

An R Package to Compute Cluster Estimated Standard Error (CESE)

Diogo Ferrari

Department of Political Science
University of California, Riverside
diogo.ferrari@ucr.edu

John E. Jackson

Department of Political Science
University of Michigan, Ann Arbor
jjacksn@umich.edu

October 19, 2020

Words in text: 2502
Words in headers: 29
Words outside text (captions, etc.): 32
Number of headers: 9
Number of floats/tables/figures: 1
Number of math inlines: 122
Number of math displayed: 23

1. Abstract

This paper presents an implementation in R of the Cluster Estimated Standard Error (CESE) proposed by [Jackson \(2019\)](#). 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 Error (CRSE) ([Greene, 2012](#); [Eicker, 1967](#); [White, 1980](#); [Liang and Zeger, 1986](#); [MacKinnon and Webb, 2017](#)), which underestimates the standard errors in most of the situations encountered in practice ([Esarey and Menger, 2018](#)).

2. Introduction

A common problem in regression analysis that requires correction of the estimated standard error 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 error of the regression coefficients will be biased downward, which leads to smaller estimated confidence intervals, and therefore lower 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 $i = 1, \dots, n$ 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 $X^T X$ is invertible, the first order condition gives the solution for that optimization problem:

$$\hat{\beta} = (X^T X)^{-1} X^T y.$$

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 properties¹. 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 $f(e | \theta)$, that is:

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

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

¹Note 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.

$$\hat{\beta} = (X^T X)^{-1} X^T y = (X^T X)^{-1} X^T (X\beta + e) = \beta + (X^T X)^{-1} X^T e$$

which gives:

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

and

$$\Sigma_{\hat{\beta}} = \text{Var}[\hat{\beta} | X, \theta] = (X^T X)^{-1} X^T \text{Var}[e | \theta] X (X^T X)^{-1}. \quad (2)$$

Assumptions about $f(e | \theta)$ 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:

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

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

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

with the estimated variance of e given by (Greene, 2012):

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

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

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

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 (Wilkinson and Rogers, 1973; Chambers, 1992; R Core Team, 2018).

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's 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 ($se_g(\hat{\beta})$):

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

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 estimated 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 \(2016\)](#) 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 ([Cameron et al., 2008](#); [Harden, 2011](#); [MacKinnon and Webb, 2017](#)). These methods only correct the confidence intervals, but they do not provide an estimate for $\hat{\Sigma}$ (see [Harden \(2011\)](#) for an exception). As a result, hypothesis tests about the nullity of multiple coefficients and interactions are infeasible. For models with interaction terms, it means marginal effects plots of the interactive terms cannot be constructed because they require the full Σ_{β} matrix. Only presenting one confidence interval ignores current best practice recommendations that eschew the use of pre-set p-values, such as those derived from $\alpha = 0.05$, in favor of reporting standard errors and letting readers decide what level of uncertainty they prefer. (see [Wasserstein et al. \(2019\)](#)). This practice requires estimating and reporting the elements of Σ_{β} as computing different confidence intervals via the commonly proposed bootstrapping requires the original data.

Two methods provide an estimate for Σ , which eliminate those problems: Cluster Robust Standard Errors (CRSE) and Cluster Estimated Standard Errors (CESE). The next section reviews these two main approaches to correct the estimated standard errors of $\hat{\beta}$. The following section presents our implementation of CESE in R and compare it with the already existing implementation of CRSE. An introductory example then explores the methods and its implementation.

3. Clustered Standard Error Corrections

3.1. 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 (Eicker, 1967; White, 1980; Liang and Zeger, 1986; Esarey and Menger, 2018). If the individual-level observations are divided into groups g (e.g., schools, countries, etc.), and $g = 1, \dots, G$, we can rewrite the estimated variance of $\hat{\beta}$ in equation (2) as:

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

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 $\hat{\Sigma}_g^{\text{CRSE}}$ as follows:

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

The R package `sandwich` provides some functions to estimate clustered standard errors using the CRSE solution (Zeileis, 2004).

MacKinnon and Webb (2017) show that there are three necessary conditions for CRSE to be consistent: (a) infinite number of clusters, (b) homogeneity across clusters in the stochastic term distributions; and (c) an equal number of observations per cluster. Moreover, authors have shown that CRSE are biased downward for small samples and possibly for large samples as well (MacKinnon and Webb, 2017; Esarey and Menger, 2018). Jackson (2019) also shows other conditions that lead the $\hat{\Sigma}_g^{\text{CRSE}}$ 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 we discuss next.

3.2. Cluster Estimated Standard Errors (CESE)

Jackson (2019) 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 Jackson (2019) for details):

$$\begin{aligned} \Sigma_g = \mathbb{E}[e_g e_g^T] &= \sigma_g^2 (I_g - P_g) + \rho_g \left[\mathbf{1}_g \mathbf{1}_g^T - (I_g - P_g) - (P_g \mathbf{1}_g \mathbf{1}_g^T + \mathbf{1}_g \mathbf{1}_g^T P_g) \right. \\ &\quad \left. + X_g (X^T X)^{-1} \left(\sum_{g=1}^G X_g^T \mathbf{1}_g \mathbf{1}_g^T X_g \right) (X^T X)^{-1} X_g \right] \end{aligned} \quad (7)$$

where $\mathbf{1}_g$ is a unitary column vector, I_g is a $g \times g$ identity matrix, and $P_g = X_g(X_g^T X_g)^{-1} X_g^T$. Equation (7) can be rewritten 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.

Jackson (2019) 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. 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.

We 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 illustrating how to use the package in practice.

4. Implementation and Example

4.1. 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 error by country, one can use:

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

To cluster by country and gender, simply use

```
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. We include five types of correction. In particular, `type` can be either "HC0", "HC1", "HC2", "HC3", and "HC4" (Hayes and Cai, 2007). Denote e_c the corrected residuals. Each option produce the following correction:

$$\begin{aligned}
\text{HC0: } e_{ic} &= e_i \\
\text{HC1: } e_{ic} &= e_i \left(\sqrt{\frac{n}{n-k}} \right) \\
\text{HC2: } e_{ic} &= e_i \left(\frac{1}{\sqrt{1-h_{ii}}} \right) \\
\text{HC3: } e_{ic} &= e_i \left(\frac{1}{1-h_{ii}} \right) \\
\text{HC4: } e_{ic} &= e_i \left(\frac{1}{\sqrt{(1-h_{ii})^{\delta_i}}} \right)
\end{aligned}$$

where k is the number of covariates, h_{ii} is the i^{th} diagonal element of the matrix $P = X(X^T X)^{-1} X^T$, 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 \(2019\)](#), we use $\hat{\sigma}_g^2 = (\hat{\rho}_g + 0.02)$ in those cases.

4.2. Example with application

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

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` ([Zeileis, 2004](#)), and those produced by the `ceser` package, which contains our implementation of the CESE method proposed by [Jackson \(2019\)](#). 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 illustrate how to use the `ceser` package, and to compare the three estimates of the standard error (raw, CRSE, and CESE), we use the data set `dcese` provided with the `ceser` package. The data set was used in [Jackson \(2019\)](#) and comes from [Elgie et al. \(2014\)](#). It contains information of 310 ($i=1, \dots, 310$) observations across 51 countries ($g=1, \dots, 51$). 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. \(2014\)](#) 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 error corrections, including CRSE. We reproduce their study here, and include the estimation of the standard errors using CESE as in [Jackson \(2019\)](#).

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. After loading the package and the data,

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

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

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

The estimated raw values of the variance covariance matrix obtained by running the standard R function from the `stats` package (R Core Team, 2018) 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_eneq	0.02532	0.001833	-0.00007513	-0.0007721	-0.0009086
	eneg	logmag	logmag_eneq		
(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_eneq	-0.0119029	-0.0299552	0.01778317		

The CRSE, using countries as the grouping variable, obtained using the `vcovCL()` function of the `sandwich` package (Zeileis, 2004) 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_enege -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_enege
 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_enege
 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_enege
 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` plays nicely with the `lmtest` package and the function `coefstest()` 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_enege, 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: <0.00000000000000002

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

Similarily, 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_neg   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, particularly 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.

Covariate	Estimate	Std. Errors			
		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_neg	0.3605	0.1334	0.2883	0.3473	0.2463

Covariate	Estimate	Confidence Intervals			
		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_neg	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.

References

- Cameron, A. C., Gelbach, J. B., and Miller, D. L. (2008). Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics*, 90(3):414–427.
- Chambers, J. M. (1992). Linear models. In Chambers, J. M. and Hastie, T. J., editors, *Statistical Models in S*, chapter 4. Wadsworth Brooks/Cole.
- Eicker, F. (1967). Limit theorems for regressions with unequal and dependent errors. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 59–82.
- Elgie, R., Bucur, C., Dolez, B., and Laurent, A. (2014). Proximity, candidates, and presidential power: How directly elected presidents shape the legislative party system. *Political Research Quarterly*, 67(3):467–477.
- Esarey, J. and Menger, A. (2018). Practical and effective approaches to dealing with clustered data. *Political Science Research and Methods*, pages 1–19.
- Greene, W. H. (2012). *Econometric analysis*. Pearson Prentice Hall, Upper Saddle River, NJ.
- Harden, J. J. (2011). A bootstrap method for conducting statistical inference with clustered data. *State Politics & Policy Quarterly*, 11(2):223–246.
- Hayes, A. F. and Cai, L. (2007). Using heteroskedasticity-consistent standard error estimators in ols regression: An introduction and software implementation. *Behavior research methods*, 39(4):709–722.
- Imbens, G. W. and Kolesar, M. (2016). Robust standard errors in small samples: Some practical advice. *Review of Economics and Statistics*, 98(4):701–712.
- Jackson, J. (2019). Corrected standard errors with clustered data. *Political Analysis*, forthcoming.
- Liang, K.-Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22.
- MacKinnon, J. G. and Webb, M. D. (2017). Wild bootstrap inference for wildly different cluster sizes. *Journal of Applied Econometrics*, 32(2):233–254.
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Wasserstein, R. L., Schirm, A. L., and Lazar, N. A. (2019). Moving to a world beyond “ $p < 0.05$ ”.
- White, H. L. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):817–838.
- Wilkinson, G. N. and Rogers, C. E. (1973). Symbolic descriptions of factorial models for analysis of variance. *Applied Statistics*, (22):392–9.
- Zeileis, A. (2004). Econometric computing with hc and hac covariance matrix estimators.