svy brr — Balanced repeated replication for survey data

Description

svy brr performs balanced repeated replication (BRR) 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 BRR methodology. Any Stata estimation command listed in [SVY] **svy estimation** may be used with svy brr. User-written programs that meet the requirements in [P] **program properties** may also be used.

Quick start

Estimate population mean of v1 using BRR standard-error estimates with sampling weight wvar1 and replicate weights in variables with prefix rwvar

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

BRR estimate of the standard error of the difference between the means of v2 and v3

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

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

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

Same as above

```
brr 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 > Balanced repeated replications estimation

Syntax

[svy] brr *exp_list* [, *svy_options brr_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 columns and 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.

brr_options	Description
Main	
hadamard(*matrix*)	Hadamard matrix
fay(#)	Fay's adjustment
Options	
saving(*filename*[, ...])	save results to *filename*; save statistics in double precision; save results to *filename* every # replications
mse	use MSE formula for variance
Reporting	
verbose	display the full table legend
nodots	suppress replication dots
dots(#)	display dots every # replications
noisily	display any output from *command*
trace	trace *command*
title(*text*)	use *text* as title for BRR results
Advanced	
nodrop	do not drop observations
reject(*exp*)	identify invalid results
dof(#)	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.

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

`hadamard(`*matrix*`)` specifies the Hadamard matrix to be used to determine which PSUs are chosen for each replicate.

`fay(`*#*`)` specifies Fay's adjustment (Judkins 1990), where 0 ≤ # ≤ 2, but excluding 1. This option overrides the `fay(`*#*`)` 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 `double`s, meaning 8-byte reals. By default, they are saved as `float`s, 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 brr` compute the variance by using deviations of the replicates from the observed value of the statistics based on the entire dataset. By default, `svy brr` 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. An “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 BRR results; the default title is “BRR 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 $= N_{\mathrm{psu}} - N_{\mathrm{strata}}$.

Remarks and examples

BRR was first introduced by McCarthy (1966, 1969a, 1969b) as a method of variance estimation for designs with two PSUs in every stratum. The BRR variance estimator tends to give more reasonable variance estimates for this design than the linearized variance estimator, which can result in large values and undesirably wide confidence intervals.

In BRR, the model is fit multiple times, once for each of a balanced set of combinations where one PSU is dropped from each stratum. The variance is estimated using the resulting replicated point estimates. Although the BRR method has since been generalized to include other designs, Stata’s implementation of BRR requires two PSUs per stratum.

To protect the privacy of survey participants, public survey datasets may contain replicate-weight variables instead of variables that identify the PSUs and strata. These replicate-weight variables are adjusted copies of the sampling weights. For BRR, the sampling weights are adjusted for dropping one PSU from each stratum; see [SVY] **Variance estimation** for more details.

▷ Example 1: BRR replicate-weight variables

The survey design for the NHANES II data (McDowell et al. 1981) is specifically suited to BRR; there are two PSUs in every stratum.

```
. use https://www.stata-press.com/data/r19/nhanes2

. svydescribe

Survey: Describing stage 1 sampling units

Sampling weights: finalwgt
             VCE: linearized
     Single unit: missing
        Strata 1: strata
 Sampling unit 1: psu
           FPC 1: <zero>

                                    Number of obs per unit
 Stratum   # units     # obs       Min      Mean       Max
──────────────────────────────────────────────────────────
       1         2       380       165     190.0       215
       2         2       185        67      92.5       118
       3         2       348       149     174.0       199
       4         2       460       229     230.0       231
       5         2       252       105     126.0       147
  (output omitted)
      29         2       503       215     251.5       288
      30         2       365       166     182.5       199
      31         2       308       143     154.0       165
      32         2       450       211     225.0       239
──────────────────────────────────────────────────────────
      31        62    10,351        67     167.0       288
```

Here is a privacy-conscious dataset equivalent to the one above; all the variables and values remain, except `strata` and `psu` are replaced with BRR replicate-weight variables. The BRR replicate-weight variables are already `svyset`, and the default method for variance estimation is `vce(brr)`.

```
. use https://www.stata-press.com/data/r19/nhanes2brr

. svyset

Sampling weights: finalwgt
             VCE: brr
             MSE: off
     BRR weights: brr_1 .. brr_32
     Single unit: missing
        Strata 1: <one>
 Sampling unit 1: <observations>
           FPC 1: <zero>
```

Suppose that we were interested in the population ratio of weight to height. Here we use `total` to estimate the population totals of `weight` and `height` and the `svy brr` prefix to estimate their ratio and variance; we use `total` instead of `ratio` (which is otherwise preferable here) to illustrate how to specify an *exp_list*.

```
. svy brr WtoH = (_b[weight]/_b[height]): total weight height
(running total on estimation sample)

BRR replications (32): .........10.........20.........30.. done

BRR results                                     Number of obs   =      10,351
                                                Population size = 117,157,513
                                                Replications    =          32
                                                Design df       =          31

      Command: total weight height
         WtoH: _b[weight]/_b[height]

------------------------------------------------------------------------------
             |                 BRR
             | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
        WtoH |   .4268116   .0008904   479.36   0.000     .4249957    .4286276
------------------------------------------------------------------------------
```

The `mse` option causes `svy brr` to use the MSE form of the BRR variance estimator. This variance estimator will tend to be larger than the previous because of the addition of the familiar squared bias term in the MSE; see [SVY] **Variance estimation** for more details. The header for the column of standard errors in the table of results is BRR * for the BRR variance estimator using the MSE formula.

```
. svy brr WtoH = (_b[weight]/_b[height]), mse: total weight height
(running total on estimation sample)

BRR replications (32): .........10.........20.........30.. done

BRR results                                     Number of obs   =      10,351
                                                Population size = 117,157,513
                                                Replications    =          32
                                                Design df       =          31

      Command: total weight height
         WtoH: _b[weight]/_b[height]

------------------------------------------------------------------------------
             |               BRR *
             | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
        WtoH |   .4268116   .0008904   479.36   0.000     .4249957    .4286276
------------------------------------------------------------------------------
```

The bias term here is too small to see any difference in the standard errors.

◁

▷ Example 2: Survey data without replicate-weight variables

For survey data with the PSU and strata variables but no replication weights, `svy brr` can compute adjusted sampling weights within its replication loop. Here the `hadamard()` option must be supplied with the name of a Stata matrix that is a Hadamard matrix of appropriate order for the number of strata in your dataset (see the following technical note for a quick introduction to Hadamard matrices).

There are 31 strata in nhanes2.dta, so we need a Hadamard matrix of order 32 (or more) to use svy brr with this dataset. Here we use h32 (from the following technical note) to estimate the population ratio of weight to height by using the BRR variance estimator.

```
. use https://www.stata-press.com/data/r19/nhanes2

. svy brr, hadamard(h32): ratio (WtoH: weight/height)
(running ratio on estimation sample)

BRR replications (32): .........10.........20.........30.. done

Survey: Ratio estimation

Number of strata = 31             Number of obs    =      10,351
Number of PSUs   = 62             Population size = 117,157,513
                                  Replications     =          32
                                  Design df        =          31

         WtoH: weight/height

--------------------------------------------------------------
             |                 BRR
             |      Ratio   std. err.     [95% conf. interval]
-------------+------------------------------------------------
        WtoH |   .4268116   .0008904      .4249957    .4286276
--------------------------------------------------------------
```

◁

❑ Technical note

A Hadamard matrix is a square matrix with r rows and columns that has the property

$$H_r' H_r = r I_r$$

where I_r is the identity matrix of order r. Generating a Hadamard matrix with order $r = 2^p$ is easily accomplished. Start with a Hadamard matrix of order 2 (H_2), and build your H_r by repeatedly applying Kronecker products with H_2. Here is the Stata code to generate the Hadamard matrix for the previous example.

```
matrix h2 = (-1, 1 \ 1, 1)
matrix h32 = h2
forvalues i = 1/4 {
        matrix h32 = h2 # h32
}
```

svy brr consumes Hadamard matrices from left to right, so it is best to make sure that r is greater than the number of strata and that the last column is the one consisting of all 1s. This will ensure full orthogonal balance according to Wolter (2007).

❑

❑ Technical note

When the svy brr prefix is used with a user-defined program and when the expression list is _b, svy brr calls

```
set coeftabresults off
```

before entering the replication loop to prevent Stata from performing unnecessary calculations. This means that, provided option noisily is not specified, estimation commands will not build or post the coefficient table matrix r(table).

If your program calls an estimation command and needs `r(table)` to exist to perform properly, then your program will need to call

```
    set coeftabresults on
```

before calling other estimation commands.

❑

Stored results

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

Scalars

`e(N_reps)`	number of replications
`e(N_misreps)`	number of replications with missing values
`e(k_exp)`	number of standard expressions
`e(k_eexp)`	number of `_b`/`_se` expressions
`e(k_extra)`	number of extra estimates added to `_b`
`e(fay)`	Fay's adjustment

Macros

`e(cmdname)`	command name from *command*
`e(cmd)`	same as `e(cmdname)` or `brr`
`e(vce)`	`brr`
`e(brrweight)`	`brrweight()` variable list

Matrices

`e(b_brr)`	BRR means
`e(V)`	BRR variance estimates

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

Methods and formulas

See [SVY] **Variance estimation** for details regarding BRR variance estimation.

References

Judkins, D. R. 1990. Fay's method for variance estimation. *Journal of Official Statistics* 6: 223–239.

McCarthy, P. J. 1966. "Replication: An approach to the analysis of data from complex surveys". In *Vital and Health Statistics*, ser. 2, no. 14. Hyattsville, MD: National Center for Health Statistics.

———. 1969a. "Pseudoreplication: Further evaluation and application of the balanced half-sample technique". In *Vital and Health Statistics*, ser. 2, no. 31. Hyattsville, MD: National Center for Health Statistics.

———. 1969b. Pseudo-replication: Half-samples. *Revue de l'Institut International de Statistique* 37: 239–264. https://doi.org/10.2307/1402116.

McDowell, A., A. Engel, J. T. Massey, and K. Maurer. 1981. "Plan and operation of the Second National Health and Nutrition Examination Survey, 1976–1980". In *Vital and Health Statistics*, ser. 1, no. 15. Hyattsville, MD: National Center for Health Statistics.

Wolter, K. M. 2007. *Introduction to Variance Estimation*. 2nd ed. New York: Springer. https://doi.org/10.1007/978-0-387-35099-8.

Also see

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

[SVY] **svy bootstrap** — Bootstrap 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

[U] 20 Estimation and postestimation commands

For suggested citations, see the FAQ on citing Stata documentation.