svy bootstrap — Bootstrap for survey data

Syntax | Menu | Description
--- | --- | ---
Options | Remarks and examples | Stored results
Methods and formulas | References | Also see

Syntax

```plaintext
svy bootstrap `exp_list' [ , `svy_options' `bootstrap_options' `eform_option' ] : command
```

### svy_options

**Description**

- `if/in`  
  - `subpop([varname] [if])` identify a subpopulation

#### Reporting

- `level(#)` set confidence level; default is `level(95)`
- `noheader` suppress table header
- `nolegend` suppress table legend
- `noadjust` do not adjust model Wald statistic
- `nocnsreport` do not display constraints

#### display_options

- control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

- `coeflegend` display legend instead of statistics

*coeflegend* is not shown in the dialog boxes for estimation commands.
### bootstrap_options Description

<table>
<thead>
<tr>
<th>Main</th>
<th>bootstrap mean-weight adjustment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bsn(#)</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options</th>
<th>save results to filename; save statistics in double precision; save results to filename every # replications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>_saving(filename[, ...])</strong></td>
<td></td>
</tr>
<tr>
<td><strong>mse</strong></td>
<td>use MSE formula for variance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reporting</th>
<th>display the full table legend</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>verbose</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advanced</th>
<th>do not drop observations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nodrop</strong></td>
<td></td>
</tr>
<tr>
<td><strong>reject(exp)</strong></td>
<td>identify invalid results</td>
</tr>
</tbody>
</table>

### Advanced Options

<table>
<thead>
<tr>
<th><strong>dof(#)</strong></th>
<th>design degrees of freedom</th>
</tr>
</thead>
</table>

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

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 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.
svy bootstrap — Bootstrap for survey data

Description

_svy bootstrap_ performs bootstrap replication for complex survey data. Typing

```
.svy bootstrap exp_list: command
```

executes _command_ once for each replicate, using sampling weights that are adjusted according to the bootstrap methodology.

_command_ defines the statistical command to be executed. Most Stata commands and user-written programs can be used with _svy bootstrap_ as long as they follow standard Stata syntax, allow the _if_ qualifier, and allow _pweight_ and _iweight_; see [U] 11 Language syntax. The by prefix may not be part of _command_.

_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.

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_ indicates that _filename_ be overwritten if it exists. This option is not shown on 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_ suppresses display of the replication dots. By default, one dot character is printed for each successful replication. A red ‘x’ is printed if _command_ returns with an error, and ‘e’ is printed if one of the values in _exp_list_ is missing.
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, $df = N_{\text{psu}} - N_{\text{strata}}$.

Remarks and examples


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 http://www.stata-press.com/data/r13/nmihs_bs

. svyset
    pweight: finwgt
    VCE: bootstrap
    MSE: off
    bsrweight: bsrw1 bsrw2 bsrw3 bsrw4 bsrw5 bsrw6 bsrw7 bsrw8 bsrw9 bsrw10
              bsrw11 bsrw12 bsrw13 bsrw14 bsrw15 bsrw16 bsrw17 bsrw18 bsrw19
              (output omitted)
              bsrw989 bsrw990 bsrw991 bsrw992 bsrw993 bsrw994 bsrw995
              bsrw996 bsrw997 bsrw998 bsrw999 bsrw1000

Single unit: missing
Strata 1: <one>
    SU 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 = 9946
Population size = 3895562
Replications = 1000

<table>
<thead>
<tr>
<th>Observed</th>
<th>Bootstrap</th>
<th>Normal-based</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std. Err.</td>
</tr>
<tr>
<td>birthwgt</td>
<td>3355.452</td>
<td>6.520637</td>
</tr>
</tbody>
</table>

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)`.
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 = 9946
Population size = 3895562
Replications = 200

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
<th>Bootstrap</th>
<th>Normal-based</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std. Err.</td>
<td>[95% Conf. Interval]</td>
</tr>
<tr>
<td>birthwgt</td>
<td>3355.452</td>
<td>5.712574</td>
<td>3344.256 3366.649</td>
</tr>
</tbody>
</table>
```

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
- \( e(N_{\text{reps}}) \): number of replications
- \( e(N_{\text{misreps}}) \): number of replications with missing values
- \( e(k_{\text{exp}}) \): number of standard expressions
- \( e(k_{\text{eexp}}) \): number of \_b/\_se expressions
- \( e(k_{\text{extra}}) \): number of extra estimates added to \_b
- \( e(bsn) \): bootstrap mean-weight adjustment

Macros
- \( e(cmdname) \): command name from command
- \( e(cmd) \): same as \( e(cmdname) \) or bootstrap
- \( e(vce) \): bootstrap
- \( e(exp#) \): \#th expression
- \( e(bsrweight) \): bsrweight() variable list

Matrices
- \( e(b_{\text{bs}}) \): bootstrap means
- \( e(V) \): 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


Also see

[SVY] svy postestimation — Postestimation tools for svy
[R] bootstrap — Bootstrap sampling and estimation
[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
[U] 20 Estimation and postestimation commands
[SVY] poststratification — Poststratification for survey data
[SVY] subpopulation estimation — Subpopulation estimation for survey data
[SVY] variance estimation — Variance estimation for survey data