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

Susanne Berger
University of Innsbruck

Nathaniel Graham
Trinity University Texas

Achim Zeileis
University of Innsbruck

Abstract

Clustered covariances or clustered standard errors are very widely used to account for correlated or clustered data, especially in economics, political sciences, or 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. Now, this shortcoming has been corrected in **sandwich** (starting from version 2.4.0): Based on methods for two generic functions (`estfun()` and `bread()`), clustered and panel covariances are now provided in `vcovCL()` and `vcovPL()`, respectively. These are directly applicable to models from many packages, e.g., including **MASS**, **pscl**, **countreg**, **betareg**, among others. Some empirical illustrations are provided as well as an assessment of the methods’ performance in a simulation study.

Keywords: clustered data, clustered covariance matrix estimators, 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 2017), 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), **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

There is a range of popular strategies for dealing with 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, also leading naturally to a clustered covariance (e.g., available in the **geepack** package for R, Halekoh, Højsgaard, and Yan 2005). 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 – then also known as quasi-maximum likelihood (QML) estimator or, in GEE jargon, as independence working model – but adjusts the covariance matrix 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), (3) and clustered sandwich covariances for clustered or panel data (see e.g., Cameron and Miller 2015).

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.

2.1. R packages for sandwich covariances

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 regresion models. `vcovHC()` is a wrapper to `sandwich()` combined with `meatHC()` and `bread()` to compute general HC covariances ranging from HC0 to HC5. `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 2011) in function `hccm()`. Like `vcovHC()` from **sandwich** this is limited to the cross-section case without clustering, though.

2.2. 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 2016), **clubSandwich** (Pustejovsky 2016), and **clusterSEs** (Esarey 2017), 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.

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 $i = 1, \dots, n$ 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 $\hat{\theta} \approx \mathcal{N}(\theta, n^{-1} S(\theta))$. Moreover, the covariance matrix is of sandwich type with a meat matrix $M(\theta)$ between two slices of bread $B(\theta)$:

$$\begin{aligned} S(\theta) &= B(\theta) \cdot M(\theta) \cdot B(\theta) & (1) \\ B(\theta) &= \left(\mathsf{E}[-\psi'(y, x, \theta)]\right)^{-1} & (2) \\ M(\theta) &= \mathsf{VAR}[\psi(y, x, \theta)]. & (3) \end{aligned}$$

An estimating function

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

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):

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

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

$$\hat{B} = \left(\frac{1}{n} \sum_{i=1}^{n} -\psi'(y_i, x_i, \hat{\theta})\right)^{-1}, \qquad (6)$$

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

$$\begin{aligned}
\hat{M} &= \frac{1}{n} \sum_{i=1}^{n} \psi(y_i, x_i, \hat{\theta}) \psi(y_i, x_i, \hat{\theta})^\top & (7)\\
\hat{M}_{\text{HAC}} &= \frac{1}{n} \sum_{i,j=1}^{n} w_{|i-j|}\, \psi(y_i, x_i, \hat{\theta}) \psi(y_j, x_j, \hat{\theta})^\top & (8)\\
\hat{M}_{\text{HC}} &= \frac{1}{n} X^\top \begin{pmatrix} \omega(r(y_1, x_1^\top \theta)) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \omega(r(y_n, x_n^\top \theta)) \end{pmatrix} X. & (9)
\end{aligned}$$

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 $r(y_1, x_1^\top \hat{\theta}), \dots, r(y_n, x_n^\top \hat{\theta})$ and may also depend on hat values as well as degrees of freedom (Zeileis 2006b). The HC type of the model in Equation 9 has to be of a form that allows factorization of the scores

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

into empirical working residuals times the regressor vector. 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 $\hat{M}_{\text{HAC}}$ from Equation 8, and `meatHC()` for $\hat{M}_{\text{HC}}$ 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 $1, \dots, n$ observations but an observation (i, g) from $1, \dots, n_g$ observations within cluster/group g (with $g = 1, \dots, G$ and $n = n_1 + \dots + n_G$). 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:

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

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

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

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, but this comes at the cost that the number of clusters G must approach infinity, not just the number of observations n (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 similar to the corresponding cross-section data estimators. HC0 to HC3 bias corrections for cluster g are defined as

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

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

$$\mathrm{HC2}: \quad (I_{n_g} - H_{gg})^{-0.5} \tag{16}$$

$$\mathrm{HC3}: \quad (I_{n_g} - H_{gg})^{-1}, \tag{17}$$

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. Thus, it is completely straightforward to add the factors for HC0 and HC1 to $\hat{M}_{\mathrm{CL}}$ (Equation 12). However, but is more demanding to apply the cluster generalizations of HC2 and HC3 (due to Kauermann and Carroll 2001; Bell and McCaffrey 2002) for which the correction factors from Equations 16 and 17 have to be applied to the (working) residuals prior to computing the clustered meat matrix. More precisely, the empirical working residuals $r(y_g, x_g^\top \hat{\theta})$ in group g are adjusted via

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

with $\alpha = -0.5$ for HC2 and $\alpha = -1$ 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 in terms of methods in R, it is not sufficient 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 $\text{diag}H$) 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. It can therefore 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, Gelbach, and Miller (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 up or substracted off, 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 up, and the two-way clustered covariance matrix with clusters formed by the intersection of *id* and *time* is substacted off

$$\hat{M}_{\text{CL}(id,time)} \quad = \quad \hat{M}_{\text{CL}(id)} + \hat{M}_{\text{CL}(time)} - \hat{M}_{\text{CL}(id \cap time)}. \tag{20}$$

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 up, whereas those with an even number are substracted.

Petersen (2009), Thompson (2011) and Ma (2014) suggest to substract the standard sandwich estimator in case that the clusters formed by the intersection of *id* and *time* do only contain a single observation. As Ma (2014) argues, the standard sandwich estimator does not take into account any bias adjustment, such that the cluster bias correction $\frac{G}{G-1}$ is larger than one. Thus, covariances are underestimated if always the clustered covariance matrix with cluster bias correction $\frac{G}{G-1}$ is substracted as the last substracted matrix instead of standard sandwich covariances.

3.3. Clustered covariances for panel data

The information of 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 2014, for a unifying view).

To reflect that the data are now panel data with time ordering within each cluster/group/id we change our notation to an index (i, t) for $i = 1, \dots, n_t$ observations at time $t = 1, \dots, T$ (with $n = n_1 + \dots + n_T$). Note that compared to the notation from the clustered case above, the variable i now denotes the group/id (e.g., firm) which was previously denoted by g.

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 2014). The meat is composed of

$$\hat{M}_{\mathrm{PL}}^{NW} = \frac{1}{n} \sum_{i,j=1}^{n} w_{|i-j|} \, \psi(y_i, x_i, \hat{\theta}) \psi(y_j, x_j, \hat{\theta})^\top. \tag{21}$$

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

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 cluster/id within each time period t. This yields a panel sandwich estimator where the meat is computed as

$$\hat{M}_{\mathrm{PL}} = \frac{1}{n} \sum_{t=1}^{T} \sum_{i,j=1}^{n_t} w_{|i-j|} \, \psi(y_{it}, x_{it}, \hat{\theta}) \psi(y_{jt}, x_{jt}, \hat{\theta})^\top, \tag{22}$$

The weights $w_{|i-j|}$ 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 2014). They are robust against panel heteroscedasticity and contemporaneously correlation, with the crucial assumption that contemporaneous correlation accross cluster follows a fixed pattern (Millo 2014; 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 that now both cross section *and* and time ordering are considered, index (t, g) is employed for the observation from cluster/group $g = 1, \dots, G$ at time $t = 1, \dots, n_g$. 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 vectors for each cluster g: $r(y_1, x_1^\top \hat{\theta}), \dots, r(y_G, x_G^\top \hat{\theta})$. For balanced data these can be arranged in a $T \times G$ matrix,

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

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 `betareg`, `clm`, `coxph`, `crch`, `glm`, `hurdle`, `lm`, `mlm`, `mlogit`, `nls`, `polr`, `rlm`, `survreg`, or `zeroinfl` from packages **stats** (R Core Team 2017), **betareg** (Cribari-Neto and Zeileis 2010; Grün *et al.* 2012), **crch** (Messner, Mayr, and Zeileis 2016), **MASS** (Venables and Ripley 2002), **mlogit** (Croissant 2013), **ordinal** (Christensen 2015), and **survival** (Therneau 2017). 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 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 substract the basic cross-section HC0 covariance matrix as the last substracted 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, specificially 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 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 to specify a variable indicating the cluster/group/id variable 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 with both variables: the cluster/group/id and the time/ordering variable, respectively.

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 $\text{floor}[4 \cdot (\frac{T}{100})^{2/9}]$ (Newey and West 1994).

The `adjust` argument allows to make a finite sample adjustment, which amounts to multiplication with $n/(n-k)$, 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, subsample = FALSE,
 sandwich = TRUE, fix = FALSE, ...)
```

based on

```
meatPC(x,  cluster = NULL, order.by = NULL, subsample = 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 `subsample = TRUE`, a balanced subset of the panel is employed, whereas for `subsample = FALSE`, a pairwise balanced sample 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 elementwise. 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.

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 provided. In Section 5.1 it is illustrated how these can now be easily obtained.

To replicate further classic results, the benchmark data from Petersen (2009) is considered in Section 5.2. This focuses on the classical linear regression case with model errors that are correlated within clusters. It is shown how the results from various other R packages (**multiwayvcov**, **plm**, **geepack**, **pcse**) can be replicated using the new **sandwich** package. One- and two-way clustered standard errors from `vcovCL()` from **sandwich** are compared to those obtained using `cluster.vcov()` from **multiwayvcov** package. Furthermore, one-way clustered standard errors can also be obtained from packages **plm** and **geepack**, and are additionally benchmarked against the results from **sandwich**. Function `vcovSCC()` from **plm** gives Driscoll and Kraay type standard errors, the estimates are compared to `vcovPL()` from **sandwich**. Also, panel-corrected standard errors can be estimated by function `vcovPC()` from **pcse** and are benchmarked against `vcovPC()` from **sandwich**.

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

In this section, in a further example we will make use of the object-orientation of `vcovCL()`, and estimate clustered standard errors for a count data hurdle model.

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. Still, one limitation of standard count data models is that the zeros and the nonzeros (positives) are assumed to come from the same data-generating process. From an economic perspective, there is a difference in determinants of "first innovation" and "continuing innovation". The rationale behind this is the notion of nonlinearities in the innovation process. In case that the first innovation is especially hard to obtain in comparison to succeeding innovations, hurdle models offer a useful way that allows for a distinction to be made between these two processes (Berger *et al.* 2017). Therefore, Berger *et al.* (2017) 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. The Aghion *et al.* (2013) data are available in the **sandwich** package.

```
R> data("InstInnovation", package = "sandwich")
```

Hurdle models are fitted with the `hurdle` function from the **pscl** package (Zeileis *et al.* 2008). Here, the count model family chosen is a negative binomial, and the zero hurdle model family is a binomial with logit link.

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

Below, a comparison of "standard" standard errors, basic sandwich standard errors and clustered standard errors for an exemplary hurdle model is shown. Standard errors are clustered by companies, with a total of 803 clusters.

```
R> library("sandwich")
R> vc <- list(
+     "standard" = vcov(h.innov),
+     "basic" = sandwich(h.innov),
+     "CL-1" = vcovCL(h.innov, cluster = InstInnovation$company)
+   )
R> sapply(vc, function(x) sqrt(diag(x)))
```

```
                                standard       basic        CL-1
count_(Intercept)            0.224989395 0.603427666 0.894454123
count_institutions           0.001602172 0.002466434 0.004302931
count_log(capital/employment) 0.054661378 0.081753967 0.152141612
count_log(sales)             0.012616546 0.032207092 0.049991706
zero_(Intercept)             0.146024566 0.153040605 0.269084403
zero_institutions            0.001350772 0.001347274 0.002086965
zero_log(capital/employment)  0.033364941 0.035876845 0.064805276
zero_log(sales)              0.016423263 0.016589737 0.027764751
```

What can be observed is that when the data are clustered, basic standard errors can greatly overstate estimator precision. Then, for the exemplary hurdle model, clustered standard errors are scaled up by factors between 1.48 and 1.86 even compared to standard sandwich standard errors.

5.2. Petersen (2009)

Benchmark data for testing the clustered standard error estimates in the linear model is a simulated data set[1] provided by Petersen (2009), containing 500 firms over 10 years. The

---

[1] http://www.kellogg.northwestern.edu/faculty/petersen/htm/papers/se/test_data.txt

data include 4 variables, `firm`, `year`, dependent variable `y` and explanatory variable `x` (see also Graham *et al.* 2016).

The data are available in **sandwich** as well as in **multiwayvcov**.

```
R> data("PetersenCL", package = "sandwich")
```

A linear model is fitted with `lm()`,

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

and testing the estimated coefficients with `coeftest()` from package **lmtest** gives

```
R> library("lmtest")
R> coeftest(p_lm)

t test of coefficients:

            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.029680   0.028359  1.0466   0.2954
x           1.034833   0.028583 36.2041   <2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

However, the data are clustered by firms (`firm`), with a total of 500 clusters. But ignoring dependency by using “standard” standard errors results in an overstate of estimator precision.

One-way clustered standard errors

Thus, one-way clustered standard errors clustered by `firm` are employed. It can be observed that clustered standard errors are larger than “standard” standard errors.

```
R> library("sandwich")
R> coeftest(p_lm, vcov = vcovCL(p_lm, cluster = PetersenCL$firm))

t test of coefficients:

            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.029680   0.067013  0.4429   0.6579
x           1.034833   0.050596 20.4530   <2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Next, it is shown how the results from various other R packages (**multiwayvcov**, **plm**, **geepack**, **pcse**) can be replicated using the new **sandwich** package.

First, `vcovCL()` from **sandwich** and `cluster.vcov()` from **multiwayvcov** are compared.

```
R> library("multiwayvcov")
```

Both functions are closely related, as `vcovCL()` is partly based on `cluster.vcov()`.

```
R> vc <- list(
+   "sandwich" = vcovCL(p_lm, cluster = PetersenCL$firm),
+   "multiwayvcov" = cluster.vcov(p_lm, cluster = PetersenCL$firm)
+   )
R> sapply(vc, function(x) sqrt(diag(x)))

              sandwich multiwayvcov
(Intercept) 0.06701270   0.06701270
x           0.05059573   0.05059573
```

The bias correction in `vcovCL()` is set to `cadjust = TRUE` and `type = "HC1"` by default. The same type of bias correction is achieved for `cluster.vcov()` by default.

Second, `vcovCL()` is compared to packages **plm** and **geepack**, who are also able to calculate clustered covariances.

```
R> library("plm")
R> library("geepack")
```

A pooling model is estimated with `plm()`, `geeglm()` fits a generalized estimating equation (GEE) with independence correlation structure.

```
R> p_plm <- plm(y ~ x, data = PetersenCL, model = "pooling",
+    indexes = c("firmid", "year"))
R> p_gee <- geeglm(y ~ x, data = PetersenCL, id = PetersenCL$firm,
+    corstr = "independence", family = gaussian)
```

As there is no `vcov()`-method specified for `geeglm()`, the convenience function `vcov.geeglm()` is provided to make this contribution.

```
R> vcov.geeglm <- function(object) {
+   vc <- object$geese$vbeta
+   rownames(vc) <- colnames(vc) <- names(coef(object))
+   return(vc)
+   }
```

Nevertheless, in order to accomplish exactly the same value for clustered standard errors using `plm` and `geepack`, the bias correction $\frac{G}{G-1}\frac{n-1}{n-k}$ has to be omitted from the clustered covariance matrix estimated by `vcovCL()`. This is achieved by setting the function arguments `cadjust = FALSE` and `type = "HC0"`.

```
R> vc <- list(
+   "sandwich" = vcovCL(p_lm, cluster = PetersenCL$firm,
+    cadjust = FALSE, type = "HC0"),
+   "plm" = vcovHC(p_plm, type = "HC0", cluster = "group"),
+   "geepack" = vcov(p_gee)
+   )
R> sapply(vc, function(x) sqrt(diag(x)))
```

```
               sandwich        plm    geepack
(Intercept) 0.06693896 0.06693896 0.06693896
x           0.05054005 0.05054005 0.05054005
```

All packages examined produce the same clustered standard errors for `lm` model objects. For packages **plm** and **geepack**, one has to omit the HC bias correction factor as well as the cluster bias correction. In general, differences in clustered standard errors often coincide with different types of bias corrections.

Two-way clustered standard errors

Two-way clustered standard errors with cluster dimensions `firm` as well as `year` are, at least for the explanatory variable `x`, a bit larger than one-way clustered standard errors clustered by `firm`.

It can be observed that `vcovCL()` from **sandwich** and `cluster.vcov()` from **multiwayvcov** deliver equivalent results.

```
R> cluster <- cbind(PetersenCL$firm, PetersenCL$year)
R> vc <- list(
+   "sandwich" = vcovCL(p_lm, cluster = cluster, cadjust = TRUE, type = "HC1"),
+   "multiwayvcov" = cluster.vcov(p_lm, cluster = cluster, use_white = FALSE,
+   df_correction = TRUE)
+   )
R> sapply(vc, function(x) sqrt(diag(x)))
```

```
               sandwich multiwayvcov
(Intercept) 0.06506392   0.06506392
x           0.05355802   0.05355802
```

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

Driscoll and Kraay standard errors

For Driscoll and Kraay standard errors, `vcovPL()` and `vcovSCC` from **plm** deliver equivalent results, given the same lag length and bias-correction. Here, the maximum lag length is chosen (which is equal to `length(PetersenCL$year) - 1`) with a `HC1` bias correction.

```
R> vc <- list(
+   "sandwich" = vcovPL(p_lm, cluster = PetersenCL$firm,
+    adjust = TRUE, lag = "max"),
+   "plm" = plm::vcovSCC(p_plm, maxlag = 9, inner = "cluster",
+    type = "HC1")
+   )
R> sapply(vc, function(x) sqrt(diag(x)))
```

```
              sandwich        plm
(Intercept) 0.01619300 0.01619300
x           0.01426406 0.01426406
```

Panel-corrected standard errors

Panel-corrected standard errors can as well be calculated from **sandwich** in function `vcovPC()`, which gives the same results as `vcovPC` from **pcse**.

```
R> library("pcse")
R> vc <- list(
+   "sandwich" = sandwich::vcovPC(p_lm, cluster = PetersenCL$firm,
+    order.by = PetersenCL$year),
+   "pcse" = pcse::vcovPC(p_lm, groupN = PetersenCL$firm,
+    groupT = PetersenCL$year)
+   )
R> sapply(vc, function(x) sqrt(diag(x)))
```

```
              sandwich       pcse
(Intercept) 0.02220064 0.02220064
x           0.02527598 0.02527598
```

6. Simulation

Next, we run a Monte Carlo simulation to assess the methods' performance in a simulation study. The aim is to test clustered standard errors beyond linear and generalized linear models. For the linear model, there are a couple of simulation studies in the literature (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 less for generalized linear models (see for example Miglioretti and Heagerty 2007) and, up to our knowledge, none for models beyond `lm()` and `glm()`.

6.1. Simulation design

Parameters ρ and G vary systematically, where ρ determines the strength of cluster correlation and varies from 0 to 0.9. The number of clusters G ranges from 10 to $50, 100, 150, 200$ to 250. Numerous studies (Green and Vavreck 2008; Arceneaux and Nickerson 2009; Harden 2011) confirmed that for the linear regression case, the higher the number of clusters, the less standard errors are biased. Furthermore, only balanced cluster are investigated, with the number of observations per cluster fixed to 5.

Linear predictor

The linear predictor is

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

with a link function h and expected value μ_{ig}. In order to introduce cluster correlation in the response, two options are investigated. The first option is to take the marginal model Equation 25 and introduce correlation via a Gaussian copula. The second option is to add a random effect in Equation 25. For the linear model with an identity link function $h(\cdot)$, both options amount to the same thing. Though for models other than `lm()`, the alternatives are different. The copula option is favored in the simulation exercises, as for models other than `lm()`, including a random effect comes along with including a bias.

We analyze three regressor variables

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

Regressor $x_{1,ig}$ in Equation 26 is composed of a linear combination of a random draw at cluster level ($\mathcal{N}_g$) and a random draw at individual level ($\mathcal{N}_{ig}$). Regressor $x_{2,g}$ in Equation 27 is composed of a random draw at cluster level, and regressor $x_{3,ig}$ in Equation 28 consists of a random draw at individual level. In most of the simulations, only a single regressor $x_{1,ig}$ is used.

In line with Harden (2011), this setup is used to introduce correlated (26), clustered (27) and uncorrelated regressors (28). The cluster correlation of $x_{1,ig}$ is controlled by parameter ρ_x, and set to 0.25 per default. This implies at least some within cluster correlation of regressor $x_{1,ig}$. For $\rho_x = 1$, regressors $x_{1,ig}$ and $x_{2,g}$ are equivalent. Furthermore, if $\rho_x = 0$, regressor $x_{1,ig}$ corresponds to regressor $x_{3,ig}$.

The vector of coefficients is fixed to

$$
\begin{aligned}
\beta_1 &= (0, 0.85, 0.5, 0.7)^\top && (29)\\
\beta_2 &= (0, 0.85, 0, 0)^\top && (30)
\end{aligned}
$$

even though these values can be interchanged without influencing the results[2].

Response distributions examined include Gaussian (`gaussian`), binomial with a logit link (`binomial(logit)`), Poisson (`poisson`), zero-truncated Poisson (`zerotrunc`), Beta (`beta`) and zero-inflated Poisson (`zip`). `vcovCL()` allows estimation of clustered covariances for all abovementioned responses (and many more).

Sandwich covariances

Covariances being compared to each other include "standard" covariances without heteroscedasticity and without autocorrelation (`standard`), basic sandwich covariances (`basic`), Driscoll and Kraay type covariances (`PL`), panel-corrected covariances (`PC`) a la Beck and Katz (1995) and clustered covariances with HC0 to HC3 adjustment (`CL-0`, `CL-1`, `CL-2`, `CL-3`). In addition, covariances from a random effects model (`random`) and from a GEE with exchangeable correlation structure (`gee`) are estimated.

Outcome measure

In order to assess the validity of statistical inference, the empirical coverage rate is the outcome measure of interest. If standard errors are estimated accurately, the empirical coverage rate of

---

[2] Values are equivalent to Harden (2011).

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

Table 1: Covariance matrices for a Gaussian response in `sim-CL.R`.

the 95% confidence interval should be close to 0.95. Values less than 0.925 will be considered to have underestimated standard errors towards a Type I error, while values greater than 0.975 will be considered to have overestimated standard errors towards a Type II error.

Simulation code

The 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 (at most) three regressors `x1`, `x2`, `x3` as well as cluster dimensions `id` and `time`. `fit()` is responsible for the model fitting, the covariances as well as further outcomes (bias, mad, power and coverage). `sim()` conducts all simulations and provides for parallelization support.

Table 1 shows exemplary how the different types of covariances are calculated for a Gaussian response. A pooled model (`m`), a random effects model (`m_re`) and a GEE with an exchangeable correlation structure (`m_gee`) is fitted. The random effects model is specified as a model with random intercepts. Model fittig for random effects models and GEEs requires the packages **lme4** and **geepack**, respectively. However, it is not possible to fit all model types for all mentioned response distributions. There are some limitations for random effects models as well as for GEEs.

Furthermore, to calculate clustered standard errors, the square root of the diagonal elements of the covariance matrices in Table 1 is calculated.

6.2. Results

The questions which should be discussed and answered by the simulation study are:

- Experiment I: How do clustered covariances perform for different types of regressors for a Gaussian response distribution?
- Experiment II: How do clustered covariances perform for glm response distributions?
- Experiment III: How do clustered covariances perform for more general model response distributions?

- Experiment IV: Which type of HC0–HC3 bias correction performs best?

Experiment I

Figure 1 shows the results from Experiment I and plots the coverage probabilities for coefficients `x1` (Equation 26), `x2` (Equation 27) and `x3` (Equation 28) on the y-axis. On the x-axis the cluster correlation of the response ρ is shown, running from 0 (no correlation) to 0.9 (high correlation). Depending on the regressors, the implications of running the cluster correlation ρ from 0 to 0.9 on the coverage of different covariance estimators are substantial.

Starting with regressor `x1` at $\rho = 0$, all standard errors except the Driscoll and Kraay standard errors perform well and are close to the "true" standard error. As the number of observations per cluster is small with only $N_g = 5$, Driscoll and Kraay standard errors are substantially underestimated, even at a ρ of 0. In a Monte-Carlo study, Driscoll and Kraay (1998) put the minimum of $N_g > 20 - 25$. Panel-corrected standard errors for regressor `x1` underestimate the "true" standard errors, that applies all the more the higher ρ. To a smaller extent, the same holds true for "standard" standard errors as well as basic sandwich standard errors. As noted by Hoechle (2007), PCSE can be quite imprecise if the crosss-sectional dimension is large compared to the time dimension.

It can be observed that for regressor `x2`, Driscoll and Kraay standard errors, PCSE, and to a smaller extent also "standard" standard errors and basic sandwich standard errors are getting worse the larger the within cluster correlation. "Standard" standard errors are biased downwards because of the wrong assumption of independent observations (see also Harden 2011).

For regressor `x3`, independent of the value of ρ, all methods perform well (except again the Driscoll and Kraay estimator).

In summary, it holds for the linear model that if a regressor exhibits within cluster correlation even to a small extent, "standard" standard errors and basic sandwich standard errors that do not take the cluster dimension into account, deteriorate with an increasing ρ. In addition, PCSE and Driscoll and Kraay standard errors do not perform sufficiently well because of a too small N_g.

As the effects of regressor `x1` are in between the effects of regressors `x2` (varies at cluster level) and `x3` (varies at individual level), we limit ourselves to a single regressor `x1` in the following simulation experiments.

Experiment II

Figure 2 illustrates the results from Experiment II. The y-axis represents again the coverage, and ρ is plotted on the x-axis. Three response distributions are compared: Gaussian, binomial (with a logit link) and Poisson. Unlike in Experiment I, only a single regressor `x1` is included, with a cluster correlation ρ_x of 0.25. The left panel depicts the coverage for the Gaussian response, and thus corresponds to the left panel in Figure 1.

Results are qualitatively similar for the binomial and Poisson responses. The only exception are the standard errors from a random effects model, where the coverage is moving faster downwards than for the other standard errors. One can thus conclude that clustered standard errors will also work for glm's.

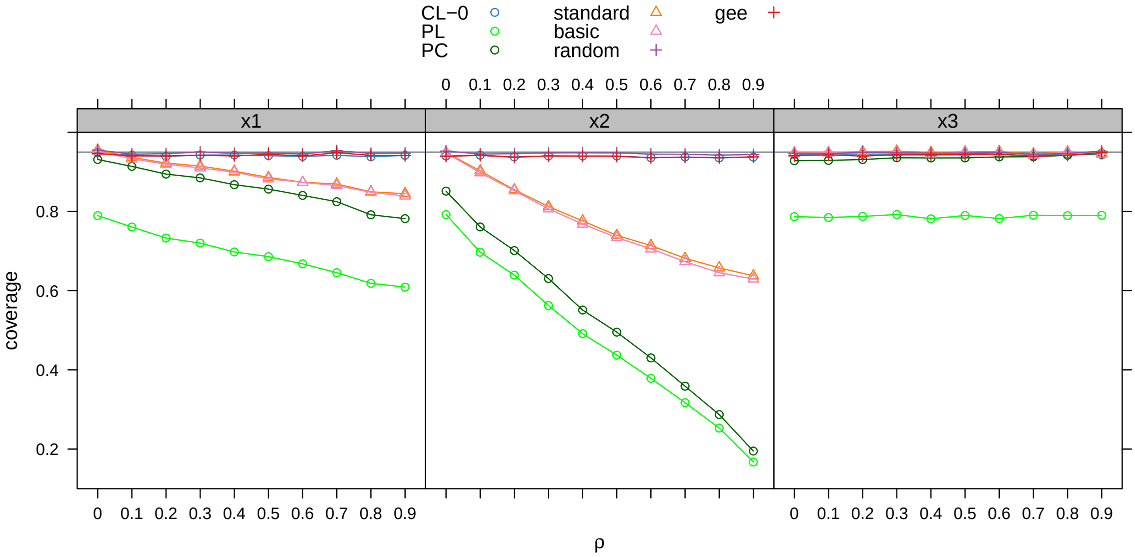


Figure 1: Experiment I

Experiment I explores how various types of (clustered) covariances perform for the linear regression model with Gaussian response and data composed of $G = 100$ (balanced) clusters each with $N_g = 5$ observations. The simulations of Experiment I are composed of $10,000$ replications. The regressors are included in the model, `x1` is composed of a linear combination of random draw at cluster level and a random draw at individual level with cluster correlation $\rho_x = 0.25$. `x2` consists of a random draw at cluster level, `x3` of a random draw at individual level. The coverage is plotted on the y-axis, the cluster correlation ρ is plotted on the x-axis. In a nutshell, if a regressor exhibits within cluster correlation even to a small extent, basic sandwich standard errors and "standard" standard errors that do not take the cluster dimension into account deteriorate with an increasing ρ. In addition, PCSE and Driscoll and Kraay standard errors do not perform sufficiently well because of a too small N_g. Nevertheless, clustered standard errors as well as standard errors from a random effects model and from a GEE with exchangeable correlation structure perform well.

Experiment III

Figure 3 shows the outcome of Experiment III and reveals the strength of the new function `vcovCL()` that comes with the feature that allows to estimate clustered covariances for models beyond glms. The y-axis represents again the coverage, and ρ is plotted on the x-axis. Comparisons are made between beta regression, zero-truncated Poisson and zero-inflated Poisson (ZIP). For these response distributions, clustered standard errors perform well and remain nearly constant even for an increasing ρ. Standard errors that do not take into account the cluster correlation, like "standard" standard errors and basic sandwich standard errors underestimate the true standard errors the more the larger ρ.

Experiment IV

Figure 4 depicts the findings of Experiment IV. The y-axis represents again the coverage, but in contrast to the other simulation experiments, the number of clusters G is plotted on the x-axis, ranging from 10 to 50 clusters, and in further steps of 50 up to 250 clusters. Gaussian, binomial and Poisson responses are compared with each other, with the focus on clustered standard errors with HC0–HC3 types of bias correction.

In most cases, all of the standard errors are underestimated for $G = 10$ clusters (except

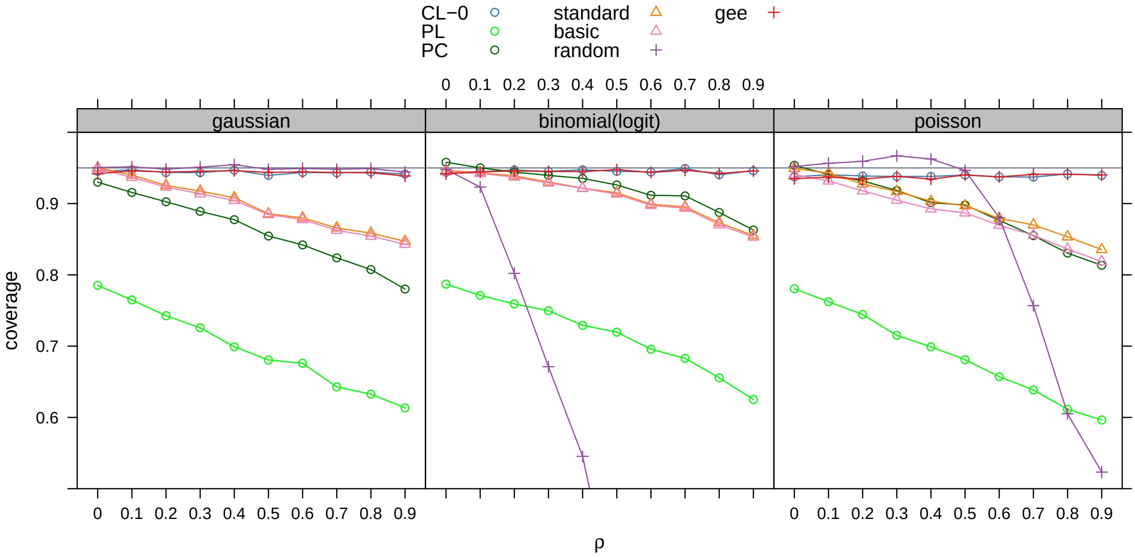


Figure 2: Experiment II

Experiment II explores how various types of (clustered) covariances perform for Gaussian, binomial and Poisson responses, where the data are composed of $G = 100$ (balanced) clusters each with $N_g = 5$ observations. The simulations of Experiment II are composed of $10,000$ replications. There is only a single regressor included in the model, `x1` is composed of a linear combination of random draw at cluster level and a random draw at individual level with cluster correlation $\rho_x = 0.25$. The coverage is plotted on the y-axis, the cluster correlation ρ is plotted on the x-axis. Compared to the Gaussian response, results are qualitatively similar for the binomial and Poisson responses. The only exception are the standard errors from a random effects model, where the coverage is moving faster downwards than for the other standard errors. One can thus conclude that clustered standard errors will also work for glm's.

clustered standard errors with HC3 bias correction for the binomial response). The larger the number of clusters G, the better the coverage and the less standard errors are underestimated. It can be observed that about 50 clusters is often enough for accurate inference. This result is well known in the literature (Arceneaux and Nickerson 2009; Petersen 2009; Harden 2011; Cameron and Miller 2015, among others).

Additionally, it can be observed that the higher the number of clusters, the less the different types of HC bias correction make the difference. However, for a small number of clusters, the HC3 correction works best, followed by HC2, HC1 and HC0. For the linear model, Long and Ervin (2000) suggest to use HC3 for small samples (with less than 250 observations).

Computation details

The packages **sandwich**, **countreg**, **geepack**, **lattice**, **lme4**, **lmtest**,**multiwayvcov**, **plm** and **pscl** are required for the applications in this paper. For replication of the simulation exercises, the R script `sim-CL.R` is required.

R version 3.4.1 has been used for computations. Package versions that have been employed are **sandwich** 2.4–0, **countreg** 0.2–0, **geepack** 1.2–1, **lattice** 0.20–35, **lme4** 1.1–13, **lmtest** 0.9–35, **multiwayvcov** 1.2.3, **plm** 1.6–6, and **pscl** 1.4.9 have been used.

R itself and all packages (except **countreg**) used are available from CRAN at `https://CRAN.`

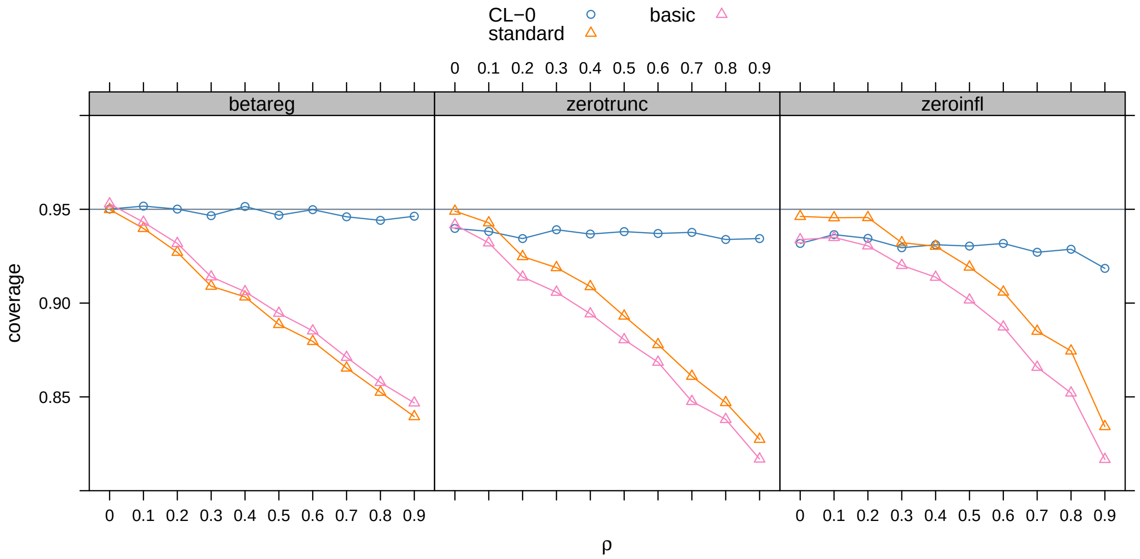


Figure 3: Experiment III

Experiment III explores how various types of (clustered) covariances perform for Beta, zero-truncated Poisson and zero-inflated Poisson (ZIP) responses, where the data are composed of $G = 100$ (balanced) clusters each with $N_g = 5$ observations. The simulations of Experiment III are composed of $10,000$ replications. There is only a single regressor included in the model, `x1` is composed of a linear combination of random draw at cluster level and a random draw at individual level with cluster correlation $\rho_x = 0.25$. The coverage is plotted on the y-axis, the cluster correlation ρ is plotted on the x-axis. Comparisons are made between beta regression, zero-truncated Poisson and zero-inflated Poisson. For these response distributions, clustered standard errors perform well and remain nearly constant even for an increasing ρ. Standard errors that do not take into account the cluster correlation, like "standard" standard errors and basic sandwich standard errors underestimate the true standard errors the more the larger ρ.

`R-project.org/`. **countreg** is accessible from `https://R-Forge.R-project.org/projects/countreg/`.

References

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`.

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`.

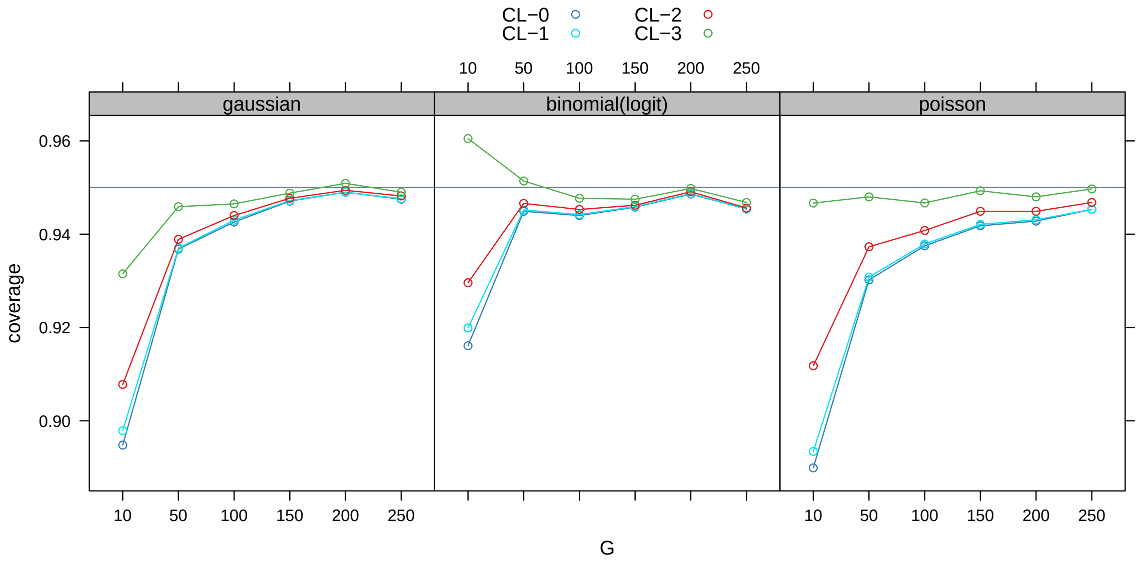


Figure 4: Experiment IV

Experiment IV explores how clustered covariances with HC0-HC3 correction perform for Gaussian, binomial and Poisson responses, the cluster correlation is fixed at $\rho = 0.25$. The data are composed of $G = 10, 50, 100, \dots, 250$ (balanced) clusters each with $N_g = 5$ observations. The simulations of Experiment IV are composed of $10,000$ replications. There is only a single regressor included in the model, `x1` is composed of a linear combination of random draw at cluster level and a random draw at individual level with cluster correlation $\rho_x = 0.25$. The coverage is plotted on the y-axis, the number of clusters G on the x-axis.

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.

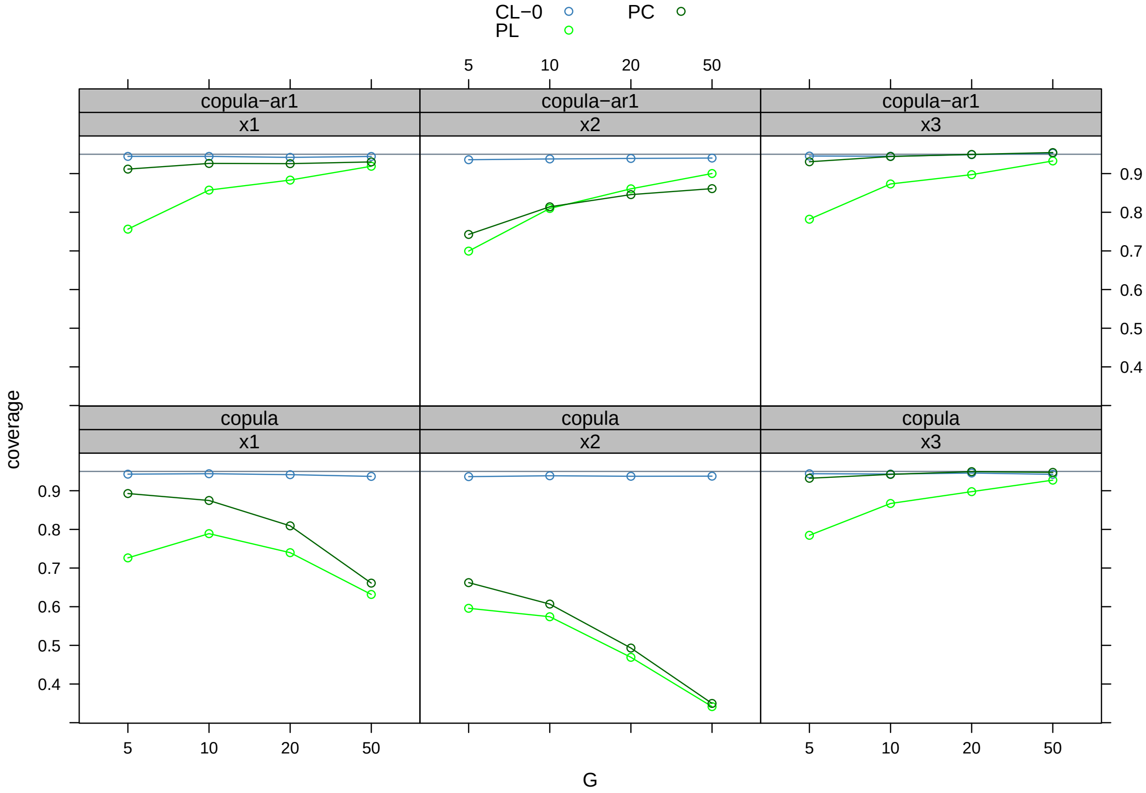
Berger S, Stocker H, Zeileis A (2017). "Innovation and Institutional Ownership Revisited: An Empirical Investigation with Count Data Models." *Empirical Economics*, **52**(4), 1675–1688. doi:10.1007/s00181-016-1118-0.

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 (2015). ***ordinal**: Regression Models for Ordinal Data.* R package version 2015.6-28, 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 (2013). ***mlogit**: Multinomial Logit Model.* R package version 0.2-4, URL `https://CRAN.R-project.org/package=mlogit`.

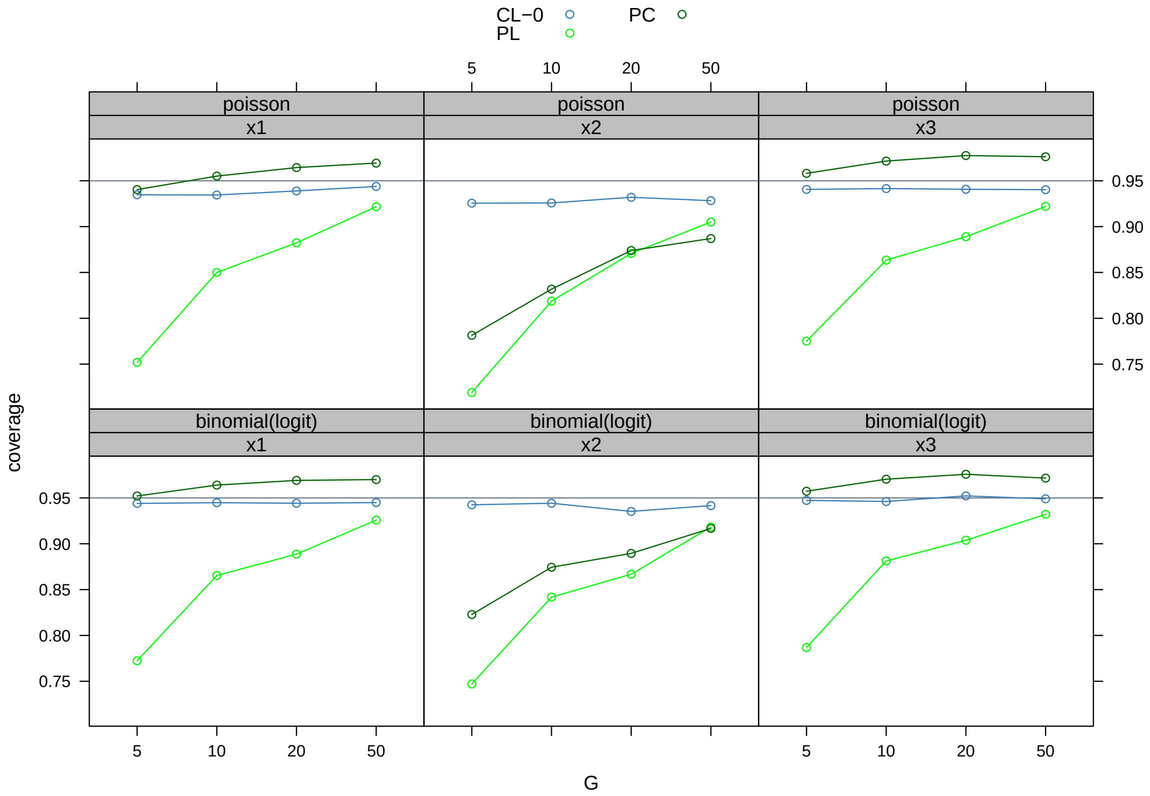
Croissant Y, Millo G (2008). "Panel Data Econometrics in R: The **plm** Package." *Journal of Statistical Software*, **27**(2). `doi:10.18637/jss.v027.i02`.

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`.

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 (2017). ***clusterSEs**: Calculate Cluster-Robust p-Values and Confidence Intervals.* R package version 2.3.3, URL `https://CRAN.R-project.org/package=clusterSEs`.

Fox J, Weisberg S (2011). *An R Companion to Applied Regression.* 2nd edition. Sage, 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 (2016). ***lfe**: Linear Group Fixed Effects.* R package version 2.5-1998, URL `https://CRAN.R-project.org/package=lfe`.

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`.

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`.

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`.

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 (2014). "Robust Standard Error Estimators for Panel Models: A Unifying Approach." Munich Personal RePEc Archive.

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`.

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 J (2016). ***clubSandwich**: Cluster-Robust (Sandwich) Variance Estimators with Small-Sample Corrections.* R package version 0.2.2, URL `https://CRAN.R-project.org/package=clubSandwich`.

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

Therneau TM (2017). ***survival**: Survival Analysis.* Version 2.41-3, 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`.

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, 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`.

A. Simulation of correlated data with a random cluster effect

Introducing within cluster error correlation via a random cluster effect instead of a copula comes along with introducing a bias, which also affects the coverage. The linear predictor in Equation 25 with a random cluster effect v_g is

$$h(\mu_{ig}) = \beta_0 + \beta_1 \cdot x_{1,ig} + \beta_2 \cdot x_{2,g} + \beta_3 \cdot x_{3,ig} + v_g, \tag{31}$$

where v_g is a random draw at cluster level $v_g \sim \mathcal{N}_g(0, \frac{\rho}{1-\rho})$. ρ ranges from 0 to 0.9 and determines again the importance of the random cluster effect.

Contrasting coverage and bias plots for models except the linear model depicts the problem caused by this approach.

Figure 5 constrasts coverage (in the upper panel) and bias (in the lower panel) for Gaussian, binomial (with a logit link) and Poisson response distributions with a single regressor `x1`. Regressor `x1` exhibits a cluster correlation ρ_x of 0.25. Each of the 10,000 simulated datasets consists of 100 clusters with 5 observations per cluster.

For the Gaussian response, the random cluster effect does not introduce a bias, but for the binomial and Poisson responses, the random cluster effect comes along with a negative bias. It can be observed that the bias is rising with the random cluster effect ρ.

Introducing within cluster correlation via a copula as is done for all simulation exercises in Section 6 does not introduce a bias. This can be observed in Figure 6, where estimators are unbiased for the response distributions and thus no further deteriorate the coverage.

Affiliation:

Susanne Berger, Achim Zeileis
Department of Statistics
Faculty of Economics and Statistics
Universität Innsbruck
Universitätsstr. 15
6020 Innsbruck, Austria
E-mail: `Susanne.Berger@uibk.ac.at`, `Achim.Zeileis@R-project.org`
URL: `https://www.uibk.ac.at/statistics/personal/berger/`, `https://eeecon.uibk.ac.at/~zeileis/`

Nathaniel Graham
Department of Finance and Decision Sciences
Trinity University Texas
One Trinity Place
San Antonio, Texas 78212, United States of America
E-mail: `npgraham1@npgraham1.com`
URL: `https://sites.google.com/site/npgraham1/`

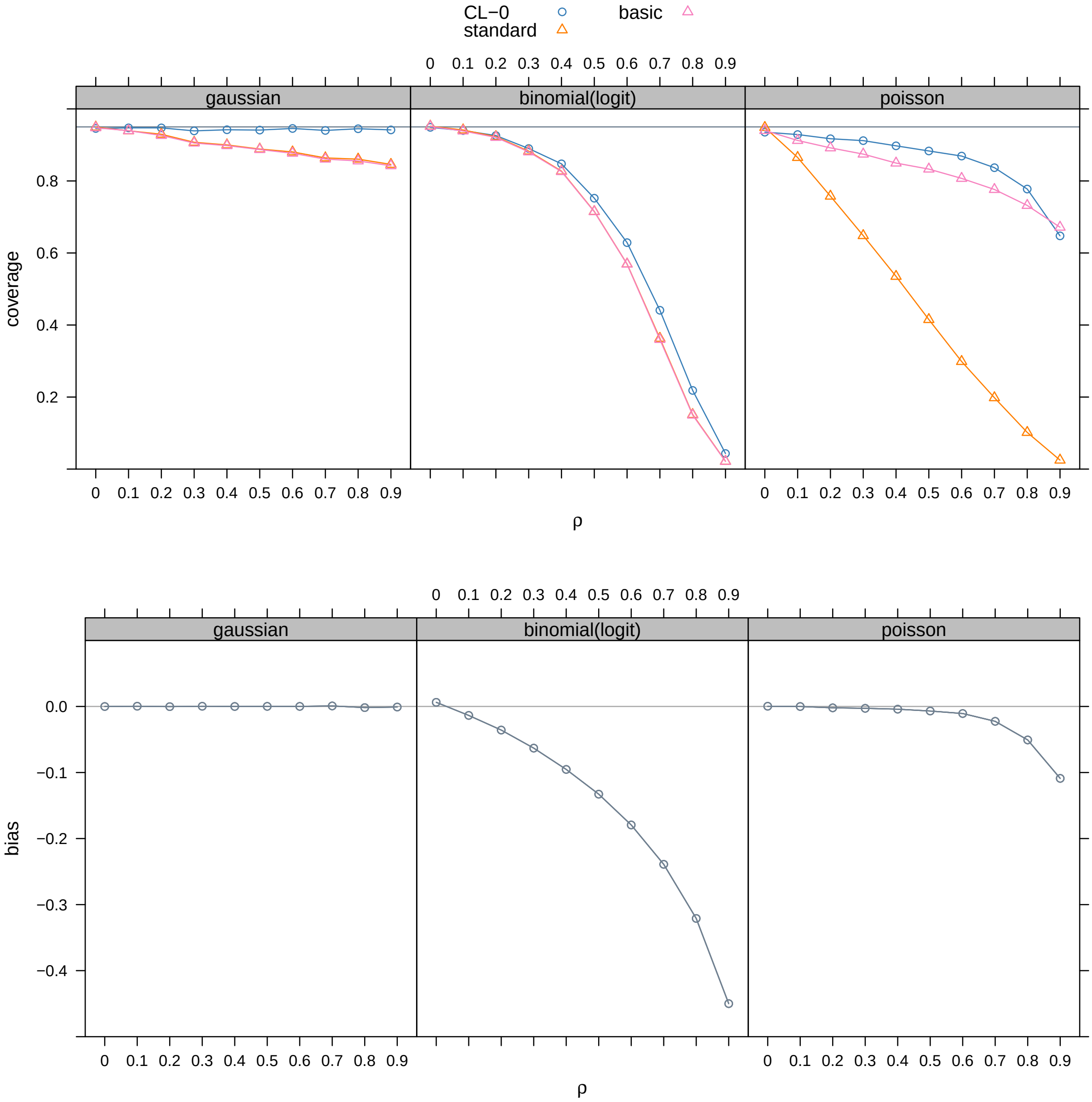


Figure 5: Introducing correlation via a random cluster effect

Explores how various types of (clustered) covariances perform for Gaussian, binomial and Poisson responses, where the data are composed of $G = 100$ (balanced) clusters each with $N_g = 5$ observations. The simulation is composed of $10,000$ replications. There is only a single regressor included in the model, `x1` is composed of a linear combination of random draw at cluster level and a random draw at individual level with cluster correlation $\rho_x = 0.25$. The coverage is plotted on the y-axis in the upper panel, the bias in the lower panel. The cluster correlation ρ is plotted on the x-axis in both panels.

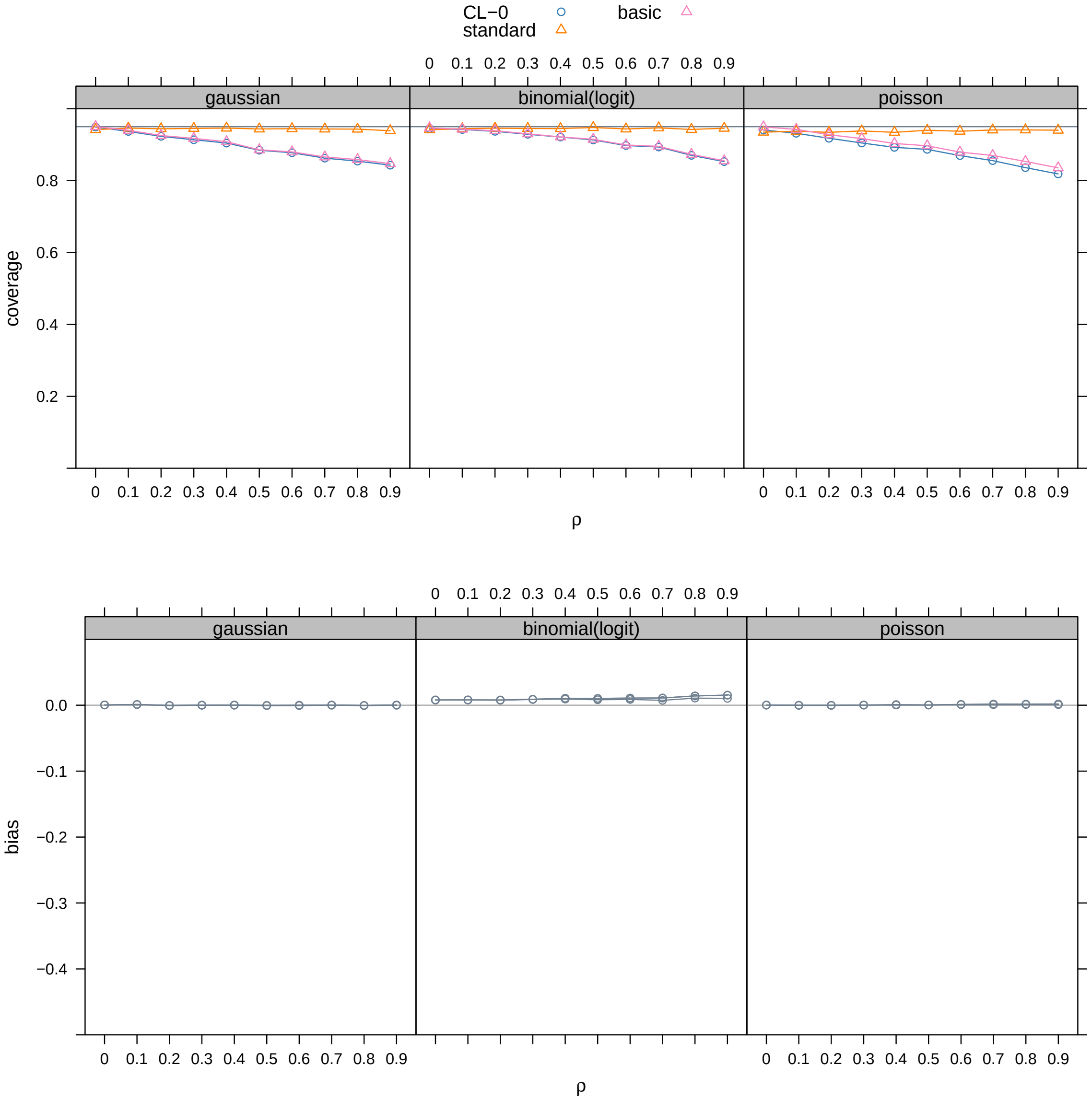


Figure 6: Introducing correlation via a normal copula

Explores how various types of (clustered) covariances perform for Gaussian, binomial and Poisson responses, where the data are composed of $G = 100$ (balanced) clusters each with $N_g = 5$ observations. The simulation is composed of $10,000$ replications. There is only a single regressor included in the model, `x1` is composed of a linear combination of random draw at cluster level and a random draw at individual level with cluster correlation $\rho_x = 0.25$. The coverage is plotted on the y-axis in the upper panel, the bias in the lower panel. The cluster correlation ρ is plotted on the x-axis in both panels.