Introduction to Complex Sample Design: Workbook

Anthony Rafferty

Workshop 1. Using svyset commands in Stata: Weighting and Clustering

1.0 Introduction

The topics covered in the first workshop are:

- How to declare the complex sample design features of you survey to Stata using the **svyset** command. We will focus for now on identifying the primary sampling units and weights (as this often satisfies for most purposes). Stratification and secondary sampling units are considered in workshop 2.
- How to create summary statistics such as frequencies, means, and cross-tabulations incorporating complex survey design (**svy**: commands).
- Conducting sub-population analysis correctly.
- Basic modelling and estimating design effects using svyset. In particular, we will focus on the effects of clustering on standard errors and on the statistical significance of findings.

Getting Started

- **Bold bullet points are instructions for the workshop tasks.** The do file (complex samples do file) also contains the commands for the following examples. The numbers of the sections correspond to the numbers of the sections in this workbook. Alternatively, you can type or copy and paste syntax from this document into the command box in Stata.
- **Do file "complex samples do file" is in C:**/work/
- **Access via my computer in start menu.**
- **The data and an electronic copy of the workbook are in this folder as well.**
- Use Stata 10 go to start menus select: all programs/faculty of humanities/econometric & statistical/stata10 and you will find stata 10.
- Set memory before starting (See below) so big enough to hold dataset.

Health Survey for England (HSE)

The Health Survey for England (HSE) is a series of annual surveys about the health of people living in England. Since 1994 the survey has been carried out by the Joint Health Surveys Unit of the National Centre for Social Research and the Department of Epidemiology and Public Health, Royal Free and University College Medical School, London. The survey is sponsored by the Department of Health.

Sample Design

The sample for the HSE was drawn in two stages. At the first stage a random sample of primary sampling units (PSUs), based on postcode sectors, was selected. Within each selected PSU, a random sample of postal addresses (known as delivery points) was then drawn. To maximise the precision of the sample, it was selected using a

method called stratified sampling. The list of PSUs in England was ordered by local authority and, within each local authority, by the percentage of households in the 2001 Census with ahead of household in a non-manual occupation (NS-SEC groups 1-3).

The sample of PSUs was then selected by sampling from the list at fixed intervals from a random starting point. 900 PSUs were selected with probability proportional to the total number of addresses within them. Selecting PSUs with probability proportional to number of addresses and sampling a fixed number of addresses in each ensures that an efficient (equal probability) sample of addresses is obtained. Once selected the 900 PSUs were randomly allocated to one of two groups; 720 PSUs were allocated to the core sample and 180 PSUs were allocated to the additional child boost sample. The core PSUs contained sampled addresses for both the core and child boost sample, the additional child boost PSUs only.

- **Q. Recall, in a sentence, how would you describe the sample design of the HSE?**
  - **single stage/multi-stage-**
  - **(implicitly/explicitly/) (proportionately/ disproportionately) stratified/unstratified?**
  -
- **Let us take a quick look at the data. Type:**

```
clear
set memory 400m
cd c:\work\
use hse07.dta

log using workshop
```

- **The log file will keep a record of your output**
- **In command box, type:**

```
describe
```

- **This gives a description of the data**.

The key variables we will be using are:

VARIABLE NAME	DESCRIPTION
Sample Design	
area[1]	Primary Sampling Unit (PSU)
hserial	Secondary Sampling Unit (SSU)
int_wt	Interview weight. (other weights for different analyses are available[2] but we will focus on the main weight.
cluster	Stratification variable (original format)
samptype	Whether main or child boost sample
Other Variables Used	
age	age
sex	sex
topqual2	Highest Qualification
econact	Economic Activity Status
cksalt	Whether add salt whilst cooking? (yes/no/salt alternative)
gor07	Government Office Region (GOR) (called gor06 in the 07 data we will use in workshop 2 and 3)

1.1. Declaring your sample design using svyset

The command **svyset** (declare data as survey data) is used to identify the sample design features of your data to Stata.

Single-stage design syntax:

svyset [psu] [weight] [, design_options options]

1.1.1. Weighting using svyset

- **We will first consider how to specify the weighting variable and consider its implications for disproportionate stratification/ sampling. Type:**

```
svyset [pweight=wt_int]
```

- **A summary of the specified design will be displayed.**

[1] In 2006, for one year it was called 'psu'.

[2] In 2007, non-response weights have been calculated for both adults and children. Four sets of non-response weights have been generated in total. Firstly a household weight was calculated to adjust for non-contact and for refusals of entire households (hhld_wt). In addition, three sets of weights have been calculated to adjust (a) non-response among individuals in responding households, interview weight (int_wt), (b) non-response to the nurse visit stage and (nurse_wt), (c) refusal to give a blood sample (blood_wt). There are also weights specific to individual years to use with given booster samples.

- **pweight means probability weight**
- **Note in the 07 HSE, there are several weights for different purposes. Type:**

```
describe wt_*
```

- **Recall there is also a child booster sample and the core sample:**

```
tabulate samptype
```

The sample for the HSE 2007 thus comprises of two components the core (general population) sample and a boost sample of children aged 2-15. The weight (wt_int) is constructed for analysing the core sample (ignoring the boost) and so assigns the child boost sample a zero weight:

- **Type:**

```
tabulate wt_int if samptype == 2
```

The weight for the analysis of the core and child boost together (wt_child) in contrast assigns a positive weight value to the child boost cases. However, as children are disproportionately over-sampled, this weight (wt_child) will act to weight down their influence so that they are proportional in the sample to their population size. Otherwise, they will have an impact on estimates disproportionate to their population size. This demonstrates the importance of applying weights where disproportionate stratification or sampling has taken place.

1.1.2. Including the Primary Sampling Unit (PSU)

- **'area' is the 07 PSU variable. Type:**

```
svyset area [pweight = wt_int]
```

1.2 Descriptive statistics and sub-population analysis

Once you have svyset your data, most survey design commands can be executed by prefixing command lines with **svy:**

We will give examples of commands here in the workshop, but a more exhaustive list is provided Stata manual or by typing : help svyset.

1.2.1. Summary statistics

- **First, estimate the mean of age assuming simple random sampling (but omitting child boost sample). Type:**

```
mean age if samptyp ==1
```

- **Now see what happens when we apply the weights (but not the PSU). Type:**

```
svyset [pweight=wt_int]
svy: mean age
```

- **Note the standard error remains similar but a greater effect of weighting is witnessed on the point estimate (i.e. the mean).**
- **Now applying weights and PSU ('area'. Type:**

```
svyset area [pweight=wt_int]
svy: mean age
```

- **The estimate of the mean remains identical to when only the weights are applied but the standard errors (and so confidence intervals) are much bigger.**
- **To find out the design effect and factor, type:**

```
estat effects
```

- **The standard error of our estimate is around 43% bigger when we take into account clustering.**

1.2.2. Frequencies (one-way tables)

- **These are produced using the 'svy: tab' command (tabulate). Type:**

```
svy: tab sex
```

- **Compare standard simple random sample approach. Type:**

```
tab sex if samptyp==1
```

1.2.3. Cross-tabulation (two-way tables)

- **The tab command is also used for two-way tables (cross-tabs)**

```
svy: tab topqual2 sex
```

- **Standard table options such as row percent, column percent, and confidence intervals can be specified:**

```
svy: tab topqual2 sex, row cell
```

1.2.4. Sub-population analysis

In many analyses, you may wish to focus on a sub-population, such as men or women, or a specific age group. A standard approach to this would be to use the **'if'** command (e.g. tab age if sex ==1), or to drop unwanted cases. However, svyset commands require information on the entire population size to calculate standard errors. Such approaches should therefore be avoided, and instead, the '**subpop**' command should be used (although in practice it often does not make much difference).

- **We first need a binary variable coded as 1=subpopulation of interest, 0 = not subpopulation of interest (missing if we don't know). In the data, we will generate a recode of our variable 'sex' (currently 1=men 2==women, recoding women to 0). Type:**

```
recode sex (2=0), ge(male)
```

- **Next, apply the subpop command:**

```
svy, subpop (male): mean age
```

- **This can be applied to most commands. For means, alternatively, we can use the 'over' command:**

```
svy: mean age, over (male)
```

1.3 Basic multivariate analysis

In the final part of the first workshop, we will consider an example of using svyset for logistic regression. Similar procedures apply to most standard modelling techniques:

- **We will first create a binary outcome variable (1=adds salt when cooking, 2=doesn't add salt/uses salt alternative). Type:**

```
ta cksalt

recode cksalt (3=0) (2=0), ge(cksalt_b)
ta cksalt_b cksalt
```

- **Model without clustering applied:**

```
*(enter code as one line)

xi: logistic cksalt_b sex i.gor07 i.econact age
i.topqual2 [pweight=wt_int]
```

- **Store estimate so we can use it later (call it 'model1):**

```
est store model1
```

- **Model with PSU specified:**

```
svyset area [pweight = wt_int]

xi: svy: logistic cksalt_b sex i.gor07 i.econact age
i.topqual2

est store model2
```

- **Compare two models using estimates table command to examine standard errors and significance. Type:**

```
est table model1 model2 , se p eform
```

- **Specified table options: se gives standard error, p give p value, eform gives exponentiated coefficients (odds ratios)**

- **Compare just with stars indicating significance. Spot the difference better between the models?:**

```
est table model1 model2 , star eform
```

- **Examine design effects (will estimate for last model run):**

```
estat effects
```

Due to clustering in the selection procedure, individuals are not selected independently. This results in correlation within clusters that can inflate variances of estimates compared to those obtained from a simple random sample (SRS) of the same size.

Workshop 2: Stratification

2.0. Introduction

Workshop 2 considers the sometimes more labour intensive, but often analytically less rewarding topic of stratification. Stratification, as we shall see, typically has less of an impact on design effects.

The topics covered are:

- How to inspect and, if necessary, prepare a stratification variable for inclusion in your analysis using the **svydes** command and other general Stata commands.
- The effects of stratification on standard errors.

2.1. Effects of Stratification

Generally speaking, the effects of clustering on the efficiency of survey estimates tend to be greater than those of stratification, meaning that ignoring stratification can be less of a concern than ignoring clustering, although weights should not be ignored, often even more so where disproportionate stratification has been employed (see workshop 1).

In the syntax for svyset, stratification is declared as a design option. Recall the command structure:

- svyset [psu] [weight] [, design_options options]
- we will use the 06 HSE for this practical
- PSU is called PSU in 06 not area; and Government Office Region variable is gor06, not gor07.
- **Load data, type:**

```
clear

set memory 400m

cd c:\work\

use c:\work\hse06.data
```

- **recreate out salty dependent variable:**

```
recode cksalt (3=0) (2=0), ge (cksalt_b)
```

- **Confusingly, the stratification variable in the HSE is called cluster (!). In the command box, type:**

```
svyset psu [pweight = wt_int], strata(cluster)
```

- A summary of the specified design will be displayed.

- **run model with stratification specified:**

```
xi: svy: logistic cksalt_b sex i.gor06 i.econact age
i.topqual2
est store m1
estat effects
```

- **Now lets compare to when we just specified the PSU (and not the strata variable)**

```
svyset psu [pweight = wt_int]
xi: svy: logistic cksalt_b sex i.gor06 i.econact age
i.topqual2
est store m2
estat effects
est table m1 m2, eform se
```

- **stars to see significance easily**

```
est table m1 m2, eform star
```

Standard errors slightly smaller when we included stratification, but overall not a big enough change in the current example to alter statistical significance of coefficients?

Identifying singleton stratum and inspecting your sample structure

A common problem encountered when including stratification in your analysis is that of **singleton stratum**. When there is only one PSU within a stratum, there is not enough information from which to compute the stratum's variance, making it impossible to compute the variance of an estimated parameter in a stratified clustered design.

When this happens, the standard errors fail to be calculated, and at the bottom of the table, you should get the following message:

```
Note: missing standard errors because of stratum with
single sampling unit.
```

- In such cases, further work is therefore required to detect and handle singleton stratum.

- **Firstly, the svydes (survey describe) command can be used to inspect your stratification variable, looking for singleton stratum. Type:**

```
svydes
```

```
Survey: Describing stage 1 sampling units

      Survey: Describing stage 1 sampling units

      pweight: wt_int
          VCE: linearized
  Single unit: missing
     Strata 1: cluster
         SU 1: psu
        FPC 1: <zero>

                                    #Obs per Unit
                                -----------------------------
Stratum    #Units      #Obs       min        mean       max
--------  --------  --------  --------  --------  --------
       1         2        24         8      12.0        16
       2         2        61        30      30.5        31
       3         2        51        21      25.5        30
       4         2        64        29      32.0        35
       5         2        12         2       6.0        10
       6         2         9         2       4.5         7
       7         2        13         6       6.5         7
       8         2        20         9      10.0        11
       9         2        31        15      15.5        16

 [Omitted]……..etc
```

- Explanation of Output: **Stratum** is the stratum id number given by the strata variable;
- **#units** is the number of PSUs in the strata and **#Obs** the number of observations in a given stratum. The other columns give some summary statistics on the number of observations.
- The important thing to note here: if strata have have singleton PSUs then #units will =1. This means they only include one PSU- its also indicated by a **(*)**
- In our current example, our stratification variable looks fine (no *)

Strata with singleton PSUs can arise for several reasons:

- For an estimator, or list of covariates, singleton strata may result from missing cases. For example, for a mean, there may be missing cases for all the observations in a particular stratum except for those in a single PSU.

- **In such cases, use the svydes command with a list of variables you are interested in to check whether there are singleton stratum. Type**

```
svydes sex gor06 econact age topqual2 cksalt_b
```

- Another source of singleton stratum is if observations are dropped from a model as they are not in an estimation sample such as in logit or probit models because a variable or group of variables perfectly predicts an outcome. In such a case, **logit** or **probit** would just terminate with an error message. To verify that this is the problem and to identify which observations are being dropped, use **logit** or **probit** with **pweight**s and the **cluster()** option (the clusters are the same thing as PSUs). You can then use the **e(sample)** function to identify the estimation sample.
- Another possible reason : The data depositor has made an error when deriving the strata variable is another potential reason. If the strata variable on an ESDS supported dataset looks suspect, contact ESDS and we will check it out for you.

Example where there are singleton stratum:

- **So everything is ok above. However, I will work through the following 'hypothetical' example where this is not the case to show you how to handle the problem, as this is a query qe get:**
- **On another dataset, through svyset, I get:**

			#Obs per Unit		
Stratum	#Units	#Obs	min	mean	max
1	2	37	18	18.5	19
2	2	30	14	15.0	16
3	2	27	10	13.5	17
4	1*	3	3	3.0	3
5	2	33	12	16.5	21
6	2	29	13	14.5	16
7	2	26	12	13.0	14
8	2	23	10	11.5	13
9	2	25	10	12.5	15
10	2	43	16	21.5	27
11	2	59	22	29.5	37
12	2	42	19	21.0	23
13	2	21	4	10.5	17

- Note the * indicating singleton stratum (this is on 06 data)
- To deal with singleton stratum, we need to identify which cases belong to them. The list command can be used to identify case numbers don't type (its 06 data example: `List area cluster if cluster == 4`

```
      +----------------+
      | area   cluster |
      |----------------|
7022. |  539         4 |
7023. |  539         4 |
7024. |  539         4 |
      +----------------+
```

Once you have identified cases in singleton stratum, there are a number of ways of dealing with singleton Strata[3]:

- Firstly, you can treat them as missing and error, deleting them from your sample.
- Singleton strata can be specified as 'certainty units' that are centred and/or scaled using the **singleunit (method)** option on for the svyset command[4]. See help svyset.
- Alternatively, you can group with other singleton strata to treat them like they belong to the other strata. For example, we could use the **recode** cases in cluster=4 to collapse value 4 in the cluster variable into 3 **Don't type this**!:

```
ge cluster_c = cluster
replace cluster = 3 in 7022/7024
```

- We can then repeat this procedure for all singleton strata identified by svydes, collapsing them into other stratum.
- Ouch.

[3] See **http://www.stata.com/support/faqs/stat/stratum.html**

[4] See: http://repec.org/snasug08/gutierrez_survey.pdf

Workshop 3. Further topics

3.0. Introduction

In this workshop, we will consider the following:

- Incorporating (or ignoring) multi-stage design (e.g. secondary sampling unit features) into your analysis and the 'ultimate cluster method'.
- Comparing linearization and replicate methods to complex sample analysis.
- A brief comparison of model-based and design-based approaches (this will be worked through at the front of class- in no great detail).

We are using the 06 HSE data.

3.1. Incorporating (or ignoring) multi-stage design features in your analysis

The Health Survey for England (HSE) is a multi-stage stratified random sample. The primary sampling unit variable is **area** and the Secondary Sampling Unit (SSU) is **address (hserial).** The following example considers what happens when we try to specify the secondary sampling unit:

Recall that the syntax for specifying multi-stage designs is as follows:

svyset psu [weight] [, design_options] [|| ssu , design_options] ... [options]

The key thing here to note is that the different stages of the sample are separated by to lines ||. You can therefore specify the design options of different stages.

- **In the following example, we will specify the primary sampling unit, secondary sampling units, and weights. Type:**

```
svyset psu [pweight=wt_int], strata(cluster)||hserial
```

- **In the output screen, you should get the following message:**

```
stage 1 is sampled with replacement, all further stages
will be ignored
```

- When using statistics packages that compute complex standard errors from multi-stage clustered samples it is generally only necessary to have a PSU variable in the dataset. Any clustering after the first stage generally does not have to be identified - the variance between PSUs automatically incorporates later stages of clustering[5]. This is referred to as the 'ultimate cluster method'.

[5] See: http://www2.napier.ac.uk/depts/fhls/peas/clustering.asp#intro

- The main exception to this rule is where a multi-stage sample design and Finite Population Correction (FPC) is specified, then information from further sampling units is required.
- Finite Population Correction adjusts design effects downwards to account for the effects of increased sample sizes, although to have a substantial effect a very large sample size is needed.
- However, because you did not specify a Finite Population Correction (FPC) at the first stage, the sampling information at the second stage is irrelevant to variance estimation.
- Overall FPC often has little or minor impact on survey research. We will not be going into this in any further detail here, but those interested in pursuing the topic further outside the workshop are directed to the PEAS website: http://www2.napier.ac.uk/depts/fhls/peas/finitepop.asp

3.2. Comparing design-based estimation approaches

There are two common alternative approaches used in Stata for variance/ covariance estimation for complex survey designs/ These are linearization (Taylor-Series) techniques, and replicate methods (BRR, Jackknife). Linearization is the default method, although alternative estimation approaches can be specified easily in **svyset** using the **vce ()** option. In the following example, we will compare results from these different methods:

- **First we will specify the default (linearization) method. Type:**
  (This is default so we do not have to explicitly specify the vce() option).

```
svyset psu [pweight=wt_int]

xi: svy: logistic cksalt_b sex i.gor06 i.econact age
i.topqual2

estat effects
```

- **We will now repeat the example using the Jackknife replicate method. Type:**

```
svyset psu [pweight=wt_int], vce(jackknife)
```

- **What is different about the svyset description given in the output screen when you enter this command?**

```
xi: svy: logistic cksalt_b sex i.gor06 i.econact age
i.topqual2

estat effects
```

The replicate methods are computationally demanding and generally take longer to run. In the above models, the estimation method chosen (reassuringly) does not make much of a difference to our estimates of standard errors.

3.3. A Brief comparison of model based and design-based approaches

The following example will be worked through at the front of class at the end of Workshop 2.

```
xi: xtlogit cksalt_b sex i.gor06 i.econact age i.topqual2 ,
i(psu)
```

Interpretation

- Lnsigu : psu level variance
- Sigmau standard deviation (square root lnsigu)
- rho: intra class correlation coefficient (portion of total variance accounted for by clustering (%)).
- Chi bar: significance of random effect.

Pros and cons of model based approach:

- model based more efficient (smaller standard errors)
- however assumptions on population structure, wrong model, wrong results
- stratification is included as covariate, however we may be interested in an estimate unadjusted by the stratification variable(s)
- design based arguably is easier for the inexperienced
- although there may be circumstances where your sampling units are of substantive interest