

svy bootstrap — Bootstrap for survey data

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
References	Also see		

Description

`svy bootstrap` performs nonparametric bootstrap estimation of specified statistics (or expressions) for a Stata command or a user-written program. The command is executed once for each replicate using sampling weights that are adjusted according to the bootstrap methodology. Any Stata estimation command listed in [\[SVY\] svy estimation](#) may be used with `svy bootstrap`. User-written programs that meet the requirements in [\[P\] program properties](#) may also be used.

Quick start

Estimate population mean of `v1` using bootstrap standard-error estimates and variables with prefix `rwvar` as the bootstrap replicate weights

```
svyset [pweight=wvar1], bsrweight(rwvar*)
svy bootstrap _b: mean v1
```

Same as above

```
svyset [pweight=wvar1], bsrweight(rwvar*) vce(bootstrap)
svy: mean v1
```

As above, and specify that 3 replicates were used to calculate each bootstrap replicate weight

```
svy, bsn(3): mean v1
```

Bootstrap standard error of the difference between the means of `v2` and `v3` using either `svyset` command above

```
svy bootstrap (_b[v2]-_b[v3]): mean v2 v3
```

As above, but name the result `diff` and save results from each replication to `mydata.dta`

```
svy bootstrap diff=(_b[v2]-_b[v3]), saving(mydata): mean v2 v3
```

Note: Any estimation command meeting the requirements specified in the *Description* may be substituted for `mean` in the examples above.

Menu

Statistics > Survey data analysis > Resampling > Bootstrap estimation

Syntax

svy bootstrap *exp_list* [, *svy_options bootstrap_options eform_option*] : *command*

<i>svy_options</i>	Description
--------------------	-------------

if/in

<u>subpop</u> ([<i>varname</i>] [<i>if</i>])	identify a subpopulation
--	--------------------------

Reporting

<u>level</u> (#)	set confidence level; default is <code>level(95)</code>
<u>noheader</u>	suppress table header
<u>nolegend</u>	suppress table legend
<u>noadjust</u>	do not adjust model Wald statistic
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
<u>coeflegend</u>	display legend instead of statistics

`coeflegend` is not shown in the dialog boxes for estimation commands.

<i>bootstrap_options</i>	Description
--------------------------	-------------

Main

<u>bsn</u> (#)	bootstrap mean-weight adjustment
----------------	----------------------------------

Options

<u>saving</u> (<i>filename</i> [, ...])	save results to <i>filename</i> ; save statistics in double precision; save results to <i>filename</i> every # replications
<u>mse</u>	use MSE formula for variance

Reporting

<u>verbose</u>	display the full table legend
<u>nodots</u>	suppress replication dots
<u>dots</u> (#)	display dots every # replications
<u>noisily</u>	display any output from <i>command</i>
<u>trace</u>	trace <i>command</i>
<u>title</u> (<i>text</i>)	use <i>text</i> as title for bootstrap results

Advanced

<u>nodrop</u>	do not drop observations
<u>reject</u> (<i>exp</i>)	identify invalid results
<u>dof</u> (#)	design degrees of freedom

svy requires that the survey design variables be identified using `svyset`; see [SVY] `svyset`.

command defines the statistical command to be executed. The `by` prefix cannot be part of *command*.

`collect` is allowed; see [U] 11.1.10 Prefix commands.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Warning: Using `if` or `in` restrictions will often not produce correct variance estimates for subpopulations. To compute estimates for subpopulations, use the `subpop()` option.

svy bootstrap requires that the bootstrap replicate weights be identified using `svyset`.

exp_list specifies the statistics to be collected from the execution of *command*. *exp_list* is required unless *command* has the `svyb` program property, in which case *exp_list* defaults to `_b`; see [P] [program properties](#). The expressions in *exp_list* are assumed to conform to the following:

```
exp_list contains      (name: elist)
                      elist
                      eexp
elist contains        newvarname = (exp)
                      (exp)
eexp is               specname
                      [eqno]specname
specname is          _b
                    _b []
                    _se
                    _se []
eqno is              ##
                    name
```

exp is a standard Stata expression; see [U] [13 Functions and expressions](#).

Distinguish between `[]`, which are to be typed, and `[][]`, which indicate optional arguments.

Options

svy_options; see [SVY] [svy](#).

Main

`bsn(#)` specifies that `#` bootstrap replicate-weight variables were used to generate each bootstrap mean-weight variable specified in the `bsrweight()` option of `svyset`. The default is `bsn(1)`. The `bsn()` option of `svy bootstrap` overrides the `bsn()` option of `svyset`; see [SVY] [svyset](#).

Options

`saving(filename [, suboptions])` creates a Stata data file (`.dta` file) consisting of (for each statistic in *exp_list*) a variable containing the replicates.

`double` specifies that the results for each replication be saved as `doubles`, meaning 8-byte reals. By default, they are saved as `floats`, meaning 4-byte reals. This option may be used without the `saving()` option to compute the variance estimates by using double precision.

`every(#)` specifies that results be written to disk every `#`th replication. `every()` should be specified in conjunction with `saving()` only when *command* takes a long time for each replication. This will allow recovery of partial results should some other software crash your computer. See [P] [postfile](#).

`replace` specifies that *filename* be overwritten if it exists. This option does not appear in the dialog box.

`mse` specifies that `svy bootstrap` compute the variance by using deviations of the replicates from the observed value of the statistics based on the entire dataset. By default, `svy bootstrap` computes the variance by using deviations of the replicates from their mean.

Reporting

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

`nodots` and `dots(#)` specify whether to display replication dots. By default, one dot character is displayed for each successful replication. A red ‘x’ is displayed if *command* returns an error, and an ‘e’ is displayed if at least one value in *exp_list* is missing. You can also control whether dots are displayed using `set dots`; see [R] [set](#).

`nodots` suppresses display of the replication dots.

`dots(#)` displays dots every # replications. `dots(0)` is a synonym for `nodots`.

`noisily` requests that any output from *command* be displayed. This option implies the `nodots` option.

`trace` causes a trace of the execution of *command* to be displayed. This option implies the `noisily` option.

`title(text)` specifies a title to be displayed above the table of bootstrap results; the default title is “Bootstrap results”.

eform_option; see [R] [eform_option](#). This option is ignored if *exp_list* is not `_b`.

Advanced

`nodrop` prevents observations outside `e(sample)` and the `if` and `in` qualifiers from being dropped before the data are resampled.

`reject(exp)` identifies an expression that indicates when results should be rejected. When *exp* is true, the resulting values are reset to missing values.

`dof(#)` specifies the design degrees of freedom, overriding the default calculation, $df = N_{psu} - N_{strata}$.

Remarks and examplesstata.com

The bootstrap methods for survey data used in recent years are largely due to [McCarthy and Snowden \(1985\)](#), [Rao and Wu \(1988\)](#), and [Rao, Wu, and Yue \(1992\)](#). For example, [Yeo, Mantel, and Liu \(1999\)](#) cites [Rao, Wu, and Yue \(1992\)](#) as the method for variance estimation used in the National Population Health Survey conducted by Statistics Canada.

In the survey bootstrap, the model is fit multiple times, once for each of a set of adjusted sampling weights. The variance is estimated using the resulting replicated point estimates.

▷ Example 1

Suppose that we need to estimate the average birthweight for the population represented by the National Maternal and Infant Health Survey (NMIHS) ([Gonzalez, Krauss, and Scott 1992](#)).

In [SVY] [svy estimation](#), the dataset `nmihs.dta` contained the following design information:

- Primary sampling units are mothers; that is, PSUs are individual observations—there is no separate PSU variable.
- The `finalwgt` variable contains the sampling weights.
- The `stratan` variable identifies strata.
- There is no variable for the finite population correction.

`nmihs_bs.dta` is equivalent to `nmihs.dta` except that the stratum identifier variable `stratan` is replaced by bootstrap replicate-weight variables. The replicate-weight variables are already `svyset`, and the default method for variance estimation is `vce(bootstrap)`.

```
. use https://www.stata-press.com/data/r17/nmihs_bs
. svyset
  Sampling weights: finwgt
                   VCE: bootstrap
                   MSE: off
Bootstrap weights: bsrw1 .. bsrw1000
  Single unit: missing
    Strata 1: <one>
  Sampling unit 1: <observations>
    FPC 1: <zero>
```

Now we can use `svy: mean` to estimate the average birthweight for our population, and the standard errors will be estimated using the survey bootstrap.

```
. svy, nodots: mean birthwgt
Survey: Mean estimation      Number of obs   =    9,946
                           Population size = 3,895,562
                           Replications   =    1,000
```

	Observed mean	Bootstrap std. err.	Normal-based [95% conf. interval]	
birthwgt	3355.452	6.520637	3342.672	3368.233

From these results, we are 95% confident that the mean birthweight for our population is between 3,343 and 3,368 grams.

◀

To accommodate privacy concerns, many public-use datasets contain replicate-weight variables derived from the “mean bootstrap” described by [Yung \(1997\)](#). In the mean bootstrap, each adjusted weight is derived from more than one bootstrap sample. When replicate-weight variables for the mean bootstrap are `svyset`, the `bsn()` option identifying the number of bootstrap samples used to generate the adjusted-weight variables should also be specified. This number is used in the variance calculation; see [\[SVY\] Variance estimation](#).

▷ Example 2

`nmihs_mbs.dta` is equivalent to `nmihs.dta` except that the strata identifier variable `stratan` is replaced by mean bootstrap replicate-weight variables. The replicate-weight variables and variance adjustment are already `svyset`, and the default method for variance estimation is `vce(bootstrap)`.

```

. use https://www.stata-press.com/data/r17/nmihs_mbs
. svyset
  Sampling weights: finwgt
                   VCE: bootstrap
                   MSE: off
Bootstrap weights: mbsrw1 .. mbsrw200
Bootstrap samples: 5
  Single unit: missing
  Strata 1: <one>
  Sampling unit 1: <observations>
  FPC 1: <zero>

```

Notice that the 200 mean bootstrap replicate-weight variables were generated from 5 bootstrap samples; in fact, the mean bootstrap weight variables in `nmihs_mbs.dta` were generated from the bootstrap weight variables in `nmihs_bs.dta`.

Here we use `svy: mean` to estimate the average birthweight for our population.

```

. svy, nodots: mean birthwgt
Survey: Mean estimation      Number of obs   =    9,946
                           Population size = 3,895,562
                           Replications   =     200

```

	Observed mean	Bootstrap std. err.	Normal-based [95% conf. interval]	
birthwgt	3355.452	5.712574	3344.256	3366.649

The standard error and confidence limits differ from the [previous example](#). This merely illustrates that the mean bootstrap is not numerically equivalent to the standard bootstrap, even when the replicate-weight variables are generated from the same resampled datasets.

Stored results

In addition to the results documented in [SVY] [svy](#), `svy bootstrap` stores the following in `e()`:

Scalars

<code>e(N_reps)</code>	number of replications
<code>e(N_misreps)</code>	number of replications with missing values
<code>e(k_exp)</code>	number of standard expressions
<code>e(k_eexp)</code>	number of <code>_b/_se</code> expressions
<code>e(k_extra)</code>	number of extra estimates added to <code>_b</code>
<code>e(bsn)</code>	bootstrap mean-weight adjustment

Macros

<code>e(cmdname)</code>	command name from <i>command</i>
<code>e(cmd)</code>	same as <code>e(cmdname)</code> or <code>bootstrap</code>
<code>e(vce)</code>	<code>bootstrap</code>
<code>e(exp#)</code>	<i>#</i> th expression
<code>e(bsrweight)</code>	<code>bsrweight()</code> variable list

Matrices

<code>e(b_bs)</code>	bootstrap means
<code>e(V)</code>	bootstrap variance estimates

When *exp_list* is `_b`, `svy bootstrap` will also carry forward most of the results already in `e()` from *command*.

Methods and formulas

See [SVY] [Variance estimation](#) for details regarding bootstrap variance estimation.

References

- Gonzalez, J. F., Jr., N. Krauss, and C. Scott. 1992. Estimation in the 1988 National Maternal and Infant Health Survey. *Proceedings of the Section on Statistics Education, American Statistical Association* 343–348.
- Kolenikov, S. 2010. Resampling variance estimation for complex survey data. *Stata Journal* 10: 165–199.
- McCarthy, P. J., and C. B. Snowden. 1985. The bootstrap and finite population sampling. In *Vital and Health Statistics*, 1–23. Washington, DC: U.S. Government Printing Office.
- Rao, J. N. K., and C. F. J. Wu. 1988. Resampling inference with complex survey data. *Journal of the American Statistical Association* 83: 231–241. <https://doi.org/10.2307/2288945>.
- Rao, J. N. K., C. F. J. Wu, and K. Yue. 1992. Some recent work on resampling methods for complex surveys. *Survey Methodology* 18: 209–217.
- Yeo, D., H. Mantel, and T.-P. Liu. 1999. Bootstrap variance estimation for the National Population Health Survey. In *Proceedings of the Survey Research Methods Section*, 778–785. American Statistical Association.
- Yung, W. 1997. Variance estimation for public use files under confidentiality constraints. In *Proceedings of the Survey Research Methods Section*, 434–439. American Statistical Association.

Also see

[SVY] **svy postestimation** — Postestimation tools for svy

[SVY] **svy brr** — Balanced repeated replication for survey data

[SVY] **svy jackknife** — Jackknife estimation for survey data

[SVY] **svy sdr** — Successive difference replication for survey data

[SVY] **Calibration** — Calibration for survey data

[SVY] **Poststratification** — Poststratification for survey data

[SVY] **Subpopulation estimation** — Subpopulation estimation for survey data

[SVY] **Variance estimation** — Variance estimation for survey data

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

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