Title

stata.com

svy jackknife — Jackknife estimation for survey data

Description

`svy jackknife` performs jackknife estimation of specified statistics (or expressions) for a Stata command or a community-contributed command. The command is executed once for each replicate using sampling weights that are adjusted according to the jackknife methodology. Any Stata estimation command listed in [SVY] **svy estimation** may be used with `svy jackknife`. Community-contributed commands that meet the requirements in [P] **program properties** may also be used.

Quick start

Estimate population mean of `v1` using jackknife standard-error estimates with sampling weight `wvar1` and sampling units identified by `su1`

```
svyset su1 [pweight = wvar1]
svy jackknife _b: mean v1
```

Same as above, but with jackknife replication weights in variables with prefix `rwvar`

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

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

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

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

```
svy jackknife 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 > Jackknife estimation

Syntax

```
svy jackknife exp_list [ , svy_options jackknife_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.

jackknife_options	Description
Main	
`eclass`	number of observations is in `e(N)`
`rclass`	number of observations is in `r(N)`
`n(`*exp*`)`	specify *exp* that evaluates to number of observations used
Options	
`saving(`*filename*[`, ...`]`)`	save results to *filename*; save statistics in double precision; save results to *filename* every # replications
`keep`	keep pseudovalues
`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 jackknife 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*.

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 svyj 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

eclass, rclass, and n(*exp*) specify where *command* stores the number of observations on which it based the calculated results. We strongly advise you to specify one of these options.

eclass specifies that *command* store the number of observations in e(N).

rclass specifies that *command* store the number of observations in r(N).

n(*exp*) allows you to specify an expression that evaluates to the number of observations used. Specifying n(r(N)) is equivalent to specifying the rclass option. Specifying n(e(N)) is equivalent to specifying the eclass option. If *command* stores the number of observations in r(N1), specify n(r(N1)).

If you specify none of these options, svy jackknife will assume eclass or rclass depending upon which of e(N) and r(N) is not missing (in that order). If both e(N) and r(N) are missing, svy jackknife assumes that all observations in the dataset contribute to the calculated result. If that assumption is incorrect, then the reported standard errors will be incorrect. For instance, say that you specify

```
. svy jackknife coef=_b[x2]: myreg y x1 x2 x3
```

where myreg uses e(n) instead of e(N) to identify the number of observations used in calculations. Further assume that observation 42 in the dataset has x3 equal to missing. The 42nd observation plays no role in obtaining the estimates, but svy jackknife has no way of knowing that and will use the wrong N. If, on the other hand, you specify

```
. svy jackknife coef=_b[x2], n(e(n)): myreg y x1 x2 x3
```

Then svy jackknife will notice that observation 42 plays no role. The n(e(n)) option is specified because myreg is an estimation command, but it stores the number of observations used in e(n) (instead of the standard e(N)). When svy jackknife runs the regression omitting the 42nd observation, svy jackknife will observe that e(n) has the same value as when svy jackknife previously ran the regression by using all the observations. Thus svy jackknife will know that myreg did not use the observation.

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.

keep specifies that new variables be added to the dataset containing the pseudovalues of the requested statistics. For instance, if you typed

```
. svy jackknife coef=_b[x2], eclass keep: regress y x1 x2 x3
```

Then the new variable coef would be added to the dataset containing the pseudovalues for _b[x2]. Let b be defined as the value of _b[x2] when all observations are used to fit the model, and let $b(j)$ be the value when the jth observation is omitted. The pseudovalues are defined as

$$\text{pseudovalue}_j = N \times \{b - b(j)\} + b(j)$$

where N is the number of observations used to produce b.

keep implies the nodrop option.

mse specifies that svy jackknife compute the variance by using deviations of the replicates from the observed value of the statistics based on the entire dataset. By default, svy jackknife computes the variance by using deviations of the pseudovalues 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, 'e' is printed if one of the values in *exp_list* is missing, 'n' is printed if the sample size is not correct, and a yellow 's' is printed if the dropped sampling unit is outside the subpopulation sample.

`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 jackknife results; the default title is "Jackknife 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, $\text{df} = N_{\text{psu}} - N_{\text{strata}}$.

Remarks and examples

stata.com

The jackknife is

- an alternative, first-order unbiased estimator for a statistic;
- a data-dependent way to calculate the standard error of the statistic and to obtain significance levels and confidence intervals; and
- a way of producing measures called pseudovalues for each observation, reflecting the observation's influence on the overall statistic.

The idea behind the simplest form of the jackknife—the one implemented in [R] **jackknife**—is to repeatedly calculate the statistic in question, each time omitting just one of the dataset's observations. Assume that our statistic of interest is the sample mean. Let y_j be the jth observation of our data on some measurement y, where $j = 1, \ldots, N$ and N is the sample size. If $\overline{y}$ is the sample mean of y using the entire dataset and $\overline{y}_{(j)}$ is the mean when the jth observation is omitted, then

$$\overline{y} = \frac{(N-1)\,\overline{y}_{(j)} + y_j}{N}$$

Solving for y_j, we obtain

$$y_j = N\,\overline{y} - (N-1)\,\overline{y}_{(j)}$$

These are the pseudovalues that `svy: jackknife` calculates. To move this discussion beyond the sample mean, let $\widehat{\theta}$ be the value of our statistic (not necessarily the sample mean) using the entire dataset, and let $\widehat{\theta}_{(j)}$ be the computed value of our statistic with the jth observation omitted. The pseudovalue for the jth observation is

$$\widehat{\theta}^{*}_{j} = N\,\widehat{\theta} - (N-1)\,\widehat{\theta}_{(j)}$$

The mean of the pseudovalues is the alternative, first-order unbiased estimator mentioned above, and the standard error of the mean of the pseudovalues is an estimator for the standard error of $\widehat{\theta}$ (Tukey 1958, Shao and Tu 1995).

When the jackknife is applied to survey data, primary sampling units (PSUs) are omitted instead of observations, N is the number of PSUs instead of the sample size, and the sampling weights are adjusted owing to omitting PSUs; see [SVY] **variance estimation** for more details.

Because of privacy concerns, many public survey datasets contain jackknife replication-weight variables instead of variables containing information on the PSUs and strata. These replication-weight variables are the adjusted sampling weights, and there is one replication-weight variable for each omitted PSU.

▷ Example 1: Jackknife with information on PSUs and strata

Suppose that we were interested in a measure of association between the weight and height of individuals in the population represented by the NHANES II data (McDowell et al. 1981). To measure the association, we will use the slope estimate from a linear regression of `weight` on `height`. We also use `svy jackknife` to estimate the variance of the slope.

```
. use http://www.stata-press.com/data/r15/nhanes2
. svyset
      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: strata
         SU 1: psu
        FPC 1: <zero>
. svy jackknife slope = _b[height]: regress weight height
(running regress on estimation sample)
Jackknife replications (62)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
............
Linear regression
Number of strata   =        31                  Number of obs      =       10,351
Number of PSUs     =        62                  Population size    =  117,157,513
                                                Replications       =           62
                                                Design df          =           31
      command:  regress weight height
        slope:  _b[height]
           n():  e(N)
------------------------------------------------------------------------------
             |             Jackknife
             |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       slope |   .8014753   .0160281    50.00   0.000     .7687858    .8341648
------------------------------------------------------------------------------
```

▷ Example 2: Jackknife replicate-weight variables

`nhanes2jknife.dta` is a privacy-conscious dataset equivalent to `nhanes2.dta`; all the variables and values remain, except that `strata` and `psu` are replaced with jackknife replicate-weight variables. The replicate-weight variables are already `svyset`, and the default method for variance estimation is `vce(jackknife)`.

```
. use http://www.stata-press.com/data/r15/nhanes2jknife
. svyset
      pweight: finalwgt
          VCE: jackknife
          MSE: off
    jkrweight: jkw_1 .. jkw_62
  Single unit: missing
     Strata 1: <one>
         SU 1: <observations>
        FPC 1: <zero>
```

Here we perform the same analysis as in the previous example, using jackknife replication weights.

```
. svy jackknife slope = _b[height], nodots: regress weight height
Linear regression
Number of strata   =        31                Number of obs     =       10,351
                                               Population size   =  117,157,513
                                               Replications      =           62
                                               Design df         =           31
      command:  regress weight height
        slope:  _b[height]
```

	Coef.	Jackknife Std. Err.	t	P>|t|	[95% Conf.	Interval]
slope	.8014753	.0160281	50.00	0.000	.7687858	.8341648

The `mse` option causes `svy jackknife` to use the MSE form of the jackknife 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 `Jknife *` for the jackknife variance estimator, which uses the MSE formula.

```
. svy jackknife slope = _b[height], mse nodots: regress weight height
Linear regression
Number of strata   =        31                Number of obs     =       10,351
                                               Population size   =  117,157,513
                                               Replications      =           62
                                               Design df         =           31
      command:  regress weight height
        slope:  _b[height]
```

	Coef.	Jknife * Std. Err.	t	P>|t|	[95% Conf.	Interval]
slope	.8014753	.0160284	50.00	0.000	.7687852	.8341654

◁

Stored results

In addition to the results documented in [SVY] **svy**, `svy jackknife` 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`

Macros

`e(cmdname)`	command name from *command*
`e(cmd)`	same as `e(cmdname)` or `jackknife`
`e(vce)`	`jackknife`
`e(exp#)`	#th expression
`e(jkrweight)`	`jkrweight()` variable list

Matrices

`e(b_jk)`	jackknife means
`e(V)`	jackknife variance estimates

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

Methods and formulas

See [SVY] **variance estimation** for details regarding jackknife variance estimation.

References

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. *Vital and Health Statistics* 1(15): 1–144.

Shao, J., and D. Tu. 1995. *The Jackknife and Bootstrap*. New York: Springer.

Tukey, J. W. 1958. Bias and confidence in not-quite large samples. Abstract in *Annals of Mathematical Statistics* 29: 614.

Also see

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

[R] **jackknife** — Jackknife estimation

[SVY] **svy bootstrap** — Bootstrap for survey data

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

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

[U] 20 Estimation and postestimation commands

[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