Journal of
open research software

CESER: An R Package to Compute Cluster Estimated Standard Errors

SOFTWARE METAPAPER

DIOGO FERRARI

]u[ubiquity press

ABSTRACT

This paper presents an implementation in R of the Cluster Estimated Standard Errors (CESE) proposed by [12]. The method estimates the covariance matrix of the estimated coefficients of linear models in grouped data sets with correlation among observations within groups. Cluster Estimated Standard Errors (CESE) is an alternative solution for the classical Cluster Robust Standard Errors (CRSE) [8, 13, 15], which underestimates the standard errors in most of the situations encountered in practice [7].

CORRESPONDING AUTHOR:

Diogo Ferrari

Assistant Professor, Department of Political Science, University of California, Riverside, US

diogo.ferrari@ucr.edu

KEYWORDS:

Clustered robust standard errors; Clustered data; Confidence Intervals; Regression Analysis

TO CITE THIS ARTICLE:

Ferrari D 2021 CESER: An R Package to Compute Cluster Estimated Standard Errors. *Journal of Open Research Software*, 9: 32. DOI: *https://doi.org/10.5334/jors.355*

(1) OVERVIEW

INTRODUCTION

A common problem in regression analysis that requires correction of the estimated standard errors of the regression coefficients is the correlation between the residuals in observations that share some observed grouping features. For instance, people that live in the same city, state, or country can display a more similar behaviour than people randomly sampled from different cities, states, or countries. The example extends for any data in which some observations have shared characteristics or belong to the same collective entity or institutional setting. For instance, people from the same school, patients from the same hospital, or groups of the same gender or race can behave more similarly than people across those groups. The within-group correlation can be caused by unobserved shared characteristics of the observations in the groups, such as some unobserved school-specific educational policies, or the unobserved patterns of behavior of doctors in different hospitals.

Non-zero within-group correlations violate a common assumption of classical multivariate regression models, namely that the residuals are independent, or simply uncorrelated. If one mistakenly assumes the residuals are independent/uncorrelated, the estimated standard errors of the regression coefficients will be biased downward, which leads to smaller estimated confidence intervals, and therefore higher chances to reject the hypothesis that the coefficients are null. It can misguide researchers and lead them to be overconfident that their working hypothesis of non-zero effect is true. We can see that easily with a simple example.

Suppose we estimate the following population regression model:

$$y = X\beta + \varepsilon$$

where $X \in (1, \mathbb{R}^k)$, $\beta \in \mathbb{R}^{(k+1)\times 1}$, $y \in \mathbb{R}$, and the last element is the error (or deviance) term $\varepsilon \in \mathbb{R}$. We collect $i = 1, \ldots, n$ observations to estimate β, which gives the statistical equation for each i with the following residuals e:

$$y_i = X_i\beta + e_i.$$

We usually take X as given (measured without error) and use the OLS estimator $\hat{\beta}$ of β, which is obtained by finding the argument that minimizes the square residuals (e) between observed outcome (y) and the outcome if no error had occurred ($X\beta$):

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(e^T e) = \underset{\beta}{\operatorname{argmin}}(y - X\beta)^T(y - X\beta).$$

Assuming X^TX is invertible, the first order condition gives the solution for that optimization problem:

$$\hat{\beta} = (X^TX)^{-1}X^Ty.$$

Up to this point, if we were simply computing an OLS point estimate of β using $\hat{\beta}$, no assumptions would be needed about the distribution of the residuals (e_i). We impose assumptions about the distribution of e to go one step further and make inferences about $\hat{\beta}$ and investigate its statistical properties.[1] The distribution of our estimator $\hat{\beta}$, and therefore our inferences, comes from the assumptions about the distribution of e. Denote that distribution generically by $f(e \mid \theta)$, that is:

$$e \sim f(e \mid \theta).$$

We can easily derive the first and second moments of $\hat{\beta}$:

$$\hat{\beta} = \left(X^TX\right)^{-1}X^Ty = \left(X^TX\right)^{-1}X^T(X\beta + e) = \beta + \left(X^TX\right)^{-1}X^Te$$

which gives:

$$\mu_\beta = \mathbb{E}[\hat{\beta} \mid X, \theta] = \beta + (X^TX)^{-1}X^T\mathbb{E}[e \mid \theta] \tag{1}$$

and

$$\Sigma_{\hat{\beta}} = \mathbb{V}\mathrm{ar}[\hat{\beta} \mid X, \theta] = (X^TX)^{-1}X^T\mathbb{V}\mathrm{ar}[e \mid \theta]X(X^TX)^{-1}. \tag{2}$$

Assumptions about $f(e \mid \theta)$ will give the small sample properties of the estimator $\hat{\beta}$. The classical assumption is that all residuals e comes from the same normal distribution with mean zero, and that they are uncorrelated. That is:

$$e \sim \mathcal{N}(0, \sigma^2 I) \tag{3}$$

If we assume that $\mathbb{E}[e \mid \theta] = 0$, as in the expression (3), then $\hat{\beta}$ is unbiased ($\mathbb{E}[\hat{\beta} \mid X, \theta] = \beta$), and its standard error is simply:

$$se(\hat{\beta}) = \sqrt{(X^TX)^{-1}\hat{\sigma}^2} \tag{4}$$

with the estimated variance of e given by [8]:

$$\hat{\sigma}^2 = \frac{(y - X\hat{\beta})^T(y - X\hat{\beta})}{n - (K+1)}.$$

Equation (4) provides the exact confidence interval for $\hat{\beta}$:

$$CI[\hat{\beta}] = (\hat{\beta} - t * \mathrm{se}(\hat{\beta}), \hat{\beta} + t * \mathrm{se}(\hat{\beta})). \tag{5}$$

In the expression (5), the value of t comes from a t-distribution and it is given by:

$$p(T < |t|) = 1 - \alpha.$$

The common practice is to choose $\alpha = 0.05$, which gives the 95% confidence interval of $\hat{\beta}$.

The standard output of the `lm()` function to estimate linear models in R assumes the zero-mean normal distribution with uncorrelated residuals, which gives the estimated standard errors shown in equation (4) above [21, 3, 18].

The clustering problem emerges in grouped data. Consider that each observation *i* belongs to a group *g* that there are *G* groups in the data; and that the error terms, (*e*), for individual observations in the same group are correlated. Following the examples above, let us say that multiple observations come from the same schools, hospitals, or countries. It is likely that the assumption of independence of the residuals is violated because individuals of the same group probably share some unobserved characteristics that affect their behavior, which creates a non-zero correlation between the residuals *within* the observed groups. Then, keeping all the other assumptions of the classical regression model, the distribution of the disturbances can be more generally denoted by:

$$e \sim \mathcal{N}(0, \Sigma).$$

In this case the standard errors of $\hat{\beta}$ under the assumption of independence or zero correlation of the residuals ($se(\hat{\beta})$) differ from the standard errors computed when the within-group correlations are taken into account ($se_g(\hat{\beta})$):

$$se(\hat{\beta}) = \sqrt{(X^TX)^{-1}\hat{\sigma}^2} \neq \sqrt{(X^TX)^{-1}(X^T\hat{\Sigma}X)(X^TX)^{-1}} = se_g(\hat{\beta})$$

Typically, $se(\hat{\beta}) < se_g(\hat{\beta})$. It means that assuming uncorrelated residuals produces confidence intervals of $\hat{\beta}$ that are smaller than the true ones, and that the researcher will be overconfident about the range of values of the linear coefficients that seem consistent with the data.

There are some approaches to deal with that problem. One is to adjust the confidence intervals. Imbens and Kolesar [11] adjust the number of the degree of freedom of the *t*-distribution, producing larger values of *t* used to construct the confidence intervals. Another approach uses bootstrap methods [2, 9, 15, 19] (see also [17]). For lack of space, below we review briefly only two other approaches, the Cluster Robust Standard Errors (CRSE), which is widely-used by practitioners, and the Cluster Estimated Standard Errors (CESE) proposed by Jackson [12], whose implementation in R is originally presented in this paper, alongside an applied example and a brief discussion of cases in which one of these two methods, CRSE or CESE, may be preferred.

CLUSTERED STANDARD ERRORS CORRECTIONS

Cluster Robust Standard Errors (CRSE)

The CRSE is the routine solution used by researchers to deal with the estimation of clustered standard errors in grouped data [5, 20, 13, 7]. If the individual-level observations are divided into groups *g* (e.g., schools, countries, etc.), and $g = 1, ..., G$, we can rewrite the estimated variance of $\hat{\beta}$ in equation (2) as:

$$\hat{\Sigma}_\beta = (X^TX)^{-1}\left[\sum_{g=1}^{G} X_g^T\hat{\Sigma}_g X_g\right](X^TX)^{-1} \tag{6}$$

The key problem is how to estimate $\hat{\Sigma}_g$, the variance-covariance matrix of the residuals for group *g*. The CRSE solution is to use the raw estimated residuals from the OLS estimates of β, and compute $\hat{\Sigma}_g$ using y_g and X_g, the output variable and the covariates, respectively, of observations in group *g*. It gives the CRSE estimator $\hat{\Sigma}_g^{CRSE}$ as follows:

$$\hat{\Sigma}_g^{CRSE} = (y_g - X_g\hat{\beta})(y_g - X_g\hat{\beta})^T = e_g e_g^T$$

In practice, to compute the CRSE we don't need to estimate Σ_g. We just need to compute the covariance matrix of the scores $\hat{s}_g = X_g^T e_g$ for each group *g*, and use $X_g^T\hat{\Sigma}_g X_g = X_g^T e_g e_g^T X_g$. The R package `sandwich` provides some functions to estimate clustered standard errors using the CRSE solution [22], and the package `clubSandwich` provides many other functionalities, including some to improve performance with small samples [16].

Djogbenou et al. [4] demonstrate the asymptotic validity under general conditions for the CRSE solution. Some limits include poor reliability of the estimated errors if the number of clusters is small and sensitivity both to heterogeneity across clusters and variability of cluster sizes. Djogbenou et al. [4] provide an extensive treatment of the topic. The CRSE can be biased downward for small samples and possibly for large samples as well and seriously underestimate the true standard errors in many cases [15, 7, 14]. Jackson [12] also shows other conditions that lead the $\hat{\Sigma}_g^{CRSE}$ to provide values that underestimate the true Σ_β, and therefore the confidence intervals of the regression coefficients. The author proposes an alternative approach to estimate Σ_g called CESE, which I discuss next.

Cluster Estimated Standard Errors (CESE)

Jackson [12] proposes an approach labeled CESE to estimate the standard errors in grouped data with within-group correlation in the residuals. The approach is based on the estimated expectation of the product of the residuals. Assuming that the residuals have the same variance-covariance matrix within the groups, if we denote by $\sigma_{ig} = \sigma_g^2$ and $\rho_{ig} = \rho_g$ the variance and the covariance, respectively, of the residuals within the group *g*, then the expectation of the product of the residuals is given by (see [12] for details):

$$\Sigma_g = \mathbb{E}[e_g e_g^T] = \sigma_g^2(I_g - P_g) + \rho_g\Big[\iota_g\iota_g^T - (I_g - P_g) - (P_g\iota_g\iota_g^T + \iota_g\iota_g^T P_g) + X_g(X^TX)^{-1}\left(\sum_{g=1}^{G} X_g^T\iota_g\iota_g^T X_g\right)(X^TX)^{-1}X_g\Big] \quad (7)$$

where ι_g is a unitary column vector, I_g is a g × g identity matrix, and $P_g = X_g\left(X^TX\right)^{-1}X_g^T$. Equation (7) can be rewriten concisely as:

$$\Sigma_g = \sigma_g^2 Q_{1g} + \rho_g Q_{2g}. \quad (8)$$

The equation above explicitly shows that the expectation of the cross-product of the residuals is a function the data through Q_{1g} and Q_{2g} and the unknown variance σ_g^2 and correlation ρ_g of the residuals e_g in each group *g*. The CESE solution is to explore the linear structure of equation (8) and to estimate σ_g^2 and ρ_g *as if* the estimated values of $e_g e_g^T$ were random deviances from their expectations. Denote ξ that deviance. Then

$$\begin{aligned} e_g e_g^T &= \mathbb{E}[e_g e_g^T] + \xi \\ &= \sigma_g^2 Q_{1g} + \rho_g Q_{2g} + \xi \\ &= \Sigma_g + \xi. \end{aligned} \quad (9)$$

The estimates of σ_g^2 and ρ_g are obtained using the OLS estimator. That is, if we denote $\Omega_g = (\sigma_g^2, \rho_g)^T$, q_{1g} (or q_{2g}) the vectorized diagonal and lower triangle of Q_{1g} (or Q_{2g}) stacked into a $n_g(n_g + 1)/2$ column vector, $q_g = [q_{1g}, q_{2g}]$, and s_{eg} the corresponding elements of $e_g e_g^T$ stacked into a column vector as well, then the OLS CESE estimator $\hat{\Omega}_g = (\hat{\sigma}_g^2, \hat{\rho}_g)^T$ of the variance and correlation of the residuals in group *g* is given by

$$\hat{\Omega}_g = \underset{\Omega_g}{\operatorname{argmin}} (s_{eg} - q_g\Omega_g)^T(s_{eg} - q_g\Omega_g).$$

As pointed above for the OLS estimator of β, if we assume that $q_g^T q_g$ is invertible, the first order condition gives:

$$\hat{\Omega}_g = (q_g^T q_g)^{-1} q_g^T s_{eg}. \quad (10)$$

We can rewrite the equation (10) as:

$$\begin{bmatrix} \hat{\sigma}_g^2 \\ \hat{\rho}_g \end{bmatrix} = \begin{bmatrix} q_{1g}^T q_{1g} & q_{1g}^T q_{2g} \\ q_{2g}^T q_{1g} & q_{2g}^T q_{2g} \end{bmatrix}^{-1} \begin{bmatrix} q_{1g}^T s_{eg} \\ q_{2g}^T s_{eg} \end{bmatrix}. \quad (11)$$

As explained above for the OLS estimates of β, the estimators of σ_g^2 and ρ_g do not require *per se* any assumption on ξ, unless we want to construct confidence intervals for the estimates of those parameters.

The CESE is attractive when its assumptions hold and the CRSE is believed to be unreliable. Jackson [12] shows that CESE produces larger standard errors for the coefficients and much more conservative confidence intervals than the CRSE, which is known to be biased downward in the cases mentioned above. CESE is also less sensitive to the number of clusters and to the heterogeneity of the clusters, which can be a problem for both CRSE and bootstrap methods.

However, it is important to notice, that the CESE is not a replacement for the CRSE because these two methods are based on different parametric assumptions. The CESE requires some assumptions that can be considered stronger than the CRSE approach, as equations (7) to (11) indicate (see more details in [12]). Each approach may be better suited to different situations. One example is the CESE assumption that the residuals have the same variance-covariance matrix within the groups. For instance, if we cluster by geographic location, but individual data is observed at different points in time as in Bertrand et al. [1], then the assumption of the same within-cluster residual variation is probably violated, and we would have to cluster the standard errors by time as well. Another example: When one uses fixed-effect models for the clusters, and the correlation of the residuals comes only from cluster-level effect, the cluster fixed effects explain all the variation in at the cluster-level, and the term ρ_g will be close to zero. In that case, CESE may be a less appealing alternative. However, when the limitations of the CRSE discussed above are a problem, the CESE is a better choice and produces more conservative standard errors.

I implemented CESE in R. It is available in the package named `ceser`. The next section presents some details of the implementation as well as an example ilustrating how to use the software in practice.

IMPLEMENTATION AND ARCHITECTURE

Computing the CESE

The package `ceser` provides a function `vcovCESE()` that takes the output of the function `lm()` (or any other that produces compatible outputs) and computes the Cluster Estimated Standard Errors (CESE). The basic structure of the function is:

```
R> vcovCESE(mod, cluster = NULL, type=NULL)
```

The parameter `mod` receives the output of the `lm()` function. The parameter `cluster` can receive a right-hand side R formula with the summation of the variables in the data that will be used to cluster the standard errors. For instance, if one wants to cluster the standard errors by country, one can use:

```
R> vcovCESE(..., cluster = ~ country, ...)
```

To cluster by country and gender, simply use (note that it means that each cluster contains observation for one gender and one country):

```
R> vcovCESE(..., cluster = ~ country + gender, ...)
```

The parameter `cluster` can also receive, instead of a formula, a string vector with the name of the variables that contain the groups to cluster the standard errors. If `cluster` = `NULL`, each observation is considered its own group to cluster the standard errors.

The parameter `type` receives the procedure to use for heterokedasticity correction. Heterokedasticity occurs when the diagonal elements of Σ are not constant across observations. The correction can also be used to deal with underestimation of the true variance of the residuals due to leverage produced by outliers. The package includes five types of correction. In particular, `type` can be either "HC0", "HC1", "HC2", "HC3", and "HC4" [10]. Denote e_c the corrected residuals. Each option produce the following corretion:

$$\text{HC0:} \quad e_{ic} = e_i$$

$$\text{HC1:} \quad e_{ic} = e_i \left(\sqrt{\frac{n}{n-k}} \right)$$

$$\text{HC2:} \quad e_{ic} = e_i \left(\frac{1}{\sqrt{1-h_{ii}}} \right)$$

$$\text{HC3:} \quad e_{ic} = e_i \left(\frac{1}{1-h_{ii}} \right)$$

$$\text{HC4:} \quad e_{ic} = e_i \left(\frac{1}{\sqrt{(1-h_{ii})^{\delta_i}}} \right)$$

where k is the number of covariates, h_{ii} is the i^{th} diagonal element of the matrix $P = X(X^TX)^{-1}X^T$), and $\delta_i = \min(4, h_{ii}\frac{n}{k})$.

The estimation also corrects for cases in which $\rho_g > \sigma^2 g$. Following Jackson [12], we use $\hat{\sigma}_g^2 = (\hat{\rho}_g + 0.02)$ in those cases.

Example with application

In applied regression analyses, the practioner is usually interested in estimating the linear coefficients and their standard errors to evaluate if the confidence interval of the point estimates of the coefficients includes the null value. It means that two quantites of interest are $\hat{\beta}$ and se($\hat{\beta}$).

In this section, we compare the standard output of the `lm()` function with the standard errors of the linear coefficients produced by the CRSE, as computed by the widely used R package `sandwich` [22], and those produced by the `ceser` package, which contains my implementation of the CESE method proposed by Jackson [12]. As discussed in the previous section, in general the CESE should be more conservative, produce larger estimates of the standard errors, and result in wider confidence intervals.

To ilustrate how to use the `ceser` package, and to compare the three estimates of the standard errors (raw, CRSE, and CESE), we use the data set `dcese` provided with the `ceser` package. The data set was used in Jackson [12] and comes from Elgie et al. [6]. It contains information of 310 (i = 1, ..., 310) observations across 51 countries (g = 1, ..., 51). The outcome variable is the number of effective legislative parties (`enep`). The explanatory variables are: the number of presidential candidates (`enpc`); a measure of presidential power (`fapres`); the proximity of presidential and legislative elections (`proximity`); the effective number of ethnic groups (`eneg`); the log of average district magnitudes (`logmag`); an interaction term between the number of presidential candidates and the presidential power (`enpcfapres` = `enpc` × `fapres`), and another interaction term between the log of the district magnitude and the number of ethnic groups (`logmag_eneg` = `logmag` × `eneg`). Elgie et al. [6] present regression analyses showing a strong relationship between `enpc` and `fapres`, `enpc`, and their interaction. The effective number of legislative parties increases with the number of presidential candidates, but decreases with presidential power. The interactive term has a positive coefficient, implying the negative association between the number of legislative parties and presidential power attenuates as the number of candidates increases. They use a variety of standard errors corrections, including CRSE. We reproduce their study here, and include the estimation of the standard errors using CESE as in Jackson [12].

Let us start with the functions that provide the variance covariance matrix of the estimated coefficients $\hat{\beta}$. For all the examples below, we use the HC3 correction. The ***Table 1*** below uses also HC1 for comparison. Let us start by loading the package and the data:

```
R> library(ceser)
R> data(dcese)
```

Before estimating the linear model, we need to sort the data using the cluster variables (this is necessary to estimate the CESE using the `ceser` package, but it is not necessary to estimate the CRSE using the `sandwish` package). In our example, we will cluster the data by country. Hence:

```
R> dcese = dcese[order(df$country), ]
```

Estimate the linear model using the `lm()` function.

```
R> mod = lm(enep ~ enpc + fapres + enpcfapres
                 + proximity + eneg + logmag
                 + logmag_eneg , data=dcese)
```

The estimated raw values of the variance covariance matrix obtained by running the standard R function from the `stats` package [18] are:

```
R> vcov (mod)
```

	(Intercept)	enpc	fapres	enpcfapres	proximity
(Intercept)	0.34193	-0.080109	-0.06498717	0.0227605	-0.0416369
enpc	-0.08011	0.035697	0.02401318	-0.0102825	0.0059204
fapres	-0.06499	0.024013	0.02734250	-0.0090018	-0.0004345
enpcfapres	0.02276	-0.010283	-0.00900179	0.0036430	-0.0014388
proximity	-0.04164	0.005920	-0.00043452	-0.0014388	0.0776196
eneg	-0.03580	-0.001477	-0.00251785	0.0007025	-0.0039084
logmag	-0.05448	-0.006981	0.00017420	0.0021400	-0.0023836
logmag_eneg	0.02532	0.001833	-0.00007513	-0.0007721	-0.0009086

	eneg	logmag	logmag_eneg
(Intercept)	-0.0358050	-0.0544826	0.02532042
enpc	-0.0014768	-0.0069809	0.00183259
fapres	-0.0025179	0.0001742	-0.00007513
enpcfapres	0.0007025	0.0021400	-0.00077214
proximity	-0.0039084	-0.0023836	-0.00090860
eneg	0.0218856	0.0222887	-0.01190289
logmag	0.0222887	0.0606796	-0.02995518
logmag_eneg	-0.0119029	-0.0299552	0.01778317

The CRSE, using countries as the grouping variable, obtained using the `vcovCL()` function of the `sandwich` package [22] are:

```
R> library(sandwich)
R> vcovCL(mod, cluster = ~country, type="HC3")
```

	(Intercept)	enpc	fapres	enpcfapres	proximity
(Intercept)	0.376409	-0.0929549	-0.06620	0.022499	-0.0315432
enpc	-0.092955	0.0930327	0.05081	-0.026847	0.0000196
fapres	-0.066198	0.0508080	0.07437	-0.024184	-0.0177849
enpcfapres	0.022499	-0.0268474	-0.02418	0.010785	0.0020836
proximity	-0.031543	0.0000196	-0.01778	0.002084	0.1029317
eneg	0.001905	-0.0165885	-0.02183	0.007097	-0.0200007
logmag	-0.030573	-0.0642203	-0.04945	0.022924	-0.0285040
logmag_eneg	-0.002075	0.0124010	0.02094	-0.007229	0.0317879

	eneg	logmag	logmag_eneg
(Intercept)	0.001905	-0.03057	-0.002075
enpc	-0.016589	-0.06422	0.012401
fapres	-0.021832	-0.04945	0.020940
enpcfapres	0.007097	0.02292	-0.007229
proximity	-0.020001	-0.02850	0.031788
eneg	0.027519	0.06041	-0.039241
logmag	0.060413	0.27344	-0.158061
logmag_eneg	-0.039241	-0.15806	0.120629

In a similar fashion, the CESE are obtained by simply running the function `vcovCESE()` of the `ceser` package:

```
R> vcovCESE(mod, cluster = ~country, type="HC3")
```

```
               (Intercept)         enpc        fapres    enpcfapres    proximity
(Intercept)        1.59804   -0.3565890     -0.326045     0.0928614   -0.086959
enpc              -0.35659    0.1254735      0.104834    -0.0354704   -0.003333
fapres            -0.32604    0.1048342      0.143206    -0.0389794   -0.017879
enpcfapres         0.09286   -0.0354704     -0.038979     0.0126978    0.003218
proximity         -0.08696   -0.0033328     -0.017879     0.0032179    0.139695
eneg              -0.08737    0.0028258     -0.007081     0.0010940   -0.005680
logmag            -0.22422    0.0009845      0.006688     0.0038080    0.009776
logmag_eneg        0.08381   -0.0058250     -0.011500     0.0008569    0.004472
                      eneg       logmag   logmag_eneg
(Intercept)     -0.087372   -0.2242235     0.0838093
enpc             0.002826    0.0009845    -0.0058250
fapres          -0.007081    0.0066880    -0.0115004
enpcfapres       0.001094    0.0038080     0.0008569
proximity       -0.005680    0.0097761     0.0044718
eneg             0.039433    0.0481561    -0.0231003
logmag           0.048156    0.2244237    -0.1048418
logmag_eneg     -0.023100   -0.1048418     0.0606626
```

Note that the estimated standard errors are ordered as expected. The raw standard errors are smaller than CRSE, which by its turn are smaller than CESE for almost all coefficients:

The standard errors for each method are:

```
R> sqrt(diag(vcov(mod)))

 (Intercept)       enpc         fapres   enpcfapres   proximity
     0.58475    0.18894        0.16536      0.06036     0.27860
        eneg     logmag    logmag_eneg
     0.14794    0.24633        0.13335

R> sqrt(diag(vcovCL(mod, cluster=~country, type="HC3")))

 (Intercept)       enpc         fapres   enpcfapres   proximity
      0.6135     0.3050         0.2727       0.1039      0.3208
        eneg     logmag    logmag_eneg
      0.1659     0.5229         0.3473

R> sqrt(diag(vcovCESE(mod, cluster=~country, type="HC3")))

 (Intercept)       enpc         fapres   enpcfapres    proximity
      1.2641     0.3542         0.3784       0.1127       0.3738
        eneg     logmag    logmag_eneg
      0.1986     0.4737         0.2463
```

Summary tables with the raw standard errors, CRSE, and CESE are easy to produce. The package `lmtest` is specially useful for that purpose. The package `ceser` integrates nicely with the `lmtest` package and the function `coeftest()` of that package, which can be used to create summary tables with the different standard errors. The raw estimates are:

```
R> summary(mod)

Call:
lm (formula = enep ~ enpc + fapres + enpcfapres + proximity +
     eneg + logmag + logmag_eneg, data = dcese)

Residuals:
      Min       1Q   Median       3Q      Max
   -3.559   -0.819   -0.361    0.377    9.039
```

```
Coefficients:

                Estimate  Std.Error  t value    Pr (>|t|)
 (Intercept)       2.7043     0.5848     4.62   0.0000056  ***
 enpc              0.3040     0.1889     1.61     0.10871
 fapres           -0.6118     0.1654    -3.70     0.00026  ***
 enpcfapres        0.2078     0.0604     3.44     0.00066  ***
 proximity        -0.0224     0.2786    -0.08     0.93589
 eneg             -0.0657     0.1479    -0.44     0.65748
 logmag           -0.1815     0.2463    -0.74     0.46193
 logmag_eneg       0.3605     0.1334     2.70     0.00727   **
  ---
codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.65 on 291 degrees of freedom
Multiple R-squared: 0.378, Adjusted R-squared: 0.363
F-statistic: 25.3 on 7 and 291 DF, p-value: <0.0000000000000002
```

We can obtain the summary with CRSE by country by running:

```
R> library(lmtest)
R> coeftest(mod, vcov = vcovCL, cluster = ~ country, type=”HC3”)

t test of coefficients:

               Estimate   Std.Error  t value  Pr(>|t|)
 (Intercept)      2.7043     0.6135     4.41  0.000015  ***
 enpc             0.3040     0.3050     1.00     0.320
 fapres          -0.6118     0.2727    -2.24     0.026    *
 enpcfapres       0.2078     0.1039     2.00     0.046    *
 proximity       -0.0224     0.3208    -0.07     0.944
 eneg            -0.0657     0.1659    -0.40     0.693
 logmag          -0.1815     0.5229    -0.35     0.729
 logmag_eneg      0.3605     0.3473     1.04     0.300
  ---
codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Similary, to use CESE instead of CRSE, simply run

```
R> coeftest(mod, vcov = vcovCESE, cluster = ~ country, type=”HC3”)

t test of coefficients:

                Estimate  Std.Error  t value   Pr (>|t|)
 (Intercept)       2.7043     1.2641     2.14       0.033  *
 enpc              0.3040     0.3542     0.86       0.391
 fapres           -0.6118     0.3784    -1.62       0.107
 enpcfapres        0.2078     0.1127     1.84       0.066  .
 proximity        -0.0224     0.3738    -0.06       0.952
 eneg             -0.0657     0.1986    -0.33       0.741
 logmag           -0.1815     0.4737    -0.38       0.702
 logmag_eneg       0.3605     0.2463     1.46       0.144
  ---
codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 1 shows how the confidence intervals differ for the different estimates of the standard error of the coefficients. The CRSE are shown with both the HC_1 and HC_3 adjustments to the residuals. We can see how the CESE is more conservative, particulary for the two covariates, `fapres` (presidential power) and `enpcfapres` [the interaction between effective number of legislative parties (enpc) and presidential power (fapres)]. For them, the null value is consistent with the data when the CESE is used, but not if the other standard errors are adopted for the computation of the confidence intervals.

The user should note that the performance of the estimation is not yet optimized to handle large data sets. There are two reasons for the suboptimal performance. The first is that the current implementation uses only high-level functions in R. The second is that the software avoids storing large matrices by using some nested loops during the estimation. Future versions of the package will implement the core functions in C++ and provide compiled code with the package to improve the performance. Nevertheless, the package is fully functional, and the performance tests shows that on average the a data set with 1000 observations take 5.3 seconds to estimate, with 3000 it takes 85 seconds, and with 5000 observations it takes around 5.5 minutes to compute the standard errors.

QUALITY CONTROL

The package has been thoroughly quality checked and tested. The package structure successfully passes all CRAN R CMD checks and all continuous integration checks implemented in Travis, including checks to build the package on Windows, Linux, and macOS. The results of the checks can be found on Travis website *https://travis-ci.org/github/DiogoFerrari/ceser*.

(2) AVAILABILITY

OPERATING SYSTEM

CESER is written in R (>=2.1) and run in any operational system that supports R Statistical Software. R can be obtained freely from *https://www.r-project.org/*.

PROGRAMMING LANGUAGE

R Statistical Software 2.1 or higher.

ADDITIONAL SYSTEM REQUIREMENTS

There is no additional requirements.

COVARIATE	STD. ERRORS				
	ESTIMATE	RAW	$CRSE_{HC1}$	$CRSE_{HC3}$	CESE
(Intercept)	2.7043	0.5848	0.4886	0.6135	1.2641
enpc	0.3040	0.1889	0.2517	0.3050	0.3542
fapres	−0.6118	0.1654	0.2038	0.2727	0.3784
enpcfapres	0.2078	0.0604	0.0826	0.1039	0.1127
proximity	−0.0224	0.2786	0.2544	0.3208	0.3738
eneg	−0.0657	0.1479	0.1415	0.1659	0.1986
logmag	−0.1815	0.2463	0.4387	0.5229	0.4737
logmag_eneg	0.3605	0.1334	0.2883	0.3473	0.2463
COVARIATE	**CONFIDENCE INTERVALS**				
	ESTIMATE	**RAW**	**$CRSE_{HC1}$**	**$CRSE_{HC3}$**	**CESE**
(Intercept)	2.7043	(1.558, 3.85)	(1.747, 3.662)	(1.502, 3.907)	(0.227, 5.182)
enpc	0.3040	(−0.066, 0.674)	(−0.189, 0.797)	(−0.294, 0.902)	(−0.39, 0.998)
fapres	−0.6118	(−0.936, −0.288)	(−1.011, −0.212)	(−1.146, −0.077)	(−1.354, 0.13)
enpcfapres	0.2078	(0.089, 0.326)	(0.046, 0.37)	(0.004, 0.411)	(−0.013, 0.429)
proximity	−0.0224	(−0.568, 0.524)	(−0.521, 0.476)	(−0.651, 0.606)	(−0.755, 0.71)
eneg	−0.0657	(−0.356, 0.224)	(−0.343, 0.212)	(−0.391, 0.259)	(−0.455, 0.324)
logmag	−0.1815	(−0.664, 0.301)	(−1.041, 0.678)	(−1.206, 0.843)	(−1.11, 0.747)
logmag_eneg	0.3605	(0.099, 0.622)	(−0.205, 0.926)	(−0.32, 1.041)	(−0.122, 0.843)

Table 1 Comparing raw standard errors, CRSE, and CESE.

DEPENDENCIES

The package depends on the following R packages: magrittr, purrr, dplyr, tibble, lmtest.

LIST OF CONTRIBUTORS

- Diogo Ferrari, Department of Political Science, University of California, Riverside
- John E. Jackson, Department of Political Science, University of Michigan, Ann Arbor

SOFTWARE LOCATION

Archive

Name: Cluster Estimated Standard Errors in R (CESER)
Persistent identifier: *10.5281/zenodo.4107151*
Licence: MIT
Publisher: Diogo Ferrari
Version published: v1.0.0
Date published: 10/19/2020

Code repository

Name: ceser
Identifier: *https://doi.org/10.5281/zenodo.4107151*
Licence: MIT
Date published: 10/19/2020

LANGUAGE

English

(3) REUSE POTENTIAL

Firstly, the adoption of methods that deal with clustered standard errors is ubiquitous in social sciences. Currently, available packages in R only provide traditional ways (CRSE) to estimate regression models with clustered standard errors, as discussed above. The CESER package provides an easy-to-use implementation of a new method, namely CESER, as proposed in Jackson [12]. It is important to note that the method implemented in our package is not bounded by any specific subfield. The package is of direct interest to any researcher using regression models.

The Cluster Estimated Standard Errors in R (CESER) package is fully compatible with other R packages widely used to compute regression models in economics, psychology, political science, sociology, and many other disciplines. Those packages include the built-in R module `stats` to complete linear models, as well as some extensions such as `glm, lmtest, lme4`. Researchers using those packages can seamlessly use our package to deal with clustered standard errors. The CESER package is well-documented and contains working examples for a copy-and-paste experimentation. Moreover, code examples are provided at the package author's personal website, including a code vignette explaining the package usage. As presented in the paper, the output of the main estimation function follows standard R format and can be manipulated by popular external packages for data visualization and reports, including tidyverse, kable, pipe computing, and ggplot2. Hence, our package can easily be reused or extended.

There are three main options for those interested in extending or contributing to the package. First, we provide full open access to the source code in the package's GitHub repository. Users can either open a ticket requesting extensions or suggesting changes. They can also make changes to their local version of the code and open a pull request for software extension or modification using the GitHub website. Finally, users are welcome to e-mail to the principal author and request further enhancements.

NOTE

1 Note the assumptions about the distribution of e is needed upfront if we are deriving a maximum likelihood estimator (MLE) of β instead of the OLS estimator.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATION

Diogo Ferrari *orcid.org/0000-0003-2454-0776*
Assistant Professor, Department of Political Science, University of California, Riverside, US

REFERENCES

1. **Bertrand M, Duflo E, Mullainathan S.** How much should we trust differences-in-differences estimates? *The Quarterly journal of economics,* 2004; 119(1): 249–275. DOI: *https://doi.org/10.1162/003355304772839588*
2. **Cameron AC, Gelbach JB, Miller DL.** Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics,* 2008; 90(3): 414–427. DOI: *https://doi.org/10.1162/rest.90.3.414*
3. **Chambers JM.** Linear models. In Chambers JM, Hastie TJ (Eds.), *Statistical Models in S*, chapter 4. Wadsworth Brooks/Cole; 1992.
4. **Djogbenou AA, MacKinnon JG, Nielsen MØ.** Asymptotic theory and wild bootstrap inference with clustered errors. *Journal of Econometrics,* 2019; 212(2): 393–412. DOI: *https://doi.org/10.1016/j.jeconom.2019.04.035*

5. **Eicker F.** Limit theorems for regressions with unequal and dependent errors. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability,* 1967; 1: 59–82.
6. **Elgie R, Bucur C, Dolez B, Laurent A.** Proximity, candidates, and presidential power: How directly elected presidents shape the legislative party system. *Political Research Quarterly,* 2014; 67(3): 467–477. DOI: *https://doi.org/10.1177/1065912914530514*
7. **Esarey J, Menger A.** Practical and effective approaches to dealing with clustered data. *Political Science Research and Methods,* 2018; 1–19. DOI: *https://doi.org/10.1017/psrm.2017.42*
8. **Greene WH.** Econometric analysis. Upper Saddle River, NJ: Pearson Prentice Hall; 2012.
9. **Harden JJ.** A bootstrap method for conducting statistical inference with clustered data. *State Politics & Policy Quarterly*, 2011; 11(2): 223–246. DOI: *https://doi.org/10.1177/1532440011406233*
10. **Hayes AF, Cai L.** Using heteroskedasticity-consistent standard error estimators in ols regression: An introduction and software implementation. *Behavior research methods,* 2007; 39(4): 709–722. DOI: *https://doi.org/10.3758/BF03192961*
11. **Imbens GW, Kolesar M.** Robust standard errors in small samples: Some practical advice. *Review of Economics and Statistics*, 2016; 98(4): 701–712. DOI: *https://doi.org/10.1162/REST_a_00552*
12. **Jackson J.** Corrected standard errors with clustered data. *Political Analysis,* 2020; 28(3): 318–339. DOI: *https://doi.org/10.1017/pan.2019.38*
13. **Liang K-Y, Zeger SL.** Longitudinal data analysis using generalized linear models. *Biometrika,* 1986; 73(1): 13–22. DOI: *https://doi.org/10.1093/biomet/73.1.13*
14. **MacKinnon JG.** How cluster-robust inference is changing applied econometrics. *Canadian Journal of Economics,* 2019; 52(3): 851–881. DOI: *https://doi.org/10.1111/caje.12388*
15. **MacKinnon JG, Webb MD.** Wild bootstrap inference for wildly different cluster sizes. *Journal of Applied Econometrics,* 2017; 32(2): 233–254. DOI: *https://doi.org/10.1002/jae.2508*
16. **Pustejovsky J.** *clubSandwich: Cluster-Robust (Sandwich) Variance Estimators with Small-Sample Corrections.* R package version 0.5.3; 2021.
17. **Pustejovsky JE, Tipton E.** Small-sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models. *Journal of Business & Economic Statistics,* 2018; 36(4): 672–683. DOI: *https://doi.org/10.1080/07350015.2016.1247004*
18. **R Core Team.** *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing; 2018.
19. **Roodman D, Nielsen MØ, MacKinnon JG, Webb MD.** Fast and wild: Bootstrap inference in stata using boottest. *The Stata Journal,* 2019; 19(1): 4–60. DOI: *https://doi.org/10.1177/1536867X19830877*
20. **White HL.** A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica,* 1980; 48(4): 817–838. DOI: *https://doi.org/10.2307/1912934*
21. **Wilkinson GN, Rogers CE.** Symbolic descriptions of factorial models for analysis of variance. *Applied Statistics,* 1973; 22: 392–9. DOI: *https://doi.org/10.2307/2346786*
22. **Zeileis A.** Econometric computing with hc and hac covariance matrix estimators; 2004. DOI: *https://doi.org/10.18637/jss.v011.i10*

TO CITE THIS ARTICLE:
Ferrari D 2021 CESER: An R Package to Compute Cluster Estimated Standard Errors. *Journal of Open Research Software,* 9: 32. DOI: *https://doi.org/10.5334/jors.355*

Submitted: 19 October 2020 Accepted: 06 October 2021 Published: 30 November 2021