STATA TECHNICAL BULLETIN

May 1996

STB-31

A publication to promote communication among Stata users



Subscriptions are available from Stata Corporation, email stata@stata.com, telephone 979-696-4600 or 800-STATAPC, fax 979-696-4601. Current subscription prices are posted at www.stata.com/bookstore/stb.html.

Previous Issues are available individually from StataCorp. See www.stata.com/bookstore/stbj.html for details.

Submissions to the STB, including submissions to the supporting files (programs, datasets, and help files), are on a nonexclusive, free-use basis. In particular, the author grants to StataCorp the nonexclusive right to copyright and distribute the material in accordance with the Copyright Statement below. The author also grants to StataCorp the right to freely use the ideas, including communication of the ideas to other parties, even if the material is never published in the STB. Submissions should be addressed to the Editor. Submission guidelines can be obtained from either the editor or StataCorp.



The *Stata Technical Bulletin* (ISSN 1097-8879) is published six times per year by Stata Corporation. Stata is a registered trademark of Stata Corporation.

Contents of this issue

## an59 Statement from the new editor

H. Joseph Newton, Stata Technical Bulletin

This issue marks the beginning of my term as editor of the *Stata Technical Bulletin*. I approach this task with excitement and some trepidation. The yeoman work of Sean Becketti and his Associate Editors is a hard act to follow but I will do my best.

I'd like to say a few words about myself and then discuss what I see of the future for the STB. I received my PhD in statistics at SUNY/Buffalo under the direction of Emanuel Parzen in the area of multiple time series analysis. I came to the Department of Statistics in 1978 as an assistant professor and am now professor and head of the department (you can check out our web pages at `http://stat.tamu.edu`).

The main thrust of my research activities remain in the area of time series analysis though recently I have spent considerable effort in space-time processes and computational statistics. I am the President of the Interface Foundation of North America, the non-profit corporation that organizes the annual Symposium on the Interface of Computing Science and Statistics. I have been the New Directions in Computing editor of the *American Statistician* for several years. In 1988 I published a time series text and software package called *TIMESLAB* (Wadsworth and Brooks/Cole). Thus computing is something I care very much about.

Since Stata Corporation moved from California to College Station a few years ago, I have been fortunate to get to know Bill Gould and the other people who have put Stata together, and I can only say that they are true workaholics who are obsessed with getting things right. When I was asked to become editor of the STB, it was very easy to say yes because of the quality of the people I will be working with.

I have several goals for the STB, goals that every editor of every publication strives for. First, I'd like to expand the range of articles while preserving the features that have made the STB useful to its readers. I am particularly interested in time series and spatial statistics. I'd love to hear from anyone who has written ado files for time series analysis. Articles on data management, interesting data sets, and education are always of great interest.

Speaking of education, the movement to using advanced computer interfaces in teaching statistics continues to grow with leaps and bounds. Using the World Wide Web for instruction will become more and more important. One example of this is our department at Texas A&M where we have established a computer lab with 25 PCs to be used in all of our undergraduate instruction (approximately 1,100 students per semester use the lab at least once a week). We use StataQuest and its dialog boxes and good graphics. I have written a set of ado files which illustrate many of the concepts in introductory courses. I am hoping that the STB will be a means for Stata users to communicate work they have done in this area.

In addition to expanding the range of articles, I hope to increase the number of authors. I have been monitoring Statalist on the Internet and have been very impressed with the care that many correspondents have used in the discussion. I have seen several discussions that would make excellent reading in the STB. Don't be surprised if I invite people to submit an expanded version of something on statalist for publication in the STB.

I welcome any comments or suggestions you or your co-workers, students, and others may have. I will do my best to make the STB something you find useful. If it's not, please let me know.

## an60 STB-25 – STB-30 available in bound format

Patricia Branton, Stata Corporation, stata@stata.com

The fifth year of the *Stata Technical Bulletin* (issues 25–30) has been reprinted in a bound book called *The Stata Technical Bulletin Reprints, Volume 5*. The volume of reprints is available from StataCorp for $25, plus shipping. Or, you may purchase all five reprint volumes for $100, plus shipping. Authors of inserts in STB-25 – STB-30 will automatically receive the book at no charge and need not order.

This book of reprints includes everything that appeared in issues 25–30 of the STB. As a consequence, you do not need to purchase the reprints if you saved your STBs. However, many subscribers find the reprints useful since they are bound in a volume that is the same size as the Stata manuals. Our primary reason for reprinting the STB, though, is to make it easier and cheaper for new users to obtain back issues. For those not purchasing the reprints, note that *zz6* in this issue provides a cumulative index for the fifth year of the original STBs.

| svy1 | Some basic concepts for design-based analysis of complex survey data |
|---|---|

John L. Eltinge, Texas A&M University, FAX 409-845-3144, EMAIL jeltinge@stat.tamu.edu
William M. Sribney, Stata Corporation, FAX 409-696-4601, EMAIL tech_support@stata.com

Several articles (*svy1–5*) in this issue cover "design-based" analyses of complex survey data. Before we launch into the details of syntax and formulas, it is useful to outline some fundamental concepts and principles. This will lead to brief explanations of

1. What "complex sample designs" are, and why we use them.
2. How "design-based" analyses of complex survey data differ from "independent, identically distributed (iid) based" analyses, and why the design-based approach is preferred.

For reasons of space, this discussion will be brief and informal. For more detailed introductions to complex survey data analysis, see, e.g., Scheaffer et al. (1996), Stuart (1984), and Williams (1978). Advanced treatments and discussion of important special topics are given by Cochran (1977), Särndal et al. (1992), Skinner et al. (1989), Thompson (1992), and Wolter (1985).

## Complex sample designs: Clusters, strata, and selection probabilities

When we carry out a large-scale survey, we often cannot afford to select our interviewed units through simple random sampling. For example, as part of a hypertension study, suppose we plan to select 2000 persons; we will interview and take blood-pressure readings on each. If we selected them through simple random sampling from the full U.S. population of 260 million people, our interviewers would have to travel to hundreds of different locations, and our travel costs would be prohibitively high.

We can reduce these costs substantially by selecting a relatively small number of counties (80, say), and then carrying out our interviews and examinations exclusively within those selected counties. This strategy is called "cluster sampling," and the individual counties are called "clusters" or "primary sampling units" (PSUs). Taking the cost-reduction strategy further, we can select a few city blocks within each selected county, select a few households within each city block, and then select a few persons within each household. The resulting extension of cluster sampling is called "multistage sampling," and the finer-level selected units are called secondary sampling units (city blocks), tertiary sampling units (households), and elements (individual interviewed persons), respectively.

Clustering is good because it reduces our interview costs. However, there is also a downside to cluster sampling: a cluster sample of 2000 persons, say, generally will produce estimators that are less precise than we could obtain from a simple random sample of 2000 persons. To recover some of this lost precision, we use a refinement known as "stratification." To carry out stratification, we note that the U.S. contains about 3000 counties, and that we can group these counties into 20–40 "strata," say, based on population size and socioeconomic factors. We select a few counties within each stratum, and then select city blocks within counties, and so on, as before. The resulting design is called "stratified multistage sampling."

Finally, we need to specify the way in which we will select our counties, city blocks, households, and persons. Under our stratified multistage design, one option is to use simple random sampling at each stage; i.e., select a simple random sample of counties within each stratum, then select a simple random sample of blocks within each selected county, and so on. Because different counties contain different numbers of people, this design gives some people a higher probability of selection than other people. Another option is to use random sampling with *unequal* selection probabilities at some stages; for example, for counties A and B in stratum 1, we may give county A a higher probability of selection than county B. This again generally gives some people a higher probability of selection than others. This may be desirable, for example, when we estimate parameters for special subpopulations, but the unequal selection probabilities will add a further complication to our analyses.

## Four essential ideas for design-based analysis

Over the past several decades, the sample survey literature has developed a broad set of methods for the analysis of data collected through a complex design. Within that literature, four themes are intertwined.

1. *Farewell to iid.* Due to the combined effects of clustering, stratification, and unequal selection probabilities, it generally is not plausible to view our survey observations as independent and identically distributed. For that reason, it generally is not appropriate to analyze complex survey data with methods based on iid assumptions (e.g., the `ci` and `regress` commands in Stata).
2. *Design-induced randomness.* Development of an alternative analysis method requires us to re-examine the basic concept of randomness, and its practical impact on our evaluation of bias, precision, and confidence interval coverage. Evaluation of a design based analysis method is based on the "randomness" of our observations *induced by our complex sampling design.*

3. *Accounting for the complex survey design.* The design-based approach leads directly to estimation and inference methods intended to "account for the sample design." The main features of these methods are "probability weighted" point estimators and variance estimators that account for stratification, clustering, and unequal selection probabilities.

4. *How exactly did you get your data?* As we carry out a design-based analysis, it is essential to identify clearly the strata and clusters used in our complex sample design.

## Design-based inference

Viewed at a very general level, analysis methods for complex survey data follow the same general strategy as in other areas of applied statistics. For example, we want a point estimator that is approximately unbiased and that has a relatively small variance, and we want to construct confidence intervals that have coverage probabilities close to their nominal levels $1 - \alpha$. However, to discuss unbiasedness, variance, and coverage probability under a complex design, we need a clear vision of the sense in which our observations are "random."

The design-based approach views our population as fixed and views the "randomness" of our observations as arising *only from the random-selection process of our complex design.* Under this approach, our hypertension survey is a snapshot of the fixed hypertensive status of each of 260 million Americans at a given time, and the "randomness" of our 2000 observations arises only from the fact that we randomly selected certain counties, city blocks, households, and individuals. Expectations and variances then are evaluated *with respect to the sampling design.* For example, an expectation is defined to be the "long-run average" we would obtain if we used our design to sample many times from the original population.

Note especially that the design-based approach is conceptually different from model-based approaches that are commonly used in other areas of applied statistics and that also can be used for some survey analyses. Under model-based approaches (including both iid-based approaches and more sophisticated modeling work), expectations, variances, and coverage probabilities are evaluated with respect to a specific model or class of models. In particular, we focus on models (including specific assumptions regarding heteroscedasticity of observations and correlation among observations) that we consider appropriate for our data, based on a mixture of prior experience and diagnostic checking. Thus, the "randomness" considered in a model-based approach is the randomness associated with the model we believe we have.

By contrast, the "randomness" considered under the design-based approach is restricted to the randomness we are sure we have: the specific random process by which we selected our sample units. In this sense, we can describe the design-based approach as very parsimonious or perhaps even minimalist. As with other minimalist endeavors, the design-based approach has a powerful intellectual appeal, within reasonable limits.

In practical terms, it is common to view design-based methods as more "robust" (e.g., against model failure), and model-based methods as potentially more "efficient" (e.g., having smaller reported standard errors); see, e.g., Kott (1991, Section 9). That's a good summary, but we shouldn't push it too far. Good design-based analysts pay careful attention to efficiency issues, and good model-based analysts devote serious effort to robustness issues and model checks. Our own view is that both approaches, carefully implemented, have appropriate places in the analysis of complex survey data. The `svy` commands introduced in *svy2–5* implement design-based methods. In future releases, we plan to provide a broad range of commands to implement both design-based and model-based methods with appropriate diagnostics. This will allow users to carry out side-by-side analyses and model checks, and thus obtain a clear view of the merits and limitations of each approach for their particular survey.

## What's wrong with iid-based methods?

In general, it is not appropriate to use iid-based analysis methods with data collected through a complex sample design. There are two fundamental problems. First, an iid-based point estimator generally will be biased under a complex design. This bias is of special concern for cases in which there is association between the selection probabilities and the observed values. For example, in the U.S., older adults tend to have hypertension more often than younger adults. Thus, if our sample design gave older adults a higher probability of selection than younger adults, then we would have a strong reason for concern regarding selection bias in a standard iid-based estimator of hypertension prevalence.

Second, variance estimators based on iid assumptions also tend to be biased, primarily because they do not account fully for changes in variability induced by the use of stratification, clustering and unequal selection probabilities. For example, in heavily clustered designs like our hypertension example, iid-based variance estimators tend to underestimate the true variances. Consequently, direct application of iid-based analysis methods to clustered-design data may lead to confidence intervals with coverage rates that are substantially lower than the nominal rate $1 - \alpha$; and equivalently, associated hypothesis tests may have Type I error rates that are substantially higher than their nominal level $\alpha$. The latter coverage- and error-rate problems can be of serious concern even for cases in which selection bias appears to be relatively minor.

Taken together, these results indicate that if we use an iid-based variance estimator with cluster-survey data, we run a substantial risk of understating the uncertainty attached to our estimates. This is consistent with what we would expect intuitively.

For example, recall that in our hypertension example, we avoided simple random sampling of 2000 persons because it would have been too expensive. Instead, we selected our 2000 persons through a cluster-sample design, which led to tolerable interview and travel costs. It is not surprising that the less-expensive cluster sample of 2000 persons contains somewhat less information than a simple random sample of the same size. In an informal sense, if we used iid-based variance estimators with our heavily clustered hypertension survey data, we would, in essence, be ignoring this loss of information.

### Design-based methods

The design-based approach leads to point estimators and variance estimators that differ from those we would use under the assumption of iid observations. In an informal sense, these design methods give us point estimators that adjust for the effects of unequal selection probabilities, and variance estimators that account for the losses (or gains) in precision induced by the sampling design.

Specifically, design-based point estimators use weighted sums of our observations, where the weight for a given survey element (e.g., an interviewed person) is inversely proportional to that element's probability of selection. For example, suppose that in our hypertension survey, the selection probability for each rural resident was only one-half of the selection probability for each urban resident. Then our point estimators will give each "rural" observation twice the weight of each "urban" observation.

In addition, design-based variance estimators account for changes in variability associated with stratification, clustering, and unequal selection probabilities. These modified variance estimators are then used to compute design-based confidence intervals and test statistics. In this issue of *Stata Technical Bulletin*, articles *svy2 and svy4* give some details regarding computation of design-based point estimators and variance estimators.

### Distinctions between strata and clusters

Analyses of survey data depend heavily on the proper identification of the complex design that was used to collect our data. At a minimum, we need to identify our selection probabilities, our strata, and our primary sampling units. This identification can be complicated by the fact that the survey literature often uses the term "cluster" for a primary sampling unit and by the fact that some government surveys and some large health studies use the terms "stratum" and "cluster" for groupings that are very different from those described previously. Consequently, it is useful to note three points regarding the distinction between a "stratum" and a "cluster" in design-based survey analysis.

1. Each element (e.g., person) in our population belongs to exactly one primary sampling unit (e.g., a county). The set of primary sampling units (*PSU*s) is partitioned into one or more strata, so that each *PSU* belongs to exactly one stratum. For a given survey design, the resulting "stratum boundaries" are viewed as fixed or predetermined. We will select one or more PSUs from each of our strata.

2. As noted previously, design-based survey analysts evaluate estimator performance under hypothetical repeated sampling from the specified design. Under this concept of "repeated sampling," the same strata are used each time. Consequently, there is no "randomness" associated with the particular strata encountered in a given survey dataset.

3. On the other hand, a given survey generally will select only a few PSUs from a large number of PSUs contained in a given stratum. The set of specific PSUs included in a specific survey dataset is a result of the random-sampling process. If we used the same sample design repeatedly, then we would anticipate selection of a different set of PSUs each time.

To illustrate the distinction between strata and clusters, consider a group of 50 school districts in southeast Texas. The population of interest covered all school principals in each of these 50 districts. The main questionnaire variables involved the principals' interest in a continuing-education program. Three possible designs were as follows; these are simplified versions of designs that were considered for a 1990 survey.

*Option 1: Stratified random sampling.* Within each of the 50 districts, select two principals through simple random sampling. Under this design, each of the districts is a separate stratum, each principal is a primary sampling unit, and two PSUs are selected from each stratum. There is no subsampling within a selected PSU. Repeated use of this design would always use the same 50 strata, but would result in the selection of a new set of principals each time.

*Option 2: Simple random sampling of districts.* Select five of the 50 districts through simple random sampling without replacement. Interview all principals contained in a given selected district. Under this design, there is only one stratum, each of the districts is a separate primary sampling unit and we select five of these PSUs. In addition, there is only one stage of sampling, selection of districts. Each principal is an element within a selected PSU. Repeated use of this design will result in the selection of different sets of five districts, and thus in selection of different sets of principals.

*Option 3: Stratified multistage sampling of districts and principals.* Group the 50 districts into two strata such that stratum 1 contains the 15 districts with the highest enrollment and stratum 2 contains the remaining 35 districts. Within stratum 1, select 3 of the 15 districts through simple random sampling without replacement, and similarly select 7 of the 35 districts

in stratum 2. Within each of the ten selected districts, select twelve principals through simple random sampling without replacement. Under this design, we have two strata. We select three PSUs from the first stratum and seven PSUs from the second. The primary sampling units are the school districts, and the secondary sampling units are the individual principals. Repeated use of this design will always use the same two strata, but will result in the selection of different sets of districts and of principals within districts.

## Closing remarks: Current and future svy commands

The next four articles in this *Stata Technical Bulletin* outline a set of commands for design-based analysis of survey data. *svy2* covers estimation of means, totals, ratios, and proportions using the commands `svymean`, `svytotal`, `svyratio`, and `svyprop`. *svy2* presents syntax and computational formulas, and also discusses some special features, e.g., design effects, that are used heavily in the analysis of complex survey data. *svy3* introduces some simple descriptive tools that are useful for initial exploration of survey datasets, e.g., stratum and PSU counts, and missing-data rates. *svy4* covers more complicated analyses involving linear regression, logistic regression, and probit modeling, with the commands `svyreg`, `svylogit`, and `svyprobt`. *svy5* supplements *svy2* and *svy4* with the additional commands `svylc` and `svytest` for estimation of linear combinations of parameters and for associated hypothesis tests. [*Editor's note: The ado files for the commands described in svy2–5 are all contained the directory svy1.*]

In future releases, we plan to present additional `svy` commands for estimation of distribution functions and quantiles, contingency table analyses, missing-data adjustments, auxiliary-data-based estimates of means and totals, and other important analyses.

## References

Cochran, W. G. 1977. *Sampling Techniques.* 3d ed. New York: John Wiley & Sons.

Kott, P. S. 1991. A model-based look at linear regression with survey data. *American Statistician* 45: 107–112.

Särndal, C.-E., B. Swensson, and J. Wretman. 1992. *Model Assisted Survey Sampling.* New York: Springer-Verlag.

Scheaffer, R. L., W. Mendenhall, and L. Ott. 1996. *Elementary Survey Sampling.* 5th ed. Boston: Duxbury Press.

Skinner, C. J., D. Holt, and T. M. F. Smith, eds. 1989. *Analysis of Complex Surveys.* New York: John Wiley & Sons.

Stuart, A. 1984. *The Ideas of Sampling.* 3d ed. New York: Macmillan.

Thompson, S. K. 1992. *Sampling.* New York: John Wiley & Sons.

Williams, B. 1978. *A Sampler on Sampling.* New York: John Wiley & Sons.

Wolter, K. M. 1985. *Introduction to Variance Estimation.* New York: Springer-Verlag.

| svy2 | Estimation of means, totals, ratios, and proportions for survey data |
|---|---|

John L. Eltinge, Texas A&M University, FAX 409-845-3144, EMAIL jeltinge@stat.tamu.edu
William M. Sribney, Stata Corporation, FAX 409-696-4601, EMAIL tech_support@stata.com

The syntax for the `svymean`, `svytotal`, and `svyratio` commands is

`svymean` *varlist* [*weight*] [`if` *exp*] [`in` *range*] [`,` *options*]

`svytotal` *varlist* [*weight*] [`if` *exp*] [`in` *range*] [`,` *options*]

`svyratio` *varname* [`/`] *varname* [*varname* [`/`] *varname* ...]
[*weight*] [`if` *exp*] [`in` *range*] [`,` *options*]

where the common *options* are

`strata(`*varname*`)` `psu(`*varname*`)` `fpc(`*varname*`)`
`by(`*varlist*`)` `subpop(`*expression*`)` `srssubpop` `nolabel`
{ `complete` | `available` } `float`
`level(#)` `ci` `deff` `deft` `meff` `meft` `obs` `size`

svymean, svyratio, and svytotal typed without arguments redisplay previous results. Any of the following options can be used when redisplaying results:

level(#) ci deff deft meff meft obs size

The syntax for the svyprop command is

svyprop *varlist* [*weight*] [if *exp*] [in *range*] [, strata(*varname*) psu(*varname*)
fpc(*varname*) by(*varlist*) subpop(*expression*) nolabel format(*%fmt*) ]

All of these commands allow pweights and iweights.

Warning: Use of if or in restrictions will not produce correct variance estimates for subpopulations in many cases. To compute estimates for subpopulations, use the by() or subpop() options.

[*Editor's note: The ado files for these commands can be found in the directory svy1.*]

## Description

The svy commands are designed for use with complex survey data. The survey design may or may not be stratified. Depending on the original sample design, primary sampling units (PSUs) may consist of clusters of observations (e.g., counties or city blocks) or the PSUs may be individual observations. Typically, there are sampling weights proportional to the inverse of the probability of being selected. In Stata syntax, sampling weights are referred to as pweights—short for "probability weights".

These commands produce estimates of finite-population means, totals, ratios, and proportions. Associated variance estimates, design effects (deff and deft), and misspecification effects (meff and meft) are also computed. Estimates for multiple subpopulations can be obtained using the by() option. The subpop() option will give estimates for a single subpopulation defined by an expression.

For linear regression, logistic regression, and probit estimation with survey data, see *svy4* in this issue.

## Options

strata(*varname*) specifies the name of a variable (numeric or string) that contains stratum identifiers. strata() can also be specified with the varset command; see the following examples.

psu(*varname*) specifies the name of a variable (numeric or string) that contains identifiers for the primary sampling unit (i.e., the cluster). psu() can also be specified with the varset command.

fpc(*varname*) requests a finite population correction for the variance estimates. If the variable specified has values less than or equal to 1, it is interpreted as a stratum sampling rate $f_h = n_h/N_h$, where $n_h$ = number of PSUs sampled from stratum $h$ and $N_h$ = total number of PSUs in the population belonging to stratum $h$. If the variable specified has values greater than or equal to $n_h$, it is interpreted as containing $N_h$. fpc() can also be specified with the varset command.

by(*varlist*) specifies that estimates be computed for the subpopulations defined by different values of the variable(s) in the by *varlist*.

subpop(*expression*) specifies that estimates be computed for the single subpopulation defined by the observations for which the specified expression is true. Note that observations with missing values for the variable(s) in this expression may have to be omitted explicitly using an if statement; see the following examples.

srssubpop can only be specified if by() or subpop() is specified. srssubpop requests that deff and deft be computed using an estimate of simple-random-sampling variance for sampling within a subpopulation. If srssubpop is not specified, deff and deft are computed using an estimate of simple-random-sampling variance for sampling from the entire population.

nolabel can only be specified if by() is specified. nolabel requests that numeric values rather than value labels be used to label output for subpopulations. By default, value labels are used.

{ complete | available } specifies how missing values are to be handled. complete specifies that only observations with complete data should be used; i.e., any observation that has a missing value for any of the variables in the *varlist* is omitted from the computation. available specifies that all available nonmissing values be used for each estimate.

If neither complete nor available is specified, available is the default when there are missing values and there are two or more variables in the *varlist* (or four or more for svyratio). If there are missing values and two or more variables (or four or more for svyratio), complete must be specified to compute the covariance or to use svytest (for hypothesis

tests) or svylc (estimates for linear combinations) after running the command (see *svy5* for a description of the svylc and svytest commands).

The complete computation requires memory for a number of variables equal to the number of variables in the *varlist* times the number of subpopulations. The available computation has no such requirements and can be specified to reduce memory requirements when covariance estimates are not needed.

float specifies that computations with the complete option be done in float precision rather than in double precision (the default) to reduce memory requirements.

format(*%fmt*) (svyprop only) specifies the display format for the proportion estimates and their standard errors. The default is %9.6f.

The following options can be specified initially or when redisplaying results:

level(#) specifies the confidence level (i.e., nominal coverage rate), in percent, for confidence intervals. The default is level(95) or as set by set level.

ci requests that confidence intervals be displayed. If no display options are specified, then, by default, confidence intervals are displayed.

deff requests that the design-effect measure deff be displayed. If no display options are specified, then, by default, deff is displayed.

deft requests that the design-effect measure deft be displayed. See the following section, *Some fine points about deff and deft*, for a discussion of deff and deft.

meff requests that the meff measure of misspecification effects be displayed.

meft requests that the meft measure of misspecification effects be displayed. See the following section, *Misspecification effects: meff and meft*, for a discussion of meff and meft.

obs requests that the number of observations used for the computation of the estimate be displayed for each row of estimates.

size requests that the estimate of the (sub)population size be displayed for each row of estimates. The (sub)population size estimate equals the sum of the weights for those observations used for the mean/total/ratio estimate.

## Setting the strata, psu, fpc, and pweight variables

For our first example, we use data from the National Maternal and Infant Health Survey (NMIHS) (Gonzalez et al. 1992, Johnson 1995). This dataset has a stratification variable stratan. Primary sampling units are mothers; i.e., PSUs are individual observations—there is no separate PSU variable. The sampling weights are given by the variable finwgt. We do not include a finite population correction for this analysis.

We use the varset command to set the stratification variable and the pweight variable.

```
. varset strata stratan
. varset pweight finwgt
```

Typing varset alone shows what has been set.

```
. varset
strata is stratan
pweight is finwgt
```

If we save the dataset, these settings will be remembered next time we use this dataset.

```
. save nmihs, replace
file nmihs.dta saved
. use nmihs, clear
. varset
strata is stratan
pweight is finwgt
```

We can now use the svy commands without having to specify explicitly the pweights and the strata() option.

```
. svymean birthwgt
Survey mean estimation
pweight:  finwgt                          Number of obs    =      9946
Strata:   stratan                         Number of strata =         6
PSU:      <observations>                  Number of PSUs   =      9946
                                          Population size  = 3895561.7
```

```
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
birthwgt |   3355.452    6.402741    3342.902    3368.003    1.142614
------------------------------------------------------------------------------
```

Alternatively, we could have set the `pweights` and `strata` when we issued the command.

```
. svymean birthwgt [pweight=finwgt], strata(stratan)
```

No matter which of these methods are used initially to set `pweights` and `strata`, the settings are remembered and do not have to be specified in subsequent use of any of the `svy` commands. We will illustrate the other features of the `varset` command in later examples; see the on-line help for `varset` for a syntax diagram.

### Estimates for subpopulations

Estimates for subpopulations can be obtained using the `by()` option.

```
. svymean birthwgt, by(race)
Survey mean estimation
pweight:  finwgt                                  Number of obs    =      9946
Strata:   stratan                                 Number of strata =         6
PSU:      <observations>                          Number of PSUs   =      9946
                                                  Population size  = 3895561.7
------------------------------------------------------------------------------
Mean   Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------------+--------------------------------------------------------------
birthwgt       |
      nonblack |    3402.32    7.609532    3387.404    3417.236    1.443763
         black |   3127.834    6.529814    3115.035    3140.634    .1720408
------------------------------------------------------------------------------
```

Note: One may wish to specify the `srssubpop` option for the computation of deff in this case; see the section *Some fine points about deff and deft* which appears later in this article.

Any number of variables can be used in the `by()` option.

```
. svymean birthwgt, by(race marital)
Survey mean estimation
pweight:  finwgt                                  Number of obs    =      9946
Strata:   stratan                                 Number of strata =         6
PSU:      <observations>                          Number of PSUs   =      9946
                                                  Population size  = 3895561.7
------------------------------------------------------------------------------
Mean      Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
------------------+-----------------------------------------------------------
birthwgt          |
nonblack   single |   3291.045    20.18795    3251.472    3330.617    1.684919
nonblack  married |   3426.407    8.379497    3409.982    3442.833    1.477145
   black   single |   3073.122    8.752553    3055.965    3090.279    .1954712
   black  married |   3221.616    12.42687    3197.257    3245.975    .2368791
------------------------------------------------------------------------------
```

In the example above, the variables `race` and `marital` have value labels. `race` has the value 0 labeled "nonblack" (i.e., white and other) and 1 labeled "black"; `marital` has the value 0 labeled "single" and 1 labeled "married". Value labels on the `by` variables make for better looking and more readable output when producing estimates for subpopulations. See [5d] label in the reference manual for information on creating value labels.

The `subpop()` option can be used to specify a single subpopulation.

```
. svymean birthwgt, subpop(race==1)
Survey mean estimation
pweight:  finwgt                                  Number of obs    =      9946
Strata:   stratan                                 Number of strata =         6
PSU:      <observations>                          Number of PSUs   =      9946
Subpop.:  race==1                                 Population size  = 3895561.7
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
birthwgt |   3127.834    6.529814    3115.035    3140.634    .1720408
------------------------------------------------------------------------------
```

The `subpop()` option takes an expression—just like the `if` syntax in Stata.

```
. svymean birthwgt, subpop(age < 20)
```

You can combine `subpop()` with `by()`.

```
. svymean birthwgt, subpop(race==0) by(marital age20)
Survey mean estimation
pweight:  finwgt                                  Number of obs    =      9946
Strata:   stratan                                 Number of strata =         6
PSU:      <observations>                          Number of PSUs   =      9946
Subpop.:  race==0                                 Population size  = 3895561.7
------------------------------------------------------------------------------
Mean      Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
------------------+-----------------------------------------------------------
birthwgt          |
  single   age20+ |   3312.012     24.2869    3264.405    3359.619    1.635331
  single   age<20 |   3244.709    36.85934    3172.457    3316.961    1.876781
 married   age20+ |   3434.923    8.674633    3417.919    3451.927    1.487806
 married   age<20 |   3287.301    34.15988    3220.341    3354.262    1.585084
------------------------------------------------------------------------------
```

When `by()` variables have missing values, the observations with missing values are automatically omitted from the analysis. In contrast to this, the `subpop()` option requires special handling when the variables in the `subpop()` expression have missing values. When you use the `subpop()` option, there is no way for Stata to know what observations, if any, should be omitted. You must explicitly tell Stata which observations to omit because of missing data. For example, our NMIHS dataset has a variable `bwgrp` (birthweight groups: 1 = very low, 2 = low, 3 = normal) which is missing for 7 observations. We omit them using an `if` statement when we compute estimates for the `bwgrp==1` subpopulation.

```
. svymean age if bwgrp~=., subpop(bwgrp==1)
Survey mean estimation
pweight:  finwgt                                  Number of obs    =      9946
Strata:   stratan                                 Number of strata =         6
PSU:      <observations>                          Number of PSUs   =      9946
Subpop.:  bwgrp==1                                Population size  = 3895561.7
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
     age |   25.79211     .158161    25.48208    26.10214    .0867546
------------------------------------------------------------------------------
```

## Warning about the use of if and in

One might be tempted to use the following standard Stata syntax to get subpopulation estimates:

```
. svymean birthwgt if highbp==1
Survey mean estimation
pweight:  finwgt                                  Number of obs    =       595
Strata:   stratan                                 Number of strata =         6
PSU:      <observations>                          Number of PSUs   =       595
                                                  Population size  = 186196.71
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
birthwgt |   3202.483     28.7201    3146.077     3258.89    1.060747
------------------------------------------------------------------------------
```

The above, however, gives incorrect variance estimates for this study design. One should use `by()` or `subpop` to get subpopulation estimates in this case.

```
. svymean birthwgt, subpop(highbp==1)
Survey mean estimation
pweight:  finwgt                                  Number of obs    =      9946
Strata:   stratan                                 Number of strata =         6
PSU:      <observations>                          Number of PSUs   =      9946
Subpop.:  highbp==1                               Population size  = 3895561.7
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
birthwgt |   3202.483    33.29483    3137.219    3267.748    1.140812
------------------------------------------------------------------------------
```

The different variance estimates are due to the way `if` works in Stata. For all commands in Stata, using `if` is equivalent to deleting all observations that do not satisfy the `if` expression and then running the commands. In contrast with this, the `svy` commands compute subpopulation variance estimates in a way that accounts for which sample units were and were not contained in the subpopulation of interest. Thus, the `svy` commands must also have access to those observations not in the subpopulation to compute the variance estimates. The survey literature refers to these variance estimators as "unconditional" variance estimators. See, e.g., Cochran (1977, Section 2.13) for a further discussion of these unconditional variance estimators and some alternative "conditional" variance estimators.

For survey data, there are only a few circumstances that necessitate the use of `if`. For example, if one suspected laboratory error for a certain set of measurements, then it might be proper to use `if` to omit these observations from the analysis. Another use of `if` was described in the previous section: `if` was used in conjunction with the `subpop()` option to omit observations with missing values.

### Options for displaying results: ci deff deft meff meft obs size

We now use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981) as our example. First, we set the `strata`, `psu`, and `pweight` variables.

```
. varset strata strata
. varset psu psu
. varset pweight finalwgt
```

We will estimate the population means for total serum cholesterol (`tcresult`) and serum triglycerides (`tgresult`).

```
. svymean tcresult tgresult
Survey mean estimation
pweight:  finalwgt                              Number of obs(*) =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
tcresult |   213.0977    1.127252    210.7986    215.3967    5.602499
tgresult |    138.576    2.071934    134.3503    142.8018    2.356968
------------------------------------------------------------------------------
(*) Some variables contain missing values.
```

If we want to see how many nonmissing observations there are for each variable, we can redisplay the results specifying the `obs` option.

```
. svymean, obs
Survey mean estimation
pweight:  finalwgt                              Number of obs(*) =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.        Obs
---------+--------------------------------------------------------------------
tcresult |   213.0977    1.127252       10351
tgresult |    138.576    2.071934        5050
------------------------------------------------------------------------------
(*) Some variables contain missing values.
```

The `svymean`, `svytotal`, and `svyratio` commands allow you to display any or all of the following: `ci` (confidence intervals), `deff`, `deft`, `meff`, `meft`, `obs`, and `size` (estimated (sub)population size).

```
. svymean, ci deff deft meff meft obs size
Survey mean estimation
pweight:  finalwgt                              Number of obs(*) =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
tcresult |   213.0977    1.127252    210.7986    215.3967    5.602499
tgresult |    138.576    2.071934    134.3503    142.8018    2.356968
------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------
    Mean |       Deft        Meff        Meft         Obs   Pop. Size
---------+--------------------------------------------------------------------
tcresult |    2.36696     5.39262    2.322202       10351   1.172e+08
tgresult |   1.535242    2.328208    1.525847        5050    56820832
------------------------------------------------------------------------------
(*) Some variables contain missing values.
```

If none of these options are specified, `ci` and `deff` are the default. Note that there is no control over the order in which the options are displayed; they are always displayed in the order shown here.

We can also give display options when we first issue a command.

```
. svymean tcresult tgresult, deff meff obs
```

## Using svytotal and svyratio

All of our examples to this point have used `svymean`. The `svytotal` command has exactly the same syntax. In our NHANES II dataset, `heartatk` is a variable that is 1 if a person has ever had a heart attack and 0 otherwise. We estimate the total numbers of persons who have had heart attacks by `sex` in the population represented by our data.

```
. svytotal heartatk, by(sex)
Survey total estimation
pweight:  finalwgt                              Number of obs    =     10349
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.171e+08
------------------------------------------------------------------------------
Total  Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------------+--------------------------------------------------------------
heartatk       |
          Male |    2304839    200231.3     1896465     2713213    1.567611
        Female |    1178437    109020.5    956088.2     1400786    .9000898
------------------------------------------------------------------------------
```

The syntax for the `svyratio` command only differs from `svymean` and `svytotal` in the way the varlist can be specified. All the options are the same. In our NHANES II dataset, the variable `tcresult` contains total serum cholesterol and the variable `hdresult` contains serum levels of high-density lipoproteins (HDL). We can use `svyratio` to estimate the ratio of the total of `hdresult` to the total of `tcresult`.

```
. svyratio hdresult/tcresult
Survey ratio estimation
pweight:  finalwgt                              Number of obs    =      8720
Strata:   newstr                                Number of strata =        30
PSU:      newpsu                                Number of PSUs   =        60
                                                Population size  =  98725345
------------------------------------------------------------------------------
      Ratio        |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
-------------------+--------------------------------------------------------------
hdresult/tcresult |   .2336173    .0024621     .228589    .2386457    7.724248
------------------------------------------------------------------------------
```

Out of the 10351 NHANES II subjects with a `tcresult` reading, only 8720 had an `hdresult` reading. Consequently, `svyratio` used only the 8720 observations that had nonmissing values for both variables. In your own datasets, if you encounter substantial missing-data rates, it is generally a good idea to look into the reasons for the missing-data phenomenon, and to consider the potential for problems with nonresponse bias in your analysis.

Note that the slash `/` in the *varlist* for `svyratio` is optional. We could have typed

```
. svyratio hdresult tcresult
```

`svyratio` or `svymean` can be used to estimate means for subpopulations. Consider the following example. In our NHANES II dataset, we have a variable `female` (equal to 1 if female and 0 otherwise) and a variable `iron` containing iron levels. Suppose that we wish to estimate the ratio of total iron levels in females to total number of females in the population. We can do this by first creating a new variable `firon` that represents iron levels in females; i.e., the variable equals iron level if the subject is female and zero if male. We can then use `svyratio` to estimate the ratio of the total of `firon` to the total of `female`.

```
. gen firon = female*iron
. svyratio firon/female
```

```
Survey ratio estimation
pweight:  finalwgt                                Number of obs    =     10351
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.172e+08
------------------------------------------------------------------------------
      Ratio       |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
------------------+-----------------------------------------------------------
   firon/female   |   97.16247    .6743344    95.78715    98.53778    2.014025
------------------------------------------------------------------------------
```

This estimate can be obtained more easily using `svymean` with the `subpop` option. The computation that is carried out is identical.

```
. svymean iron, subpop(female==1)
Survey mean estimation
pweight:  finalwgt                                Number of obs    =     10351
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
Subpop.:  female==1                               Population size  = 1.172e+08
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
    iron |   97.16247    .6743344    95.78715    98.53778    2.014025
------------------------------------------------------------------------------
```

## Estimating proportions

Estimating proportions can be done using the `svymean` or `svyprop` commands. The mean of `heartatk` (`heartatk` equals 1 if a person has ever had a heart attack and 0 otherwise) is an estimate of the proportion of persons who have had heart attacks.

```
. svymean heartatk
Survey mean estimation
pweight:  finalwgt                                Number of obs    =     10349
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.171e+08
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
heartatk |   .0297383    .0018484    .0259684    .0335081    1.225296
------------------------------------------------------------------------------
```

We could have also obtained this estimate using `svyprop`.

```
. svyprop heartatk
------------------------------------------------------------------------------
pweight:  finalwgt                              Number of obs      =     10349
Strata:   strata                                Number of strata   =        31
PSU:      psu                                   Number of PSUs     =        62
                                                Population size    = 1.171e+08
------------------------------------------------------------------------------
Survey proportions estimation
 heartatk      _Obs   _EstProp    _StdErr
        0      9873   0.970262   0.001848
        1       476   0.029738   0.001848
```

The `svymean` command produces more output than `svyprop`—it calculates covariance, deff, meff, etc. In order to do these computations, `svymean` requires that you specify a separate dummy (0/1) variable for each proportion you wish to estimate. With `svyprop`, we do not have to create dummy variables, nor does the command create them internally. `svyprop` will estimate proportions for all of the categories defined by the unique values of the variables in its *varlist*. It can handle any number of categories, but provides less output.

```
. svyprop race agegrp
------------------------------------------------------------------------------
pweight:  finalwgt                              Number of obs      =     10351
Strata:   strata                                Number of strata   =        31
PSU:      psu                                   Number of PSUs     =        62
                                                Population size    = 1.172e+08
------------------------------------------------------------------------------
```

```
Survey proportions estimation
     race    agegrp      _Obs   _EstProp    _StdErr
    White  age20-29      1975   0.241082   0.006800
    White  age30-39      1411   0.178603   0.006491
    White  age40-49      1120   0.148661   0.004936
    White  age50-59      1129   0.147948   0.006418
    White  age60-69      2552   0.120860   0.004560
    White   age 70+       878   0.042001   0.003132
    Black  age20-29       286   0.031459   0.004918
    Black  age30-39       179   0.020485   0.002754
    Black  age40-49       124   0.015115   0.001964
    Black  age50-59       140   0.015160   0.002668
    Black  age60-69       260   0.009703   0.001380
    Black   age 70+        97   0.003583   0.000753
    Other  age20-29        59   0.007917   0.002237
    Other  age30-39        32   0.005213   0.002072
    Other  age40-49        28   0.004587   0.001669
    Other  age50-59        22   0.004052   0.002168
    Other  age60-69        48   0.002926   0.002152
    Other   age 70+        11   0.000645   0.000540
```

`svyprop` also allows the `by()` and `subpop()` options for subpopulation estimates.

```
. svyprop agegrp, by(race)
------------------------------------------------------------------------------
pweight:  finalwgt                            Number of obs      =     10351
Strata:   strata                              Number of strata   =        31
PSU:      psu                                 Number of PSUs     =        62
                                              Population size    = 1.172e+08
------------------------------------------------------------------------------
Survey proportions estimation
-> race=White
  agegrp      _Obs   _EstProp    _StdErr
age20-29      1975   0.274220   0.007286
age30-39      1411   0.203153   0.006602
age40-49      1120   0.169096   0.004680
age50-59      1129   0.168285   0.005908
age60-69      2552   0.137473   0.004102
 age 70+       878   0.047774   0.003265
-> race=Black
  agegrp      _Obs   _EstProp    _StdErr
age20-29       286   0.329388   0.019606
age30-39       179   0.214495   0.012963
age40-49       124   0.158266   0.013822
age50-59       140   0.158738   0.012046
age60-69       260   0.101601   0.007740
 age 70+        97   0.037513   0.005293
-> race=Other
  agegrp      _Obs   _EstProp    _StdErr
age20-29        59   0.312426   0.047763
age30-39        32   0.205729   0.027031
age40-49        28   0.181028   0.019953
age50-59        22   0.159901   0.024910
age60-69        48   0.115473   0.038194
 age 70+        11   0.025442   0.011067
```

## Changing the strata, psu, fpc, and pweight variables

The NHANES II dataset contains a special sampling weight for use with the `lead` variable. We can change the `pweight` by setting it again using `varset`.

```
. varset pweight leadwt
. svymean lead, by(sex race)
Survey mean estimation
pweight:  leadwt                              Number of obs    =      4948
Strata:   strata                              Number of strata =        31
PSU:      psu                                 Number of PSUs   =        62
                                              Population size  = 1.129e+08
```

```
------------------------------------------------------------------------------
Mean      Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
------------------+-----------------------------------------------------------
lead              |
    Male    White |   16.78945    .3010539    16.17544    17.40345    4.641307
    Male    Black |   19.70286    .7448225    18.18379    21.22194    1.950369
    Male    Other |   16.16566    .9394023    14.24973    18.08158    1.293194
  Female    White |   11.80468    .2447241    11.30556    12.30379    5.920213
  Female    Black |   12.92722    .5255033    11.85545    13.99899    3.946779
  Female    Other |   11.74192     .655919    10.40417    13.07968    1.492849
------------------------------------------------------------------------------
```

To change the pweight back to `finalwgt`, we type

```
. varset pweight finalwgt
```

Remember that typing `varset` alone displays the settings.

```
. varset
strata is strata
psu is psu
pweight is finalwgt
```

## Finite population correction (FPC)

A finite population correction (FPC) accounts for the reduction in variance that occurs when we sample *without* replacement from a finite population, as compared to sampling *with* replacement. The `fpc()` option of the `svy` commands computes an FPC for cases of simple random sampling or stratified random sampling; i.e., for sample designs that use simple random sampling without replacement of PSUs within each stratum with no subsampling within PSUs. The `fpc()` option is not intended for use with designs that involve subsampling within PSUs.

Consider the following dataset.

```
. list
      strata       psu    weight        nh        Nh         x
  1.       1         1         3         5        15       2.8
  2.       1         2         3         5        15       4.1
  3.       1         3         3         5        15       6.8
  4.       1         4         3         5        15       6.8
  5.       1         5         3         5        15       9.2
  6.       2         1         4         3        12       3.7
  7.       2         2         4         3        12       6.6
  8.       2         3         4         3        12       4.2
```

We first set the `strata`, `psu`, and `pweight`s.

```
. varset strata strata
. varset psu psu
. varset pweight weight
```

In this dataset, the variable `nh` is the number of PSUs per stratum that were sampled, `Nh` is the total number of PSUs per stratum in the sampling frame (i.e., the population), and `x` is our survey item of interest. If we wish to use a finite population correction in our computations, we set `fpc` to `Nh`, the variable representing the total number of PSUs per stratum in the population.

```
. varset fpc Nh
. varset
strata is strata
psu is psu
pweight is weight
fpc is Nh
. svymean x
Survey mean estimation
pweight:  weight                                Number of obs    =         8
Strata:   strata                                Number of strata =         2
PSU:      psu                                   Number of PSUs   =         8
FPC:      Nh                                    Population size  =        27
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
       x |   5.448148    .6160407    3.940751    6.955545    .9853061
------------------------------------------------------------------------------
Finite population correction (FPC) assumes simple random sampling without
replacement of PSUs within each stratum with no subsampling within PSUs.
```

If we want to redo the computation without an FPC, we must clear the `fpc` using `varset` and run the estimation command again.

```
. varset fpc, clear
. svymean x
Survey mean estimation
pweight:  weight                                  Number of obs    =         8
Strata:   strata                                  Number of strata =         2
PSU:      psu                                     Number of PSUs   =         8
                                                  Population size  =        27
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
       x |   5.448148    .7412683     3.63433    7.261966    1.003906
------------------------------------------------------------------------------
```

Including an FPC always reduces the variance estimate. However, when the $N_h$ are large relative to the $n_h$, the reduction in estimated variance due to the FPC is small.

Rather than having a variable that represents the total number of PSUs per stratum in the sampling frame, we sometimes have a variable that represents a sampling rate $f_h = n_h/N_h$. If we have a variable that represents a sampling rate, we set it the same way to get an FPC. The commands are smart; if the `fpc` variable is less than or equal to 1, it is interpreted as a sampling rate; if it is greater than or equal to $n_h$, it is interpreted as containing $N_h$.

## Some fine points about deff and deft

The ratio deff (Kish 1965) is intended to compare the variance obtained under our complex survey design to the variance that we would have obtained if we had collected our observations through simple random sampling. deff is defined as

$$\text{deff} = \widehat{V}/\widehat{V}_{\text{srswor}}$$

where $\widehat{V}$ is the design-based estimate of variance (i.e., this is what the `svy` commands compute—the displayed standard error is the square root of $\widehat{V}$) and $\widehat{V}_{\text{srswor}}$ is an estimate of what the variance would be if a similar survey were conducted using simple random sampling (srs) without replacement (wor) with the same number of sample elements as in the actual survey. In other words, $\widehat{V}_{\text{srswor}}$ is an estimate of the variance for a hypothetical simple-random-sampling design in place of the complex design that we actually used.

deft is defined as (Kish 1995)

$$\text{deft} = \sqrt{\widehat{V}/\widehat{V}_{\text{srswr}}}$$

where $\widehat{V}_{\text{srswr}}$ is computed in the same way as $\widehat{V}_{\text{srswor}}$ except now the hypothetical simple-random-sampling design is with replacement (wr).

Computationally, $\widehat{V}_{\text{srswor}} = (1 - m/\widehat{M})\widehat{V}_{\text{srswr}}$, where $m$ is the number of sampled elements (i.e., the number of observations in the dataset) and $\widehat{M}$ is the estimated total number of elements in the population. For many surveys, the term $(1 - m/\widehat{M})$ is very close to 1, so that in these cases $\widehat{V}_{\text{srswor}}$ and $\widehat{V}_{\text{srswr}}$ are almost equal. Furthermore, if the `fpc()` option is not specified or set with `varset`, we do not compute any finite population corrections, so the estimates produced for $\widehat{V}_{\text{srswor}}$ and $\widehat{V}_{\text{srswr}}$ are exactly equal. To summarize: deft is exactly the square root of deff when `fpc` is not set, and it is somewhat smaller than the square root of deff when the user sets an `fpc`.

*The srssubpop option for deff and deft with subpopulations*

When there are subpopulations, the `svy` commands can compute design effects with respect to one of two different hypothetical simple random sampling designs. The first hypothetical design is one in which simple random sampling is conducted across the full population. This scheme is the default for deff and deft computed by the `svy` commands. The second hypothetical design is one in which the simple random sampling is conducted entirely within the subpopulation of interest. This second scheme is used for deff and deft when the `srssubpop` option is specified.

Deciding which scheme is preferable depends on the nature of the subpopulations. If one reasonably can imagine identifying members of the subpopulations prior to sampling them, then the second scheme is preferable. This case arises primarily when the subpopulations are strata or groups of strata. Otherwise, one may prefer to use the first scheme.

Here is an example of using first scheme (i.e., the default) with the NHANES II data.

```
. svymean iron, by(sex)
```

```
Survey mean estimation

pweight:  finalwgt                                Number of obs    =     10351
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.172e+08
------------------------------------------------------------------------------
Mean   Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------------+--------------------------------------------------------------
iron           |
          Male |   104.7969     .557267    103.6603    105.9334    1.360971
        Female |   97.16247    .6743344    95.78715    98.53778    2.014025
------------------------------------------------------------------------------
```

Here is the same example rerun using the second scheme; i.e., specifying the `srssubpop` option.

```
. svymean iron, by(sex) srssubpop

Survey mean estimation

pweight:  finalwgt                                Number of obs    =     10351
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.172e+08

------------------------------------------------------------------------------
Mean   Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------------+--------------------------------------------------------------
iron           |
          Male |   104.7969     .557267    103.6603    105.9334    1.348002
        Female |   97.16247    .6743344    95.78715    98.53778    2.031321
------------------------------------------------------------------------------
```

Because the NHANES II did not stratify on sex, we consider it problematic to consider design effects with respect to simple random sampling of the female (or male) subpopulation. Consequently, we would prefer to use the first scheme here, although the values of deff differ little between the two schemes in this case.

For other examples (generally involving heavy oversampling or undersampling of specified subpopulations), the differences in deff for the two schemes can be much more dramatic. Consider the NMIHS data and compute the mean of `birthwgt` by `race`.

```
. svymean birthwgt, by(race) deff obs

Survey mean estimation

pweight:  finwgt                                  Number of obs    =      9946
Strata:   stratan                                 Number of strata =         6
PSU:      <observations>                          Number of PSUs   =      9946
                                                  Population size  = 3895561.7

------------------------------------------------------------------------------
Mean   Subpop. |   Estimate    Std. Err.       Deff         Obs
---------------+--------------------------------------------------------------
birthwgt       |
      nonblack |    3402.32    7.609532    1.443763        4724
         black |   3127.834    6.529814    .1720408        5222
------------------------------------------------------------------------------

. svymean birthwgt, by(race) deff obs srssubpop

Survey mean estimation

pweight:  finwgt                                  Number of obs    =      9946
Strata:   stratan                                 Number of strata =         6
PSU:      <observations>                          Number of PSUs   =      9946
                                                  Population size  = 3895561.7

------------------------------------------------------------------------------
Mean   Subpop. |   Estimate    Std. Err.       Deff         Obs
---------------+--------------------------------------------------------------
birthwgt       |
      nonblack |    3402.32    7.609532    .8268418        4724
         black |   3127.834    6.529814    .5289629        5222
------------------------------------------------------------------------------
```

Since the NMIHS survey was stratified on race, marital status, age, and birthweight, we consider it plausible to consider design effects computed with respect to simple random sampling within an individual race group. Consequently, in this case, we would recommend the second scheme; i.e., we would use the `srssubpop` option.

### Misspecification effects: meff and meft

Misspecification effects are used to assess biases in variance estimators that are computed under the wrong assumptions. The survey literature (e.g., Scott and Holt 1982, p. 850; Skinner 1989) define misspecification effects with respect to a general set of "wrong" variance estimators. The current `svy` commands consider only one specific form: variance estimators computed under the incorrect assumption that our *observed* sample was selected through simple random sampling.

The resulting "misspecification effect" measure is informative primarily in cases for which an unweighted point estimator is approximately unbiased for the parameter of interest. See Eltinge and Sribney (1996) for a detailed discussion of extensions of misspecification effects that are appropriate for *biased* point estimators. An expanded set of misspecification effect measures is planned for future versions of the `svy` commands.

Note that the definition of misspecification effect is in contrast with the earlier definition of design effect. For a design effect, we compare our complex-design-based variance estimate with an estimate of the true variance that we would have obtained under a hypothetical true simple random sample.

The `svy` commands use the following definitions for $\mathrm{meff}_\mathrm{c}$ and $\mathrm{meft}_\mathrm{c}$:

$$\mathrm{meff}_\mathrm{c} = \widehat{V}/\widehat{V}_\mathrm{msp}$$

$$\mathrm{meft}_\mathrm{c} = \sqrt{\mathrm{meff}_\mathrm{c}}$$

where $\widehat{V}$ is the appropriate design-based estimate of variance and $\widehat{V}_\mathrm{msp}$ is the variance estimate computed with a misspecified design—namely, ignoring the sampling weights, stratification, and clustering. In other words, $\widehat{V}_\mathrm{msp}$ is what the `ci` command used without weights would naively compute.

Here we request that the misspecification effects be displayed for the estimation of mean zinc levels using our NHANES II data.

```
. svymean zinc, by(sex) meff meft
Survey mean estimation
pweight:  finalwgt                              Number of obs    =      9202
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.043e+08
------------------------------------------------------------------------------
Mean   Subpop. |   Estimate    Std. Err.       Meff        Meft
---------------+--------------------------------------------------------------
zinc           |
          Male |   90.74543    .5850741    6.282539    2.506499
        Female |    83.8635    .4689532    6.326477    2.515249
------------------------------------------------------------------------------
```

If we run `ci` without weights, we get the standard errors that are $(\widehat{V}_\mathrm{msp})^{1/2}$.

```
. ci zinc, by(sex)
->  sex=Male
Variable |     Obs         Mean    Std. Err.       [95% Conf. Interval]
---------+--------------------------------------------------------------
    zinc |    4375     89.53143    .2334228         89.0738    89.98906
->  sex=Female
Variable |     Obs         Mean    Std. Err.       [95% Conf. Interval]
---------+--------------------------------------------------------------
    zinc |    4827     83.76652     .186444        83.40101    84.13204
. di .5850741/.2334228
2.5064994
. di (.5850741/.2334228)^2
6.2825391
. di .4689532/.186444
2.5152496
. di (.4689532/.186444)^2
6.3264806
```

### Memory requirements

These commands are implemented as ado files. For reasons of computational efficiency, the `svymean`, `svytotal`, and `svyratio` commands will create new variables of type `double` when the `complete` option is specified or is the default (as is

the case when there are no missing values). The `complete` computation requires room for $K = k_{\rm var} n_{\rm subpop}$ variables of type `double`, where $k_{\rm var}$ equals the number of variables in the *varlist* and $n_{\rm subpop}$ equals the number of subpopulations.

If you get the following error message:

```
. svymean age height weight iron zinc, by(sex agegrp)
no room to add more variables
try using "available" option
r(902);
```

either run the command again using the `available` option or specify fewer variables in the *varlist*. Both alternatives are suitable when you do not wish to compute the covariance matrix. If you do wish to compute the covariance, either use the `float` option (which reduces memory requirements in half to $K$ variables of type `float`) or resize Stata's memory area to a sufficient `maxvar` and `width` (see [4] memory in the reference manual).

These special requirements will be unnecessary in a future release of Stata when fast, memory-efficient, internal versions of these commands will be produced.

## Methods and Formulas

The current `svy` commands use the relatively simple variance estimators outlined below. See, for example, Cochran (1977) and Wolter (1985) for some methodological background on these variance estimators. In some cases, some authors prefer to use other variance estimators that, for example, account separately for variance components at different stages of sampling, use finite population corrections with some unequal-probability and multistage designs, and include other special design features.

In addition, the current `svy` commands use "linearization" based variance estimators for nonlinear functions like sample ratios. Alternative variance estimators that use replication methods—for example, jackknifing or balanced repeated replication—may be included in future `svy` versions.

*Totals*

All of the computations done by the `svytotal`, `svymean`, `svyratio`, and `svyprop` commands are essentially based on the formulas for totals.

Let $h = 1, \ldots, L$ enumerate the strata in the survey, and let $(h, i)$ denote the $i$ th primary sampling unit (PSU) in stratum $h$ for $i = 1, \ldots, N_h$, where $N_h$ is the total number of PSUs in stratum $h$ in the population. Let $M_{hi}$ be the number of elements in PSU $(h, i)$, and let $M = \sum_{h=1}^{L} \sum_{i=1}^{N_h} M_{hi}$ be the total number of elements in the population.

Let $Y_{hij}$ be a survey item for element $j$ in PSU $(h, i)$ in stratum $h$; e.g., $Y_{hij}$ might be income for adult $j$ in block $i$ in county $h$. The associated population total is

$$Y = \sum_{h=1}^{L} \sum_{i=1}^{N_h} \sum_{j=1}^{M_{hi}} Y_{hij} \tag{1}$$

Let $y_{hij}$ be the items for those elements selected in our sample; here $h = 1, \ldots, L$; $i = 1, \ldots, n_h$; and $j = 1, \ldots, m_{hi}$. The total number of elements in the sample (i.e., the number of observations in the dataset) is $m = \sum_{h=1}^{L} \sum_{i=1}^{n_h} m_{hi}$.

Our estimator $\widehat{Y}$ for the population total $Y$ is

$$\widehat{Y} = \sum_{h=1}^{L} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} w_{hij} y_{hij} \tag{2}$$

where $w_{hij}$ are the user-specified sampling weights (`pweights` or `iweights`). Our estimator $\widehat{M}$ for the total number of elements in the population is simply the sum of the weights:

$$\widehat{M} = \sum_{h=1}^{L} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} w_{hij}$$

$\widehat{M}$ is labeled "Population size" on the output of the commands.

To compute an estimate of the variance of $\widehat{Y}$, we first define $z_{yhi}$ and $\overline{z}_{yh}$ by

$$z_{yhi} = \sum_{j=1}^{m_{hi}} w_{hij} y_{hij} \qquad \text{and} \qquad \overline{z}_{yh} = \frac{1}{n_h} \sum_{i=1}^{n_h} z_{yhi}$$

Our estimate for the variance of $\widehat{Y}$ is

$$\widehat{V}(\widehat{Y}) = \sum_{h=1}^{L}(1-f_h)\frac{n_h}{n_h-1}\sum_{i=1}^{n_h}(z_{yhi}-\overline{z}_{yh})^2 \tag{3}$$

The factor $(1-f_h)$ is the finite population correction. If the user does not set an `fpc` variable, $f_h = 0$ is used in the formula. If an `fpc` variable is set and is greater than or equal to $n_h$, the variable is assumed to contain the values of $N_h$, and $f_h$ is given by $f_h = n_h/N_h$. If the `fpc` variable is less than or equal to 1, it is assumed to contain the values of $f_h$. As discussed earlier, nonzero values of $f_h$ in formula (3) are intended for use *only* with simple random sampling or stratified random sampling with no subsampling within PSUs.

If the *varlist* given to `svytotal` contains two or more variables and the `complete` option is specified or is the default, the covariance of the variables is computed. For estimated totals $\widehat{Y}$ and $\widehat{X}$ (notation for $X$ is defined similarly to that of $Y$), our covariance estimate is

$$\widehat{\mathrm{Cov}}(\widehat{Y},\widehat{X}) = \sum_{h=1}^{L}(1-f_h)\frac{n_h}{n_h-1}\sum_{i=1}^{n_h}(z_{yhi}-\overline{z}_{yh})(z_{xhi}-\overline{z}_{xh}) \tag{4}$$

*Ratios, means, and proportions*

Let $R = Y/X$ be a population ratio that we wish to estimate, where $Y$ and $X$ are population totals defined as in (1). Our estimate for $R$ is $\widehat{R} = \widehat{Y}/\widehat{X}$. Using the delta method (i.e., a first-order Taylor expansion), the variance of the approximate distribution of $\widehat{R}$ is

$$\frac{1}{X^2}\left[V(\widehat{Y}) - 2R\,\mathrm{Cov}(\widehat{Y},\widehat{X}) + R^2\,V(\widehat{X})\right]$$

Direct substitution of $\widehat{X}$, $\widehat{R}$, and expressions (3) and (4) lead to the variance estimator

$$\widehat{V}(\widehat{R}) = \frac{1}{\widehat{X}^2}\left[\widehat{V}(\widehat{Y}) - 2\widehat{R}\,\widehat{\mathrm{Cov}}(\widehat{Y},\widehat{X}) + \widehat{R}^2\,\widehat{V}(\widehat{X})\right] \tag{5}$$

If we define the following "ratio residual"

$$d_{hij} = \frac{1}{\widehat{X}}(y_{hij} - \widehat{R}\,x_{hij}) \tag{6}$$

and replace $y_{hij}$ with $d_{hij}$ in our variance formula (3), we get the right hand side of equation (7) below. Simple algebra shows that this is identical to (5).

$$\widehat{V}(\widehat{R}) = \sum_{h=1}^{L}(1-f_h)\frac{n_h}{n_h-1}\sum_{i=1}^{n_h}(z_{dhi}-\overline{z}_{dh})^2 \tag{7}$$

To extend our variance estimators from ratios to other parameters, note that means are simply ratios with $X_{hij} = 1$ and proportions are simply means with $Y_{hij}$ equal to a $0/1$ variable. Similarly, estimates for a subpopulation $\mathcal{S}$ are obtained by computing estimates for $Y_{\mathcal{S}hij} = I_{(h,i,j)\in\mathcal{S}}Y_{hij}$ and $X_{\mathcal{S}hij} = I_{(h,i,j)\in\mathcal{S}}X_{hij}$ where $I_{(h,i,j)\in\mathcal{S}}$ equals 1 if element $(h,i,j)$ is a member of subpopulation $\mathcal{S}$ and 0 otherwise.

*Weights*

When computing finite population corrections (i.e., when an `fpc` variable is set) or when estimating totals, the `svy` commands assume your weights are the weights appropriate for estimation of a population total. For example, the sum of your weights should equal an estimate of the size of the relevant population. When an `fpc` is not set, the commands `svymean`, `svyratio`, and `svyprop` are invariant to the scale of the weights; i.e., these commands give the same results no matter what the scale of weights.

*Confidence intervals*

Let $n = \sum_{h=1}^{L} n_h$ be the total number of PSUs in the sample. The customary "degrees of freedom" attributed to our test statistic are $d = n - L$. Hence, under regularity conditions, an approximate $100(1-\alpha)\%$ confidence interval for a parameter $\theta$ (e.g., $\theta$ could be a total $Y$ or ratio $R$) is $\widehat{\theta} \pm t_{1-\alpha/2,d}\,[\widehat{V}(\widehat{\theta})]^{1/2}$.

Cochran (1977, Section 2.8) and Korn and Graubard (1990) give some theoretical justification for the use of $d = n - L$ in computation of univariate confidence intervals and $p$-values. However, for some cases, inferences based on the customary $n - L$

degrees-of-freedom calculation may be excessively liberal. For example, the resulting confidence intervals may have coverage rates substantially less than the nominal $1-\alpha$. This problem generally is of greatest practical concern when the population of interest has a very skewed or heavy-tailed distribution, or is concentrated in a small number of PSUs. In some of these cases, the user may want to consider constructing confidence intervals based on alternative degrees-of-freedom terms based on the Satterthwaite (1941, 1946) approximation and modifications thereof; see, e.g., Cochran (1977, p. 96) and Jang and Eltinge (1995).

*deff and deft*

deff is estimated as (Kish 1965)

$$\text{deff} = \frac{\widehat{V}(\widehat{\theta})}{\widehat{V}_{\text{srswor}}(\widetilde{\theta}_{\text{srs}})}$$

where $\widehat{V}(\widehat{\theta})$ is the design-based estimate of variance from formula (3) for a parameter $\theta$, and $\widehat{V}_{\text{srswor}}(\widetilde{\theta}_{\text{srs}})$ is an estimate of the variance for an estimator $\widetilde{\theta}_{\text{srs}}$ that would be obtained from a similar hypothetical survey conducted using simple random sampling (srs) without replacement (wor) with the same number of sample elements $m$ as in the actual survey. If $\theta$ is a total $Y$, we calculate

$$\widehat{V}_{\text{srswor}}(\widetilde{\theta}_{\text{srs}}) = (1-f)\frac{\widehat{M}}{m-1}\sum_{h=1}^{L}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\big(y_{hij}-\widehat{\overline{Y}}\big)^2 \tag{8}$$

where $\widehat{\overline{Y}} = \widehat{Y}/\widehat{M}$. The factor $(1-f)$ is a finite population correction. If the user sets an `fpc`, we use $f = m/\widehat{M}$; if the user does not specify an `fpc`, $f = 0$ is used. If $\theta$ is a ratio $R$, we replace $y_{hij}$ in (8) with $d_{hij}$ from (6). Note that $\sum_{h=1}^{L}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}d_{hij} = 0$, so that $\widehat{\overline{Y}}$ is replaced with zero.

deft is estimated as (Kish 1995)

$$\text{deft} = \sqrt{\frac{\widehat{V}(\widehat{\theta})}{\widehat{V}_{\text{srswr}}(\widetilde{\theta}_{\text{srs}})}}$$

where $\widehat{V}_{\text{srswr}}(\widetilde{\theta}_{\text{srs}})$ is an estimate of the variance for an estimator $\widetilde{\theta}_{\text{srs}}$ obtained from a similar survey conducted using simple random sampling (srs) with replacement (wr). $\widehat{V}_{\text{srswr}}(\widetilde{\theta}_{\text{srs}})$ is computed using (8) with $f = 0$.

When we are computing estimates for a subpopulation $\mathcal{S}$ and the `srssubpop` option is *not* specified (i.e., the default), formula (8) is used with $y_{\mathcal{S}hij} = I_{(h,i,j)\in\mathcal{S}}\, y_{hij}$ in place of $y_{hij}$. Note that the sums in (8) are still calculated over all elements in the sample regardless of whether they belong to the subpopulation. This is because we assume, by default, that the simple random sampling is done across the full population.

When the `srssubpop` option is specified, we assume that the simple random sampling is carried out within subpopulation $\mathcal{S}$. In this case, we use (8) with the sums restricted to those elements belonging to the subpopulation; $m$ is replaced with $m_{\mathcal{S}}$, the number of sample elements from the subpopulation; $\widehat{M}$ is replaced with $\widehat{M}_{\mathcal{S}}$, the sum of the weights from the subpopulation; and $\widehat{\overline{Y}} = \widehat{Y}/\widehat{M}$ is replaced with $\widehat{\overline{Y}}_{\mathcal{S}} = \widehat{Y}_{\mathcal{S}}/\widehat{M}_{\mathcal{S}}$, the weighted mean across the subpopulation.

*meff and meft*

$\text{meff}_{\text{c}}$ and $\text{meft}_{\text{c}}$ are estimated as

$$\text{meff}_{\text{c}} = \frac{\widehat{V}(\widehat{\theta})}{\widehat{V}_{\text{msp}}(\widehat{\theta}_{\text{msp}})}$$

$$\text{meft}_{\text{c}} = \sqrt{\text{meff}_{\text{c}}}$$

where $\widehat{V}(\widehat{\theta})$ is the design-based estimate of variance from formula (3) for a parameter $\theta$. In addition, $\widehat{\theta}_{\text{msp}})$ and $\widehat{V}_{\text{msp}}(\widehat{\theta}_{\text{msp}})$ are the point estimator and variance estimator based on the incorrect assumption that our observations were obtained through simple random sampling with replacement—in other words, they are the estimators obtained by simply ignoring weights, stratification, and clustering. When $\theta$ is a mean $\overline{Y}$, the estimator and its variance estimate are computed using the standard formulas for an unweighted mean:

$$\widehat{\overline{Y}}_{\text{msp}} = \frac{1}{m}\sum_{h=1}^{L}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} y_{hij}$$

$$\widehat{V}_{\text{msp}}(\widehat{\overline{Y}}_{\text{msp}}) = \frac{1}{m(m-1)}\sum_{h=1}^{L}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}}\big(y_{hij}-\widehat{\overline{Y}}_{\text{msp}}\big)^2$$

When $\theta$ is a total $Y$, $\widehat{Y}_{\rm msp} = \widehat{M}\,\widehat{\overline{Y}}_{\rm msp}$ and $\widehat{V}_{\rm msp}(\widehat{Y}_{\rm msp}) = \widehat{M}^2\,\widehat{V}_{\rm msp}(\widehat{\overline{Y}}_{\rm msp})$. When $\theta$ is a ratio $R = Y/X$, $\widehat{R}_{\rm msp} = \widehat{\overline{Y}}_{\rm msp}/\widehat{\overline{X}}_{\rm msp}$ and the estimator (5) with $\widehat{V}_{\rm msp}(\widehat{\overline{Y}}_{\rm msp})$, etc., is used to compute $\widehat{V}_{\rm msp}(\widehat{R}_{\rm msp})$.

When we compute meff and meft for a subpopulation, we simply restrict our sums to those elements belonging to the subpopulation and use $m_{\mathcal{S}}$ and $\widehat{M}_{\mathcal{S}}$ in place of $m$ and $\widehat{M}$.

## Saved Results

When the `available` option is specified or when `available` is the default (which is the case when there are missing values and multiple variables), the matrices `S_E_b` and `S_E_V` (both row vectors) are created which contain, respectively, the mean, total, or ratio estimates and design-based variance $\widehat{V}$.

The full covariance matrix is not computed when the `available` option is used. The full covariance matrix is only computed when the `complete` option is specified or when `complete` is the default (which is the case when there are no missing values). In this case, the mean, total, or ratio estimates and covariance matrix are posted to Stata's internal areas for holding estimation results. The estimates can be retrieved using

```
. matrix m = get(_b)
. matrix V = get(VCE)
```

where `m` and `V` can be replaced by names of the user's choice; see [6m] get in the reference manual for details.

`svymean`, `svytotal`, `svyratio` save the following results in the `S_E_` macros:

| | |
|---|---|
| `S_E_nobs` | number of observations $m$ |
| `S_E_nstr` | number of strata $L$ |
| `S_E_npsu` | number of sampled PSUs $n$ |
| `S_E_npop` | estimate of population size $\widehat{M}$ |
| `S_E_nby` | number of subpopulations |
| `S_E_cmd` | command name (e.g., `svymean`) |
| `S_E_depv` | `Mean`, `Total`, or `Ratio` |
| `S_E_vl` | *varlist* |
| `S_E_by` | `by()` *varlist* |
| `S_E_sub` | `subpop()` expression |
| `S_E_lab` | label indicator or labels |
| `S_E_wgt` | weight type |
| `S_E_exp` | weight variable or expression |
| `S_E_str` | `strata()` variable |
| `S_E_psu` | `psu()` variable |
| `S_E_fpc` | `fpc()` variable |

In addition, the following matrices are created by `svymean`, `svytotal`, and `svyratio`.

| | |
|---|---|
| `S_E_Vmsp` | misspecification (co)variance $\widehat{V}_{\rm msp}$ |
| `S_E_Vsrs` | simple-random-sampling-without-replacement (co)variance $\widehat{V}_{\rm srswor}$ |
| `S_E_Vswr` | simple-random-sampling-with-replacement (co)variance $\widehat{V}_{\rm srswr}$ (only created when `fpc()` option is specified) |
| `S_E_deff` | vector of deff estimates |
| `S_E_deft` | vector of deft estimates |
| `S_E_meft` | vector of meft estimates |
| `S_E_npop` | vector of subpopulation size estimates |
| `S_E_nobs` | vector of numbers of nonmissing observations |
| `S_E_err` | vector of error flags |

The following matrices are created when the `available` option is the default or is specified explicitly.

| | |
|---|---|
| `S_E_b` | vector of mean, total, or ratio estimates |
| `S_E_V` | vector of design-based variance estimates $\widehat{V}$ |
| `S_E_nstr` | vector of numbers of strata |
| `S_E_npsu` | vector of numbers of sampled PSUs |

## Acknowledgment

We thank Wayne Johnson of the National Center for Health Statistics for providing the NMIHS and NHANES II datasets.

## References

Cochran, W. G. 1977. *Sampling Techniques.* 3d ed. New York: John Wiley & Sons.

Eltinge, J. L. and W. M. Sribney. 1996. Accounting for point-estimation bias in assessment of misspecification effects, confidence-set coverage rates and test sizes. Unpublished manuscript. Department of Statistics, Texas A & M University.

Gonzalez Jr., J. F., N. Krauss, and C. Scott. 1992. Estimation in the 1988 National Maternal and Infant Health Survey. In *Proceedings of the Section on Statistics Education, American Statistical Association*, 343–348.

Jang, D. S. and J. L. Eltinge. 1995. Empirical assessment of the stability of variance estimators based on a two-clusters-per-stratum design. Technical Report #225, Department of Statistics, Texas A&M University. Submitted for publication.

Johnson, W. 1995. Variance estimation for the NMIHS. Technical document. National Center for Health Statistics, Hyattsville, MD.

Kish, L. 1965. *Survey Sampling*. New York: John Wiley & Sons.

——. 1995. Methods for design effects. *Journal of Official Statistics* 11: 55–77.

Korn, E. L., and B. I. Graubard. 1990. Simultaneous testing of regression coefficients with complex survey data: use of Bonferroni t statistics. *The 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* 15(1). National Center for Health Statistics, Hyattsville, MD.

Satterthwaite, F. E. 1941. Synthesis of variance. *Psychometrika* 6: 309–316.

——. 1946. An approximate distribution of estimates of variance components. *Biometrics Bulletin* 2: 110–114.

Scott, A. J. and D. Holt. 1982. The effect of two-stage sampling on ordinary least squares methods. *Journal of the American Statistical Association* 77: 848–854.

Skinner, C. J. 1989. Introduction to Part A. In *Analysis of Complex Surveys*, ed. C. J. Skinner, D. Holt, and T. M. F. Smith, 23–58. New York: John Wiley & Sons.

Wolter, K. M. 1985. *Introduction to Variance Estimation*. New York: Springer-Verlag.

## svy3 | Describing survey data: sampling design and missing data

John L. Eltinge, Texas A&M University, FAX 409-845-3144, EMAIL jeltinge@stat.tamu.edu
William M. Sribney, Stata Corporation, FAX 409-696-4601, EMAIL tech_support@stata.com

The syntax for the `svydes` command is

```
svydes [varlist] [weight] [if exp] [in range] [, strata(varname) psu(varname)
                                                 fpc(varname) bypsu ]
```

`pweights` and `iweights` are allowed.

[*Editor's note: The ado file for this command can be found in the directory svy1.*]

### Description

Sample-survey data are typically stratified. Within each stratum, there are primary sampling units (PSUs), which may be either clusters of observations or individual observations. `svydes` displays a table that describes the strata and PSUs in the dataset. By default, one row of the table is produced for each stratum. Displayed for each stratum are the number of PSUs, the range and mean of the number of observations per PSU, and the total number of observations. If the `bypsu` option is specified, `svydes` will display the number of observations in each PSU for every PSU in the dataset.

If a *varlist* is specified, `svydes` will report the number of PSUs that contain at least one observation with complete data (i.e., no missing values) for all variables in the *varlist*. These are precisely the PSUs that would be used to compute estimates for the variables in *varlist* using the `svy` estimation commands: `svymean`, `svytotal`, `svyratio`, `svyprop`, `svyreg`, `svylogit`, and `svyprobt`. The variance estimation formulas for these `svy` estimation commands require at least two PSUs per stratum. If there are some strata with only a single PSU, an error message is displayed.

```
. svymean x
stratum with only one PSU detected
r(499);
. svydes x
```

The stratum (or strata) with only one PSU can be located from the table produced by '`svydes x`'. After locating this stratum, it can be "collapsed" into an adjacent stratum, and then variance estimates can be computed. See the following examples for an illustration of the procedure.

For details on the `svy` estimation commands, see the articles *svy2* and *svy4* in this issue.

### Options

`strata(`*varname*`)` specifies the name of a variable (numeric or string) that contains stratum identifiers. `strata()` can also be specified with the `varset` command.

`psu(`*varname*`)` specifies the name of a variable (numeric or string) that contains identifiers for the primary sampling unit (i.e., the cluster). `psu()` can also be specified with the `varset` command.

`fpc(`*varname*`)` can be set here or with the `varset` command. If an `fpc` variable has been specified, `svydes` checks the `fpc` variable for missing values. Other than this, `svydes` does not use the `fpc` variable. See the article *svy2* for details on the `fpc`.

`bypsu` specifies that results be displayed for each PSU in the dataset; i.e., a separate line of output is produced for every PSU. This option can only be used when a PSU variable has been specified using the `psu()` option or set with `varset`.

Note: weights are checked for missing values, but are not otherwise used by `svydes`.

## Examples

We use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981) as our example. First, we set the `strata`, `psu`, and `pweight` variables.

```
. varset strata strata
. varset psu psu
. varset pweight finalwgt
```

Typing `svydes` will show us the strata and PSU arrangement of the dataset.

```
. svydes
pweight:  finalwgt
Strata:   strata
PSU:      psu
                                        #Obs per PSU
 Strata                          ----------------------------
 strata     #PSUs     #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
       6         2       298       131     149.0       167
       7         2       476       206     238.0       270
       8         2       338       158     169.0       180
       9         2       244       100     122.0       144
      10         2       262       119     131.0       143
      11         2       275       120     137.5       155
      12         2       314       144     157.0       170
      13         2       342       154     171.0       188
      14         2       405       200     202.5       205
      15         2       380       189     190.0       191
      16         2       336       159     168.0       177
      17         2       393       180     196.5       213
      18         2       359       144     179.5       215
      20         2       285       125     142.5       160
      21         2       214       102     107.0       112
      22         2       301       128     150.5       173
      23         2       341       159     170.5       182
      24         2       438       205     219.0       233
      25         2       256       116     128.0       140
      26         2       261       129     130.5       132
      27         2       283       139     141.5       144
      28         2       299       136     149.5       163
      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     10351        67     167.0       288
```

Our NHANES II dataset has 31 strata (stratum 19 is missing) and 2 PSUs per stratum.

The variable `hdresult` contains serum levels of high-density lipoproteins (HDL). If we try to estimate the mean of `hdresult`, we get an error.

```
. svymean hdresult
stratum with only one PSU detected
r(499);
```

Running `svydes` with `hdresult` as its *varlist* will show us which stratum or strata have only one PSU.

```
. svydes hdresult
```

```
pweight:  finalwgt
Strata:   strata
PSU:      psu
                          #Obs with  #Obs with      #Obs per included PSU
 Strata     #PSUs     #PSUs  complete   missing  ------------------------------
 strata   included  omitted    data      data       min      mean       max
--------  --------  --------  --------  --------  --------  --------  --------
       1         1         1       114       266       114     114.0       114
       2         1         1        98        87        98      98.0        98
       3         2         0       277        71       116     138.5       161
       4         2         0       340       120       160     170.0       180
       5         2         0       173        79        81      86.5        92
       6         2         0       255        43       116     127.5       139
       7         2         0       409        67       191     204.5       218
       8         2         0       299        39       129     149.5       170
       9         2         0       218        26        85     109.0       133
      10         2         0       233        29       103     116.5       130
      11         2         0       238        37        97     119.0       141
      12         2         0       275        39       121     137.5       154
      13         2         0       297        45       123     148.5       174
      14         2         0       355        50       167     177.5       188
      15         2         0       329        51       151     164.5       178
      16         2         0       280        56       134     140.0       146
      17         2         0       352        41       155     176.0       197
      18         2         0       335        24       135     167.5       200
      20         2         0       240        45        95     120.0       145
      21         2         0       198        16        91      99.0       107
      22         2         0       263        38       116     131.5       147
      23         2         0       304        37       143     152.0       161
      24         2         0       388        50       182     194.0       206
      25         2         0       239        17       106     119.5       133
      26         2         0       240        21       119     120.0       121
      27         2         0       259        24       127     129.5       132
      28         2         0       284        15       131     142.0       153
      29         2         0       440        63       193     220.0       247
      30         2         0       326        39       147     163.0       179
      31         2         0       279        29       121     139.5       158
      32         2         0       383        67       180     191.5       203
--------  --------  --------  --------  --------  --------  --------  --------
      31        60         2      8720      1631        81     145.3       247
                              ----------------
                                     10351
```

Both of `strata==1` and `strata==2` have only one PSU with nonmissing values of `hdresult`. Since this dataset has only 62 PSUs, the `bypsu` option will give a manageable amount of output:

```
. svydes hdresult, bypsu
pweight:  finalwgt
Strata:   strata
PSU:      psu
                  #Obs with  #Obs with
 Strata      PSU    complete   missing
 strata      psu      data      data
--------  --------  --------  --------
       1         1         0       215
       1         2       114        51
       2         1        98        20
       2         2         0        67
       3         1       161        38
       3         2       116        33
(output omitted)
      32         1       180        59
      32         2       203         8
--------  --------  --------  --------
      31        62      8720      1631
                  ----------------
                         10351
```

It is rather striking that there are two PSUs without any values for `hdresult`. All other PSUs have only a moderate number of missing values. Obviously, in a case such as this, a data analyst should first try to ascertain the reason why these data are missing. The answer here (Johnson 1995) is that HDL measurements could not be collected until the third survey location. Thus, there are no `hdresult` data for the first two locations: `strata==1 & psu==1` and `strata==2 & psu==2`.

Assuming that we wish to go ahead and analyze the `hdresult` data, we must "collapse" strata—that is, merge them together—so that every stratum has at least two PSUs with some nonmissing values. We can accomplish this by collapsing `strata==1` into `strata==2`. To perform the stratum collapse, we create a new strata identifier `newstr` and a new PSU identifier `newpsu`. This is easy to do using basic commands in Stata.

```
. gen newstr = strata
. gen newpsu = psu
. replace newpsu = psu + 2 if strata==1
(380 real changes made)
. replace newstr = 2 if strata==1
(380 real changes made)
```

We set the new strata and PSU variables.

```
. varset strata newstr
. varset psu newpsu
```

We use `svydes` to check what we have done.

```
. svydes hdresult, bypsu
pweight:  finalwgt
Strata:   newstr
PSU:      newpsu
                    #Obs with  #Obs with
 Strata      PSU     complete    missing
 newstr    newpsu      data       data
--------  --------  ---------  --------
       2         1         98         20
       2         2          0         67
       2         3          0        215
       2         4        114         51
       3         1        161         38
       3         2        116         33
(output omitted)
      32         1        180         59
      32         2        203          8
--------  --------  ---------  --------
      30        62       8720       1631
                    ----------
                         10351
```

The new stratum, `newstr==2`, has 4 PSUs, 2 of which contain some nonmissing values of `hdresult`. This is sufficient to allow us to estimate the mean of `hdresult`.

```
. svymean hdresult
Survey mean estimation
pweight:  finalwgt                              Number of obs    =      8720
Strata:   newstr                                Number of strata =        30
PSU:      newpsu                                Number of PSUs   =        60
                                                Population size  =  98725345
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
hdresult |   49.67141    .3830147    48.88919    50.45364    6.257131
------------------------------------------------------------------------------
```

## References

Johnson, C. L. 1995. Personal communication.

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* 15(1). National Center for Health Statistics, Hyattsville, MD.

| svy4 | Linear, logistic, and probit regressions for survey data |
|---|---|

John L. Eltinge, Texas A&M University, FAX 409-845-3144, EMAIL jeltinge@stat.tamu.edu
William M. Sribney, Stata Corporation, FAX 409-696-4601, EMAIL tech_support@stata.com

The syntax for the `svyreg`, `svylogit`, and `svyprobt` commands is

`svyreg` *varlist* [*weight*] [`if` *exp*] [`in` *range*] [`,` *options* ]

`svylogit` *varlist* [*weight*] [`if` *exp*] [`in` *range*] [`,` `or` *maximize_options options* ]

`svyprobt` *varlist* [*weight*] [`if` *exp*] [`in` *range*] [`,` *maximize_options options* ]

where the common *options* are

`noconstant` `strata(`*varname*`)` `psu(`*varname*`)` `fpc(`*varname*`)`
`subpop(`*expression*`)` `srssubpop` `noadjust` `float`
`level(#)` `prob` `ci` `deff` `deft` `meff` `meft`

These commands share the features of all estimation commands. The commands typed without arguments redisplay previous results. The following options can be given when redisplaying results:

`or level(#) prob ci deff deft meff meft`

`svyreg` allows `pweight`s and `iweight`s. `svylogit` and `svyprobt` allow only `pweight`s.

Warning: Use of `if` or `in` restrictions will not produce correct variance estimates for subpopulations in many cases. To compute estimates for a subpopulation, use the `subpop()` option.

[*Editor's note: The ado files for these commands can be found in the directory svy1.*]

## Description

These commands estimate regression models for complex survey data. `svyreg` estimates linear regression. `svylogit` estimates pseudo-maximum-likelihood logistic regression. `svyprobt` estimates a pseudo-maximum-likelihood probit model (see Skinner et al. (1989, Section 3.4.4) for a discussion of pseudo-MLEs). The dependent variable for `svylogit` and `svyprobt` should be a 0/1 variable (or, more precisely, a zero/not zero variable).

The commands allow any or all of the following: probability sampling weights, stratification, and clustering. Associated variance estimates, design effects (deff and deft), and misspecification effects (meff and meft) are computed. The `subpop()` option will give estimates for a single subpopulation defined by an expression.

Many of the options here are the same as those for the `svymean`, `svytotal`, and `svyratio` commands. The article *svy2* in this issue should be read first for an understanding of these shared `svy` command options.

## Options

`noconstant` estimates a model without the constant term (intercept).

`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, $d$ = 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 discussion of the Wald test and adjustments thereof.

`float` specifies that covariance computations be done in float precision rather than double (the default). Using double precision requires room for $k+1$ variables of type `double`, where $k$ is the number of variables in the model. Using the `float` option requires room for $k+1$ variables of type `float`. Coefficient estimates are always computed in double precision.

*maximize_options* (`svylogit` and `svyprobt` only) control the maximization process; see [7] maximize in the reference manual. You should never have to specify them.

The following options can be specified initially or when redisplaying results:

`or` (`svylogit` only) reports the estimated coefficients transformed to odds ratios, i.e., $\exp(\widehat{b})$ rather than $\widehat{b}$. Standard errors and confidence intervals are similarly transformed.

`level(#)` specifies the confidence level (i.e., nominal coverage rate), in percent, for confidence intervals.

`prob` requests that the $t$ statistic and $p$-value be displayed. The degrees of freedom for the $t$ statistic are $d$ = total number of sampled PSUs minus the total number of strata (regardless of the number of terms in the model). If no display options are specified, then, by default, the $t$ statistic and $p$-value are displayed.

`ci` requests that confidence intervals be displayed. If no display options are specified, then, by default, confidence intervals are displayed.

`deff` requests that the design-effect measure deff be displayed; see *svy2* for details.

`deft` requests that the design-effect measure deft be displayed; see *svy2* for details.

`meff` requests that the meff measure of misspecification effects be displayed; see *svy2* for details.

`meft` requests that the meft measure of misspecification effects be displayed; see *svy2* for details.

See the article *svy2* in this issue for a description of the remaining options:

`strata(`*varname*`) psu(`*varname*`) fpc(`*varname*`) subpop(`*expression*`) srssubpop`

## Examples

We use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981) as our example. First, we set the `strata`, `psu`, and `pweight` variables.

```
. varset pweight leadwt
. varset strata strata
. varset psu psu
```

Once the `strata`, `psu`, and `pweight` variables are set, we can use `svyreg` just as we would `regress` with nonsurvey data.

```
. svyreg loglead age female black orace region2-region4 smsa1 smsa2
Survey linear regression
pweight:  leadwt                                  Number of obs    =      4948
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.129e+08
                                                  F(   9,     23)  =    134.62
                                                  Prob > F         =    0.0000
                                                  R-squared        =    0.2443
------------------------------------------------------------------------------
 loglead |      Coef.    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   .0028425    .0004282      6.638   0.000      .0019691    .0037159
  female |  -.3641964    .0112612    -32.341   0.000     -.3871637   -.3412291
   black |   .1462126    .0277811      5.263   0.000      .0895527    .2028725
   orace |  -.0754489    .0370151     -2.038   0.050     -.1509418    .0000439
 region2 |  -.0206953    .0456639     -0.453   0.654     -.1138274    .0724369
 region3 |  -.1272598    .0528061     -2.410   0.022     -.2349586   -.0195611
 region4 |  -.0374591    .0422001     -0.888   0.382     -.1235268    .0486085
   smsa1 |   .1038586    .0432539      2.401   0.023      .0156417    .1920755
   smsa2 |   .0995561    .0365985      2.720   0.011      .0249129    .1741993
   _cons |   2.623901    .0421096     62.311   0.000      2.538018    2.709784
------------------------------------------------------------------------------
```

If we wish to test joint hypotheses after the regression, we can use the `svytest` command. See *svy5* for details on this command.

Running logistic regressions with `svylogit` is as simple as running the `logit` command. Note that, just like, `logit` the dependent variable should be a 0/1 variable (or, more precisely, a zero/not zero variable).

```
. svylogit highlead age female black orace
(sum of wgt is   1.1292e+08)
Survey logistic regression
pweight:  leadwt                                  Number of obs    =      4948
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.129e+08
                                                  F(   4,     28)  =     33.23
                                                  Prob > F         =    0.0000
------------------------------------------------------------------------------
highlead |      Coef.    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   .0085094    .0037505      2.269   0.030      .0008602    .0161587
  female |  -2.111908    .1801479    -11.723   0.000     -2.479322   -1.744494
   black |   .6037818    .2298884      2.626   0.013      .1349213    1.072642
   orace |  -.1988237    .4337411     -0.458   0.650     -1.083445     .685797
   _cons |  -2.498023    .2124733    -11.757   0.000     -2.931365   -2.064681
------------------------------------------------------------------------------
```

We can redisplay the results as odds ratios using the `or` option.

```
. svylogit, or
Survey logistic regression
pweight:  leadwt                                  Number of obs    =      4948
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.129e+08
                                                  F(   4,     28)  =     33.23
                                                  Prob > F         =    0.0000
------------------------------------------------------------------------------
highlead | Odds Ratio    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   1.008546    .0037826      2.269   0.030      1.000861     1.01629
  female |   .1210068    .0217991    -11.723   0.000         .0838    .1747333
   black |   1.829023    .4204711      2.626   0.013      1.144447    2.923093
   orace |   .8196944    .3555351     -0.458   0.650      .3384278    1.985354
------------------------------------------------------------------------------
```

To estimate a model for a subpopulation, the `subpop()` option is used.

```
. svylogit highlead age female, subpop(black==1) or
(sum of wgt is   1.0490e+07)
Survey logistic regression
pweight:  leadwt                                  Number of obs    =      4948
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.129e+08
Subpopulation no. of obs =       506              F(   2,     30)  =     13.32
Subpopulation size       =  10490430              Prob > F         =    0.0001
------------------------------------------------------------------------------
highlead | Odds Ratio    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   1.015155    .0082557      1.850   0.074      .9984561    1.032133
  female |   .0281831    .0204156     -4.927   0.000      .0064322    .1234857
------------------------------------------------------------------------------
```

This time we specified the `or` option when we first issued the command. Note that using '`if black==1`' to model the subpopulation would not give the same result. All of the discussion in the section *Warning about the use of if and in* in the article *svy2* applies to the variance estimates for `svyreg`, `svylogit`, and `svyprobt`. Also remember that if there are observations with missing values for any of the variables that determine the subpopulation, they must be explicitly omitted. For example, if the variable `black` had missing values, we would estimate the subpopulation model using

```
. svylogit highlead age female if black~=., subpop(black==1) or
```

## Methods and Formulas

The commands `svyreg`, `svylogit`, and `svyprobt` use variants on the basic weighted-point-estimation methods used by `svytotal`. In addition, these regression commands use "linearization"-based variance estimators that are natural extensions of the variance estimator used in `svytotal`. For general methodological background on regression and generalized-linear-model analyses of complex survey data, see, for example, Binder (1983), Cochran (1977), Fuller (1975), Godambe (1991), Kish and Frankel (1974), Särndal et al. (1992), Shao (1996), and Skinner et al. (1989). The notation and development presented below is adapted from Binder (1983).

*Linear regression*

We use here the same notation as in the *Methods and Formulas* section of the article *svy2*; that section should be read first. Again, we let $(h, i, j)$ index the elements in the population, where $h = 1, \ldots, L$ are the strata, $i = 1, \ldots, N_h$ are the PSUs in stratum $h$, and $j = 1, \ldots, M_{hi}$ are the elements in PSU $(h, i)$. The regression coefficients $\beta = (\beta_0, \beta_1, \ldots, \beta_k)$ are viewed as fixed finite-population parameters that we wish to estimate. These parameters are defined with respect to an outcome variable $Y_{hij}$ and a $k+1$-dimensional row vector of explanatory variables $X_{hij} = (X_{hij0}, \ldots, X_{hijk})$. As in non-survey work, we often have $X_{hij0}$ identically equal to unity, so that $\beta_0$ is an intercept coefficient. Within a finite-population context, we can formally define the regression coefficient vector $\beta$ as the solution to the vector estimating equation

$$G(\beta) = X'Y - X'X\beta = 0 \tag{1}$$

where $Y$ is the vector of outcomes for the full population and $X$ is the matrix of explanatory variables for the full population. Assuming $(X'X)^{-1}$ exists, the solution to (1) is $\beta = (X'X)^{-1}X'Y$.

Given observations $(y_{hij}, x_{hij})$ collected through a complex sample design, we need to estimate $\beta$ in a way that accounts for the sample design. To do this, note that the matrix factors $X'X$ and $X'Y$, can be viewed as matrix population totals. For example, $X'Y = \sum_{h=1}^{L}\sum_{i=1}^{N_h}\sum_{j=1}^{M_{hi}} X_{hij}Y_{hij}$. Thus, we estimate $X'X$ and $X'Y$ with the weighted estimators

$$\widehat{X'X} = \sum_{h=1}^{L}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}x'_{hij}x_{hij} = X'_{\rm s}WX_{\rm s}$$

and

$$\widehat{X'Y} = \sum_{h=1}^{L}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}x'_{hij}y_{hij} = X'_{\rm s}WY_{\rm s}$$

where $X_{\rm s}$ is the matrix of explanatory variables for the sample, $Y_{\rm s}$ is the outcome vector for the sample, and $W = \mathrm{diag}(w_{hij})$ is a diagonal matrix containing the sampling weights $w_{hij}$. The corresponding coefficient estimator is

$$\widehat{\beta} = (\widehat{X'X})^{-1}\widehat{X'Y} = (X'_{\rm s}WX_{\rm s})^{-1}X'_{\rm s}WY_{\rm s} \tag{2}$$

Note that equation (2) is what the `regress` command with `aweights` or `iweights` computes for point estimates.

The coefficient estimator $\widehat{\beta}$ can also be defined as the solution to the weighted sample estimating equation

$$\widehat{G}(\beta) = \widehat{X'Y} - \widehat{X'X}\beta = X'_{\rm s}WY_{\rm s} - X'_{\rm s}WX_{\rm s}\beta = 0$$

We can write $\widehat{G}(\beta)$ as

$$\widehat{G}(\beta) = \sum_{h=1}^{L}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}d_{hij} \tag{3}$$

where $d_{hij} = x'_{hij}e_{hij}$ and $e_{hij} = y_{hij} - x_{hij}\beta$ is the regression residual associated with sample unit $(h, i, j)$. Thus, $\widehat{G}(\beta)$ can be viewed as a special case of a total estimator.

Our variance estimator for $\widehat{\beta}$ is based on the following "linearization" argument. A first-order matrix Taylor expansion shows that

$$\widehat{\beta} - \beta \doteq -\left[\frac{\partial\widehat{G}(\beta)}{\partial\beta}\right]^{-1}\widehat{G}(\beta)$$

Thus, our variance estimator for $\widehat{\beta}$ is

$$\widehat{V}(\widehat{\beta}) = \left\{\left[\frac{\partial\widehat{G}(\beta)}{\partial\beta}\right]^{-1}\widehat{V}(\widehat{G}(\beta))\left[\frac{\partial\widehat{G}(\beta)}{\partial\beta}\right]^{-\mathrm{T}}\right\}\Bigg|_{\beta=\widehat{\beta}} = [X'_{\rm s}WX_{\rm s}]^{-1}\,\widehat{V}(\widehat{G}(\beta))\big|_{\beta=\widehat{\beta}}\,[X'_{\rm s}WX_{\rm s}]^{-1}$$

Viewing $\widehat{G}(\beta)$ as a total estimator according to equation (3), the variance estimator $\widehat{V}(\widehat{G}(\beta))\big|_{\beta=\widehat{\beta}}$ can be computed using equation (3) from *svy2* with $y_{hij}$ replaced by $d_{hij}$ and with $\widehat{\beta}$ used to estimate $e_{hij}$.

*Logistic regression and probit estimation*

To develop notation for our estimators in `svylogit` and `svyprobt`, suppose that we observed $(Y_{hij}, X_{hij})$ for the entire population, and that $(Y_{hij}, X_{hij})$ arose from a certain logistic regression or probit model. Let $l(\beta; Y_{hij}, X_{hij})$ be the associated "log-likelihood" under this model. Then, for our finite population, we define the parameter $\beta$ by the vector estimating equation

$$G(\beta) = \sum_{h=1}^{L}\sum_{i=1}^{N_h}\sum_{j=1}^{M_{hi}} S(\beta; Y_{hij}, X_{hij}) = 0$$

where $S = \partial l/\partial\beta$ is the score vector; i.e., the first derivative with respect to $\beta$ of $l(\beta; Y_{hij}, X_{hij})$. Then, the "pseudo-maximum-likelihood" estimator $\widehat{\beta}$ is the solution to the weighted sample estimating equation

$$\widehat{G}(\beta) = \sum_{h=1}^{L}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}S(\beta; y_{hij}, x_{hij}) = 0 \tag{4}$$

See Skinner et al. (1989, Section 3.4.4) for a discussion of pseudo-MLEs. Note that the solution $\widehat{\beta}$ of equation (4) is what the `logit` or `probit` command with `aweights` produces for point estimates.

Again, we use a first-order matrix Taylor series expansion to produce the variance estimator for $\widehat{\beta}$

$$\widehat{V}(\widehat{\beta}) = \left\{ \left[\frac{\partial \widehat{G}(\beta)}{\partial \beta}\right]^{-1} \widehat{V}(\widehat{G}(\beta)) \left[\frac{\partial \widehat{G}(\beta)}{\partial \beta}\right]^{-\mathrm{T}} \right\}\Bigg|_{\beta=\widehat{\beta}} = H^{-1}\widehat{V}(\widehat{G}(\beta))\big|_{\beta=\widehat{\beta}} H^{-1}$$

where $H$ is the Hessian matrix for the weighted sample log-likelihood. We can write $\widehat{G}(\beta)$ as

$$\widehat{G}(\beta) = \sum_{h=1}^{L}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij} d_{hij}$$

where $d_{hij} = x'_{hij} s_{hij}$ and $s_{hij}$ is the score function for element $(h,i,j)$. The term $s_{hij}$ is computed by rewriting the sample log-likelihood $l(\beta; y_{hij}, x_{hij})$ as a function of $z_{hij} = x'_{hij}\beta$:

$$s_{hij} = \frac{\partial l(z_{hij}; y_{hij})}{\partial z_{hij}}$$

Thus, again, $\widehat{G}(\beta)$ can be viewed as a special case of a total estimator, and the variance estimator $\widehat{V}(\widehat{G}(\beta))\big|_{\beta=\widehat{\beta}}$ is computed using equation (3) from *svy2* with $y_{hij}$ replaced by $d_{hij}$ and with $\widehat{\beta}$ used to estimate $s_{hij}$.

## References


Binder, D. A. 1983. On the variances of asymptotically normal estimators from complex surveys. *International Statistical Review* 51: 279–292.

Cochran, W. G. 1977. *Sampling Techniques.* 3d ed. New York: John Wiley & Sons.

Fuller, W. A. 1975. Regression analysis for sample survey. *Sankhyā, Series C* 37: 117–132.

Godambe, V. P. ed. 1991. *Estimating Functions.* Oxford: Clarendon Press.

Gonzalez Jr., J. F., N. Krauss, and C. Scott. 1992. Estimation in the 1988 National Maternal and Infant Health Survey. In *Proceedings of the Section on Statistics Education, American Statistical Association*, 343–348.

Johnson, W. 1995. Variance estimation for the NMIHS. Technical document. National Center for Health Statistics, Hyattsville, MD.

Kish, L. and M. R. Frankel. 1974. Inference from complex samples. *Journal of the Royal Statistical Society* A 36: 1–37.

Korn, E. L., and B. I. Graubard. 1990. Simultaneous testing of regression coefficients with complex survey data: use of Bonferroni t statistics. *The 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* 15(1). National Center for Health Statistics, Hyattsville, MD.

Särndal, C.-E., B. Swensson, and J. Wretman. 1992. *Model Assisted Survey Sampling.* New York: Springer-Verlag.

Shao, J. 1996. Resampling methods for sample surveys (with discussion). *Statistics* 27: 203–254.

Skinner, C. J. 1989. Introduction to Part A. In *Analysis of Complex Surveys*, ed. C. J. Skinner, D. Holt, and T. M. F. Smith, 23–58. New York: John Wiley & Sons.


## svy5 | Estimates of linear combinations and hypothesis tests for survey data


John L. Eltinge, Texas A&M University, FAX 409-845-3144, EMAIL jeltinge@stat.tamu.edu
William M. Sribney, Stata Corporation, FAX 409-696-4601, EMAIL tech_support@stata.com


The syntax for the `svylc` command is

`svylc` [*exp*] [`, show or level(#) deff deft meff meft`]

The command `svytest` can be used with three different syntaxes:

(1) `svytest` *exp* = *exp* [`, noadjust accumulate notest`]
(2) `svytest` *coefficientlist* [`, noadjust accumulate notest`]
(3) `svytest` [*varlist*] `, bonferroni`

In the above, *exp* is a linear expression that is valid for the `test` command; *exp* = *exp* is a linear equation that is valid for the `test` command; and *coefficientlist* is a valid coefficient list for the `test` command; see [5s] test in the reference manual.

[*Editor's note: The ado files for these commands can be found in the directory svy1.*]

## Description

`svylc` produces estimates for linear combinations of parameters after a `svy` estimation command; i.e., any of the commands `svymean`, `svytotal`, `svyratio`, `svyreg`, `svylogit`, or `svyprobt`. `svytest` tests multidimensional linear hypotheses after a `svy` estimation command. See the articles *svy2* and *svy4* in this issue for an introduction to the `svy` estimation commands.

By default, `svylc` computes the point estimate, standard error, $t$ statistic, $p$-value, and confidence interval for the specified linear combination. Design effects (deff and deft) and misspecification effects (meff and meft) can be optionally displayed; see *svy2* in this issue for a detailed description of these options.

Syntax (1) for `svytest` allows you to build up a multidimensional hypothesis consisting of any number of linear equations. Syntax (2) tests hypotheses of the form $\beta_1 = 0$, $\beta_2 = 0$, $\beta_3 = 0$, etc. Syntax (3) is only available after the regression commands `svyreg`, `svylogit`, and `svyprobt`. It computes a Bonferroni adjustment for hypotheses of the form $\beta_1 = 0$, $\beta_2 = 0$, $\beta_3 = 0$, etc. See the following examples and the *Methods and Formulas* section for details.

By default, `svytest` used with syntax (1) or (2) carries out an adjusted Wald test. Specifically, it uses the approximate $F$ statistic $(d-k+1)W/(kd)$, where $W$ is the Wald test statistic, $k$ is the dimension of the hypothesis test, and $d =$ total number of sampled PSUs minus the total number of strata. Under the null hypothesis, $(d-k+1)W/(kd) \sim F(k, d-k+1)$, where $F(k, d-k+1)$ is an $F$ distribution with $k$ numerator degrees of freedom and $d-k+1$ denominator degrees of freedom.

## Options

`show` requests that the labeling syntax for the previous `svy` estimates be displayed. This is useful when the `svy` estimation command produced estimates for subpopulations using the `by()` option. When `show` is specified, no expression *exp* is specified.

`or` (after `svylogit` only) reports the estimated coefficients transformed to odds ratios, i.e., $\exp(\widehat{b})$ rather than $\widehat{b}$. Standard errors and confidence intervals are similarly transformed.

`level(#)` specifies the confidence level (i.e., nominal coverage rate), in percent, for confidence intervals. The default is `level(95)` or as set by `set level`.

`deff` requests that the design-effect measure deff be displayed.

`deft` requests that the design-effect measure deft be displayed. See *svy2* for a discussion of deff and deft.

`meff` requests that the meff measure of misspecification effects be displayed.

`meft` requests that the meft measure of misspecification effects be displayed. See *svy2* for a discussion of meff and meft.

`noadjust` specifies that the Wald test be carried out as $W/k \sim F(k, d)$ (notation as described above). This gives the same result as the `test` command.

`accumulate` allows a hypothesis to be tested jointly with the previously tested hypotheses.

`notest` suppresses the output. This option is useful when you are interested only in the joint test of a number of hypotheses.

`bonferroni` can be specified only after estimating a model with `svyreg`, `svylogit`, or `svyprobt`. When this option is specified, `svytest` displays adjusted $p$-values for each of the coefficients corresponding to the variables in its *varlist*. Adjusted $p$-values are computed as $p_{\text{adj}} = \min(kp, 1)$, where $k$ is the number of variables specified, and $p$ is the unadjusted $p$-value (i.e., the $p$-value shown in the output of the estimation command) obtained from the statistic $t = \widehat{b}/[\widehat{V}(\widehat{b})]^{1/2}$ which is assumed to have a $t$ distribution with $d$ degrees of freedom. If no *varlist* is specified with the `bonferroni` option, adjustments are made for all terms in the model excluding the constant.

## The use of svylc when there are no by() subpopulations

We use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981) as our example. Suppose that we wish to estimate the difference of the means of systolic (variable `bpsystol`) and diastolic (variable `bpdiast`) blood pressures. First, we estimate the means, and then we use `svylc`.

```
. svymean bpsystol bpdiast
Survey mean estimation
pweight:  finalwgt                          Number of obs    =     10351
Strata:   strata                            Number of strata =        31
PSU:      psu                               Number of PSUs   =        62
                                            Population size  = 1.172e+08
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
bpsystol |   126.9458     .603462     125.715    128.1766    8.230475
 bpdiast |   81.01726    .5090314    79.97909    82.05544    16.38656
------------------------------------------------------------------------------
```

```
. svylc bpsystol - bpdiast
 ( 1)  bpsystol - bpdiast = 0.0
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |   45.92852    .2988395    153.690   0.000      45.31903    46.53801
------------------------------------------------------------------------------
```

We can also specify any of the options `deff`, `deft`, `meff`, or `meft`, or change the confidence level (i.e., nominal coverage rate) of the confidence interval.

```
. svylc bpsystol - bpdiast, level(90) deff meff
 ( 1)  bpsystol - bpdiast = 0.0
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.       t     P>|t|      [90% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |   45.92852    .2988395    153.690   0.000      45.42183    46.43521
------------------------------------------------------------------------------
------------------------------------------------------------------------------
    Mean |       Deff        Meff
---------+--------------------------------------------------------------------
     (1) |   3.835532    3.087148
------------------------------------------------------------------------------
```

`svylc` works in the same manner after using the `subpop` option.

```
. svymean bpsystol bpdiast if female~=., subpop(female==1)
Survey mean estimation
pweight:  finalwgt                              Number of obs    =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
Subpop.:  female==1                             Population size  = 1.172e+08
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
bpsystol |   124.2027    .7051858    122.7644    125.6409    5.162487
 bpdiast |   79.03227    .5207306    77.97023    80.09431    8.973799
------------------------------------------------------------------------------

. svylc bpsystol - bpdiast
 ( 1)  bpsystol - bpdiast = 0.0
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |   45.17039    .4040852    111.784   0.000      44.34625    45.99453
------------------------------------------------------------------------------
```

## Missing data: The complete and available options for svymean, svytotal, and svyratio

The `svymean`, `svytotal`, and `svyratio` commands can handle missing data in two ways. The `available` option (which is the default when there are missing values and two or more variables) uses every available nonmissing value for each variable separately. The `complete` option (which is the default when there are no missing values or only one variable) uses only those observations with nonmissing values for all variables in the *varlist*. Here is an example where `available` is the default.

```
. svymean tcresult tgresult
Survey mean estimation
pweight:  finalwgt                              Number of obs(*) =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
tcresult |   213.0977    1.127252    210.7986    215.3967    5.602499
tgresult |    138.576    2.071934    134.3503    142.8018    2.356968
------------------------------------------------------------------------------
(*) Some variables contain missing values.
```

We redisplay the results using the `obs` option to see how many observations were used for each estimate.

```
. svymean, obs
Survey mean estimation
pweight:  finalwgt                              Number of obs(*) =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.        Obs
---------+--------------------------------------------------------------------
tcresult |   213.0977    1.127252      10351
tgresult |    138.576    2.071934       5050
------------------------------------------------------------------------------
(*) Some variables contain missing values.
```

Because we estimated the mean of `tgresult` using a different set of observations than `tcresult`, we could not compute the covariance between the two, and hence, we cannot estimate the variance of the difference. So if we now ask `svylc` to estimate this difference, we get an error message.

```
. svylc tcresult - tgresult
must run svy command with "complete" option before using this command
r(301);
```

We need not know in advance whether the default was `available` or `complete`. `svylc` will always tell us if we need to run the estimation command again with the `complete` option. With the `complete` option, only those observations that have nonmissing values for both `tgresult` and `tcresult` are used when we compute the means. When this option is specified, `svymean` computes the full covariance matrix.

```
. svymean tcresult tgresult, complete
Survey mean estimation
pweight:  finalwgt                              Number of obs    =      5050
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  =  56820832
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------+--------------------------------------------------------------------
tcresult |   211.3975    1.252274    208.8435    213.9515    3.571411
tgresult |    138.576    2.071934    134.3503    142.8018    2.356968
------------------------------------------------------------------------------
```

Now we can estimate the variance of the difference.

```
. svylc tcresult - tgresult
 ( 1)  tcresult - tgresult = 0.0
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |   72.82146    2.039786     35.701   0.000      68.66129    76.98163
------------------------------------------------------------------------------
```

Complete-case analyses are essentially based on the assumption that the missingness of values within the dataset is uninformative. Caution is advised in analysis of any dataset involving a substantial amount of missing data.

## The use of svylc after svyreg, svylogit, and svyprobt

`svylc` can be used after `svyreg`, `svylogit`, and `svyprobt` to estimate a linear combination of the coefficients (i.e., the $\beta$'s).

```
. svyreg tcresult bpsystol bpdiast age age2
Survey linear regression
pweight:  finalwgt                              Number of obs    =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08
                                                F(   4,     28)  =    307.00
                                                Prob > F         =    0.0000
                                                R-squared        =    0.1945
------------------------------------------------------------------------------
```

```
tcresult |      Coef.    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+------------------------------------------------------------------
bpsystol |   .1060743    .0346796      3.059   0.005      .0353449    .1768038
 bpdiast |   .2966662    .0569594      5.208   0.000      .1804969    .4128356
     age |    3.35711    .2099842     15.987   0.000      2.928844    3.785375
    age2 |  -.0247207    .0020795    -11.888   0.000     -.0289619   -.0204796
   _cons |    83.8242    5.649261     14.838   0.000      72.30246    95.34594
------------------------------------------------------------------------------

. svylc bpsystol - bpdiast
 ( 1)  bpsystol - bpdiast = 0.0
------------------------------------------------------------------------------
tcresult |      Coef.    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+------------------------------------------------------------------
     (1) |  -.1905919    .0818056     -2.330   0.027     -.3574354   -.0237483
------------------------------------------------------------------------------
```

Note that `svyreg`, `svylogit`, and `svyprobt` always use only complete cases, so that the covariance is always computed, and `svylc` can always be run afterward.

The variable `highbp` is 1 if a person has high blood pressure and 0 otherwise. We can model it using logistic regression.

```
. svylogit highbp height weight age age2 female black
(sum of wgt is   1.1716e+08)
Survey logistic regression
pweight:  finalwgt                              Number of obs    =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08
                                                F(   6,     26)  =     87.70
                                                Prob > F         =    0.0000
------------------------------------------------------------------------------
  highbp |      Coef.    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+------------------------------------------------------------------
  height |  -.0325996    .0058727     -5.551   0.000     -.0445771   -.0206222
  weight |    .049074    .0031966     15.352   0.000      .0425545    .0555936
     age |   .1541151    .0208709      7.384   0.000      .1115486    .1966815
    age2 |  -.0010746    .0002025     -5.306   0.000     -.0014877   -.0006616
  female |   -.356497    .0885354     -4.027   0.000      -.537066    -.175928
   black |   .3429301    .1409005      2.434   0.021      .0555615    .6302986
   _cons |   -4.89574    1.159135     -4.224   0.000     -7.259813   -2.531668
------------------------------------------------------------------------------
```

We can redisplay the results expressed as odds ratios.

```
. svylogit, or
Survey logistic regression
pweight:  finalwgt                              Number of obs    =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08
                                                F(   6,     26)  =     87.70
                                                Prob > F         =    0.0000
------------------------------------------------------------------------------
  highbp | Odds Ratio    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+------------------------------------------------------------------
  height |    .967926    .0056843     -5.551   0.000      .9564019     .979589
  weight |   1.050298    .0033574     15.352   0.000      1.043473    1.057168
     age |   1.166625    .0243485      7.384   0.000      1.118008    1.217356
    age2 |    .998926    .0002023     -5.306   0.000      .9985135    .9993386
  female |   .7001246    .0619858     -4.027   0.000      .5844605    .8386784
   black |    1.40907    .1985388      2.434   0.021      1.057134    1.878171
------------------------------------------------------------------------------
```

`svylc` can be used to estimate the sum of the coefficients for `female` and `black`.

```
. svylc female + black
 ( 1)  female + black = 0.0
------------------------------------------------------------------------------
  highbp |      Coef.    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+------------------------------------------------------------------
     (1) |  -.0135669    .1653936     -0.082   0.935     -.3508894    .3237555
------------------------------------------------------------------------------
```

This result is more easily interpreted as an odds ratio.

```
. svylc female + black, or
 ( 1)  female + black = 0.0
------------------------------------------------------------------------------
  highbp | Odds Ratio    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |   .9865247    .1631648     -0.082   0.935      .7040616    1.382309
------------------------------------------------------------------------------
```

The odds ratio 0.987 is an estimate of the ratio of the odds of having high blood pressure for black females over the odds for our reference category of nonblack males (controlling for height, weight, and age).

### Subpopulations with one by() variable

The `svymean`, `svytotal`, and `svyratio` commands allow a `by()` option which produces estimates for subpopulations. Frequently, one wishes to compute estimates for differences of subpopulation estimates. It is easy to use `svylc` to compute estimates for differences or any other linear combination of estimates. The only thing one must know is the proper syntax for referencing the subpopulation estimates. In this and the next two sections, we illustrate the syntax with a series of examples.

Suppose that we wish to get an estimate of the difference in mean vitamin C levels (variable `vitaminc`) between males and females. First, we compute the means of `vitaminc` by `sex`.

```
. svymean vitaminc, by(sex)
Survey mean estimation
pweight:  finalwgt                              Number of obs    =      9973
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.129e+08
------------------------------------------------------------------------------
Mean   Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------------+--------------------------------------------------------------
vitaminc       |
          Male |   .9312051    .0169297    .8966768    .9657333    4.926449
        Female |    1.12753    .0173704    1.092103    1.162957    5.028652
------------------------------------------------------------------------------
```

Then we use the `svylc` command.

```
. svylc [vitaminc]Male - [vitaminc]Female
 ( 1)  [vitaminc]Male - [vitaminc]Female = 0.0
------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |  -.1963252     .015981    -12.285   0.000     -.2289186   -.1637318
------------------------------------------------------------------------------
```

When `svymean` or `svytotal` is used with a `by()` option, the syntax for referencing the subpopulation estimates is

[*varname*]*subpop_label*

For example, we use `[vitaminc]Male` to refer to the subpopulation estimates. This is the same syntax that is used with the `test` command when there are multiple equations; see [5s] test in the reference manual for full details.

Be sure to type the variable names and subpopulation labels exactly as they are displayed in the output. Remember that Stata is case sensitive.

```
. svylc [vitaminc]male - [vitaminc]female
male not found
r(111);
```

If there are no subpopulation labels, simply use the numbers displayed in the output.

```
. svymean vitaminc, by(sex) nolabel
Survey mean estimation
pweight:  finalwgt                              Number of obs    =      9973
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.129e+08
------------------------------------------------------------------------------
Mean   Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------------+--------------------------------------------------------------
```

```
vitaminc       |
        sex==1 |   .9312051    .0169297    .8966768    .9657333    4.926449
        sex==2 |    1.12753    .0173704    1.092103    1.162957    5.028652
-------------------------------------------------------------------------------

. svylc [vitaminc]1 - [vitaminc]2
 ( 1)  [vitaminc]1 - [vitaminc]2 = 0.0
-------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+---------------------------------------------------------------------
     (1) |  -.1963252     .015981    -12.285   0.000     -.2289186   -.1637318
-------------------------------------------------------------------------------
```

## Subpopulations with two or more by() variables

If there are two or more `by()` variables, you must refer to the subpopulations by numbers (1, 2, ...) when using `svylc`.

```
. svymean vitaminc, by(sex race)
Survey mean estimation
pweight:  finalwgt                                Number of obs    =      9973
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.129e+08
-------------------------------------------------------------------------------
Mean      Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
------------------+------------------------------------------------------------
vitaminc          |
    Male    White |   .9475117    .0168982    .9130475    .9819758    4.646413
    Male    Black |   .7382045    .0477521    .6408135    .8355955    2.165885
    Male    Other |   1.021363    .0521427     .915017    1.127708    1.739788
  Female    White |   1.151125    .0168117    1.116838    1.185413    4.032603
  Female    Black |   .9222313    .0348224    .8512105     .993252    2.915009
  Female    Other |     1.0804    .0412742    .9962202    1.164579     1.00135
-------------------------------------------------------------------------------
```

You can see the numbering scheme by running `svylc` with the `show` option.

```
. svylc, show
-------------------------------------------------------------------------------
    Mean |      Coef.   Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+---------------------------------------------------------------------
vitaminc |
       1 |   .9475117   .0168982     56.072   0.000      .9130475    .9819758
       2 |   .7382045   .0477521     15.459   0.000      .6408135    .8355955
       3 |   1.021363   .0521427     19.588   0.000       .915017    1.127708
       4 |   1.151125   .0168117     68.472   0.000      1.116838    1.185413
       5 |   .9222313   .0348224     26.484   0.000      .8512105     .993252
       6 |     1.0804   .0412742     26.176   0.000      .9962202    1.164579
-------------------------------------------------------------------------------
```

So if we want to test the hypothesis that vitamin C levels are the same in white females and black females, we need to test subpopulation 4 versus subpopulation 5.

```
. svylc [vitaminc]4 - [vitaminc]5
 ( 1)  [vitaminc]4 - [vitaminc]5 = 0.0
-------------------------------------------------------------------------------
    Mean |   Estimate    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+---------------------------------------------------------------------
     (1) |   .2288941    .0337949      6.773   0.000      .1599688    .2978193
-------------------------------------------------------------------------------
```

You can see the numbering scheme by running `svylc` with the `show` option.

```
. svylc, show
-------------------------------------------------------------------------------
    Mean |      Coef.   Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+---------------------------------------------------------------------
vitaminc |
       1 |   .9475117   .0168982     56.072   0.000      .9130475    .9819758
       2 |   .7382045   .0477521     15.459   0.000      .6408135    .8355955
       3 |   1.021363   .0521427     19.588   0.000       .915017    1.127708
       4 |   1.151125   .0168117     68.472   0.000      1.116838    1.185413
       5 |   .9222313   .0348224     26.484   0.000      .8512105     .993252
       6 |     1.0804   .0412742     26.176   0.000      .9962202    1.164579
-------------------------------------------------------------------------------
```

### The use of svylc after svyratio

Using `svylc` after `svyratio` is a little more complicated. But, again, the `show` option on `svylc` will guide you.

```
. svyratio y1/x1 y2/x2
Survey ratio estimation
pweight:  finalwgt                                Number of obs    =     10351
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.172e+08
------------------------------------------------------------------------------
      Ratio       |   Estimate    Std. Err.   [95% Conf. Interval]         Deff
------------------+-----------------------------------------------------------
      y1/x1       |   .9918905    .0102386    .9710087    1.012772    1.647415
      y2/x2       |   .9962729    .0083088    .9793269    1.013219      1.0771
------------------------------------------------------------------------------

. svylc, show
------------------------------------------------------------------------------
   Ratio |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
y1       |
      x1 |   .9918905   .0102386     96.878   0.000       .9710087    1.012772
---------+--------------------------------------------------------------------
y2       |
      x2 |   .9962729   .0083088    119.905   0.000       .9793269    1.013219
------------------------------------------------------------------------------

. svylc [y1]x1 - [y2]x2
 ( 1)  [y1]x1 - [y2]x2 = 0.0
------------------------------------------------------------------------------
   Ratio |   Estimate    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |  -.0043824    .0125921     -0.348   0.730     -.0300641    .0212993
------------------------------------------------------------------------------
```

The following examples illustrate the syntax when there are `by()` subpopulations.

```
. svyratio y1/x1, by(race)
Survey ratio estimation
pweight:  finalwgt                                Number of obs    =     10351
Strata:   strata                                  Number of strata =        31
PSU:      psu                                     Number of PSUs   =        62
                                                  Population size  = 1.172e+08
------------------------------------------------------------------------------
Ratio     Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]         Deff
------------------+-----------------------------------------------------------
y1/x1             |
            White |    .995116    .0116867    .9712807    1.018951    1.879759
            Black |   .9525558    .0381059    .8748384    1.030273    2.242268
            Other |   1.026876    .0447707    .9355659    1.118187    .8308877
------------------------------------------------------------------------------

. svylc, show
------------------------------------------------------------------------------
   Ratio |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
1        |
   White |    .995116   .0116867     85.149   0.000       .9712807    1.018951
   Black |   .9525558   .0381059     24.998   0.000       .8748384    1.030273
   Other |   1.026876   .0447707     22.936   0.000       .9355659    1.118187
------------------------------------------------------------------------------

. svylc [1]White - [1]Black
 ( 1)  [1]White - [1]Black = 0.0
------------------------------------------------------------------------------
   Ratio |   Estimate    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |   .0425602    .0439945      0.967   0.341     -.0471671    .1322875
------------------------------------------------------------------------------

. svyratio y1/x1, by(sex race)
Survey ratio estimation
```

```
pweight:  finalwgt                              Number of obs    =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08
------------------------------------------------------------------------------
Ratio     Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
------------------+-----------------------------------------------------------
y1/x1             |
    Male    White |   1.000215    .0150805    .9694585    1.030972    1.460442
    Male    Black |   .9726418    .0486307    .8734589    1.071825    1.426839
    Male    Other |   1.000358    .0732775     .850907    1.149808    1.266913
  Female    White |   .9904237    .0169396    .9558752    1.024972    2.109029
  Female    Black |   .9362548    .0409748    .8526861    1.019823    1.619815
  Female    Other |   1.056553     .082305    .8886906    1.224415    1.228803
------------------------------------------------------------------------------

. svylc, show
------------------------------------------------------------------------------
   Ratio |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
1        |
       1 |   1.000215   .0150805     66.325   0.000       .9694585    1.030972
       2 |   .9726418   .0486307     20.001   0.000       .8734589    1.071825
       3 |   1.000358   .0732775     13.652   0.000        .850907    1.149808
       4 |   .9904237   .0169396     58.468   0.000       .9558752    1.024972
       5 |   .9362548   .0409748     22.850   0.000       .8526861    1.019823
       6 |   1.056553    .082305     12.837   0.000       .8886906    1.224415
------------------------------------------------------------------------------

. svylc [1]1 - [1]4
 ( 1)  [1]1 - [1]4 = 0.0
------------------------------------------------------------------------------
   Ratio |   Estimate    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |   .0097916    .0221119      0.443   0.661     -.0353058     .054889
------------------------------------------------------------------------------
```

## Testing hypotheses with svytest

Joint hypothesis tests can be performed after `svy` estimation commands using the `svytest` command. Here we estimate a linear regression of `loglead` (log of blood lead).

```
. svyreg loglead age female black orace region2-region4
Survey linear regression
pweight:  leadwt                                Number of obs    =      4948
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.129e+08
                                                F(   7,     25)  =    186.18
                                                Prob > F         =    0.0000
                                                R-squared        =    0.2321
------------------------------------------------------------------------------
 loglead |      Coef.    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   .0027842    .0004318      6.448   0.000      .0019036    .0036649
  female |  -.3645445    .0110947    -32.857   0.000     -.3871724   -.3419167
   black |   .1783735    .0321995      5.540   0.000      .1127022    .2440447
   orace |  -.0473781    .0383677     -1.235   0.226     -.1256295    .0308733
 region2 |  -.0242082    .0384767     -0.629   0.534     -.1026819    .0542655
 region3 |  -.1646067    .0549628     -2.995   0.005      -.276704   -.0525094
 region4 |  -.0361289    .0377054     -0.958   0.345     -.1130296    .0407717
   _cons |   2.696084    .0236895    113.809   0.000      2.647769    2.744399
------------------------------------------------------------------------------
```

We can use `svytest` to test the joint significance of the region dummies: `region1` is the Northeast, `region2` is the Midwest, `region3` is the South, and `region4` is the West. We test the hypothesis that `region2` $= 0$, `region3` $= 0$, and `region4` $= 0$.

```
. svytest region2 region3 region4
Adjusted Wald test
 ( 1)  region2 = 0.0
 ( 2)  region3 = 0.0
 ( 3)  region4 = 0.0
       F(  3,    29) =    2.97
            Prob > F =    0.0480
```

The `noadjust` option on `svytest` produces an unadjusted Wald test.

```
. svytest region2 region3 region4, noadjust
Unadjusted Wald test
 ( 1)  region2 = 0.0
 ( 2)  region3 = 0.0
 ( 3)  region4 = 0.0
       F(  3,    31) =    3.18
            Prob > F =    0.0377
```

Bonferroni adjusted $p$-values can also be computed.

```
. svytest region2 region3 region4, bonferroni
Bonferroni adjustment for 3 comparisons
------------------------------------------------------------------------------
 loglead |      Coef.    Std. Err.       t      Adj. P
---------+--------------------------------------------------------------------
 region2 |  -.0242082    .0384767     -0.629    1.0000
 region3 |  -.1646067    .0549628     -2.995    0.0161 *
 region4 |  -.0361289    .0377054     -0.958    1.0000
------------------------------------------------------------------------------
```

The smallest adjusted $p$-value is a $p$-value for a test of the same joint hypothesis that we tested before; namely, `region2`$= 0$, `region3`$= 0$, and `region4`$= 0$. See the Korn and Graubard (1990) for a discussion of these three different procedures for conducting joint hypothesis tests.

The examples given above show how to use `svytest` to test hypotheses for which the coefficients are jointly hypothesized to be zero. We will now illustrate the use of `svytest` to test general hypotheses. Let us run the same regression model, only this time we will include the other region dummy `region1` and omit the constant term.

```
. svyreg loglead age female black orace region1-region4, nocons
Survey linear regression
pweight:  leadwt                                Number of obs    =      4948
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.129e+08
                                                F(   8,     24)  =   5148.74
                                                Prob > F         =    0.0000
                                                R-squared        =    0.9806
------------------------------------------------------------------------------
 loglead |      Coef.    Std. Err.       t     P>|t|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   .0027842    .0004318      6.448   0.000      .0019036    .0036649
  female |  -.3645445    .0110947    -32.857   0.000     -.3871724   -.3419167
   black |   .1783735    .0321995      5.540   0.000      .1127022    .2440447
   orace |  -.0473781    .0383677     -1.235   0.226     -.1256295    .0308733
 region1 |   2.696084    .0236895    113.809   0.000      2.647769    2.744399
 region2 |   2.671876    .0420415     63.553   0.000      2.586132     2.75762
 region3 |   2.531477    .0601017     42.120   0.000      2.408899    2.654055
 region4 |   2.659955    .0405778     65.552   0.000      2.577196    2.742714
------------------------------------------------------------------------------
```

In order to test the joint hypothesis that `region1` $=$ `region2` $=$ `region3` $=$ `region4`, we must enter the equations of the hypothesis one at a time and use the `accumulate` option.

```
. svytest region1 = region2
Adjusted Wald test
 ( 1)  region1 - region2 = 0.0
       F(  1,    31) =    0.40
            Prob > F =    0.5338
. svytest region2 = region3, accum
Adjusted Wald test
 ( 1)  region1 - region2 = 0.0
 ( 2)  region2 - region3 = 0.0
       F(  2,    30) =    4.41
            Prob > F =    0.0209
. svytest region3 = region4, accum
Adjusted Wald test
 ( 1)  region1 - region2 = 0.0
 ( 2)  region2 - region3 = 0.0
 ( 3)  region3 - region4 = 0.0
```

```
       F(  3,    29) =    2.97
            Prob > F =    0.0480
```

As expected, we get the same answer as before. Note that the Bonferroni adjustment procedure is not available for use with the above syntax. The `svytest` command can only use the Bonferroni procedure to test whether a group of coefficients are simultaneously equal to zero.

The `svytest` command can also be used after `svymean`, `svytotal`, and `svyratio`. `svytest` and `svylc` use the same syntax to reference the estimates. The only difference is that you use `svytest` with a full equation (i.e., you include an equal sign and a right-hand side for the equation), or with a list of estimates that you wish to test simultaneously equal to zero. Here is an example of the former.

```
. svymean bpsystol bpdiast, by(rural)

Survey mean estimation

pweight:  finalwgt                              Number of obs    =     10351
Strata:   strata                                Number of strata =        31
PSU:      psu                                   Number of PSUs   =        62
                                                Population size  = 1.172e+08

------------------------------------------------------------------------------
Mean   Subpop. |   Estimate    Std. Err.   [95% Conf. Interval]        Deff
---------------+--------------------------------------------------------------
bpsystol       |
      rural==0 |   126.6065    .5503138    125.4841    127.7289    4.655704
      rural==1 |   127.6753    1.261624    125.1022    130.2484    11.52492
---------------+--------------------------------------------------------------
bpdiast        |
      rural==0 |   80.90864    .4990564    79.89081    81.92648    10.94774
      rural==1 |   81.25081    .9476732    79.31802     83.1836    17.36593
------------------------------------------------------------------------------

. svytest [bpsystol]0 = [bpsystol]1

Adjusted Wald test

 ( 1)  [bpsystol]0 - [bpsystol]1 = 0.0

       F(  1,    31) =    0.71
            Prob > F =    0.4064

. svytest [bpdiast]0 = [bpdiast]1, accumulate

Adjusted Wald test

 ( 1)  [bpsystol]0 - [bpsystol]1 = 0.0
 ( 2)  [bpdiast]0 - [bpdiast]1 = 0.0

       F(  2,    30) =    0.65
            Prob > F =    0.5300
```

## Methods and Formulas

`svylc` estimates $\eta = C\theta$, where $\theta$ is a $q \times 1$ vector of parameters (e.g., population means or population regression coefficients), and $C$ is any $1 \times q$ vector of constants. The estimate of $\eta$ is $\widehat{\eta} = C\widehat{\theta}$, and the estimate of its variance is

$$\widehat{V}(\widehat{\eta}) = C\widehat{V}(\widehat{\theta})C'$$

Similarly, the simple-random-sampling variance estimator used in the computation of deff and deft is $\widehat{V}_{\rm srs}(\widetilde{\eta}_{\rm srs}) = C\widehat{V}_{\rm srs}(\widetilde{\theta}_{\rm srs})C'$. And the variance estimator used in the computation of meff and meft is $\widehat{V}_{\rm msp}(\widehat{\eta}_{\rm msp}) = C\widehat{V}_{\rm msp}(\widehat{\theta}_{\rm msp})C'$. See the *Methods and formulas* section of *svy2* for details on the computation of deff, deft, meff, and meft.

`svytest` tests the null hypothesis $\mathrm{H}_0$: $C\theta = c$, where $\theta$ is a $q \times 1$ vector of parameters, $C$ is any $k \times q$ matrix of constants, and $c$ is a $k \times 1$ vector of constants. The Wald test statistic is

$$W = (C\theta - c)'(C\widehat{V}(\widehat{\theta})C')^{-1}(C\theta - c)$$

By default, `svytest` uses

$$\frac{d-k+1}{kd}\,W \sim F(k, d-k+1)$$

to compute the $p$-value. Here $d$ = total number of sampled PSUs minus the total number of strata, and $F(k, d-k+1)$ is an $F$ distribution with $k$ numerator degrees of freedom and $d-k+1$ denominator degrees of freedom. If the `noadjust` option is

specified, the $p$-value is computed using $W/d \sim F(k,d)$. Note that the `noadjust` option gives the same results as the `test` command.

When the `bonferroni` option is specified, `svytest` displays adjusted $p$-values for each of the coefficients corresponding to the specified variables. Adjusted $p$-values are computed as $p_{\rm adj} = \min(kp, 1)$, where $k$ is the number of variables specified, and $p$ is the unadjusted $p$-value (i.e., the $p$-value shown in the output of the estimation command) obtained from the statistic $t = \widehat{b}/[\widehat{V}(\widehat{b})]^{1/2}$ which is assumed to have a $t$ distribution with $d$ degrees of freedom.

See Korn and Graubard (1990) for a detailed description of the Bonferroni adjustment technique and a discussion of the relative merits of it and of the adjusted and unadjusted Wald tests.

## Saved Results

`svylc` saves the following results in the `S_` macros:

| | |
|---|---|
| `S_1` | point estimate of linear combination |
| `S_2` | standard error (square root of design-based variance estimate) |
| `S_3` | number of strata |
| `S_4` | number of sampled PSUs |
| `S_5` | deff |
| `S_6` | deft |
| `S_7` | meft |

`svytest` saves the following results in the `S_` macros:

| | |
|---|---|
| `S_1` | $F$ numerator degrees of freedom (i.e., dimension of hypothesis test) |
| `S_2` | $F$ denominator degrees of freedom (or $t$ statistic degrees of freedom for `bonferroni`) |
| `S_3` | $F$ statistic (or maximal $t$ statistic for `bonferroni`) |

## References

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

## zz6 Cumulative index for STB-25 – STB-30

### [an] Announcements



### [crc] CRC-Provided Support Materials



### [dm] Data Management

## STB categories and insert codes

Inserts in the STB are presently categorized as follows:

*General Categories:*

| | | | |
|---|---|---|---|
| *an* | announcements | *ip* | instruction on programming |
| *cc* | communications & letters | *os* | operating system, hardware, & interprogram communication |
| *dm* | data management | *qs* | questions and suggestions |
| *dt* | data sets | *tt* | teaching |
| *gr* | graphics | *zz* | not elsewhere classified |
| *in* | instruction | | |

*Statistical Categories:*

| | | | |
|---|---|---|---|
| *sbe* | biostatistics & epidemiology | *ssa* | survival analysis |
| *sed* | exploratory data analysis | *ssi* | simulation & random numbers |
| *sg* | general statistics | *sss* | social science & psychometrics |
| *smv* | multivariate analysis | *sts* | time-series, econometrics |
| *snp* | nonparametric methods | *svy* | survey sample |
| *sqc* | quality control | *sxd* | experimental design |
| *sqv* | analysis of qualitative variables | *szz* | not elsewhere classified |
| *srd* | robust methods & statistical diagnostics | | |

In addition, we have granted one other prefix, *crc*, to the manufacturers of Stata for their exclusive use.

## International Stata Distributors

International Stata users may also order subscriptions to the *Stata Technical Bulletin* from our International Stata Distributors.

| | |
|---|---|
| Company: | Applied Statistics & Systems Consultants |
| Address: | 14/26 Yizreel St., P.O. Box 1169<br>Nazerath-Ellit 17100, Israel |
| Phone: | +972 6554254 |
| Fax: | +972 6554254 |
| Email: | sasconsl@actcom.co.il |
| Countries served: | Israel |

| | |
|---|---|
| Company: | Dittrich & Partner Consulting |
| Address: | Prinzenstrasse 2<br>D-42697 Solingen<br>Germany |
| Phone: | +49 212-3390 99 |
| Fax: | +49 212-3390 90 |
| Email: | available soon |
| Countries served: | Austria, Germany, Italy |

| | |
|---|---|
| Company: | Metrika Consulting |
| Address: | Roslagsgatan 15<br>113 55 Stockholm<br>Sweden |
| Phone: | +46-708-163128 |
| Fax: | +46-8-6122383 |
| Email: | available soon |
| Countries served: | Baltic States, Denmark, Finland, Iceland, Norway, Sweden |

| | |
|---|---|
| Company: | Ritme Informatique |
| Address: | 34 boulevard Haussmann<br>75009 Paris<br>France |
| Phone: | +33 1 42 46 00 42 |
| Fax: | +33 1 42 46 00 33 |
| Email: | ritme.inf@applelink.apple.com |
| Countries served: | Belgium, France, Luxembourg, Switzerland |

| | |
|---|---|
| Company: | Smit Consult |
| Address: | Scheidingstraat 1<br>Postbox 220<br>5150 AE Drunen<br>Netherlands |
| Phone: | +31 416-378 125 |
| Fax: | +31 416-378 385 |
| Email: | j.a.c.m.smit@smitcon.nl |
| Countries served: | Netherlands |

| | |
|---|---|
| Company: | Timberlake Consultants |
| Address: | 47 Hartfield Crescent<br>West Wickham<br>Kent BR4 9DW, U.K. |
| Phone: | +44 181 462 0495 |
| Fax: | +44 181 462 0493 |
| Email: | 100412.2603@compuserve.com |
| Countries served: | Ireland, U.K. |

| | |
|---|---|
| Company: | Timberlake Consultants<br>Satellite Office |
| Address: | Praceta do Comércio,<br>N°13–9° Dto. Quinta Grande<br>2720 Alfragide Portugal |
| Phone: | +351 (01) 4719337 |
| Telemóvel: | 0931 62 7255 |
| Email: | 100412.2603@compuserve.com |
| Countries served: | Portugal |