

Stata Survey Statistics

[stata_survey_stats_commands]

See UCLA/ATS, Stat 130 Class Notes, "Survey Sampling" for an introduction with practice data; and see Levy & Lemeshow (1999), which is the principal source of the following notes. Commands changed in Stata-8.

Stata survey-statistics format commands

```
. svydescribe
. svyset, clear

. svyset [pw=weightvar], stata(stratvar) psu(psuvar) [last term is name of variable]
. svydes
```

-
- See Stata instructions for the command *sampsi*.
 - svy commands with "if" give the wrong CI; use *subpop* or *by* instead of "if" (the same applies to bootstrapping)
 - do not use LR-test with *svystats*, or otherwise with *pweight*. Instead use either *svytest* (if # clusters < ~100) or *test* (which in this instance gives a Wald test)
 - *Finite population correction (fpc)*: computes adjusted N for se estimates ($N - n/N$); used only in simple random sampling without replacement. That is, it accounts for the reduction in variance that occurs when sampling **without** replacement from a finite population, as compared to sampling **with** replacement. Use *fpc()* option for cases of simple random sampling or stratified random sampling without replacement of psu's within each stratum with no subsampling within psu's. **Including *fpc()* reduces the variance estimate, but minimally if N-psu's is large relative to sampled-n-psu's.** To use, set *fpc()* to *Nh*, the var representing the total number of psu's per stratum in the population (e.g., *hid* in datafile comprised of individual household members). **Caution: you must know the total population-parameter of the pertinent var (e.g., total pop-N in each stratum or cluster) in order to use *fpc*, so thus we will rarely if ever use *fpc***
 - Recall that that unequal nonresponse rates across strata &/or clusters must be adjusted for
-

How to set survey-stats in Stata for various samples

simple random sample

random sample of one hospital from a population (*hospro*) of 25 hospitals; enumeration unit (i.e. population) is #births ($N=773$) in the sampled hospital during the previous year; in sum, what Stata needs for a simple random sample is (1) weighting var & (2) total enumeration units of population (N) from which sample is drawn

```
. svyset [pw=weight1], fpc(birth)          momsag.dta; sampling weight=N/n  
                                          birth=total N
```

[psu <obs> is *hospro* (total #hospitals), which=25; in this case psu is the enumeration unit, but this is not so in all cases]

```
. svymean momsag          compare with: su momsag  
. svytotal momsag        compare with: sumsum momsag  
                          egen rsum=sum(momsag)
```

random sample of 40 workers from total enumeration units (i.e. population, N : *popsize*) of 1200 workers; includes (1) weighting var & (2) total enumeration units of population (N) from which sample is drawn

```
. svyset [pw=wt1], fpc(popsize)          workers.dta; sampling weight=N/n  
                                          popsize=total N  
. svymean fvc, by(exposure)             simple random sample with subdomains
```

summary: what is required for simple random sampling is just two parameters, the total #enumeration units (N) from which the sample is drawn & the weighting var (inverse of sampling fraction n/N ; i.e. weighting var= N/n). In these cases, $\text{birth}=N$ & $\text{popsize}=N$ —i.e. using *fpc* to denote sample size obtains the parameter, N , used in computing the *finite population correction* $(N - n)/N$, which, in turn, is used to compute se estimates (Levy & Lemeshow 52-55, 75-76)

systematic random sampling

- every k -obs, when $k=N/n$ is an integer (to yield unbiased estimates); does not require prior knowledge of population size (N)
- *repeated systematic sampling*: also yields unbiased estimates & does not require prior knowledge of population size (N) (Levy & Lemeshow 101-9)
 - example (*workloss.dta*): take a systematic random sample of 18 workers from the list of 162 workers (*id*) to estimate the mean of workdays lost per worker due to acute illness (*dayslost*). Thus, sample one of every nine workers, which, however, we can do by taking six systematic samples containing three workers each (sampling interval $162/3=54$, so we take six systematic samples of 1 in 54 workers). To do this we first choose 6 random numbers between 1 and 54, and then we choose systematic samples of 1 in 54 beginning each with a random number.
 - *workloss.dta*: *cluster* is a number from 1 to 54 identifying the position of the sampling frame of the first element in the particular sample cluster; *xi* is the total of x , the workloss days for the three workers in each cluster; *wt1* is the ratio of total clusters, M , to sample clusters,

two-stage clustered sampling

```
. svyset [pw=w], psu(hospno)    il10pt2.dta
                               hospno=total # of clusters in the pop
. svy:total dxdead            dxdead=admitted or discharged dead
. svy:ratio dxdead lifethrt   lifethrt=admission of patient with
                               life-threatening illness
```

in sum: in this configuration, Stata cannot use both psu & fpc, so the latter is not used (see example for il10pt2.dta in Levy & Lemeshow)

probability proportional to size sampling

```
. svyset [pw=wstar], psu(drawing)    hospslet.dta
                                       drawing=# clusters
. svy:total lifethrt dxdead
. svy:ratio dxdead lifethrt
```

In sum: in this configuration, Stata cannot use both psu & fpc, so the latter is not used

Stata survey-set & describe procedures

```
. svyset [pw=wtvar], strata(res_ses) psu(hid)    example from World Bank Teguc &
                                                Salv: individual-level data
                                                hid: for individuals clustered by
                                                households
. svydes
. svyset, clear                                to delete svyset
```

In sum: use fpc only if you know the total pop-parameter (e.g., total pop-N for each stratum or cluster)

Stata survey statistics

```
. svy:tab xcat1 xcat2: two-way table
. svy:tab xcat1 xcat2, col
. svy:tab xcat1 xcat2, row per
. svy:test x1 x2: hypothesis test              or test if cluster size>100
. svy:lc: estimate linear combinations such as differences of means and of regression
coefficient
. svy:mean
. svy:prop
. svy:ratio
. svy:total

. svy:reg
. svy:logit
. svy:olog    eform
. svy:mlog    rrr
. svy:probt
. svy:oprobt
```

```
. svy: intrg
. svy: pois
```

Examples

```
. svydes
. svymean dwelf [note: var must not be weighted separately/individually,
but rather only via the svy-weighting procedure]
. svy: mean dwelf, subpop(lte730) var: lte730=1
. svy: mean dwelf, subpop(lte730) in f/50 (complete) (95) [or: (available) (99)]
. svy: prop id, subpop(lte730) [note: first sort id]
. svyset, clear
```

If necessary to collapse psu's & re-set survey stats:

```
. gen newstr=stratid
. gen newpsu=psuid
. replace newpsu=psuid + 2 if stratid==1
. replace newstr=2 if stratid==1
. svyset strat newstr
. svyset psu newpsu
. svydes x1, bypsu
```

```
. svy: mean x1 x2
. svy: mean x1 x2, obs
. svy: mean x1 x2, com ci deff
. svy: test x1 x2
```

complete: nonmissing only
no test with com, subpop or
by() in command
svylc requires equal # obs,
so must run com
beforehand

```
. svy: mean x1 x2, com ci deff
. svy: lc x1 - x2, deff
```

```
. svy: mean x1 x2, subpop(female)
. svy: mean x1 x2, subpop(female) obs
. svy: mean x1 x2, subpop(female) com ci deff
```

no test with com, subpop
or by() in command

Note: "subpop" only works for a binary var's 1-level. So convert any multilevel categorical var into a series of 0/1 dummy vars; or for a dummy var create another, complementary dummy var: male 0=female 1=male; female 0=male 1=female. Then specify the 1-level of any dummy var as "subpop." Alternatively, simply specify, e.g., either "by(male)" or "by(female)" to yield the data for male & female.

```
. svy: mean x1, by(female)
. svy: mean x1, by(female black)
```

```
. svy: prop x1
. svy: prop, by(female) deff
. svy: prop, subpop(black) deff
```

no obs or com options

<pre>. svy:ratio x1 x2 svy:ratio x1 x2, obs svy:ratio x1 x2, com ci deff</pre>	<p>provides more accurate mean-estimate</p>
<pre>. svy:total x1 x2 svy:total x1 x2, obs svy:total x1 x2, com ci deff svy:lc x1 - x2, deff svy:total, by(female) svy:total, by(female) obs svy:total, by(female) com ci deff svy:total, subpop(hispanic) svy:total, subpop(hispanic) com ci deff</pre>	<p>must have equal obs; run com beforehand</p>
<pre>. svy:tab x1 x2 svy:tab x1 x2, row se ci svy:tab x1 x2, row se ci format(%7.4f) svy:tab, row se ci nomarg</pre>	<p>row totals in small format row totals in large format no row totals</p>
<pre>. svy:tab x1 x2 x3, tab(x4) row svy:tab gender race, tab(income) row</pre>	<p>computes proportions relative to a specified var</p>
<pre>. svy:reg y x1 x2 x3 svy:reg y x1 x2 x3, deff linktest svy:test x1 x2 svy:lc x1 - x2, deff</pre>	
<pre>. svy:logit y x1 x2 x3 linktest svy:logit y x1 x2 x3, deff svy:logit, or di(or - 1)*100 svy:test x1 x2 svy:lc x1 - x2, deff</pre>	<p>displays previous results as odds ratio; display % change</p>
<p><i>To combine subpop() with by():</i></p> <pre>. gen black=(race==1) if race!=. svy:mean x1, subpop(black) by(marital age20)</pre>	<p><i>do not use if x1==</i></p>
<pre>. svy:mean, ci deff deff meff meff obs size</pre>	<p>deff & ci are default</p>
<pre>. svy:total x1, by(female)</pre>	<p>same syntax as svymean</p>
<pre>. svy:ratio x1 x2 svy:ratio x1/x2</pre>	<p>computes x1/x2 ratio no obs or com</p>
<p><i>[alternatively: svymean x1, subpop(x2)]</i></p>	<p>perhaps easier to do if x2 is set up appropriately</p>
<pre>. svy:prop x1 x2 svy:prop x1 x2, subpop(female) svy:prop x1, by(white female)</pre>	

```

. svyset fpc hid                                see Levy & Lemeshow
svyset
svy:mean x1

. svy:reg y x1 x2 x3 x4
svy:test x1 x2
svy:test x1 x2, b

. svy:mlogit health female black age age2
svy:test [good] female=[excellent] female, notest
svy:test [good]black=[excellent]black, accum

```

How to do a design-based analysis (Levy & Lemeshow, chap. 16)

1. Identify the following elements of the sample design: stratification; clustering vars used; pop sizes required to determine fpc's
2. Using the above info, determine the sampling weight for each sample object
3. Determine for each sample record a final sampling weight that takes into consideration any nonresponse & poststratification adjustments that are desired
4. Ensure that all stratification, clustering, & pop size data required for an appropriate design-based analysis are identified on each sample record
5. Determine the procedure & set of commands for performing the required analysis for the particular software package used
6. Run the analysis & interpret the findings

How to incorporate stratification on several vars simultaneously [Stata digest v4 #872]: the following *does not work* if unequal sampling was done at the various levels of stratification; in that case, use SUDAAN)

1. egen stratvar=group(industry region size)
2. do crosstabs of psu with each individual stratification var to see if there's overlap

How to deal with the following design: persons clustered at various sites, which in turn are stratified by geographic region--use pweight & cluster options, & include fixed effects for the regions [Stata digest v4 #872]