Cluster-robust standard errors using R

Mahmood Arai
Department of Economics
Stockholm University

March 12, 2015

1 Introduction

This note deals with estimating cluster-robust standard errors on one and two dimensions using R (see R Development Core Team [2007]). Cluster-robust standard errors are an issue when the errors are correlated within groups of observations. For discussion of robust inference under within groups correlated errors, see Wooldridge [2003], Cameron et al. [2006], and Petersen [2005] and the references therein.

Two data sets are used. The first data set is panel data from Introduction to Econometrics by Stock and Watson [2006a], chapter 10. The second data set is the Mitchell Petersen's test data for two-way clustering. The first part of this note deals with estimation of fixed-effects model using the Fatality data. The second part deals with cluster-robust standard errors.

You need to install package `lmtest` by Torsten Hothorn, Achim Zeileis, Giovanni Millo and David Mitchell, package `sandwich` by Thomas Lumley and Achim Zeileis, package `plm` by Yves Croissant and Giovanni Millo and `Ecdat` by Yves Croissant. The function `robcov` in the package `Design` by Frank E. Harrell Jr can be used for clustering in one dimension in case of an `ols-fit`. The function `plm` can be used for obtaining one-way clustered standard errors.

2 Estimating fixed-effects model

The data set `Fatality` in the package `Ecdat` cover data for 48 US states over 7 years. One way to estimate such a model is to include fixed group intercepts in the model. This is an example estimating a two-way fixed effects model.

```
> library(Ecdat)
> data(Fatality)
> LSDV <- lm(mrall ~ beertax + factor(year) + factor(state),
+                                                  data=Fatality)
```

When the number of groups are large, we run into the incidental parameter problem, implying inconsistent parameter estimates, and will have computational problems inverting a large model-matrix. A solution is to use variables measured as deviation from group mean in estimation. Using such an transformation we have to correct the degree of freedom for the number of group means that are estimated using the transformation.

Let us first we write a function that computes deviation from group means of our variables. This makes only sense for numeric variables. The following function takes a dataframe `df1` and yields a new data set including the original data and new variables computed as deviation from group means as defined by the argument `group`. The group centered variables have the same names as the original variables with a prefix `C.`.

```
> gcenter <- function(df1,group) {
+      variables <- paste(
+          rep("C", ncol(df1)),  colnames(df1), sep=".")
+      copydf <- df1
+      for (i in 1:ncol(df1)) {
+          copydf[,i] <- df1[,i] - ave(df1[,i], group,FUN=mean)}
+      colnames(copydf) <- variables
+      return(cbind(df1,copydf))}
```

Now we use this function to obtain transformed data.

```
> centerFatality <- gcenter(Fatality[,1:4], Fatality$state)
```

We can then use this transformed data and run `OLS` using the same model as before.

```
> fmlm  <- lm(C.mrall ~  C.beertax + factor(year),
+                                             data=centerFatality)
```

The variance-covariance matrix must be reweighted with $dfcw = (N - k)/((N - k) - (M - 1))$ where N is the total number of observations, M is the number of groups and K is the model rank.

```
> library(sandwich)
> M <- length(unique(Fatality$state))
> dfcw <- fmlm$df / (fmlm$df - (M -1))
> library(lmtest)
> coeftest(fmlm, dfcw*vcov(fmlm))

t test of coefficients:

                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.059553   0.027326  2.1794 0.030016 *
C.beertax        -0.639980   0.197377 -3.2424 0.001307 **
factor(year)1983 -0.079903   0.038354 -2.0833 0.037997 *
factor(year)1984 -0.072421   0.038352 -1.8883 0.059864 .
factor(year)1985 -0.123976   0.038442 -3.2250 0.001386 **
factor(year)1986 -0.037864   0.038588 -0.9813 0.327191
factor(year)1987 -0.050902   0.038974 -1.3061 0.192447
factor(year)1988 -0.051804   0.039623 -1.3074 0.191992
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The second argument reweights the variance-covariance matrix correcting the degrees of freedom accounting for the fact that data are centered around group means. The standard errors are valid under constant error variance. However, in fixed-effects models you should use cluster-robust standard errors as described in the next section – See Arellano [1987], Wooldridge [2002] and Stock and Watson [2006b]. The package `plm` can be used to compute one-way cluster-robust standard errors.

```
> library(plm)
> fmplm <- plm(mrall~ beertax + factor(year), data=Fatality)
```

The degree-of-freedom of arellano in plm using HC1 is $N/(N-K)$. In the next section we use a slightly different degree-of-freedom correction in order to replicate Stock and Watson [2006a] and Petersen [2005].

3 Cluster-robust standard errors

Two functions are presented herebelow. These functions have the following arguments:

- The fitted model fm
- A factor for the degree of freedom correction when we have estimated on deviation from group mean data, dfcw. Set this argument to 1 when such a degree of freedom correction is not necessary.
- The cluster variable or variables, cluster, cluster1, cluster2.

The functions have two parts.

For N observations, M clusters, and K =rank($\mathbf{X}$) where $\mathbf{X}$ is the matrix of regressors, $(M/(M-1)) * ((N-1)/(N-K))$ is computed as degree of freedom correction.

The second part of the function computes:

$$\mathbf{X}'\mathbf{X}^{-1}\mathbf{u}'\mathbf{u}\mathbf{X}'\mathbf{X}^{-1}$$

where $\mathbf{u}$ is a $M \times K$ matrix with rows u_j. Each row is the per cluster sum of $\mathbf{X}_j * \mathbf{e}_j$ over all individuals within each cluster. Denoting the number of observations in cluster j as N_j, $\mathbf{X}_j$ is a $N_j \times K$ matrix of regressors for cluster j, the star $*$ denotes element by elements multiplication and $\mathbf{e}_j$ is a $N_j \times 1$ vector of residuals.

The $\mathbf{X}_j * \mathbf{e}_j$ is estimated using the function estfun. Summing over observations per cluster using apply(...) yields $\mathbf{u}$. This is then used as the argument in the function sandwich to obtain the variance covariance matrix (Zeileis [2006]). The function mclx has the same structure repeated over clusters and the overlap between clusters and finally summarized as summing up over clusters minus the overlap.

The functions clx for one-way clustering.

```
> clx <-
+ function(fm, dfcw, cluster){
+          library(sandwich)
+          library(lmtest)
+          M <- length(unique(cluster))
+          N <- length(cluster)
+          dfc <- (M/(M-1))*((N-1)/(N-fm$rank))
+          u  <- apply(estfun(fm),2,
+                           function(x) tapply(x, cluster, sum))
+          vcovCL <- dfc*sandwich(fm, meat=crossprod(u)/N)*dfcw
+          coeftest(fm, vcovCL) }
```

Clustered on state, replicating Stock and Watson

```
> clx(fmlm, dfcw, Fatality$state)

t test of coefficients:

                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.059553   0.044026  1.3527  0.17709
C.beertax        -0.639980   0.385787 -1.6589  0.09809 .
factor(year)1983 -0.079903   0.037907 -2.1079  0.03580 *
factor(year)1984 -0.072421   0.047409 -1.5276  0.12758
factor(year)1985 -0.123976   0.049759 -2.4916  0.01321 *
factor(year)1986 -0.037864   0.061648 -0.6142  0.53951
factor(year)1987 -0.050902   0.068722 -0.7407  0.45941
factor(year)1988 -0.051804   0.069580 -0.7445  0.45709
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The function `mclx` for two-way clustering.

```
> mclx <-
+ function(fm, dfcw, cluster1, cluster2){
+         library(sandwich)
+         library(lmtest)
+         cluster12 = paste(cluster1,cluster2, sep="")
+         M1  <- length(unique(cluster1))
+         M2  <- length(unique(cluster2))
+         M12 <- length(unique(cluster12))
+         N   <- length(cluster1)
+         K   <- fm$rank
+         dfc1  <- (M1/(M1-1))*((N-1)/(N-K))
+         dfc2  <- (M2/(M2-1))*((N-1)/(N-K))
+         dfc12 <- (M12/(M12-1))*((N-1)/(N-K))
+         u1   <- apply(estfun(fm), 2,
+                         function(x) tapply(x, cluster1,  sum))
+         u2   <- apply(estfun(fm), 2,
+                         function(x) tapply(x, cluster2,  sum))
+         u12  <- apply(estfun(fm), 2,
+                         function(x) tapply(x, cluster12, sum))
+         vc1   <-  dfc1*sandwich(fm, meat=crossprod(u1)/N )
+         vc2   <-  dfc2*sandwich(fm, meat=crossprod(u2)/N )
+         vc12  <- dfc12*sandwich(fm, meat=crossprod(u12)/N)
+         vcovMCL <- (vc1 + vc2 - vc12)*dfcw
+         coeftest(fm, vcovMCL)}
```

The following applies the clustering functions on Mitchell Petersen's test-data. Download the `test_data.txt` from Petersens se-programming page and create a `lm` object by running `y` on `x` using the data `test`.

```
> SITE <- "http://www.kellogg.northwestern.edu/faculty/petersen/"
> URLdata <- paste(SITE, "/htm/papers/se/test_data.txt", sep="")
> VarNames <- c("firmid", "year", "x", "y")
> test <- read.table(file=URLdata, col.names=VarNames)
> fm <- lm(y ~ x, data=test)
```

To cluster on firm, the arguments are the fitted model `fm`, we need no degree of freedom correction since we have estimated the model on the original location (no transformation) implying that the second argument is `1`.The third argument specify the cluster variable. The variable for firm indicator is `firmid` in the data `test`.

```
> clx(fm,1,test$firmid)

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 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The following yields clustering on year.

```
> clx(fm,1, test$year)

t test of coefficients:

            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.029680   0.023387  1.2691   0.2045
x           1.034833   0.033389 30.9933   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

For clustering on firm and year we use the function `mclx`.

```
> mclx(fm,1, test$firmid, test$year)

t test of coefficients:

            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.029680   0.065064  0.4562   0.6483
x           1.034833   0.053558 19.3217   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

These results were obtained on a `x86_64-apple-darwin13.4.0` platform using R version 3.1.2 (2014-10-31) [R Development Core Team, 2007] with packages **lmtest** 0.9-33 (2014-01-23), **sandwich** 2.3-2 (2014-08-24), **plm** 1.4-0 (2013-12-24) and **Ecdat** 0.2-7 (2013-04-25).

References

M. Arellano. Computing robust standard errors for within-groups estimators. *Oxford Bulletin of Economics and Statistics*, 49(4):431–34, November 1987. URL http://ideas.repec.org/a/bla/obuest/v49y1987i4p431-34.html.

C. A. Cameron, J. B. Gelbach, and D. L. Miller. Robust inference with multiway clustering. *NBER Working Paper*, T0327, 2006. URL http://www.nber.org/papers/t0327.

M. A. Petersen. Estimating standard errors in finance panel data sets: Comparing approaches. NBER Working Papers 11280, National Bureau of Economic Research, Inc, Apr. 2005. URL http://ideas.repec.org/p/nbr/nberwo/11280.html.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. URL http://www.R-project.org. ISBN 3-900051-07-0.

J. H. Stock and M. W. Watson. *Introduction to Econometrics*. Addison Wesley, 2006a.

J. H. Stock and M. W. Watson. Heteroskedasticity-robust standard errors for fixed effects panel data regression. NBER Technical Working Papers 0323, National Bureau of Economic Research, Inc, June 2006b. URL http://ideas.repec.org/p/nbr/nberte/0323.html.

J. M. Wooldridge. *Econometric Analysis of Cross Section and Panel Data*. Cambridge, MA: MIT Press, 2002.

J. M. Wooldridge. Cluster-sample methods in applied econometrics. *American Economic Review*, 93:133–138, 2003.

A. Zeileis. Object-oriented computation of sandwich estimators. *Journal of Statistical Software*, i09, 2006. URL http://www.jstatsoft.org/v16/i09.