bootstrap postestimation — Postestimation tools for bootstrap

Postestimation commands

The following postestimation command is of special interest after `bootstrap`:

Command	Description
`estat bootstrap`	percentile-based and bias-corrected CI tables

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

The postestimation command is allowed if it may be used after *command*.

predict

The syntax of `predict` (and even if `predict` is allowed) following `bootstrap` depends upon the *command* used with `bootstrap`. If `predict` is not allowed, neither is `predictnl`.

margins

The syntax of `margins` (and even if `margins` is allowed) following `bootstrap` depends upon the *command* used with `bootstrap`.

estat

Description for estat

`estat bootstrap` displays a table of confidence intervals for each statistic from a bootstrap analysis.

Menu for estat

Statistics > Postestimation

Syntax for estat

`estat bootstrap` [, *options*]

options	Description
`bc`	bias-corrected CIs; the default
`bca`	bias-corrected and accelerated (BC_a) CIs
`normal`	normal-based CIs
`percentile`	percentile CIs
`all`	all available CIs
`noheader`	suppress table header
`nolegend`	suppress table legend
`verbose`	display the full table legend

`bc`, `bca`, `normal`, and `percentile` may be used together.

Options for estat

`bc` is the default and displays bias-corrected confidence intervals.

`bca` displays bias-corrected and accelerated confidence intervals. This option assumes that you also specified the `bca` option on the `bootstrap` prefix command.

`normal` displays normal approximation confidence intervals.

`percentile` displays percentile confidence intervals.

`all` displays all available confidence intervals.

`noheader` suppresses display of the table header. This option implies `nolegend`.

`nolegend` suppresses display of the table legend, which identifies the rows of the table with the expressions they represent.

`verbose` requests that the full table legend be displayed.

Remarks and examples

stata.com

▷ Example 1

The estat bootstrap postestimation command produces a table containing the observed value of the statistic, an estimate of its bias, the bootstrap standard error, and up to four different confidence intervals.

If we were interested merely in getting bootstrap standard errors for the model coefficients, we could use the bootstrap prefix with our estimation command. If we were interested in performing a thorough bootstrap analysis of the model coefficients, we could use the estat bootstrap postestimation command after fitting the model with the bootstrap prefix.

Using example 1 from [R] **bootstrap**, we need many more replications for the confidence interval types other than the normal based, so let's rerun the estimation command. We will reset the random-number seed—in case we wish to reproduce the results—increase the number of replications, and save the bootstrap distribution as a dataset called bsauto.dta.

```
. use https://www.stata-press.com/data/r17/auto
(1978 automobile data)
. set seed 1
. bootstrap _b, reps(1000) saving(bsauto) bca: regress mpg weight gear foreign
  (output omitted)
. estat bootstrap, all
Linear regression                               Number of obs     =         74
                                                Replications      =       1000
```

mpg	Observed coefficient	Bias	Bootstrap std. err.	[95% conf.	interval]	
weight	-.00613903	.0000686	.00065005	-.0074131	-.004865	(N)
				-.0073115	-.0048083	(P)
				-.0073757	-.0048444	(BC)
				-.0075498	-.0049202	(BCa)
gear_ratio	1.4571134	.0297538	1.4471522	-1.379253	4.29348	(N)
				-1.18779	4.540121	(P)
				-1.185389	4.540121	(BC)
				-1.131393	4.58386	(BCa)
foreign	-2.2216815	.1029615	1.2606565	-4.692523	.2491598	(N)
				-4.513954	.5011647	(P)
				-4.608057	.4208305	(BC)
				-4.614719	.3925043	(BCa)
_cons	36.101353	-.3122698	5.4303717	25.45802	46.74469	(N)
				24.55211	46.0322	(P)
				24.90078	46.05819	(BC)
				24.99072	46.40419	(BCa)

```
Key:    N: Normal
        P: Percentile
       BC: Bias-corrected
      BCa: Bias-corrected and accelerated
```

The estimated standard errors here differ from our previous estimates using only 100 replications by, respectively, 7%, 6%, 8%, and 4%; see example 1 of [R] **bootstrap**. So much for our advice that 50–200 replications are good enough to estimate standard errors. Well, the more replications the better—that advice you should believe.

Which of the methods to compute confidence intervals should we use? If the statistic is unbiased, the percentile (P) and bias-corrected (BC) methods should give similar results. The bias-corrected confidence interval will be the same as the percentile confidence interval when the observed value of the statistic is equal to the median of the bootstrap distribution. Thus, for unbiased statistics, the two methods should give similar results as the number of replications becomes large. For biased statistics, the bias-corrected method should yield confidence intervals with better coverage probability (closer to the nominal value of 95% or whatever was specified) than the percentile method. For statistics with variances that vary as a function of the parameter of interest, the bias-corrected and accelerated method (BC_a) will typically have better coverage probability than the others.

When the bootstrap distribution is approximately normal, all of these methods should give similar confidence intervals as the number of replications becomes large. If we examine the normality of these bootstrap distributions using, say, the `pnorm` command (see [R] **Diagnostic plots**), we see that they closely follow a normal distribution. In this case, the normal approximation would also be a valid choice. The chief advantage of the normal-approximation method is that it (supposedly) requires fewer replications than the other methods. Of course, it should be used only when the bootstrap distribution exhibits normality.

We can load `bsauto.dta` containing the bootstrap distributions for these coefficients:

```
. use bsauto
(bootstrap: regress)

. describe *

Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
_b_weight       float   %9.0g                 _b[weight]
_b_gear_ratio   float   %9.0g                 _b[gear_ratio]
_b_foreign      float   %9.0g                 _b[foreign]
_b_cons         float   %9.0g                 _b[_cons]
```

We can now run other commands, such as `pnorm`, on the bootstrap distributions. As with all standard estimation commands, we can use the `bootstrap` command to replay its output table. The default variable names assigned to the statistics in *exp_list* are `_bs_1`, `_bs_2`, . . . , and each variable is labeled with the associated expression. The naming convention for the extended expressions `_b` and `_se` is to prepend `_b_` and `_se_`, respectively, onto the name of each element of the coefficient vector. Here the first coefficient is `_b[weight]`, so `bootstrap` named it `_b_weight`.

◁

Also see

[R] **bootstrap** — Bootstrap sampling and estimation

[U] **20 Estimation and postestimation commands**