Title

svy — The survey prefix command

Syntax

svy [*vcetype*] [, *svy_options eform_option*] : *command*

vcetype	Description
SE	
linearized	Taylor-linearized variance estimation
bootstrap	bootstrap variance estimation; see [SVY] **svy bootstrap**
brr	BRR variance estimation; see [SVY] **svy brr**
jackknife	jackknife variance estimation; see [SVY] **svy jackknife**
sdr	SDR variance estimation; see [SVY] **svy sdr**

Specifying a *vcetype* overrides the default from svyset.

svy_options	Description
if/in	
subpop([*varname*] [*if*])	identify a subpopulation
SE	
dof(#)	design degrees of freedom
bootstrap_options	more options allowed with bootstrap variance estimation; see [SVY] ***bootstrap_options***
brr_options	more options allowed with BRR variance estimation; see [SVY] ***brr_options***
jackknife_options	more options allowed with jackknife variance estimation; see [SVY] ***jackknife_options***
sdr_options	more options allowed with SDR variance estimation; see [SVY] ***sdr_options***
Reporting	
level(#)	set confidence level; default is level(95)
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
noheader	suppress table header
nolegend	suppress table legend
noadjust	do not adjust model Wald statistic
noisily	display any output from *command*
trace	trace *command*
coeflegend	display legend instead of statistics

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

mi estimate may be used with svy linearized if the estimation command allows mi estimate; it may not be used with svy bootstrap, svy brr, svy jackknife, or svy sdr.

noheader, nolegend, noadjust, noisily, trace, and coeflegend are not shown in the dialog boxes for 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.

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

Description

svy fits statistical models for complex survey data. Typing

```
. svy: command
```

executes *command* while accounting for the survey settings identified by svyset.

command defines the estimation command to be executed. Not all estimation commands are supported by svy. See [SVY] **svy estimation** for a list of Stata's estimation commands that are supported by svy. See [P] **program properties** for a discussion of what is required for svy to support an estimation command. The by prefix may not be part of *command*.

Options

if/in

subpop(*subpop*) specifies that estimates be computed for the single subpopulation identified by *subpop*, which is

[*varname*] [*if*]

Thus the subpopulation is defined by the observations for which *varname* $\neq 0$ that also meet the if conditions. Typically, *varname* $= 1$ defines the subpopulation, and *varname* $= 0$ indicates observations not belonging to the subpopulation. For observations whose subpopulation status is uncertain, *varname* should be set to a missing value; such observations are dropped from the estimation sample.

See [SVY] **subpopulation estimation** and [SVY] **estat**.

SE

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

bootstrap_options are other options that are allowed with bootstrap variance estimation specified by svy bootstrap or specified as svyset using the vce(bootstrap) option; see [SVY] ***bootstrap_options***.

brr_options are other options that are allowed with BRR variance estimation specified by svy brr or specified as svyset using the vce(brr) option; see [SVY] ***brr_options***.

jackknife_options are other options that are allowed with jackknife variance estimation specified by svy jackknife or specified as svyset using the vce(jackknife) option; see [SVY] ***jackknife_options***.

sdr_options are other options that are allowed with SDR variance estimation specified by svy sdr or specified as svyset using the vce(sdr) option; see [SVY] ***sdr_options***.

Reporting

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **20.7 Specifying the width of confidence intervals**.

`nocnsreport`; see [R] **estimation options**.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(`*%fmt*`)`, `pformat(`*%fmt*`)`, `sformat(`*%fmt*`)`, and `nolstretch`; see [R] **estimation options**.

The following options are available with `svy` but are not shown in the dialog boxes:

`noheader` prevents the table header from being displayed. This option implies `nolegend`.

`nolegend` prevents the table legend identifying the subpopulations from being displayed.

`noadjust` specifies that the model Wald test be carried out as $W/k \sim F(k,d)$, where W is the Wald test statistic, k is the number of terms in the model excluding the constant term, d is the total number of sampled PSUs minus the total number of strata, and $F(k,d)$ is an F distribution with k numerator degrees of freedom and d denominator degrees of freedom. By default, an adjusted Wald test is conducted: $(d-k+1)W/(kd) \sim F(k,d-k+1)$.

See Korn and Graubard (1990) for a discussion of the Wald test and the adjustments thereof. Using the `noadjust` option is not recommended.

`noisily` requests that any output from *command* be displayed.

`trace` causes a trace of the execution of *command* to be displayed.

`coeflegend`; see [R] **estimation options**.

The following option is usually available with `svy` at the time of estimation or on replay but is not shown in all dialog boxes:

eform_option; see [R] ***eform_option***.

Remarks and examples

stata.com

The `svy` prefix is designed for use with complex survey data. Typical survey design characteristics include sampling weights, one or more stages of clustered sampling, and stratification. For a general discussion of various aspects of survey designs, including multistage designs, see [SVY] **svyset**.

Below we present an example of the effects of weights, clustering, and stratification. This is a typical case, but drawing general rules from any one example is still dangerous. You could find particular analyses from other surveys that are counterexamples for each of the trends for standard errors exhibited here.

▷ Example 1: The effects of weights, clustering, and stratification

We use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981) as our example. This is a national survey, and the dataset has sampling weights, strata, and clustering. In this example, we will consider the estimation of the mean serum zinc level of all adults in the United States.

First, consider a proper design-based analysis, which accounts for weighting, clustering, and stratification. Before we issue our `svy` estimation command, we set the weight, strata, and PSU identifier variables:

```
. use http://www.stata-press.com/data/r13/nhanes2f
. svyset psuid [pweight=finalwgt], strata(stratid)
      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: stratid
         SU 1: psuid
        FPC 1: <zero>
```

We now estimate the mean by using the proper design-based analysis:

```
. svy: mean zinc
(running mean on estimation sample)
Survey: Mean estimation
Number of strata =      31        Number of obs     =       9189
Number of PSUs   =      62        Population size   =  104176071
                                  Design df         =         31
```

	Mean	Linearized Std. Err.	[95% Conf.	Interval]
zinc	87.18207	.4944827	86.17356	88.19057

If we ignore the survey design and use `mean` to estimate the mean, we get

```
. mean zinc
Mean estimation                          Number of obs     =     9189
```

	Mean	Std. Err.	[95% Conf.	Interval]
zinc	86.51518	.1510744	86.21904	86.81132

The point estimate from the unweighted analysis is smaller by more than one standard error than the proper design-based estimate. Also, design-based analysis produced a standard error that is 3.27 times larger than the standard error produced by our incorrect analysis.

◁

▷ Example 2: Halfway is not enough—the importance of stratification and clustering

When some people analyze survey data, they say, “I know I have to use my survey weights, but I will just ignore the stratification and clustering information.” If we follow this strategy, we will obtain the proper design-based point estimates, but our standard errors, confidence intervals, and test statistics will usually be wrong.

To illustrate this effect, suppose that we used the `svy: mean` procedure with `pweights` only.

```
. svyset [pweight=finalwgt]

      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: <one>
         SU 1: <observations>
        FPC 1: <zero>

. svy: mean zinc
(running mean on estimation sample)

Survey: Mean estimation

Number of strata =       1        Number of obs     =       9189
Number of PSUs   =    9189        Population size   =  104176071
                                  Design df         =       9188

--------------------------------------------------------------
             |             Linearized
             |       Mean   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
        zinc |   87.18207   .1828747      86.82359    87.54054
--------------------------------------------------------------
```

This approach gives us the same point estimate as our design-based analysis, but the reported standard error is less than one-half the design-based standard error. If we accounted only for clustering and weights and ignored stratification in NHANES II, we would obtain the following analysis:

```
. svyset psuid [pweight=finalwgt]

      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: <one>
         SU 1: psuid
        FPC 1: <zero>

. svy: mean zinc
(running mean on estimation sample)

Survey: Mean estimation

Number of strata =       1        Number of obs     =       9189
Number of PSUs   =       2        Population size   =  104176071
                                  Design df         =          1

--------------------------------------------------------------
             |             Linearized
             |       Mean   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
        zinc |   87.18207   .7426221      77.74616    96.61798
--------------------------------------------------------------
```

Here our standard error is about 50% larger than what we obtained in our proper design-based analysis.

◁

▷ Example 3

Let's look at a regression. We model zinc on the basis of age, weight, sex, race, and rural or urban residence. We compare a proper design-based analysis with an ordinary regression (which assumes independent and identically distributed error).

Here is our design-based analysis:

```
. svyset psuid [pweight=finalwgt], strata(stratid)

      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: stratid
         SU 1: psuid
        FPC 1: <zero>

. svy: regress zinc age c.age#c.age weight female black orace rural
(running regress on estimation sample)

Survey: Linear regression

Number of strata   =        31            Number of obs     =       9189
Number of PSUs     =        62            Population size   =  104176071
                                          Design df         =         31
                                          F(   7,      25)  =      62.50
                                          Prob > F          =     0.0000
                                          R-squared         =     0.0698

------------------------------------------------------------------------------
             |             Linearized
        zinc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  -.1701161   .0844192    -2.02   0.053    -.3422901     .002058
             |
 c.age#c.age |   .0008744   .0008655     1.01   0.320    -.0008907    .0026396
             |
      weight |   .0535225   .0139115     3.85   0.001     .0251499    .0818951
      female |  -6.134161   .4403625   -13.93   0.000    -7.032286   -5.236035
       black |  -2.881813   1.075958    -2.68   0.012    -5.076244    -.687381
       orace |  -4.118051   1.621121    -2.54   0.016    -7.424349   -.8117528
       rural |  -.5386327   .6171836    -0.87   0.390    -1.797387    .7201216
       _cons |   92.47495   2.228263    41.50   0.000     87.93038    97.01952
------------------------------------------------------------------------------
```

If we had improperly ignored our survey weights, stratification, and clustering (that is, if we had used the usual Stata `regress` command), we would have obtained the following results:

```
. regress zinc age c.age#c.age weight female black orace rural
      Source |       SS       df       MS              Number of obs =    9189
-------------+------------------------------           F(  7,  9181) =   79.72
       Model |  110417.827     7  15773.9753           Prob > F      =  0.0000
    Residual |   1816535.3  9181   197.85811           R-squared     =  0.0573
-------------+------------------------------           Adj R-squared =  0.0566
       Total |  1926953.13  9188  209.724982           Root MSE      =  14.066

------------------------------------------------------------------------------
        zinc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   -.090298   .0638452    -1.41   0.157    -.2154488    .0348528
             |
 c.age#c.age |  -.0000324   .0006788    -0.05   0.962    -.0013631    .0012983
             |
      weight |   .0606481   .0105986     5.72   0.000     .0398725    .0814237
      female |  -5.021949   .3194705   -15.72   0.000    -5.648182   -4.395716
       black |  -2.311753   .5073536    -4.56   0.000    -3.306279   -1.317227
       orace |  -3.390879   1.060981    -3.20   0.001    -5.470637   -1.311121
       rural |  -.0966462   .3098948    -0.31   0.755    -.7041089    .5108166
       _cons |   89.49465   1.477528    60.57   0.000     86.59836    92.39093
------------------------------------------------------------------------------
```

The point estimates differ by 3%–100%, and the standard errors for the proper designed-based analysis are 30%–110% larger. The differences are not as dramatic as we saw with the estimation of the mean, but they are still substantial.

◁

Stored results

svy stores the following in e():

Scalars

e(N)	number of observations
e(N_sub)	subpopulation observations
e(N_strata)	number of strata
e(N_strata_omit)	number of strata omitted
e(singleton)	1 if singleton strata, 0 otherwise
e(census)	1 if census data, 0 otherwise
e(F)	model *F* statistic
e(df_m)	model degrees of freedom
e(df_r)	variance degrees of freedom
e(N_pop)	estimate of population size
e(N_subpop)	estimate of subpopulation size
e(N_psu)	number of sampled PSUs
e(stages)	number of sampling stages
e(k_eq)	number of equations in e(b)
e(k_aux)	number of ancillary parameters
e(p)	*p*-value
e(rank)	rank of e(V)

Macros

e(prefix)	svy
e(cmdname)	command name from *command*
e(cmd)	same as e(cmdname) or e(vce)
e(command)	*command*
e(cmdline)	command as typed
e(wtype)	weight type
e(wexp)	weight expression
e(wvar)	weight variable name
e(singleunit)	singleunit() setting
e(strata)	strata() variable
e(strata#)	variable identifying strata for stage #
e(psu)	psu() variable
e(su#)	variable identifying sampling units for stage #
e(fpc)	fpc() variable
e(fpc#)	FPC for stage #
e(title)	title in estimation output
e(poststrata)	poststrata() variable
e(postweight)	postweight() variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(mse)	mse, if specified
e(subpop)	*subpop* from subpop()
e(adjust)	noadjust, if specified
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins

Matrices

e(b)	estimates
e(V)	design-based variance
e(V_srs)	simple-random-sampling-without-replacement variance, $\widehat{V}_{\rm srswor}$
e(V_srssub)	subpopulation simple-random-sampling-without-replacement variance, $\widehat{V}_{\rm srswor}$ (created only when subpop() is specified)

e(V_srswr)	simple-random-sampling-with-replacement variance, $\widehat{V}_{\rm srswr}$ (created only when fpc() option is svyset)
e(V_srssubwr)	subpopulation simple-random-sampling-with-replacement variance, $\widehat{V}_{\rm srswr}$ (created only when subpop() is specified)
e(V_modelbased)	model-based variance
e(V_msp)	variance from misspecified model fit, $\widehat{V}_{\rm msp}$
e(_N_strata_single)	number of strata with one sampling unit
e(_N_strata_certain)	number of certainty strata
e(_N_strata)	number of strata
Functions	
e(sample)	marks estimation sample

svy also carries forward most of the results already in e() from *command*.

Methods and formulas

See [SVY] **variance estimation** for all the details behind the point estimate and variance calculations made by svy.

References

Korn, E. L., and B. I. Graubard. 1990. Simultaneous testing of regression coefficients with complex survey data: Use of Bonferroni t statistics. *American Statistician* 44: 270–276.

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.

Also see

[SVY] **svy estimation** — Estimation commands for survey data

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

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

[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] **svyset** — Declare survey design for dataset

[P] **_robust** — Robust variance estimates

[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

[P] **program properties** — Properties of user-defined programs