Analyzing Survey Data Using Stata 10

Roberto G. Gutierrez

Director of Statistics
StataCorp LP

2008 Summer NASUG, Chicago

1. About survey data
2. Using `svyset`
3. Data analysis
4. Bootstrapping via replicate weights
5. Concluding remarks

- All things being equal, a simple random sample gives the most efficiency per observation collected
- Oftentimes, however, "all things" are not equal
- Cost (monetary or otherwise) considerations often dictate that samples not be taking strictly at random
- Examples of this include
  - Undersampling where it is more expensive, or more homogeneous
  - Sampling groups rather than individuals (a city block, for instance)
  - Realizing your sampling frame is not indicative of the population, and weighting accordingly

- The cost of not performing a simple random sample (SRS) can be measured in terms of accuracy and precision
- Parameter estimates can be made accurate through proper weighting
- You cannot make your estimates as precise as if you took an SRS, but you can find out what precision you do have
- To get it all correct, however, there are four aspects of survey data that need to be considered and accounted for

- **Stratification** refers to the taking of two (or more) independent random samples and combining the information to make joint inference about the entire population. Each *strata* has its own variability and may be sampled at a different rate.

- **Clustered Sampling** occurs when individuals are sampled in groups rather than individually. Individuals within the same cluster (or PSU, primary sampling unit) share the same sampling fate.

- **Probability (sampling) weights** indicate weighted sampling. An individuals "p-weight" is equal to the inverse probability of being sampled, or equivalently the number in the population represented.

- A **finite population correction** (FPC) represents that we are sampling without replacement, **AND** that the population is small enough for that to matter.

- Stata 10.0 is fully "survey-capable"
- In Stata, there is a clear separation between setting the design and performing the actual analysis
- You declare the design characteristics using `svyset`
- This declaration is a one-time event. You save the survey settings along with the data
- You perform the analysis just as you would with i.i.d. data – you just have to add the `svy:` prefix
- As such, survey in Stata is as easy as learning to use `svyset`

Example

- Consider data on American high school seniors, collected following a multistage design
- Sex, race, height, and weight were recorded
- In the first stage of sampling, counties were independently selected from each state
- In the second stage, schools were selected within each chosen county
- Within each school, every attending senior took the survey
- The data are at `http://www.stata-press.com`, easily accessible from within Stata

```
. use http://www.stata-press.com/data/r10/multistage
. describe
Contains data from http://www.stata-press.com/data/r10/multistage.dta
  obs:         4,071
 vars:            11                          29 Mar 2007 00:53
 size:       122,130 (98.8% of memory free)
```

variable name	storage type	display format	value label	variable label
sex	byte	%9.0g	sex	1=male, 2=female
race	byte	%9.0g	race	1=white, 2=black, 3=other
height	float	%9.0g		height (in.)
weight	float	%9.0g		weight (lbs.)
sampwgt	double	%9.0g		sampling weight
state	byte	%9.0g		State ID (strata)
county	byte	%9.0g		County ID (PSU)
school	byte	%9.0g		School ID (SSU)
id	int	%9.0g		Person ID
ncounties	byte	%9.0g		Stage 1 FPC
nschools	int	%9.0g		Stage 2 FPC

```
Sorted by:  state  county  school
```

```
. svyset county [pw=sampwgt], strata(state) fpc(ncounties) || school, fpc(nschools)
      pweight: sampwgt
          VCE: linearized
  Single unit: missing
     Strata 1: state
         SU 1: county
        FPC 1: ncounties
     Strata 2: <one>
         SU 2: school
        FPC 2: nschools
. save highschool
file highschool.dta saved
```

- In more standard problems, the syntax is of the form

  ```
  . svyset psu_variable [pw=weight_variable], strata(strata_variable)
  ```

- Since we save the data with the survey settings as `highschool.dta`, we don't ever have to specify the design again – it is part of the dataset.

Other features of svyset include:

- You can have more than two stages, each separated by ||
- The default variance estimation is set to Taylor linearization, but you could also choose the jackknife, or balanced and repeated replication (BRR)
- You can tell Stata how you would like to treat strata with singleton PSUs
- You can treat them either as an error condition (missing), or as certainty units that can be centered and/or scaled

```
. svydescribe weight
Survey: Describing stage 1 sampling units
      pweight: sampwgt
          VCE: linearized
  Single unit: missing
     Strata 1: state
  (output omitted)
```

Stratum	#Units included	#Units omitted	#Obs with complete data	#Obs with missing data	#Obs per included Unit min	mean	max
1	2	0	92	0	34	46.0	58
2	2	0	112	0	51	56.0	61
3	2	0	43	0	18	21.5	25
4	2	0	37	0	14	18.5	23
(output omitted)							
47	2	0	67	0	28	33.5	39
48	2	0	56	0	23	28.0	33
49	2	0	78	0	39	39.0	39
50	2	0	64	0	31	32.0	33
50	100	0	4071	0	14	40.7	81

4071

To get some means and confidence intervals treating the data as a simple random sample, you would type

```
. mean height weight, over(sex)
Mean estimation                         Number of obs    =      4071

         male: sex = male
       female: sex = female
```

Over	Mean	Std. Err.	[95% Conf.	Interval]
height				
male	69.22091	.0737168	69.07639	69.36544
female	65.48295	.0615088	65.36236	65.60354
weight				
male	163.0539	.7094428	161.663	164.4448
female	138.0472	.7112746	136.6527	139.4416

To incorporate the survey design, you merely add "svy:"

```
. svy: mean height weight, over(sex)
(running mean on estimation sample)

Survey: Mean estimation

Number of strata =        50          Number of obs    =      4071
Number of PSUs   =       100          Population size  = 8.0e+06
                                      Design df        =        50

          male: sex = male
        female: sex = female
```

Over	Mean	Linearized Std. Err.	[95% Conf.	Interval]
height				
male	69.64261	.1187832	69.40403	69.88119
female	65.79278	.0709494	65.65027	65.93529
weight				
male	165.4809	1.116802	163.2377	167.7241
female	136.204	.9004157	134.3955	138.0125

How about a linear regression?

```
. generate male = (sex == 1)
. generate height2 = height^2
. svy: regress weight height height2 male
(running regress on estimation sample)
Survey: Linear regression
Number of strata   =        50                  Number of obs      =      4071
Number of PSUs     =       100                  Population size    =   8000000
                                                Design df          =        50
                                                F(   3,     48)    =    244.44
                                                Prob > F           =    0.0000
                                                R-squared          =    0.2934
```

weight	Coef.	Linearized Std. Err.	t	P>|t|	[95% Conf.	Interval]
height	-19.15831	4.694205	-4.08	0.000	-28.5869	-9.729724
height2	.16828	.0351139	4.79	0.000	.0977517	.2388083
male	14.88619	1.628219	9.14	0.000	11.61581	18.15656
_cons	666.8937	156.905	4.25	0.000	351.7408	982.0467

- This also works for nonlinear models, such as logistic regression
- Let's use the NHANES2 data

```
. use http://www.stata-press.com/data/r10/nhanes2d, clear
. svyset
      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: strata
         SU 1: psu
        FPC 1: <zero>
```

- Typing `svyset` without arguments will replay the survey settings for you

We can use these data to fit a logit model for high blood pressure, and get survey-adjusted odds ratios and standard errors

```
. svy: logistic highbp height weight age female
(running logistic on estimation sample)

Survey: Logistic regression

Number of strata   =        31                  Number of obs      =     10351
Number of PSUs     =        62                  Population size    = 1.172e+08
                                                Design df          =        31
                                                F(   4,     28)    =    178.69
                                                Prob > F           =    0.0000
```

highbp	Odds Ratio	Linearized Std. Err.	t	P>|t|	[95% Conf.	Interval]
height	.9688567	.0056821	-5.39	0.000	.9573369	.9805151
weight	1.052489	.0032829	16.40	0.000	1.045814	1.059205
age	1.050473	.0024816	20.84	0.000	1.045424	1.055547
female	.7250086	.0641185	-3.64	0.001	.6053533	.8683151

- You can also get odds ratios specific to females

```
. svy, subpop(female): logistic highbp height weight age
(running logistic on estimation sample)

Survey: Logistic regression

Number of strata   =        31                  Number of obs      =     10351
Number of PSUs     =        62                  Population size    = 1.172e+08
                                                Subpop. no. of obs =      5436
                                                Subpop. size       =  60998033
                                                Design df          =        31
                                                F(   3,     29)    =    137.05
                                                Prob > F           =    0.0000
```

highbp	Odds Ratio	Linearized Std. Err.	t	P>|t|	[95% Conf.	Interval]
height	.9765379	.0092443	-2.51	0.018	.957865	.9955749
weight	1.047845	.0044668	10.96	0.000	1.038774	1.056994
age	1.058105	.003541	16.88	0.000	1.050907	1.065352

- This is not the same as throwing away the data on males, and Stata knows this

How about jackknife standard errors?

```
. svy jackknife, subpop(female): logistic highbp height weight age
(running logistic on estimation sample)
Jackknife replications (62)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
............
Survey: Logistic regression
Number of strata   =        31                  Number of obs      =     10351
Number of PSUs     =        62                  Population size    = 1.172e+08
                                                Subpop. no. of obs =      5436
                                                Subpop. size       =  60998033
                                                Replications       =        62
                                                Design df          =        31
                                                F(   3,     29)    =    136.91
                                                Prob > F           =    0.0000
```

highbp	Odds Ratio	Jackknife Std. Err.	t	P>|t|	[95% Conf.	Interval]
height	.9765379	.0092477	-2.51	0.018	.957858	.9955821
weight	1.047845	.0044691	10.96	0.000	1.038769	1.056999
age	1.058105	.0035427	16.87	0.000	1.050904	1.065355

- When performing simultaneous tests, denominator degrees of freedom need to be adjusted for strata and PSUs

```
. test height weight
Adjusted Wald test
 ( 1)  height = 0
 ( 2)  weight = 0
       F(  2,    30) =   58.21
            Prob > F =    0.0000
. test height weight, nosvyadjust
Unadjusted Wald test
 ( 1)  height = 0
 ( 2)  weight = 0
       F(  2,    31) =   60.15
            Prob > F =    0.0000
```

- Other postestimation routines, such as linear combinations of estimates, and nonlinear tests and combinations can also be applied after survey estimation

After fitting the model, you can obtain design effects due to survey by using estat

```
. estat effects
```

highbp	Coef.	Jackknife Std. Err.	DEFF	DEFT
height	-.0237417	.0094699	1.31101	1.14499
weight	.0467353	.0042651	1.74506	1.32101
age	.0564794	.0033482	.916825	.95751
_cons	-4.507688	1.561851	1.29274	1.13699

```
. estat effects, meff meft
```

highbp	Coef.	Jackknife Std. Err.	MEFF	MEFT
height	-.0237417	.0094699	1.62184	1.27351
weight	.0467353	.0042651	2.23313	1.49437
age	.0564794	.0033482	.922923	.960689
_cons	-4.507688	1.561851	1.61274	1.26994

- Semiparametric Cox and fully-parametric (e.g., Weibull) regression models can be fit with survey data
- Declaring survival data to Stata works similarly to declaring survey data
- In the case of survival data, you declare time variable(s), censoring indicators, sampling weights, etc.
- These declarations layer over the survey declarations, and Stata makes sure there are no conflicts
- Of course, survival settings can also be saved with the data

```
. use http://www.stata-press.com/data/r10/nhefs
. svyset psu2 [pw=swgt2], strata(strata2)
      pweight: swgt2
          VCE: linearized
  Single unit: missing
     Strata 1: strata2
         SU 1: psu2
        FPC 1: <zero>
. stset age_final [pw=swgt2], fail(died)
     failure event:  died != 0 & died < .
obs. time interval:  (0, age_final]
 exit on or before:  failure
           weight:  [pweight=swgt2]
--------------------------------------------------------------------------
    14407  total obs.
     1344  event time missing (age_final>=.)                  PROBABLE ERROR
--------------------------------------------------------------------------
    13063  obs. remaining, representing
     4604  failures in single record/single failure data
   861932  total analysis time at risk, at risk from t =             0
                             earliest observed entry t =             0
                                  last observed exit t =            96
```

```
. svy: stcox former_smoker smoker male urban1 rural
(running stcox on estimation sample)

Survey: Cox regression

Number of strata   =        35                  Number of obs      =      10753
Number of PSUs     =       105                  Population size    =  178083231
                                                Design df          =         70
                                                F(   5,     66)    =      67.25
                                                Prob > F           =     0.0000
```

_t	Haz. Ratio	Linearized Std. Err.	t	P>|t|	[95% Conf.	Interval]
former_smo~r	1.239317	.0829107	3.21	0.002	1.084514	1.416217
smoker	2.691434	.1961611	13.58	0.000	2.327309	3.112529
male	1.523904	.0957688	6.70	0.000	1.344385	1.727395
urban1	.8997145	.0529653	-1.80	0.077	.8000443	1.011802
rural	.9016422	.0557823	-1.67	0.099	.7969779	1.020052

- Replicate weights are becoming increasingly popular
- Privacy is the main reason
- Instead of recording strata/PSU membership and the original weights, you keep a (large) set of weight variables reflecting repeated sampling
- These repeated samples can be based on the jackknife, balanced and repeated replication (BRR), or the bootstrap
- I'll discuss the bootstrap since, in my opinion, it is the most popular

- To perform the bootstrap with survey data, you need to install a piece of software
- This is not part of official Stata, but easily installed from the web as a "user-written" program
- The author is Jeff Pitblado (`jpitblado@stata.com`) of StataCorp, so in a way it is official
- It will eventually be part of official Stata.

- To install the bs4rw program, you can type

```
. net install http://www.stata.com/users/jpitblado/bs4rw, replace
checking bs4rw consistency and verifying not already installed...
installing into c:\ado\plus\...
installation complete.
```

- But the above assumes you know where to go. An alternative is to type

```
. findit survey bootstrap
```

  and follow the links toward installing.

- As I like to say, findit is Google for Stata

bs4rw is a prefix command, analogous to svy:. It works with all the commands that work with svy:

```
. use http://www.stata-press.com/data/r10/autorw, clear
(1978 Automobile Data)
. bs4rw, rweights(boot*): regress mpg for weight
(running regress on estimation sample)
BS4Rweights replications (300)
  (output omitted)
```

```
Linear regression                               Number of obs     =         74
                                                Replications      =        300
                                                Wald chi2(2)      =     167.11
                                                Prob > chi2       =     0.0000
                                                R-squared         =     0.6627
                                                Adj R-squared     =     0.6532
                                                Root MSE          =     3.4071
```

mpg	Observed Coef.	Bootstrap Std. Err.	z	P>|z|	Normal-based [95% Conf.	Interval]
foreign	-1.650029	1.065621	-1.55	0.122	-3.738608	.4385502
weight	-.0065879	.0005102	-12.91	0.000	-.0075879	-.0055879
_cons	41.6797	1.666637	25.01	0.000	38.41315	44.94625

- To analyze survey data means dealing with strata, clusters, weights, and finite sampling
- Stata 10.0 is "fully-functional" for survey data
- The key is to master `svyset`, and we are happy to help out here
- Multistage designs work just fine, as does Cox regression and parametric survival models
- Bootstrapping based on replicate weights available as a user-written add-on