). "Bias Reduction in Standard Errors for Linear Regression with Multi-Stage Samples." *Survey Methodology*, **28**(2), 169–181. URL `http://www.statcan.gc.ca/pub/12-001-x/2002002/article/9058-eng.pdf`.

Berger S, Stocker H, Zeileis A (2017). "Innovation and Institutional Ownership Revisited: An Empirical Investigation with Count Data Models." *Empirical Economics*, **52**(4), 1675–1688. `doi:10.1007/s00181-016-1118-0`.

Bester CA, Conley TG, Hansen CB (2011). "Inference with Dependent Data Using Cluster Covariance Estimators." *Journal of Econometrics*, **165**(2), 137–151. `doi:10.1016/j.jeconom.2011.01.007`.

Cameron AC, Gelbach JB, Miller DL (2008). "Bootstrap-Based Improvements for Inference with Clustered Errors." *The Review of Economics and Statistics*, **90**(3), 414–427. `doi:10.1162/rest.90.3.414`.

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`.

Cameron AC, Miller DL (2015). "A Practitioner's Guide to Cluster-Robust Inference." *Journal of Human Resources*, **50**(2), 317–372. `doi:10.3368/jhr.50.2.317`.

Cameron AC, Trivedi PK (2005). *Microeconometrics: Methods and Applications*. Cambridge University Press, Cambridge.

Christensen RHB (2019). **ordinal**: *Regression Models for Ordinal Data*. R package version 2019.12-10, URL `https://CRAN.R-project.org/package=ordinal`.

Cribari-Neto F, Zeileis A (2010). "Beta Regression in R." *Journal of Statistical Software*, **34**(2), 1–24. `doi:10.18637/jss.v034.i02`.

Croissant Y (2020). "Estimation of Random Utility Models in R: The **mlogit** Package." *Journal of Statistical Software*, **95**(11), 1–41. `doi:10.18637/jss.v095.i11`.

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

Davison AC, Hinkley DV (1997). *Bootstrap Methods and Their Applications*. Cambridge University Press, Cambridge.

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

Efron B (1979). "Bootstrap Methods: Another Look at the Jackknife." *The Annals of Statistics*, **7**(1), 1–26. `doi:10.1214/aos/1176344552`.

Eicker F (1963). "Asymptotic Normality and Consistency of the Least Squares Estimator for Families of Linear Regressions." *Annals of Mathematical Statistics*, **34**, 447–456. `doi:10.1214/aoms/1177704156`.

Esarey J, Menger A (2019). "Practical and Effective Approaches to Dealing with Clustered Data." *Political Science Research and Methods*, **7**(3), 541–559. `doi:10.1017/psrm.2017.42`.

Fitzmaurice GM (2014). "Multilevel Modeling of Longitudinal Data." In *Symposium on Recent Advances in Multilevel Modeling*. New York University, New York. URL `https://sites.google.com/a/stern.nyu.edu/symposium-on-recent-advances-in-multilevel-modeling/`.

Fox J, Weisberg S (2019). *An R Companion to Applied Regression*. 3rd edition. Sage Publications, Thousand Oaks.

Freedman DA (2006). "On the So-Called 'Huber Sandwich Estimator' and 'Robust Standard Errors'." *The American Statistician*, **60**(4), 299–302. `doi:10.1198/000313006x152207`.

Galbraith S, Daniel JA, Vissel B (2010). "A Study of Clustered Data and Approaches to Its Analysis." *Journal of Neuroscience*, **30**(32), 10601–10608. `doi:10.1523/jneurosci.0362-10.2010`.

Gaure S (2013). "**lfe**: Linear Group Fixed Effects." *The R Journal*, **5**(2), 104–116. `doi:10.32614/RJ-2013-031`.

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`.

Green DP, Vavreck L (2008). "Analysis of Cluster-Randomized Experiments: A Comparison of Alternative Estimation Approaches." *Political Analysis*, **16**(2), 138–152. `doi:10.1093/pan/mpm025`.

Grün B, Kosmidis I, Zeileis A (2012). "Extended Beta Regression in R: Shaken, Stirred, Mixed, and Partitioned." *Journal of Statistical Software*, **48**(11), 1–25. `doi:10.18637/jss.v048.i11`.

Halekoh U, Højsgaard S, Yan J (2005). "The R Package **geepack** for Generalized Estimating Equations." *Journal of Statistical Software*, **15**(2), 1–11. `doi:10.18637/jss.v015.i02`.

Hansen B, Lee S (2017). "Asymptotic Theory for Clustered Samples." *Working Paper 2017-18*, School of Economics, The University of New South Wales. URL `https://EconPapers.RePEc.org/RePEc:swe:wpaper:2017-18`.

Harden JJ (2011). "A Bootstrap Method for Conducting Statistical Inference with Clustered Data." *State Politics & Policy Quarterly*, **11**(2), 223–246. `doi:10.1177/1532440011406233`.

Hoechle D (2007). "Robust Standard Errors for Panel Regressions with Cross-Sectional Dependence." *Stata Journal*, **7**(3), 281–312. URL `http://www.stata-journal.com/sjpdf.html?articlenum=st0128`.

Huber PJ (1967). "The Behavior of Maximum Likelihood Estimation under Nonstandard Conditions." In LM LeCam, J Neyman (eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, Berkeley.

Jin S (2015). "The Impact of Sampling Procedures on Statistical Inference with Clustered Data." Conference Poster. URL `http://www.sas.rochester.edu/psc/polmeth/posters/Jin.pdf`.

Johnson P (2004). "Cross Sectional Time Series: The Normal Model and Panel Corrected Standard Errors." URL `http://pj.freefaculty.org/guides/stat/Regression/TimeSeries-Longitudinal-CXTS/CXTS-PCSE.pdf`.

Kauermann G, Carroll RJ (2001). "A Note on the Efficiency of Sandwich Covariance Matrix Estimation." *Journal of the American Statistical Association*, **96**(456), 1387–1396. `doi:10.1198/016214501753382309`.

Kiefer NM (1980). "Estimation of Fixed Effect Models for Time Series of Cross-Sections with Arbitrary Intertemporal Covariance." *Journal of Econometrics*, **14**(2), 195–202. `doi:10.1016/0304-4076(80)90090-1`.

Kloek T (1981). "OLS Estimation in a Model Where a Microvariable Is Explained by Aggregates and Contemporaneous Disturbances Are Equicorrelated." *Econometrica*, **49**(1), 205–207. `doi:10.2307/1911134`.

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

Long JS, Ervin LH (2000). "Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model." *The American Statistician*, **54**, 217–224. `doi:10.1080/00031305.2000.10474549`.

Ma MS (2014). "Are We Really Doing What We Think We Are Doing? A Note on Finite-Sample Estimates of Two-Way Cluster-Robust Standard Errors." Mimeo. URL `http://ssrn.com/abstract=2420421`.

Mammen E (1992). *When Does Bootstrap Work?: Asymptotic Results and Simulations*, volume 77 of *Lecture Notes in Statistics*. Springer-Verlag.

Mammen E (1993). "Bootstrap and Wild Bootstrap for High Dimensional Linear Models." *The Annals of Statistics*, **21**(1), 255–285. `doi:10.1214/aos/1176349025`.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. 2nd edition. Chapman & Hall, London. `doi:10.1007/978-1-4899-3242-6`.

Messner JW, Mayr GJ, Zeileis A (2016). "Heteroscedastic Censored and Truncated Regression with **crch**." *The R Journal*, **8**(1), 173–181.

Miglioretti DL, Heagerty PJ (2007). "Marginal Modeling of Nonnested Multilevel Data Using Standard Software." *American Journal of Epidemiology*, **165**(4), 453–463. `doi:10.1093/aje/kwk020`.

Millo G (2017). "Robust Standard Error Estimators for Panel Models: A Unifying Approach." *Journal of Statistical Software*, **82**(3), 1–27. `doi:10.18637/jss.v082.i03`.

Molenberghs G, Kenward MG, Verbeke G, Iddi S, Efendi A (2013). "On the Connections between Bridge Distributions, Marginalized Multilevel Models, and Generalized Linear Mixed Models." *International Journal of Statistics and Probability*, **2**(4), 1–21. `doi:10.5539/ijsp.v2n4p1`.

Moulton BR (1986). "Random Group Effects and the Precision of Regression Estimates." *Journal of Econometrics*, **32**(3), 385–397. `doi:10.1016/0304-4076(86)90021-7`.

Moulton BR (1990). "An Illustration of a Pitfall in Estimating the Effects of Aggregate Variables on Micro Units." *The Review of Economics and Statistics*, **72**(2), 334–338. `doi:10.2307/2109724`.

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

Newey WK, West KD (1994). "Automatic Lag Selection in Covariance Matrix Estimation." *Review of Economic Studies*, **61**, 631–653. doi:10.2307/2297912.

Nichols A, Schaffer M (2007). "Clustered Standard Errors in Stata." *United Kingdom Stata Users' Group Meetings 2007*, Stata Users Group. URL http://RePEc.org/usug2007/crse.pdf.

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

Pustejovsky JE, Tipton E (2017). "Small-Sample Methods for Cluster-Robust Variance Estimation and Hypothesis Testing in Fixed Effects Models." *Journal of Business & Economic Statistics*, **36**(4), 672–683. doi:10.1080/07350015.2016.1247004.

Pustejovsky JE, Tipton E (2018). "Small-Sample Methods for Cluster-Robust Variance Estimation and Hypothesis Testing in Fixed Effects Models." *Journal of Business & Economic Statistics*, **36**(4), 672–683. doi:10.1080/07350015.2016.1247004.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Satterthwaite FE (1946). "An Approximate Distribution of Estimates of Variance Components." *Biometrics Bulletin*, **2**(6), 110–114.

Stack Overflow (2014). "Different Robust Standard Errors of Logit Regression in Stata and R." Accessed 2018-08-07, URL https://stackoverflow.com/questions/27367974/different-robust-standard-errors-of-logit-regression-in-stata-and-r.

StataCorp (2019). *Stata Statistical Software: Release 16*. StataCorp LLC, College Station. URL https://www.stata.com/.

Therneau TM (2020). **survival**: *Survival Analysis*. R package version 3.2-3, URL https://CRAN.R-project.org/package=survival.

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.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. doi:10.1007/978-0-387-21706-2.

Webb MD (2014). "Reworking Wild Bootstrap Based Inference for Clustered Errors." *Working Paper 1315*, Queen's Economics Department. URL http://qed.econ.queensu.ca/working_papers/papers/qed_wp_1315.pdf.

White H (1980). "A Heteroskedasticity-Consistent Covariance Matrix and a Direct Test for Heteroskedasticity." *Econometrica*, **48**, 817–838. doi:10.2307/1912934.

White H (1994). *Estimation, Inference and Specification Analysis*. Cambridge University Press, Cambridge.

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 (2006a). "Implementing a Class of Structural Change Tests: An Econometric Computing Approach." *Computational Statistics & Data Analysis*, **50**, 2987–3008. doi:10.1016/j.csda.2005.07.001.

Zeileis A (2006b). "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. URL https://CRAN.R-project.org/doc/Rnews/.

Zeileis A, Kleiber C (2020). **countreg**: *Count Data Regression*. R package version 0.2-1/r164, URL https://R-Forge.R-project.org/projects/countreg/.

Zeileis A, Kleiber C, Jackman S (2008). "Regression Models for Count Data in R." *Journal of Statistical Software*, **27**(8), 1–25. doi:10.18637/jss.v027.i08.

Zeileis A, Köll S, Graham N (2020). "Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R." *Journal of Statistical Software*, **95**(1), 1–36. doi:10.18637/jss.v095.i01.

A. Simulation results for panel data with AR(1) correlations

As observed in Figures 1–2, the estimators for panel covariances (PL and `PC`) have problems with the "short" panels of only 5 observations per cluster. To assess whether the estimators perform correctly in those situations they were designed for, we take a closer look at (a) "longer" panels (with up to 50 observations per cluster), and (b) an exponentially decaying autoregressive (AR) correlation structure of order 1 instead of an exchangeable correlation structure.

Figure 5 shows the results from a supplementary simulation experiment that is analogous to Experiment I. The two differences are: (1) The number of observations per cluster is increased from 5 up to 50 and the cluster correlation is fixed at $\rho = 0.25$ (with higher values leading to qualitatively the same results). (2) Additionally, an AR(1) correlation structure is considered. Somewhat surprisingly, the standard clustered HC0 covariance performs satisfactorily in all scenarios and better than the panel estimators (PL and PC). The latter approach the
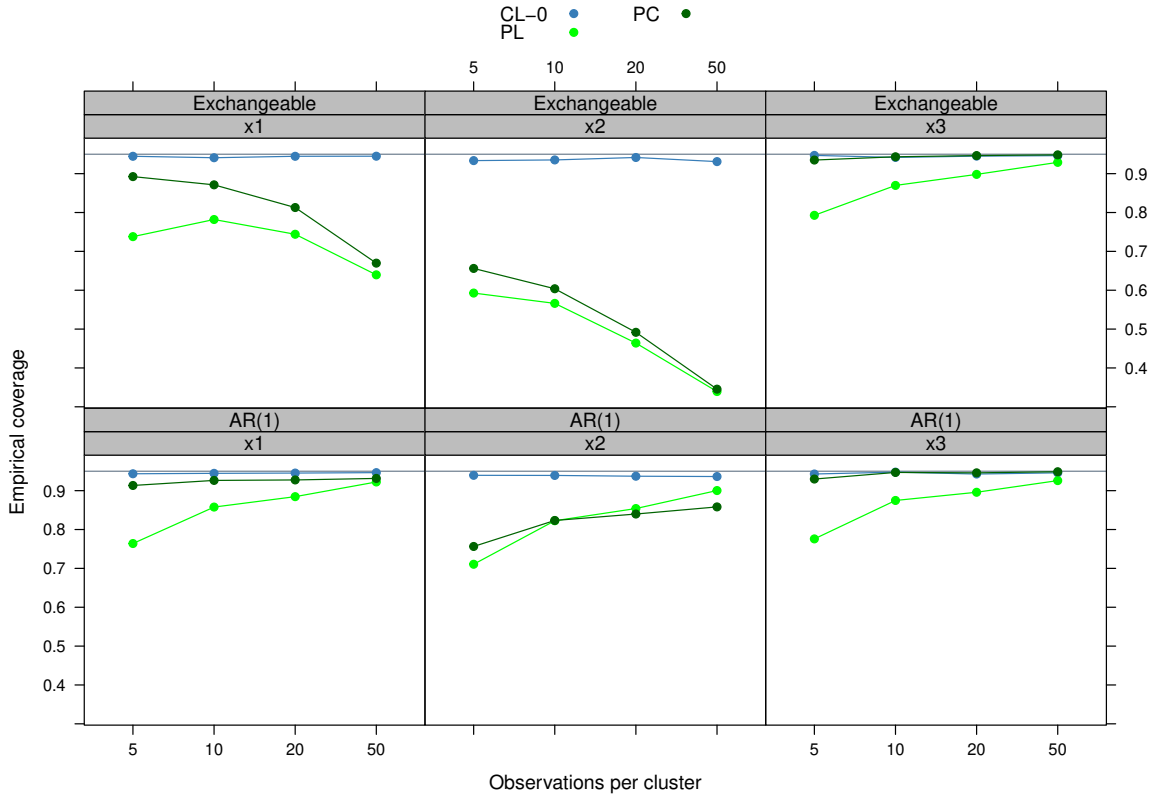

Figure 5: Supplementary simulation experiment. Gaussian response with $G = 100$ (balanced) clusters of 5, 10, 20, or 50 observations each. Regressor `x1` is correlated ($\rho_x = 0.25$), `x2` clustered, and `x3` uncorrelated. Either an exchangeable cluster correlation of $\rho = 0.25$ or an exponentially decaying AR(1) correlation structure with autoregressive coefficient $\rho = 0.25$ is used. The coverage (from 10,000 replications) is plotted on the y-axis against the number of observations per cluster on the x-axis. The horizontal reference line indicates the nominal coverage of 0.95.
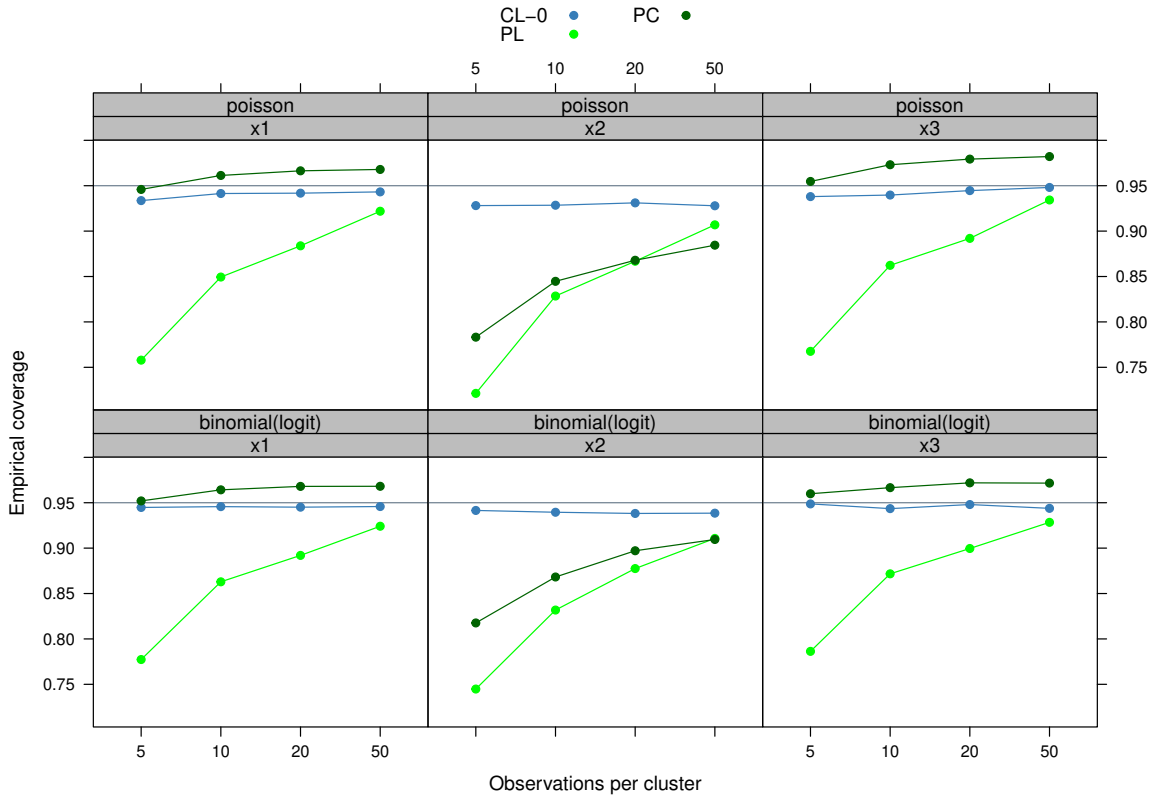
Figure 6: Supplementary simulation experiment. Poisson and binomial response with $G = 100$ (balanced) clusters of 5, 10, 20, or 50 observations each. Regressor x1 is correlated ($\rho_x = 0.25$), x2 clustered, and x3 uncorrelated. An exponentially decaying AR(1) correlation structure with autoregressive coefficient $\rho = 0.25$ is used. The coverage (from 10,000 replications) is plotted on the y-axis against the number of observations per cluster on the x-axis. The horizontal reference line indicates the nominal coverage of 0.95.

desired coverage of 0.95 when the panels become longer (i.e., the number of observations per cluster increases) and the correlation structure is AR(1). However, in case of an exchangeable correlation structure and correlated/clustered regressors, the coverage even decreases for longer panels. The reason for this is that the panel covariance estimators are based on the assumption that correlations are dying out, which is the case for an AR(1) structure, but not for an exchangeable correlation structure.

Additionally, Figure 6 brings out that the AR(1) findings are not limited to the Gaussian case but can be confirmed for binomial and Poisson GLMs as well.

Affiliation:

Achim Zeileis, Susanne Köll
Department of Statistics
Faculty of Economics and Statistics
Universität Innsbruck
Universitätsstr. 15
6020 Innsbruck, Austria
E-mail: Achim.Zeileis@R-project.org
URL: https://www.zeileis.org/

Nathaniel Graham
Division of International Banking and Finance Studies
A.R. Sanchez, Jr. School of Business
Texas A&M International University
5201 University Boulevard
Laredo, Texas 78041, United States of America
E-mail: npgraham1@npgraham1.com
URL: https://sites.google.com/site/npgraham1/