

Analyzing Complex Survey Data: Some key issues to be aware of

Richard Williams, University of Notre Dame, <https://academicweb.nd.edu/~rwilliam/>

Last revised January 24, 2015

Rather than repeat material that is already well-covered elsewhere, read the Stata Manual's *Introduction to Survey Commands* first. It explains how and why the survey design and the survey data collection need to be taken into account when doing your analysis. Pay particular attention to the introduction and skim the rest. There are just a few additional points I want to illustrate here.

Most of the analysis we have done so far assumes that cases were selected via simple random sampling – the equivalent of drawing names out of a hat. In reality, sampling schemes are often much more complicated than that. Survey data (I am quoting a lot from the Stata Manual here) are characterized by

- **sampling weights**, aka probability weights or pweights: “In sample surveys, observations are selected through a random process, but different observations may have different probabilities of selection,” e.g. blacks may be oversampled
- **cluster sampling**: “Individuals are not sampled independently in most survey designs. Collections of individuals (for example, counties, city blocks, or households) are typically sampled as a group known as a cluster.”
- **stratification**: “In surveys, different groups of clusters are often sampled separately. These groups are called strata. For example, the 254 counties of a state might be divided into two strata, say, urban counties and rural counties. Then 10 counties might be sampled from the urban stratum, and 15 from the rural stratum”

Failure to take the sampling scheme into account can lead to inaccurate point estimates and/or flawed estimates of the standard errors.

The *svyset* command and the *svy:* prefix. Your data need to be *svyset* first. The *svyset* command tells Stata everything it needs to know about the data set's sampling weights, clustering, and stratification. You only need to *svyset* your data once. Hopefully, the provider of your data has told you what you need for the *svyset* command or has even *svyset* the data for you. If not, you are going to have to do some reading or get some help to figure out how to do it yourself.

If the data are already *svyset*, then typing *svyset* by itself will show what the settings are.

```
. webuse nhanes2f, clear
. svyset

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

```
. sum finalwgt stratid psuid
```

Variable	Obs	Mean	Std. Dev.	Min	Max
finalwgt	10337	11320.85	7304.457	2000	79634
stratid	10337	16.65986	9.499389	1	32
psuid	10337	1.482151	.4997055	1	2

The pweight variable is finalwgt. The summary statistics show you that each person in the sample represents anywhere from 2,000 to 79,634 people in the population. Put another way, if the pweight for a person is 10,000, that means that the respondent had once chance in 10,000 of being selected for the sample.

Once the data are `svyset`, you need to remember to use the `svy:` prefix with your commands. For example, instead of typing

```
. reg weight height age i.female i.black
```

Source	SS	df	MS	Number of obs =	10337
Model	620082.606	4	155020.652	F(4, 10332) =	881.52
Residual	1816944.64	10332	175.856044	Prob > F =	0.0000
Total	2437027.25	10336	235.7805	R-squared =	0.2544
				Adj R-squared =	0.2542
				Root MSE =	13.261

weight	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
height	.7485279	.01966	38.07	0.000	.7099905 .7870652
age	.1237255	.0078948	15.67	0.000	.1082501 .1392009
1.female	-1.540187	.3721392	-4.14	0.000	-2.269652 -.8107221
1.black	3.679295	.4256284	8.64	0.000	2.844981 4.513609
_cons	-59.05337	3.563342	-16.57	0.000	-66.03822 -52.06853

You should type

```
. svy: reg weight height age i.female i.black
(running regress on estimation sample)
```

Survey: Linear regression

Number of strata =	31	Number of obs =	10337
Number of PSUs =	62	Population size =	117023659
		Design df =	31
		F(4, 28) =	880.32
		Prob > F =	0.0000
		R-squared =	0.2887

weight	Linearized Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
height	.7409099	.0274549	26.99	0.000	.6849151 .7969046
age	.1520647	.0117481	12.94	0.000	.1281044 .1760251
1.female	-2.924562	.6054516	-4.83	0.000	-4.159389 -1.689736
1.black	4.033541	.7436242	5.42	0.000	2.516909 5.550172
_cons	-58.19333	4.909523	-11.85	0.000	-68.20637 -48.18029

Notice some differences in the output. You are told that this sample represents a population of 117 million people. You don't get an ANOVA table anymore. The coefficients are somewhat different, reflecting the fact that cases are not being weighted equally anymore. The changes in coefficients and also the fact that clustering and stratification are taken into account affect the significance tests and confidence intervals.

However, there are some important differences between the analysis of survey and non-survey data that you need to be aware of.

Linear regression: Some of the statistics and tests you are used to using are inappropriate. There are numerous things you are used to doing with linear regression that will not work with svyset data. This is at least partly because, with survey data, assumptions that cases are independent of each other are violated. In other cases, it may be because Stata hasn't figured out how to adapt the test or procedure to svyset data.

Example 1: Wald tests work but incremental F tests do not:

```
. svy: reg weight height age
(running regress on estimation sample)
```

Survey: Linear regression

Number of strata	=	31	Number of obs	=	10337
Number of PSUs	=	62	Population size	=	117023659
			Design df	=	31
			F(2, 30)	=	1803.49
			Prob > F	=	0.0000
			R-squared	=	0.2785

		Linearized				
weight	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
height	.8486177	.0148894	56.99	0.000	.8182505	.8789849
age	.1593711	.0118398	13.46	0.000	.1352236	.1835186
_cons	-77.78301	2.514129	-30.94	0.000	-82.91061	-72.65541

```
. est store m1
```

```
. svy: reg weight height age i.female i.black
(running regress on estimation sample)
```

Survey: Linear regression

```
Number of strata =      31      Number of obs =      10337
Number of PSUs  =      62      Population size = 117023659
                                   Design df      =      31
                                   F( 4,      28)    =      880.32
                                   Prob > F         =      0.0000
                                   R-squared         =      0.2887
```

		Linearized				
weight	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
height	.7409099	.0274549	26.99	0.000	.6849151	.7969046
age	.1520647	.0117481	12.94	0.000	.1281044	.1760251
1.female	-2.924562	.6054516	-4.83	0.000	-4.159389	-1.689736
1.black	4.033541	.7436242	5.42	0.000	2.516909	5.550172
_cons	-58.19333	4.909523	-11.85	0.000	-68.20637	-48.18029

```
. est store m2
. test 1.female 1.black
```

Adjusted Wald test

```
( 1) 1.female = 0
( 2) 1.black = 0
```

```
F( 2, 30) = 20.86
Prob > F = 0.0000
```

```
. ftest m1 m2
Linearized vce not allowed
r(198);
```

So, in general you should use Wald tests for hypothesis testing. However, you can also use the `nestreg` command (without factor variables) since `nestreg` basically just does Wald tests on each model.

```
. nestreg, quietly: svy: reg weight (height age) (female black)
```

```
Block 1: height age
Block 2: female black
```

		Block		Design		Change
Block	F	df	df	Pr > F	R2	in R2
1	1803.49	2	31	0.0000	0.2785	
2	20.86	2	31	0.0000	0.2887	0.0102

Similar issues come up with techniques like logistic regression that use maximum likelihood estimation. See the appendix.

Example 2: Numerous post-estimation diagnostic commands do not work

```
. quietly svy: reg weight height age female black
. dfbeta
option dfbeta() not allowed after svy estimation
r(198);

. estat hetttest
invalid subcommand hetttest
r(321);

. estat imtest
invalid subcommand imtest
r(321);

. rvfplot
option resid not allowed
r(198);

. estat lvr2plot
invalid subcommand lvr2plot
r(321);
```

Given that these are diagnostic tests, you may want to do exploratory analyses that ignore the svysetting of the data.

Subsample analyses. Another thing to be careful of is subsample analyses, e.g. analyzing men only. With non-svy data, you usually just create an extract first which has only your desired cases; or you include an `if` qualifier with your command, e.g. something like

```
reg y x1 x2 x3 if female==0
```

With svy data, however, that kind of approach can, under certain conditions, seriously bias your results, i.e. the standard error calculations can be wrong if all the data are not available. Instead, you should use the `subpop` option to specify your sample. As UCLA explains in its Stata FAQs, “When the subpopulation option(s) is used, only the cases defined by the subpopulation are used in the calculation of the estimate, but all cases are used in the calculation of the standard errors.” UCLA further adds that “Using `if` in the `subpop` option does not remove cases from the analysis. The cases excluded from the subpopulation by the `if` are still used in the calculation of the standard errors, as they should be.” So, do something like this:

```
. svy, subpop(if female==0): reg weight height age i.black
(running regress on estimation sample)
```

Survey: Linear regression

Number of strata	=	31	Number of obs	=	10337
Number of PSUs	=	62	Population size	=	117023659
			Subpop. no. of obs	=	5428
			Subpop. size	=	60901624
			Design df	=	31
			F(3, 29)	=	243.60
			Prob > F	=	0.0000
			R-squared	=	0.1146

		Linearized				
	weight	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
	height	.6075799	.0360875	16.84	0.000	.5339789 .6811808
	age	.1899597	.014411	13.18	0.000	.1605684 .2193511
	1.black	7.695681	.9835048	7.82	0.000	5.68981 9.701552
	_cons	-41.51002	5.815098	-7.14	0.000	-53.36999 -29.65005

Some commands do not work with svy: Some commands, like `summarize`, do not work with the `svy:` prefix. Sometimes you can find an alternative command that does, e.g. `svy:mean`.

```
. webuse nhanes2f
```

```
. svy: summarize diabetes black weight height
```

`summarize` is not supported by `svy` with `vce(linearized)`; see help `svy estimation` for a list of Stata estimation commands that are supported by `svy`
`r(322);`

```
. svy: mean diabetes black weight height
```

(running mean on estimation sample)

Survey: Mean estimation

Number of strata	=	31	Number of obs	=	10335
Number of PSUs	=	62	Population size	=	116997257
			Design df	=	31

		Linearized		
		Mean	Std. Err.	[95% Conf. Interval]
	diabetes	.0342853	.0018197	.0305739 .0379966
	black	.0956367	.0127804	.0695709 .1217026
	weight	71.91131	.1670327	71.57065 72.25198
	height	168.4647	.1471856	168.1645 168.7649

Note that standard deviations are not reported by the `mean` command. `estat sd` reports subpopulation standard deviations based on the estimation results from `mean` and `svy: mean`.

```
. estat sd
```

	Mean	Std. Dev.
diabetes	.0342853	.1819697
black	.0956367	.2941067
weight	71.91131	15.43409
height	168.4647	9.702569

For information on other svy postestimation commands, type `help svy_estat`.

Another example: `svy:` won't work in combination with the `sw:` prefix, because stepwise methods are not considered appropriate with svy data.

```
. sw, pe(.05): svy: logit diabetes black weight height
svy is not supported by stepwise
r(199);
```

Also, user-written post-estimation commands may or may not work after using the `svy:` prefix (and if they do work you should try to check to see if they work correctly. For example, even Stata's old `adjust` command does not work correctly with svy data.)

Conclusion. svy data are not that hard to work with. But, you do have to understand some of the important differences that do exist. For more, see Stata's SVY Manual. Other good references (as of January 22, 2015) include

<http://www.ats.ucla.edu/stat/stata/faq/> (see the lower part of the page)

<http://www.stata.com/support/faqs/stat/#survey>

Appendix: Maximum Likelihood – and Statistics based on it – are inappropriate (optional)

With survey data, the ML assumptions that cases are independent of each other are violated. As a result, with commands like `logit` you can't get several statistics you are used to, e.g. Model LR Chi², BIC, AIC. You get F statistics and T statistics instead of chi-squares and zs. You have to do Wald tests instead of LR Chi² model contrasts. The following will illustrate this.

```
. webuse nhanes2f, clear
. * Constrained model
. svy: logit diabetes female black
(running logit on estimation sample)
```

Survey: Logistic regression

Number of strata	=	31	Number of obs	=	10335
Number of PSUs	=	62	Population size	=	116997257
			Design df	=	31
			F(2, 30)	=	15.02
			Prob > F	=	0.0000

		Linearized				[95% Conf. Interval]	
diabetes	Coef.	Std. Err.	t	P> t			
female	.2937317	.1198141	2.45	0.020	.0493693	.5380941	
black	.644294	.1157506	5.57	0.000	.408219	.8803689	
_cons	-3.582378	.100637	-35.60	0.000	-3.787628	-3.377127	

Already some key differences in the output are obvious. We got an F test instead of a Model LR Chi² statistic. For each coefficient we got a T statistic instead of a Z statistic. No Log Likelihood for the model was reported. The Population size line told us how many people are in the population that this sample represents (about 117 million). Now let's see what happens when we try to contrast nested models.

```
. est store ml
. svy: logit diabetes female black weight height
(running logit on estimation sample)
```

Survey: Logistic regression

Number of strata	=	31	Number of obs	=	10335
Number of PSUs	=	62	Population size	=	116997257
			Design df	=	31
			F(4, 28)	=	26.97
			Prob > F	=	0.0000

		Linearized				[95% Conf. Interval]	
diabetes	Coef.	Std. Err.	t	P> t			
female	-.1591092	.1481466	-1.07	0.291	-.4612563	.1430379	
black	.5026699	.1270514	3.96	0.000	.2435468	.761793	
weight	.0290168	.0033496	8.66	0.000	.0221852	.0358484	
height	-.0574331	.0081596	-7.04	0.000	-.0740747	-.0407915	
_cons	4.149243	1.295284	3.20	0.003	1.507495	6.790992	


```
. est store m2
. lrtest m1 m2, all
lrtest is not appropriate with survey estimation results
r(322);
```

The `lrtest` command does not work because the assumptions behind it are violated with complicated survey designs. (It won't even work if you include the `force` option.) Hence, we don't get a likelihood ratio chi-square contrast. We also don't get BIC or AIC statistics because the ML assumptions behind those statistics are also violated. Instead, we have to use `test` commands.

```
. * Use Wald test instead
. test weight height
```

Adjusted Wald test

```
( 1)  [diabetes]weight = 0
( 2)  [diabetes]height = 0

      F(  2,    30) =    40.50
      Prob > F =    0.0000
```