Performing Cluster Bootstrapped Regressions in R

BLINDED

October 6, 2016

A simple and quick way to perform cluster bootstrapped regressions in `R` is by using the `rms` package which has the speedy and customizable `bootcov` function. For applied researchers, this avoids the statistical programming often associated with bootstrapping.

Reading in the data

I provide an example using the commonly used High School and Beyond dataset. The following lines of code read in the dataset from the UCLA website (it is in a Stata .dta format and uses the `foreign` package to read it) and creates a subset of the data only using five of the original variables.

```
library(foreign)
hsb<-read.dta(file="http://www.ats.ucla.edu/stat/paperexamples/singer/hsb12.d
ta")
names(hsb)

##  [1] "school"        "student"       "minority"      "female"
##  [5] "ses"           "meanses"       "cses"          "mathach"
##  [9] "size"          "sector"        "pracad"        "disclim"
## [13] "himinty"       "meansesBYcses" "sectorBYcses"

hsb2<-subset(hsb,,c('school','mathach','cses','meanses','sector'))
str(hsb2)

## 'data.frame':    7185 obs. of  5 variables:
##  $ school : int  1224 1224 1224 1224 1224 1224 1224 1224 1224 1224 ...
##  $ mathach: num  5.88 19.71 20.35 8.78 17.9 ...
##  $ cses   : num  -1.1 -0.16 -0.1 -0.24 0.27 ...
##  $ meanses: num  -0.428 -0.428 -0.428 -0.428 -0.428 ...
##  $ sector : num  0 0 0 0 0 0 0 0 0 0 ...
```

Step 1: Fit the model

Before proceeding, the `rms` package must be installed. Users must use `install.packages('rms')` once if they have not already installed the package. Once installed, users can then use the `ols` and the `bootcov` functions. Prior to using the `bootcov` function, users must first fit their model using the `ols` function in `rms` (not the regular `lm` function in Base R).

We are interested in predicting math achievement using school-level socioeconomic status or SES (meanses), the school sector (sector, 1 = catholic, 0 = public), the student's centered

SES (cses), and the interaction between the sector and cses. The syntax for fitting the model and saving results into an object referred to here as `modfit` is similar to running a standard linear regression using the `lm` function. The only exception is that two options in the `ols` function are added which read `x=T` and `y=T` (see syntax below). These are needed by the `bootcov` function. The following syntax below fits the model and shows the output.

```
library(rms)
```

After loading the package, we fit a model using the `ols` function.

```
modfit<-ols(mathach~meanses+sector+cses+sector*cses,data=hsb2,x=T,y=T)
modfit

##
## Linear Regression Model
##
## ols(formula = mathach ~ meanses + sector + cses + sector * cses,
##     data = hsb2, x = T, y = T)
##                  Model Likelihood     Discrimination
##                     Ratio Test           Indexes
## Obs     7185    LR chi2   1374.49    R2       0.174
## sigma 6.2526    d.f.            4    R2 adj   0.174
## d.f.    7180    Pr(> chi2) 0.0000    g        3.262
##
## Residuals
##
##      Min       1Q   Median       3Q      Max
## -20.2821  -4.6214   0.1513   4.8252  17.6168
##
##                Coef    S.E.   t      Pr(>|t|)
## Intercept     12.1014 0.1070 113.11 <0.0001
## meanses        5.1638 0.1910  27.04 <0.0001
## sector         1.2723 0.1580   8.05 <0.0001
## cses           2.7820 0.1490  18.67 <0.0001
## sector * cses -1.3485 0.2251  -5.99 <0.0001
```

As a point of comparison, we may fit a multilevel model and compare results (this is not a necessary step but shown for comparative purposes). We use the `nlme` package and specify a random intercept model. If not already installed, users must install the `nlme` package.

```
library(nlme)
mlm<-lme(mathach~meanses+sector+cses+sector*cses,random=~1|school,data=hsb2)
summary(mlm)

## Linear mixed-effects model fit by REML
##  Data: hsb2
##        AIC      BIC    logLik
##   46531.02 46579.18 -23258.51
##
## Random effects:
##  Formula: ~1 | school
```

```
##         (Intercept) Residual
## StdDev:    1.540686 6.068631
##
## Fixed effects: mathach ~ meanses + sector + cses + sector * cses
##                  Value Std.Error   DF  t-value p-value
## (Intercept) 12.112908 0.1986474 7023 60.97692   0e+00
## meanses      5.336554 0.3689726  157 14.46328   0e+00
## sector       1.216392 0.3061145  157  3.97365   1e-04
## cses         2.782091 0.1446048 7023 19.23927   0e+00
## sector:cses -1.348549 0.2184493 7023 -6.17328   0e+00
##  Correlation:
##             (Intr) meanss sector cses
## meanses      0.245
## sector      -0.697 -0.356
## cses         0.004  0.000 -0.003
## sector:cses -0.003  0.000  0.004 -0.662
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.11736844 -0.72730506  0.01340776  0.75298197  3.03320166
##
## Number of Observations: 7185
## Number of Groups: 160
```

A comparison of standard errors of the school level variables (i.e., meanses and sector) will show that the OLS standard errors are much smaller (which are underestimated) than the standard errors from the multilevel model. These standard errors are the primary concern when analyzing clustered data.

Step 2: Run the clustered bootstrap regression

Once the model has been fit, running the clustered bootstrapped regression is straightforward. There are three important options that the `bootcov` function needs. First is the fit object (which is the `modfit` object created using the `ols` function used earlier). Second, the clustering variable has to be specified: in this case, it is the school variable which is `hsb2$school`. Last, we specify the number of bootstrapped replications (B) to use, here we specify `B = 1000` (the default is 200 if not specified). An option that users may want to add (not shown) is `pr=T` which merely shows the progress of bootstrapping (as bootstrapping, depending on model complexity, may take a few seconds and using the option provides users with onscreen feedback).

```
set.seed(123) #set for replicable results
bootcov(modfit,cluster=hsb2$school,B=1000)

##
## Linear Regression Model
##
## ols(formula = mathach ~ meanses + sector + cses + sector * cses,
##     data = hsb2, x = T, y = T)
```

```
##                             Model Likelihood     Discrimination
##                                Ratio Test           Indexes
## Obs                  7185    LR chi2    1374.49    R2         0.174
## sigma              6.2526    d.f.             4    R2 adj     0.174
## d.f.                 7180    Pr(> chi2) 0.0000    g          3.262
## Cluster on hsb2$school
## Clusters              160
##
## Residuals
##
##      Min       1Q   Median       3Q      Max
## -20.2821  -4.6214   0.1513   4.8252  17.6168
##
##                 Coef    S.E.   t     Pr(>|t|)
## Intercept      12.1014 0.1699 71.21 <0.0001
## meanses         5.1638 0.3327 15.52 <0.0001
## sector          1.2723 0.2914  4.37 <0.0001
## cses            2.7820 0.1601 17.37 <0.0001
## sector * cses -1.3485 0.2325 -5.80 <0.0001
```

A comparison of the three different model standard errors will show that the standard errors of the clustered bootstrapped regressions are closer to the ones found in the multilevel model. NOTE: the point estimates of the bootstrapped regressions are the same as the OLS regression so it is important that the model is properly specified to begin with.

Other considerations

For nested models *with a low number of clusters* with a binary predictor at level 2 (e.g., a treatment indicator where treat = 1 or 0), researchers should keep in mind that it is possible, due to the low number of clusters, to have a bootstrapped sample with clusters that are either all in the treatment condition or all in the control condition. In such a case, the treatment effect for that subsample becomes inestimable as a result of a lack of variation in the treatment variable. However, a modified bootstrap procedure is possible where the treatment and control clusters are separated into two groups and in each resampling step, clusters are sampled independently within the treatment and control groups and then combined to form the complete bootstrapped sample, ensuring the presence of both treatment and control groups in every bootstrapped sample.

In the `bootcov` function, this can be specified using the `group=` option. The `sector` variable in the HSB dataset indicates whether the school was a Catholic or public school.

```
#Creating a small dataset of 10 schools, chosen here by school id number
hsb3<-hsb2[hsb2$school %in% c('1288','1296','1308','7635','7688',
   '2990','9347','3088','6170','3610'),]
```

The standard cluster bootstrap regression is shown below using the reduced dataset of 10 schools.

```
modfit2<-ols(mathach~meanses+sector+cses+sector*cses,data=hsb3,x=T,y=T)
modfit2
```

```
##
## Linear Regression Model
##
## ols(formula = mathach ~ meanses + sector + cses + sector * cses,
##     data = hsb3, x = T, y = T)
##                   Model Likelihood     Discrimination
##                      Ratio Test            Indexes
## Obs       427    LR chi2    121.31    R2       0.247
## sigma 5.8253    d.f.            4    R2 adj   0.240
## d.f.      422    Pr(> chi2) 0.0000    g        3.721
##
## Residuals
##
##      Min       1Q   Median       3Q      Max
## -16.1331  -4.1440   0.3367   4.6686  14.8484
##
##               Coef    S.E.   t     Pr(>|t|)
## Intercept     12.2402 0.6718 18.22 <0.0001
## meanses        6.4584 1.4230  4.54 <0.0001
## sector         1.9658 1.0528  1.87 0.0626
## cses           2.4134 0.7057  3.42 0.0007
## sector * cses -0.4290 0.8884 -0.48 0.6294
```

```
set.seed(1234)
bootcov(modfit2,cluster=hsb3$school,B=1000)
```

```
## Warning in bootcov(modfit2, cluster = hsb3$school, B = 1000): fit failure
## in 4 resamples. Might try increasing maxit
```

```
##
## Linear Regression Model
##
## ols(formula = mathach ~ meanses + sector + cses + sector * cses,
##     data = hsb3, x = T, y = T)
##                             Model Likelihood     Discrimination
##                                Ratio Test            Indexes
## Obs                 427    LR chi2    121.31    R2       0.247
## sigma            5.8253    d.f.            4    R2 adj   0.240
## d.f.                422    Pr(> chi2) 0.0000    g        3.721
## Cluster on hsb3$school
## Clusters             10
##
## Residuals
##
##      Min       1Q   Median       3Q      Max
## -16.1331  -4.1440   0.3367   4.6686  14.8484
##
##               Coef    S.E.   t     Pr(>|t|)
## Intercept     12.2402 1.9066  6.42 <0.0001
## meanses        6.4584 3.9193  1.65 0.1001
```

```
## sector         1.9658 3.1872  0.62 0.5377
## cses           2.4134 0.8006  3.01 0.0027
## sector * cses -0.4290 0.9359 -0.46 0.6469
```

In the output, an error appears that out of the 1,000 replications, 4 could not be estimated. Using the group= option allows us to stratify the sample. Here is a modified version of the bootcov options. No errors now appear.

```
set.seed(1234)
bootcov(modfit2,cluster=hsb3$school,group=hsb3$sector,B=1000)

##
## Linear Regression Model
##
## ols(formula = mathach ~ meanses + sector + cses + sector * cses,
##     data = hsb3, x = T, y = T)
##                            Model Likelihood     Discrimination
##                               Ratio Test           Indexes
## Obs                 427    LR chi2     121.31    R2       0.247
## sigma            5.8253    d.f.             4    R2 adj   0.240
## d.f.                422    Pr(> chi2) 0.0000    g        3.721
## Cluster on hsb3$school
## Clusters             10
##
## Residuals
##
##      Min       1Q   Median       3Q      Max
## -16.1331  -4.1440   0.3367   4.6686  14.8484
##
##               Coef    S.E.    t     Pr(>|t|)
## Intercept     12.2402 1.4468  8.46 <0.0001
## meanses        6.4584 2.9386  2.20 0.0285
## sector         1.9658 2.3462  0.84 0.4026
## cses           2.4134 0.7038  3.43 0.0007
## sector * cses -0.4290 0.8173 -0.52 0.5999
```

Results can also be compared if a MLM is fit with the smaller sample.

```
mlm10<-lme(mathach~meanses+sector+cses+sector*cses,random=~1|school,data=hsb3
)
summary(mlm10)

## Linear mixed-effects model fit by REML
##  Data: hsb3
##       AIC      BIC   logLik
##   2700.94 2729.255 -1343.47
##
## Random effects:
##  Formula: ~1 | school
##         (Intercept) Residual
## StdDev:    1.932757 5.604522
```

```
##
## Fixed effects: mathach ~ meanses + sector + cses + sector * cses
##                    Value Std.Error  DF   t-value p-value
## (Intercept) 12.662469  1.400718 415  9.039987  0.0000
## meanses      6.308036  3.212066   7  1.963856  0.0903
## sector       1.487295  2.363630   7  0.629242  0.5492
## cses         2.413856  0.678973 415  3.555156  0.0004
## sector:cses -0.429664  0.854757 415 -0.502674  0.6155
##  Correlation:
##             (Intr) meanss sector cses
## meanses      0.627
## sector      -0.867 -0.809
## cses         0.003  0.000 -0.002
## sector:cses -0.002  0.000  0.002 -0.794
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.12908164 -0.67745680  0.03250507  0.76544828  2.91268719
##
## Number of Observations: 427
## Number of Groups: 10
```

NOTE. The bootcov function computes the p values based on the degrees of freedom of the whole model (i.e., n - k - 1, where k = number of predictors or 427 - 4 - 1 = 422). However, in actuality, there are only 10 schools with 2 predictors at level 2. In the MLM above, 7 degrees of freedom are used (i.e., G - level 2 predictors - 1 where G is the number of groups). We may want to manually compute the p values for this. At level 1, the degrees of freedom in a MLM is n - (df at level 2) - k - 1 or 427 - 7 - 4 - 1 = 415. Using OLS, the df is 422. This may be less important with level 1 variables as the total number of level 1 observations are often larger than the number of clusters.

```
### this is how the mean SES variable p value is computed where 2.2 is the t statistic
### and 422 is the df
(1-pt(2.2,422))*2

## [1] 0.02834827

### if we adjust the df to 7, the p value is now .06
(1-pt(2.2,7))*2

## [1] 0.06373102
```

END