

Survey Data Analysis with Stata

Jeff Pitblado
Associate Director, Statistical Software
StataCorp LP

JSM 2011

Outline

1	Types of data	2
2	Survey data characteristics	3
2.1	Single-stage designs	4
2.2	Multistage designs	10
2.3	Poststratification	12
2.4	Strata with a single sampling unit	14
2.5	Certainty units	17
3	Variance estimation	17
3.1	Linearization	18
3.2	Balanced repeated replication (BRR)	23
3.3	Jackknife	26
3.4	Bootstrap	29
3.5	Successive difference replication	30
3.6	Replicate weights	31
4	Estimation for subpopulations	31
5	Postestimation	34
6	Summary	40

This workshop's materials have been posted to the following website for your convenience.

<http://www.stata.com/users/jpitblado/2011jsm/>

Why survey data?

- Collecting data can be expensive and time consuming.
- Consider how you would collect the following data:
 - Smoking habits of teenagers
 - Birthweights for expectant mothers with high blood pressure
- Using stages of clustered sampling can help cut down on expense and time.

1 Types of data

Simple random sample (SRS) data

Observations are independently sampled from a data-generating process.

- Typical assumption: data are independent and identically distributed (iid).
- Make inferences about the data generating process.
- Sample variability is explained by the statistical model attributed to the data generating process.

Standard data

We will use this term to distinguish this type of data from survey data.

Correlated data

Individuals are assumed not to be independent.

Causes:

- Observations are taken over time
- Random-effects assumptions
- Cluster sampling

Treatment:

- Time-series models
- Longitudinal/panel data models
- `vce(cluster ...)` option

Survey data

Individuals are sampled from a fixed population according to a survey design.

Distinguishing characteristics of this type of data include

- the complex nature under which individuals are sampled,
- inferences made about the fixed population, and
- sample variability attributed to the survey design.

2 Survey data characteristics

Standard data

- Estimation commands for standard data include
 - `proportion`, and
 - `regress`.
- We will refer to these as *standard estimation commands*.

Survey data

- Estimation commands for survey data are governed by the **svy** prefix, for example
 - **svy: proportion**, and
 - **svy: regress**.
- **svy** requires that the data be **svyset**.

▷ Example: `proportion` and **svy: proportion**

The standard header in the output for a Stata estimation command contains a title and some information about the sample.

- `proportion` reports the sample size.
- **svy: proportion** also reports the number of strata, the primary sampling units (PSU), the estimated population size, and the design degrees of freedom.

- **svy** reports the number of strata and PSUs, even for multistage designs, to show where the design degrees of freedom come from.

$$df = N_{PSU} - N_{strata}$$

Second National Health and Nutrition Examination Survey

```
. webuse nhanes2
. proportion sex
```

Proportion estimation Number of obs = 10351

	Proportion	Std. Err.	[95% Conf. Interval]	
sex				
Male	.4748333	.0049085	.4652117	.484455
Female	.5251667	.0049085	.515545	.5347883


```
. svy: proportion sex
(running proportion on estimation sample)
Survey: Proportion estimation
Number of strata =        31        Number of obs       =       10351
Number of PSUs   =       62        Population size      =   117157513
                                         Design df            =         31
```

	Proportion	Linearized Std. Err.	[95% Conf. Interval]	
sex				
Male	.4793502	.005734	.4676557	.4910447
Female	.5206498	.005734	.5089553	.5323443

proportion reports different proportion values than **svy: proportion**. This is due to the survey design characteristics that were **svyset** when the dataset was created.

◀

2.1 Single-stage designs

Single-stage syntax

```
svyset [psu] [weight] [, strata(varname) fpc(varname)]
```

Syntax elements:

- Primary sampling units (PSU)
- Sampling weights – **pweight**
- Strata
- Finite population correction (FPC)

Sampling unit

An individual or collection of individuals from the population that can be selected for observation.

- Sampling groups of individuals is synonymous with cluster sampling.
- Cluster sampling usually results in inflated variance estimates compared to SRS.

▷ Examples

- High schools for sampling from the population of 12th graders
- Hospitals for sampling from the population of newborn infants

◁

Sampling weight

The reciprocal of the probability that an individual to be sampled.

- Probabilities are derived from the survey design.
 - sampling units
 - strata
- Typically considered to be the number of individuals in the population that a sampled individual represents.
- Reduces bias induced by the sampling design.

▷ Example

If there are 100 hospitals in our population and we choose 5 of them, the sampling weight is $20 = 100/5$. Thus a sampled hospital represents 20 hospitals in the population.

Sampling weights correct for over- and undersampling of sections of the population. Many times this over- or undersampling is intentional.

◁

Strata

In stratified designs, the population is partitioned into well-defined groups called strata.

- Sampling units are independently sampled from within each stratum.
- Stratification usually results in smaller variance estimates compared with SRS.

▷ Examples

- States of the union are typically used as strata in national surveys in the U.S.
- Demographic information such as age group, gender, and ethnicity would yield highly effective strata.

Although there is potential for improving efficiency by reducing sampling variability, it is usually not very practical to stratify on demographic information.

◁

Finite population correction (FPC)

An adjustment applied to the variance due to sampling without replacement.

- Sampling without replacement from a finite population reduces sampling variability.

□ Notes

- We will see that the FPC affects the number of components in the linearized variance estimator for multistage designs.
- We can use **svyset** to specify an SRS design.

□

▷ Examples

The following examples show the use of **svyset** for single-stage designs:

1. `auto` – specifying an SRS design.
2. `nmihs` is the National Maternal and Infant Health Survey (1988) dataset that came from a stratified design.
3. `fpc` is a simulated dataset with variables that identify the characteristics from a stratified and without-replacement clustered design.

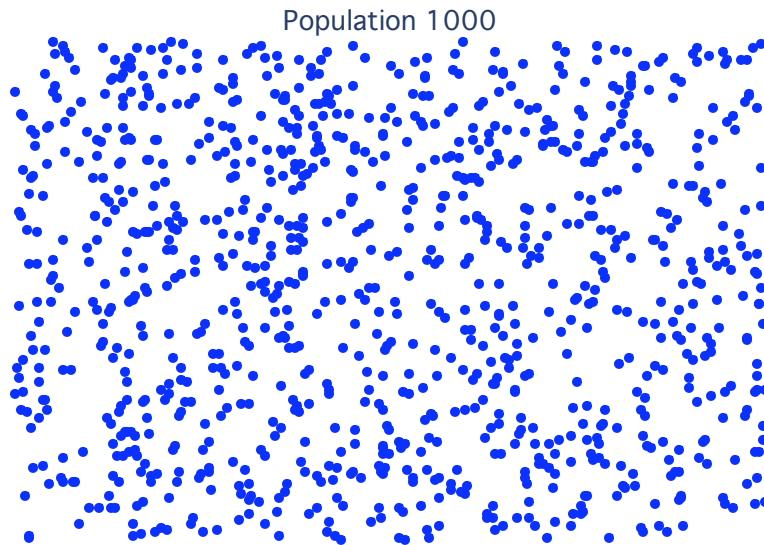
```
— The auto data that ships with Stata —
. sysuse auto
(1978 Automobile Data)
. svyset _n
      pweight: <none>
      VCE: linearized
Single unit: missing
Strata 1: <one>
SU 1: <observations>
FPC 1: <zero>

— National Maternal and Infant Health Survey —
. webuse nmihs
. svyset [pw=finwgt], strata(stratan)
      pweight: finwgt
      VCE: linearized
Single unit: missing
Strata 1: stratan
SU 1: <observations>
FPC 1: <zero>

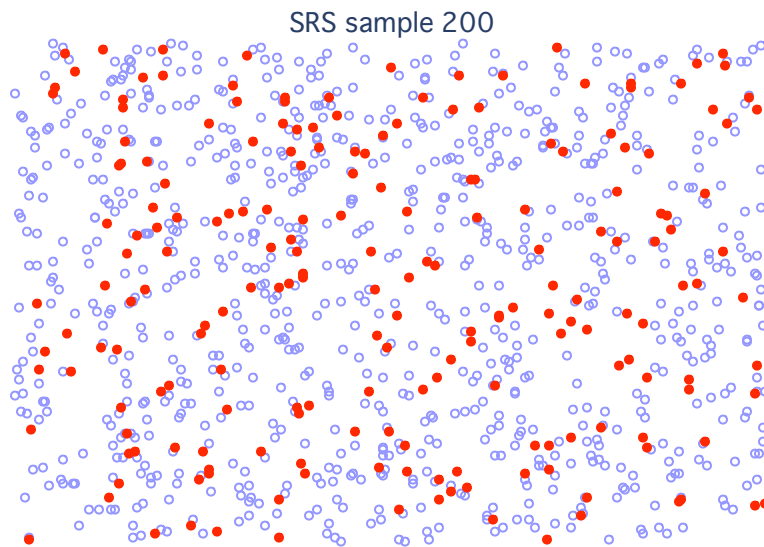
— Simulated data —
. webuse fpc
. svyset psuid [pw=weight], strata(stratid) fpc(Nh)
      pweight: weight
      VCE: linearized
Single unit: missing
Strata 1: stratid
SU 1: psuid
FPC 1: Nh
```

◀

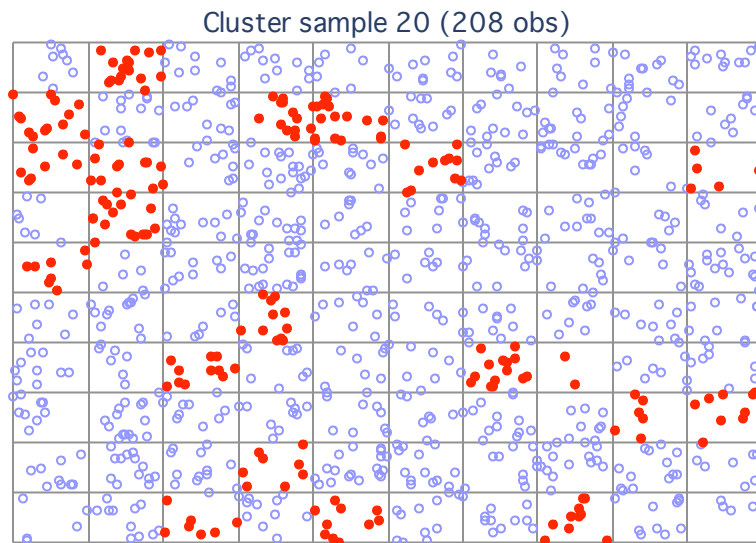
Below is a visual representation of a hypothetical population. Suppose that each dot represents an individual.



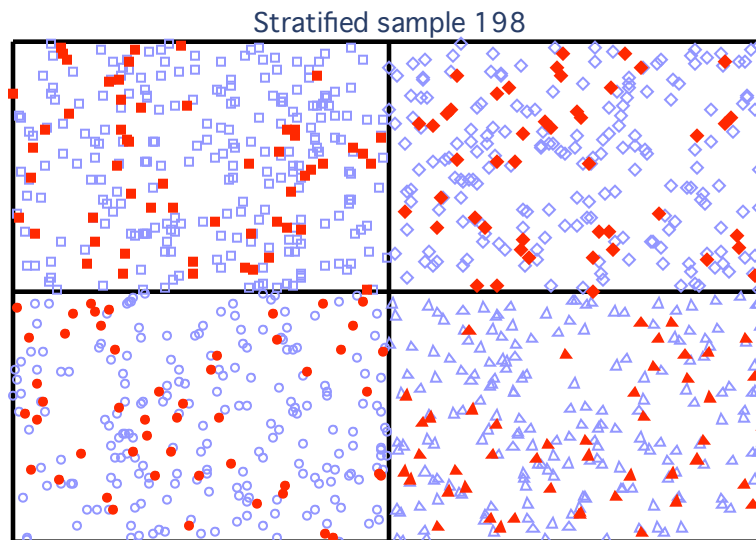
The following shows a 20% SRS. The solid dots identify sampled individuals. Together with the open dots, they represent the hypothetical population.



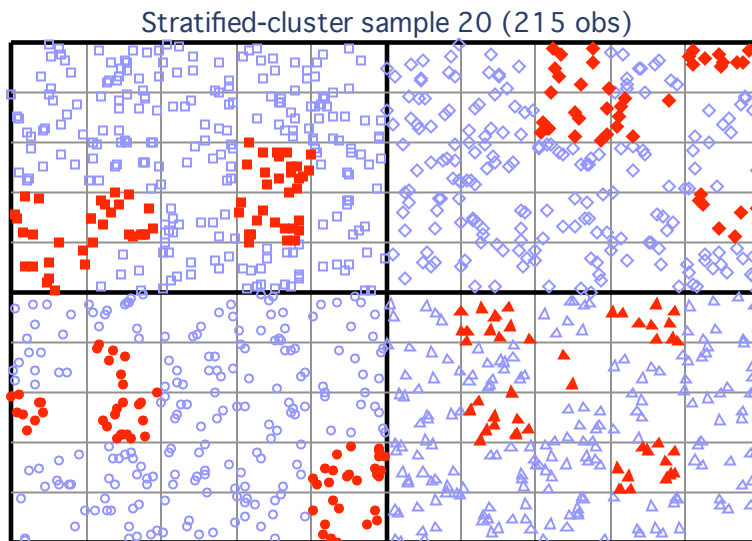
Here we partition the population into small blocks then sample 20% of the blocks. Not all blocks contain the same number of individuals, so the sample size is a random quantity.



Here we partition the population into four big regions then select a 20% sample within each region. The sample size is not exactly 20% of the population size because of unbalanced regions and rounding.



Here we reestablish the smaller blocks within the four regions then sample 20% of the blocks within each region.



2.2 Multistage designs

Let's use an example to introduce and motivate multistage designs:

► Example

Purpose

Study the smoking habits of teenagers in the U.S.

Survey design

1. Use state for strata, and counties are the PSUs.
2. The second-stage units are high schools, randomly selected within each sampled county.
3. Stratifying on gender, the final-stage units are high school seniors, randomly selected within each sampled high school.

◀

Multistage syntax

```
svyset psu [weight] [, strata(varname) fpc(varname)]  
    [|| ssu [, strata(varname) fpc(varname)]]  
    [|| ssu [, strata(varname) fpc(varname)]] ...
```

- Stages are delimited by “||”.
- SSU stands for secondary or subsequent sampling units.
- FPC is required at stage s for stage $s + 1$ to play a role in the linearized variance estimator.

□ Note

svyset will note that it is disregarding subsequent stages when an FPC is not specified for a given stage.

□

Multiple stages of cluster sampling

1. PSUs are independently selected within each stratum.
 2. Second-stage units are independently selected within each sampled PSU.
 3. Third-stage units are independently selected within each sampled second-stage unit.
- Sampling units are independently selected within each sampled SSU.
 - Stratification is also allowed at each sampling stage.

▷ Example: **svyset** for a multistage design

High school senior data

1. Counties are randomly selected within each state.
2. High schools are randomly selected within each sampled county.
3. Female and male seniors are randomly selected within each sampled high school.

```

. webuse seniors
. svyset county [pw=sampwgt], strata(state) fpc(ncounties)
  || school, fpc(nschools)
  || _n, strata(gender) fpc(nseniors)
    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
Strata 3: gender
  SU 3: <observations>
  FPC 3: nseniors

```

FPC variables

- `ncounties` is the number of counties within each category of `state`.
- `nschools` is the number of high schools within `state county`.
- `nseniors` is the number of high school seniors within `state county school sex`.

◀

2.3 Poststratification

Poststratification

A method for adjusting sampling weights, usually to account for underrepresented groups in the population. Poststratification:

- adjusts weights to sum to the poststratum sizes in the population,
- reduces bias due to nonresponse and underrepresented groups, and
- can result in smaller variance estimates.

Syntax

```
svyset ... poststrata(varname) postweight(varname)
```

□ Note

Recall that I said it is usually not very practical to stratify on demographic information such as age group, gender, and ethnicity. However, we can usually poststratify on these variables using the frequency distribution information available from census data.

□

► Example: **svyset** for poststratification

A veterinarian has 1300 clients—450 cats and 850 dogs. He would like to estimate the average annual expenses of his clientele, but he only has enough time to gather information on 50 randomly selected clients. Thus we have an SRS design; the sampling weight is $26 = 1300/50$.

The dog clients are (on average) twice as expensive as cat clients. We can use the above frequency distribution of dogs and cats to poststratify on animal type.

Cat and dog data from Levy and Lemeshow (1999)

```
. webuse poststrata
. bysort type: sum totexp
```

```
-> type = dog
```

Variable	Obs	Mean	Std. Dev.	Min	Max
totexp	32	49.85844	8.376695	32.78	66.2

```
-> type = cat
```

Variable	Obs	Mean	Std. Dev.	Min	Max
totexp	18	21.71111	8.660666	7.14	39.88

Here are the mean estimates with poststratification:

```
. svyset [pw=weight], poststrata(type) postweight(postwgt) fpc(fpc)
      pweight: weight
      VCE: linearized
      Poststrata: type
      Postweight: postwgt
      Single unit: missing
      Strata 1: <one>
      SU 1: <observations>
      FPC 1: fpc
```

```
. svy: mean totexp
(running mean on estimation sample)
```

Survey: Mean estimation

Number of strata =	1	Number of obs =	50
Number of PSUs =	50	Population size =	1300
N. of poststrata =	2	Design df =	49

	Mean	Linearized Std. Err.	[95% Conf. Interval]	
totexp	40.11513	1.163498	37.77699	42.45327

Here are the mean estimates without poststratification:

```
. svyset _n [pw=weight]
      pweight: weight
      VCE: linearized
      Single unit: missing
      Strata 1: <one>
      SU 1: <observations>
      FPC 1: <zero>
. svy: mean totexp
(running mean on estimation sample)
Survey: Mean estimation
Number of strata =      1      Number of obs   =      50
Number of PSUs   =     50      Population size =    1300
                        Design df   =      49
```

	Mean	Linearized Std. Err.	[95% Conf. Interval]	
totexp	39.7254	2.265746	35.17221	44.27859

4

2.4 Strata with a single sampling unit

How do we get stuck with strata that have only one sampling unit?

- Missing data can cause entire sampling units to be dropped from the analysis, possibly leaving a single sampling unit in the estimation sample.
- There are certainty units.
- The design is just bad.

Big problem for variance estimation

- Consider a sample with only one observation.
- **svy** reports missing standard-error estimates by default.

Finding these lonely sampling units

Use **svydescribe**

- to describe the strata and sampling units and
- to help find strata with a single sampling unit.

► Example: **svydescribe**

The Second National Health and Nutrition Examination Survey (NHANES2) data have 31 strata, each stratum contains 2 PSUs.

— Second National Health and Nutrition Examination Survey —

```
. webuse nhanes2
. svydescribe
Survey: Describing stage 1 sampling units
      pweight: finalwgt
          VCE: linearized
Single unit: missing
  Strata 1: strata
    SU 1: psu
    FPC 1: <zero>
```

Stratum	#Units	#Obs	#Obs per Unit		
			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

Some variables in this dataset have enough missing values to exhibit the lonely PSU problem:

```

—— Mean high density lipids (mg/dL) —————
. svy: mean hresult
(running mean on estimation sample)
Survey: Mean estimation
Number of strata =      31      Number of obs   =      8720
Number of PSUs  =      60      Population size = 98725345
                                   Design df      =       29

```

	Mean	Linearized Std. Err.	[95% Conf. Interval]
hresult	49.67141	.	.

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

Use **if e(sample)** after estimation commands to restrict **svydes**'s focus on the estimation sample. The **single** option will further restrict output to strata with one sampling unit:

```

—— Restrict to the estimation sample —————
. svydescribe if e(sample), single
Survey: Describing strata with a single sampling unit in stage 1
      pweight: finalwgt
           VCE: linearized
Single unit: missing
   Strata 1: strata
      SU 1: psu
     FPC 1: <zero>

```

Stratum	#Units	#Obs	#Obs per Unit		
			min	mean	max
1	1*	114	114	114.0	114
2	1*	98	98	98.0	98

2

Specifying variable names with **svydes** will result in more information about missing values:

```

—— Specifying variables for more information —————
. svydescribe hresult, single
Survey: Describing strata with a single sampling unit in stage 1
      pweight: finalwgt
           VCE: linearized
Single unit: missing
   Strata 1: strata
      SU 1: psu
     FPC 1: <zero>

```

Stratum	#Units included	#Units omitted	#Obs with complete data	#Obs with missing data	#Obs per included Unit		
					min	mean	max
1	1*	1	114	266	114	114.0	114
2	1*	1	98	87	98	98.0	98

2

4

Handling lonely sampling units

- Drop them from the estimation sample.
- **svyset** one of the ad hoc adjustments in the **singleunit()** option.
- Somehow combine them with other strata.

2.5 Certainty units

- Sampling units that are guaranteed to be chosen by the design.
- Certainty units are handled by treating each one as its own stratum with an FPC of 1.

3 Variance estimation

Stata has five variance estimation methods for survey data:

- linearization
- balanced repeated replication (BRR)
- jackknife
- bootstrap
- successive difference replication (SDR)

□ Notes

- Linearization
 - Stata's **vce(robust)** for complex data
 - The default variance estimation method for **svy**.
- Replication methods
 - Motivation
 - * Linearization can have poor performance in datasets with a small number of sampling units.
 - * Because of privacy concerns, data providers are reluctant to release strata and sampling unit information in public-use data. Thus some datasets now come packaged with weight variables for use with replication methods.
 - Concept
 - * Think of a replicate as a copy of the point estimates.
 - * The idea is to resample the data, while computing replicates from each resample, then to use the replicates to estimate the variance.

□

3.1 Linearization

Linearization

A method for deriving a variance estimator using a first order Taylor approximation of the point estimator of interest.

- Foundation: Variance of the total estimator

Syntax

```
svyset ... [vce(linearized) ]
```

- Delta method
- Huber/White/robust/sandwich estimator

Total estimator—Stratified two-stage design

- y_{hijk} is the observed value from a sampled individual.
- strata: $h = 1, \dots, L$
- PSU: $i = 1, \dots, n_h$
- SSU: $j = 1, \dots, m_{hi}$
- individual: $k = 1, \dots, m_{hij}$

$$\begin{aligned}\hat{Y} &= \sum w_{hijk} y_{hijk} \\ \hat{V}(\hat{Y}) &= \sum_h (1 - f_h) \frac{n_h}{n_h - 1} \sum_i (y_{hi} - \bar{y}_h)^2 + \\ &\quad \sum_h f_h \sum_i (1 - f_{hi}) \frac{m_{hi}}{m_{hi} - 1} \sum_j (y_{hij} - \bar{y}_{hi})^2\end{aligned}$$

- f_h is the sampling fraction for stratum h in the first stage.
- f_{hi} denotes a sampling fraction in the second stage.
- Remember that the design degrees of freedom is

$$\text{df} = N_{\text{PSU}} - N_{\text{strata}}$$

► Example: **svy: total**

Let's use our (imaginary) survey data on high school seniors to estimate the number of smokers in the population:

```
. webuse seniors
. svyset
    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
  Strata 3: gender
    SU 3: <observations>
    FPC 3: nseniors
```

— Estimate number of seniors who have smoked —

```
. svy: total smoked
(running total on estimation sample)
```

Survey: Total estimation

```
Number of strata =      50      Number of obs   =      10559
Number of PSUs   =     100      Population size = 20992929
                                   Design df      =         50
```

	Total	Linearized Std. Err.	[95% Conf. Interval]	
smoked	8347260	331155.1	7682115	9012404

— Use first stage without FPC —

```
. svyset county [pw=sampwgt], strata(state)
```

```
    pweight: sampwgt
      VCE: linearized
Single unit: missing
  Strata 1: state
    SU 1: county
    FPC 1: <zero>
```

```
. svy: total smoked
(running total on estimation sample)
```

Survey: Total estimation

```
Number of strata =      50      Number of obs   =      10559
Number of PSUs   =     100      Population size = 20992929
                                   Design df      =         50
```

	Total	Linearized Std. Err.	[95% Conf. Interval]	
smoked	8347260	346853.4	7650584	9043935

◀

Linearized variance for regression models

- The model is fit using estimating equations.
- $\hat{G}()$ is a total estimator; use Taylor expansion to get $\hat{V}(\hat{\beta})$:

$$\hat{G}(\beta) = \sum_j w_j s_j \mathbf{x}_j = \mathbf{0}$$

$$\hat{V}(\hat{\beta}) = D \hat{V}^{-1} \{ \hat{G}(\beta) \} |_{\beta=\hat{\beta}} D'$$

ML models

- $\hat{G}()$ is the gradient.
- s_j is an equation-level score.
- D is the inverse negative Hessian matrix at the solution.

Least-squares regression

- $\hat{G}()$ is the normal equations.
- s_j is a residual.
- D is the inverse of the weighted outer product of the predictors including the intercept

$$D = (\mathbf{X}' \mathbf{W} \mathbf{X})^{-1}$$

► Example: **svy: logit**

Here is an example of a logistic regression, that models the incidence of high blood pressure as a function of some demographic variables:

```

----- Second National Health and Nutrition Examination Survey -----
. webuse nhanes2
. svyset
    pweight: finalwgt
      VCE: linearized
    Single unit: missing
    Strata 1: strata
      SU 1: psu
      FPC 1: <zero>
----- Model high blood pressure on some demographics -----
. describe highbp height weight age female race

```

variable name	storage type	display format	value label	variable label
highbp	byte	%8.0g		1 if BP > 140/90, 0 otherwise
height	float	%9.0g		height (cm)
weight	float	%9.0g		weight (kg)
age	byte	%9.0g		age in years
female	byte	%8.0g		1=female, 0=male
race	byte	%9.0g	race	1=white, 2=black, 3=other

```

. svy: logit highbp height weight c.age##c.age i.female i.race, baselevel
(running logit on estimation sample)
Survey: Logistic regression
Number of strata   =          31      Number of obs       =       10351
Number of PSUs    =          62      Population size      =    117157513
                                   Design df          =           31
                                   F(   7,      25)       =       72.33
                                   Prob > F             =       0.0000

```

highbp	Coef.	Linearized Std. Err.	t	P> t	[95% Conf. Interval]	
height	-.0322128	.0057975	-5.56	0.000	-.0440368	-.0203888
weight	.0491464	.0031926	15.39	0.000	.042635	.0556578
age	.1540661	.0208216	7.40	0.000	.1116003	.196532
c.age#c.age	-.0010731	.000201	-5.34	0.000	-.0014829	-.0006632
female						
0	0	(base)				
1	-.3502998	.0861874	-4.06	0.000	-.5260801	-.1745194
race						
1	0	(base)				
2	.3461358	.1414863	2.45	0.020	.0575726	.634699
3	.1506854	.4349656	0.35	0.731	-.7364327	1.037804
_cons	-4.974867	1.168757	-4.26	0.000	-7.358563	-2.591172

◀

3.2 Balanced repeated replication (BRR)

Balanced repeated replication

For designs with two PSUs in each of L strata.

- Compute replicates by dropping a PSU from each stratum.
- Find a balanced subset of the 2^L replicates. $L \leq r < L + 4$
- The replicates are used to estimate the variance.

Syntax

```
svyset ... vce(brr) [mse]
```

□ Note

- The idea is to resample the data, compute replicates from each resample, and then use the replicates to estimate the variance.
- Balance here means that stratum specific contributions to the variance cancel out. In other words, no stratum contributes more to the variance than any other.
- We can find a balanced subset by finding a Hadamard matrix of order r .
- When the dataset contains replicate weight variables, you do not need to worry about Hadamard matrices.

□

For completeness, here is how the sampling weights are adjusted to produce BRR replicate weights.

BRR replicate weights

- w_j is the sampling weight for individual j in the first PSU of stratum h .
- H_r is a Hadamard matrix for r replications; $H_r' H_r = rI$.
- Fay's adjustment is f ; $f = 0$ by default.

The adjusted sampling weight for the i th replicate is

$$w_j^* = \begin{cases} fw_j, & \text{if } H_r[i, h] = -1 \\ (2 - f)w_j, & \text{if } H_r[i, h] = +1 \end{cases}$$

□ Note

- These replicate weights are used to produce copies of the point estimates; hence replicates of the point estimates. The replicates are then used to estimate the variance.
- **svy brr** can employ replicate-weight variables in the dataset if you **svyset** them. Otherwise, **svy brr** will automatically adjust the sampling weights to produce the replicates; however, a Hadamard matrix must be specified.

□

BRR variance formulas

- $\hat{\theta}$ denotes the point estimates.
- $\hat{\theta}_{(i)}$ denotes the i th replicate of the point estimates.
- $\bar{\theta}_{(.)}$ denotes the average of the replicates.

Default variance formula:

$$\hat{V}(\hat{\theta}) = \frac{1}{r(1-f)^2} \sum_{i=1}^r \{\hat{\theta}_{(i)} - \bar{\theta}_{(.)}\} \{\hat{\theta}_{(i)} - \bar{\theta}_{(.)}\}'$$

Mean squared error (MSE) formula:

$$\hat{V}(\hat{\theta}) = \frac{1}{r(1-f)^2} \sum_{i=1}^r \{\hat{\theta}_{(i)} - \hat{\theta}\} \{\hat{\theta}_{(i)} - \hat{\theta}\}'$$

□ Note

- The default variance formula uses deviations of the replicates from their mean.
- The MSE formula uses deviations of the replicates from the point estimates.
- `BRR *` is clickable; it takes you to a short help file informing you that you used the MSE formula for BRR variance estimation.

□

► Example: **svy brr: logit**

Let's revisit the previous logistic model fit, but use BRR for variance estimation:

```

----- Second National Health and Nutrition Examination Survey -----
. webuse nhanes2brr
. svyset [pw=finalwgt], vce(brr) mse brrweight(brr_*)
    pweight: finalwgt
      VCE: brr
      MSE: on
    brrweight: brr_1 brr_2 brr_3 brr_4 brr_5 brr_6 brr_7 brr_8 brr_9 brr_10
              brr_11 brr_12 brr_13 brr_14 brr_15 brr_16 brr_17 brr_18 brr_19
              brr_20 brr_21 brr_22 brr_23 brr_24 brr_25 brr_26 brr_27 brr_28
              brr_29 brr_30 brr_31 brr_32
    Single unit: missing
      Strata 1: <one>
        SU 1: <observations>
        FPC 1: <zero>
. svy: logit highbp height weight c.age##c.age i.female i.race
(running logit on estimation sample)
BRR replications (32)
-----|-----|-----|-----|-----|-----|
      1      2      3      4      5
.....
Survey: Logistic regression
                                     Number of obs   =      10351
                                     Population size   =  117157513
                                     Replications       =         32
                                     Design df          =         31
                                     F(   7,      25)     =       68.79
                                     Prob > F           =       0.0000

```

highbp	Coef.	BRR * Std. Err.	t	P> t	[95% Conf. Interval]	
height	-.0322128	.0058938	-5.47	0.000	-.0442333	-.0201922
weight	.0491464	.0032099	15.31	0.000	.0425997	.0556931
age	.1540661	.020759	7.42	0.000	.1117279	.1964044
c.age#c.age	-.0010731	.0002	-5.37	0.000	-.0014809	-.0006652
1.female	-.3502998	.0876341	-4.00	0.000	-.5290307	-.1715689
race						
2	.3461358	.145253	2.38	0.023	.0498903	.6423814
3	.1506854	.5561909	0.27	0.788	-.9836733	1.285044
_cons	-4.974867	1.17038	-4.25	0.000	-7.361872	-2.587863

◀

3.3 Jackknife

The jackknife

A replication method for variance estimation. It is not restricted to a specific survey design.

- Delete-1 jackknife: drop 1 PSU
- Delete- k jackknife: drop k PSUs within a stratum

Syntax

```
svyset ... vce(jackknife) [mse]
```

□ Note

- **svy jackknife** can employ replicate-weight variables in the dataset if you **svyset** them. Otherwise, **svy jackknife** will automatically adjust the sampling weights to produce the replicates using the delete-1 jackknife methodology.
- In the delete-1 jackknife, each PSU is represented by a corresponding replicate.
- The delete- k jackknife is only supported if you already have the corresponding replicate-weight variables for **svyset**. □

For completeness, here is how the sampling weights are adjusted to produce the jackknife replicate weights.

Delete-1 jackknife replicate weights

- w_{hij} is the sampling weight for individual j in PSU i of stratum h .
- Drop PSU i^* from stratum h^* .
- There are n_{h^*} replicates from stratum h^* .

The adjusted sampling weight is

$$w_{hij}^* = \begin{cases} 0 & , \text{ if } h = h^* \text{ and } i = i^* \\ \frac{n_h}{n_h - 1} w_{hij} & , \text{ if } h = h^* \text{ and } i \neq i^* \\ w_{hij} & , \text{ otherwise} \end{cases}$$

Delete- k jackknife replicate weights

- w_{hij} is the sampling weight for individual j in PSU i of stratum h .
- Drop k PSUs from stratum h^* .
- There are $c_{h^*} = \binom{n_{h^*}}{k}$ replicates from stratum h^* .

The adjusted sampling weight is

$$w_{hij}^* = \begin{cases} 0 & , \text{ if } h = h^* \text{ and } i \text{ is dropped} \\ \frac{n_h}{n_h - k} w_{hij} & , \text{ if } h = h^* \text{ and } i \text{ is not dropped} \\ w_{hij} & , \text{ otherwise} \end{cases}$$

Jackknife variance formulas

- $\hat{\theta}_{(h,i)}$ is a replicate of the point estimates from stratum h , PSU i .
- $\bar{\theta}_h$ is the average of the replicates from stratum h .
- $m_h = (n_h - 1)/n_h$ is the delete-1 multiplier for stratum h .
- $m_h = (n_h - k)/c_h k$ is the delete- k multiplier for stratum h .

Default variance formula:

$$\hat{V}(\hat{\theta}) = \sum_{h=1}^L (1 - f_h) m_h \sum_{i=1}^{n_h} \{\hat{\theta}_{(h,i)} - \bar{\theta}_h\} \{\hat{\theta}_{(h,i)} - \bar{\theta}_h\}'$$

Mean squared error (MSE) formula:

$$\hat{V}(\hat{\theta}) = \sum_{h=1}^L (1 - f_h) m_h \sum_{i=1}^{n_h} \{\hat{\theta}_{(h,i)} - \hat{\theta}\} \{\hat{\theta}_{(h,i)} - \hat{\theta}\}'$$

□ Notes

- The default variance formula uses deviations of the replicates from their mean.
- The MSE formula uses deviations of the replicates from the point estimates.
- **Jknife *** is clickable; it takes you to a short help file informing you that you used the MSE formula for jackknife variance estimation.
- Make sure to specify the correct multiplier when you **svyset** jackknife replicate weight variables.

□

▷ Example: **svy jackknife: logit**

Here we are again with our now familiar logistic model fit, in which we use the delete-1 jackknife variance estimator:

```

—— Second National Health and Nutrition Examination Survey ———
. webuse nhanes2
. svyset
    pweight: finalwgt
      VCE: linearized
Single unit: missing
  Strata 1: strata
    SU 1: psu
    FPC 1: <zero>
. svy jackknife, mse: logit highbp height weight c.age##c.age i.female i.race
(running logit 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    =  117157513
                                Replications      =        62
                                Design df         =        31
                                F( 7, 25)         =    72.21
                                Prob > F          =    0.0000

```

highbp	Coef.	Jknife * Std. Err.	t	P> t	[95% Conf. Interval]	
height	-.0322128	.0058034	-5.55	0.000	-.044049	-.0203766
weight	.0491464	.0031957	15.38	0.000	.0426286	.0556642
age	.1540661	.0208246	7.40	0.000	.1115941	.1965382
c.age#c.age	-.0010731	.0002007	-5.35	0.000	-.0014823	-.0006638
1.female	-.3502998	.0862108	-4.06	0.000	-.5261279	-.1744716
race						
2	.3461358	.1421962	2.43	0.021	.0561247	.636147
3	.1506854	.5415594	0.28	0.783	-.9538323	1.255203
_cons	-4.974867	1.171829	-4.25	0.000	-7.364828	-2.584907

◀

3.4 Bootstrap

The bootstrap

Even less restrictive on the design and parameters than the delete-1 jackknife.

- Resample the observed data by adjusting the sampling weights.
- **svy bootstrap**: requires replicate-weight variables.

Syntax

```
svyset ... vce(bootstrap) bsrweight(varlist) [bsn(#) mse]
```

Bootstrap variance formulas

- $\hat{\theta}$ denotes point estimates.
- $\hat{\theta}_{(i)}$ is the i th replicate of the point estimates.
- $\bar{\theta}_{(.)}$ is the average of the replicates.
- b is the number of bootstrap samples used to generate each replicate weight variable, default is **bsn(1)**.

Default variance formula:

$$\hat{V}(\hat{\theta}) = \frac{b}{r} \sum_{i=1}^r \{\hat{\theta}_{(i)} - \bar{\theta}_{(.)}\} \{\hat{\theta}_{(i)} - \bar{\theta}_{(.)}\}'$$

Mean squared error (MSE) formula:

$$\hat{V}(\hat{\theta}) = \frac{b}{r} \sum_{i=1}^r \{\hat{\theta}_{(i)} - \hat{\theta}\} \{\hat{\theta}_{(i)} - \hat{\theta}\}'$$

3.5 Successive difference replication

Successive difference replication (SDR)

Replication method designed for systematic samples whose sampling units are ordered.

- Resample the observed data by adjusting the sampling weights.
- **svy sdr**: requires replicate-weight variables.

Syntax

```
svyset ... vce(sdr) sdrweight(varlist) [mse]
```

SDR variance formulas

- $\hat{\theta}$ denotes the point estimates.
- $\hat{\theta}_{(i)}$ is the i th replicate of the point estimates.
- $\bar{\theta}_{(.)}$ is the average of the replicates.
- f is the sampling fraction from the **fpc()** option.

Default variance formula:

$$\hat{V}(\hat{\theta}) = (1 - f) \frac{4}{r} \sum_{i=1}^r \{\hat{\theta}_{(i)} - \bar{\theta}_{(.)}\} \{\hat{\theta}_{(i)} - \bar{\theta}_{(.)}\}'$$

Mean squared error (MSE) formula:

$$\hat{V}(\hat{\theta}) = (1 - f) \frac{4}{r} \sum_{i=1}^r \{\hat{\theta}_{(i)} - \hat{\theta}\} \{\hat{\theta}_{(i)} - \hat{\theta}\}'$$

3.6 Replicate weights

Replicate weight variable

A variable in the dataset that contains sampling weight values that were adjusted for resampling the data.

- Typically used to protect the privacy of the survey participants.
- Eliminates the need to **svyset** the strata and PSU variables.

Syntax

```
svyset ... brrweight(varlist) [fay(#)]  
svyset ... jkrweight(varlist [, ... multiplier(#)])  
svyset ... bsrweight(varlist) [bsn(#)]  
svyset ... sdrweight(varlist)
```

4 Estimation for subpopulations

Focus on a subset of the population

- Subpopulation variance estimation
 - assumes the same survey design for subsequent data collection, and
 - requires the **subpop()** option.
- Restricted-sample variance estimation
 - assumes the identified subset for subsequent data collection,
 - ignores the fact that the sample size is a random quantity, and
 - requires using the **if** or **in** restrictions.

□ Notes

- As I mentioned earlier, variability is governed by the survey design, so our variance estimates assume the design is fixed. The **subpop()** option assumes this, too.
- If we discourage you from using **if** and **in**, why does **svy** allow them?
 - You might want to restrict your sample because of known defects in some of the variables.
 - Researchers can use **if** and **in** to conduct simulation studies by simulating survey samples from a population dataset without having to use **preserve** and **restore**.
- We can illustrate the difference between these estimators with an SRS design.

□

Total from SRS data

- Data are y_1, \dots, y_n and S is the subset of observations.

$$\delta_j(S) = \begin{cases} 1, & \text{if } j \in S \\ 0, & \text{otherwise} \end{cases}$$

- Subpopulation (or restricted-sample) total:

$$\hat{Y}_S = \sum_{j=1}^n \delta_j(S) w_j y_j$$

- Sampling weight and subpopulation size:

$$w_j = \frac{N}{n}, \quad N_S = \sum_{j=1}^n \delta_j(S) w_j = \frac{N}{n} n_S$$

Variance of a subpopulation total

Sample n without replacement from a population comprised of the N_S subpopulation values with $N - N_S$ additional zeroes.

$$\hat{V}(\hat{Y}_S) = \left(1 - \frac{n}{N}\right) \frac{n}{n-1} \sum_{j=1}^n \left\{ \delta_j(S) w_j y_j - \frac{1}{n} \hat{Y}_S \right\}^2$$

Variance of a restricted-sample total

Sample n_S without replacement from the subpopulation of N_S values.

$$\tilde{V}(\hat{Y}_S) = \left(1 - \frac{n_S}{\hat{N}_S}\right) \frac{n_S}{n_S-1} \sum_{j=1}^n \delta_j(S) \left\{ y_j - \frac{1}{n_S} \hat{Y}_S \right\}^2$$

► Example: **svy**, **subpop()**

Suppose that we want to estimate the mean birthweight for mothers with high blood pressure. The **highbp** variable (in the **nmihs** data) is an indicator for mothers with high blood pressure.

In the reported results, the subpopulation information is provided in the header. Although the restricted sample results reproduce the same mean, the standard errors differ.

```

----- National Maternal and Infant Health Survey -----
. webuse nmihs
. svyset [pw=finwgt], strata(stratan)
      pweight: finwgt
        VCE: linearized
  Single unit: missing
    Strata 1: stratan
      SU 1: <observations>
    FPC 1: <zero>
----- Focus: birthweight, mothers with high blood pressure -----
. describe birthwgt highbp

```

variable name	storage type	display format	value label	variable label
birthwgt	int	%8.0g		Birthweight in grams
highbp	byte	%8.0g	hibp	High blood pressure: 1=yes,0=no

```

. label list hibp
hibp:
      0 norm BP
      1 hi BP
----- Subpopulation estimation -----
. svy, subpop(highbp): mean birthwgt
(running mean on estimation sample)
Survey: Mean estimation
Number of strata =      6      Number of obs   =    9953
Number of PSUs  =   9953      Population size = 3898922
                               Subpop. no. obs   =     595
                               Subpop. size      = 186196.7
                               Design df         =    9947

```

	Mean	Linearized Std. Err.	[95% Conf. Interval]	
birthwgt	3202.483	33.29493	3137.218	3267.748

```

----- Restricted sample estimation -----
. svy: mean birthwgt if highbp
(running mean on estimation sample)
Survey: Mean estimation
Number of strata =      6      Number of obs   =     595
Number of PSUs  =    595      Population size = 186197
                               Design df         =     589

```

	Mean	Linearized Std. Err.	[95% Conf. Interval]	
birthwgt	3202.483	28.7201	3146.077	3258.89

4

5 Postestimation

Working with estimation results

Most standard postestimation commands support **svy** results:

- **estat**
- **estimates**
- **lincom, nlcom**
- **predict, predictnl**
- **test, testnl**
- **margins, marginsplot, contrast, pwcompare**

Survey-specific features in estat

Archer–Lemeshow goodness-of-fit

- **estat gof**

Coefficient of variation

- **estat cv**

Design and misspecification effects

- **estat effects**
- **estat lceffects**

Survey design characteristics

- **estat svyset**

Marginal effects

Predictive margins and marginal effects

- **margins**

Graph results from **margins**

- **marginsplot**

Perform ANOVA-style tests on the effects of factor variables

- **contrast**

Perform pairwise comparisons of marginal means and slopes

- **pwcompare**

► Example: Postestimation

Recall the example in our discussion of variance estimators that we were modeling the incidence of high blood pressure as a function of some demographic variables.

```

----- Second National Health and Nutrition Examination Survey -----
. webuse nhanes2
. svyset
    pweight: finalwgt
      VCE: linearized
  Single unit: missing
    Strata 1: strata
      SU 1: psu
    FPC 1: <zero>
----- Model high blood pressure on some demographics -----
. describe highbp height weight age female race

```

variable name	storage type	display format	value label	variable label
highbp	byte	%8.0g		1 if BP > 140/90, 0 otherwise
height	float	%9.0g		height (cm)
weight	float	%9.0g		weight (kg)
age	byte	%9.0g		age in years
female	byte	%8.0g		1=female, 0=male
race	byte	%9.0g	race	1=white, 2=black, 3=other

```

. svy: logit highbp height weight c.age#c.age i.female i.race, baselevel
(running logit on estimation sample)
Survey: Logistic regression
Number of strata   =          31
Number of PSUs    =          62
Number of obs     =        10351
Population size   =    117157513
Design df         =           31
F(      7,      25) =        72.33
Prob > F          =         0.0000

```

highbp	Coef.	Linearized Std. Err.	t	P> t	[95% Conf. Interval]	
height	-.0322128	.0057975	-5.56	0.000	-.0440368	-.0203888
weight	.0491464	.0031926	15.39	0.000	.042635	.0556578
age	.1540661	.0208216	7.40	0.000	.1116003	.196532
c.age#c.age	-.0010731	.000201	-5.34	0.000	-.0014829	-.0006632
female						
0	0	(base)				
1	-.3502998	.0861874	-4.06	0.000	-.5260801	-.1745194
race						
0	0	(base)				
2	.3461358	.1414863	2.45	0.020	.0575726	.634699
3	.1506854	.4349656	0.35	0.731	-.7364327	1.037804
_cons	-4.974867	1.168757	-4.26	0.000	-7.358563	-2.591172

From the Archer–Lemeshow goodness-of-fit test, we find no evidence for lack of fit.

```

— Archer-Lemeshow goodness-of-fit —
. estat gof
Logistic model for highbp, goodness-of-fit test
              F(9,23) =          1.08
              Prob > F =          0.4141

```

svy: **logit** reports its coefficients in the log-odds metric; we can use the **or** reporting option to get a table of the odds ratios, instead:

```

— Report the odds ratios —
. svy: logit, or baselevels
Survey: Logistic regression
Number of strata   =          31
Number of PSUs    =          62
Number of obs     =         10351
Population size   =       117157513
Design df         =           31
F( 7, 25)         =         72.33
Prob > F          =         0.0000

```

highbp	Odds Ratio	Linearized Std. Err.	t	P> t	[95% Conf. Interval]	
height	.9683005	.0056137	-5.56	0.000	.9569187	.9798177
weight	1.050374	.0033535	15.39	0.000	1.043557	1.057236
age	1.166568	.0242898	7.40	0.000	1.118066	1.217174
c.age#c.age	.9989275	.0002008	-5.34	0.000	.9985182	.9993371
female						
0	1	(base)				
1	.7044769	.060717	-4.06	0.000	.5909168	.8398605
race						
1	1	(base)				
2	1.413595	.2000043	2.45	0.020	1.059262	1.886454
3	1.162631	.5057044	0.35	0.731	.478819	2.82301
_cons	.0069094	.0080754	-4.26	0.000	.0006371	.0749322

Of particular interest might be the **race** variable, which is a factor variable with three coded levels:

```

— Value labels of the race variable —
. describe race

```

variable name	storage type	display format	value label	variable label
race	byte	%9.0g	race	1=white, 2=black, 3=other

```

. label list race
race:
      1 White
      2 Black
      3 Other

```

From this output we can see that the odds ratio comparing blacks with whites is clearly large and statistically significant. We can use **margins** to look at the predicted probabilities of high blood pressure to see if this represents a sizable change.

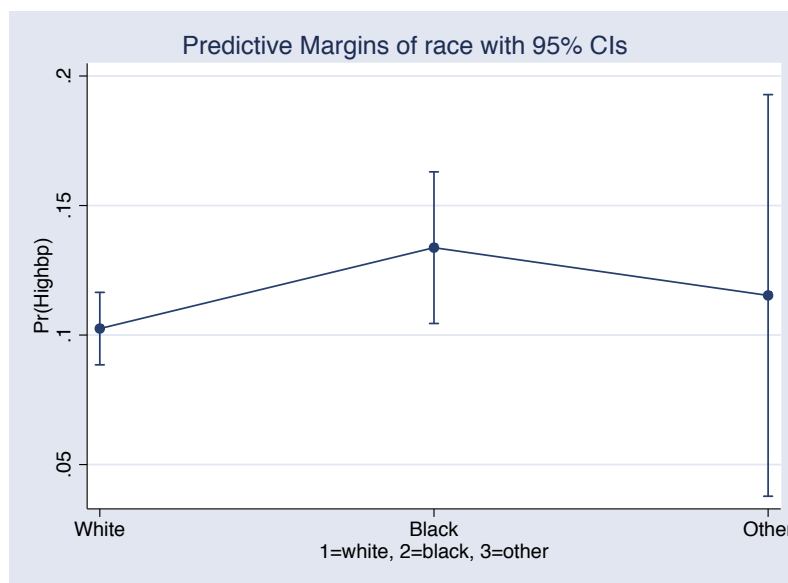
```

— Predictive margins —
. margins race, vce(unconditional)

```

Predictive margins				Number of obs	=	10351
Expression : Pr(highbp), predict()						
	Margin	Linearized Std. Err.	t	P> t	[95% Conf. Interval]	
race						
1	.1024922	.0068574	14.95	0.000	.0885065	.1164779
2	.1337316	.0143502	9.32	0.000	.1044642	.162999
3	.1152981	.0380074	3.03	0.005	.0377814	.1928148

margins computes all sorts of marginal statistics using predicted values from the currently fitted model. In this case, **margins** produces predictive margins for the probabilities of high blood pressure for each level of *race*. Computationally, predictive margins are the weighted average of the predicted values for each observation in the estimation sample. The **vce(unconditional)** option specifies that **margins** produce linearized variance estimates for each predictive margin; otherwise, the standard errors are computed using the delta method and are effectively conditional on the observed predictor (independent) variables in the model. Here is a profile plot of the predictive margins;



It is tempting to look at the overlapping confidence intervals (CI) and conclude that the difference between marginal probabilities between the first two levels of *race* is not significant at the 5% level. The problem with this comparison is that it does not account for the covariance between these two point estimates. We can use the **dydx()** option to get **margins** to compute the marginal effects of *race*:

```

— Marginal effects —
. margins, vce(unconditional) dydx(race)
Average marginal effects      Number of obs      =      10351
Expression : Pr(highbp), predict()

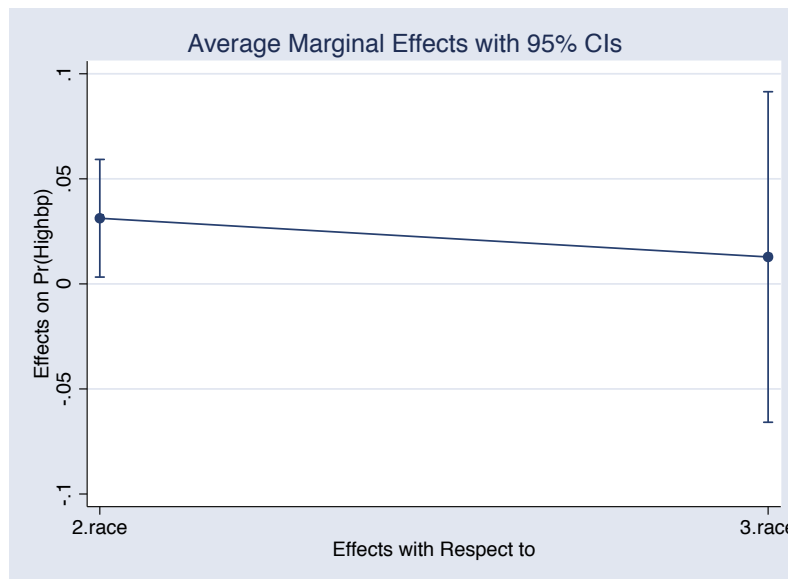
```

dy/dx w.r.t. : 2.race 3.race

	dy/dx	Linearized Std. Err.	t	P> t	[95% Conf. Interval]	
race						
2	.0312395	.0137273	2.28	0.030	.0032424	.0592366
3	.0128059	.0385697	0.33	0.742	-.0658575	.0914693

Note: dy/dx for factor levels is the discrete change from the base level.

Now we can conclude that there is a significant difference between the two marginal probabilities at the 5% level. Here is a profile plot of the marginal effects:



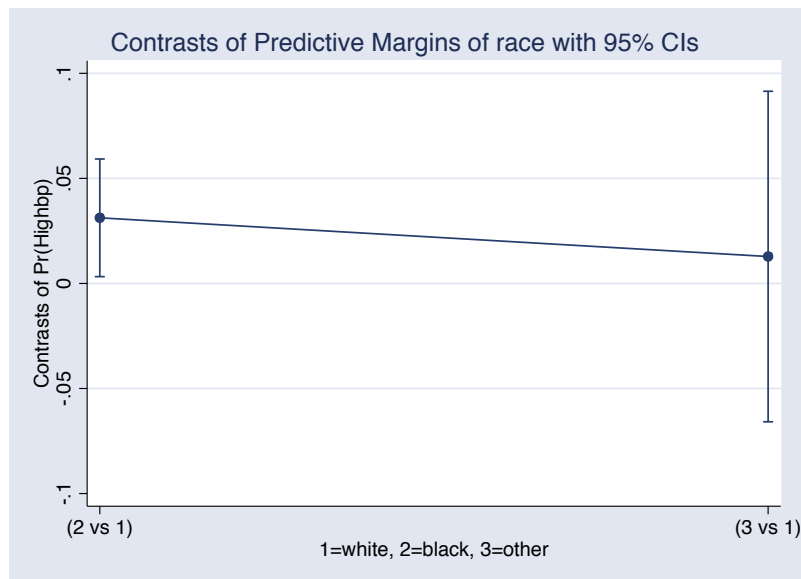
The **contrast** command and contrast operators are new in Stata 12. Also **margins** has a richer set of operators for computing discrete marginal effects. Here we use the *reference category operator* **r.** to get **margins** to compare the predictive margins at each level of **race** to the base level, **race = 1**, (white):

Marginal effects via contrasts			
. margins r.race, vce(unconditional)			
Contrasts of predictive margins			
Number of strata	=	31	Design df = 31
Number of PSUs	=	62	
Expression : Pr(highbp), predict()			
	df	F	P>F
race			
(2 vs 1)	1	5.18	0.0299
(3 vs 1)	1	0.11	0.7421
Joint	2	2.53	0.0969
Design	31		

Note: F statistics are adjusted for the survey design.

	Contrast	Linearized Std. Err.	[95% Conf. Interval]	
race				
(2 vs 1)	.0312395	.0137273	.0032424	.0592366
(3 vs 1)	.0128059	.0385697	-.0658575	.0914693

The **margins** output looks different, but the calculated marginal effects are the same. With **contrast** operators, **margins** adds a Wald table that tests each term in the margins list. The effects table also indicates which levels of the factor variable are being compared. Here is a profile plot corresponding to these marginal effects:



◀

6 Summary

1. Use **svyset** to specify the survey design for your data.
2. Use **svydes** to find strata with a single PSU.
3. Choose your variance estimation method; you can **svyset** it.
4. Use the **svy** prefix with estimation commands.
5. Use **subpop()** instead of **if** and **in**.
6. Most standard postestimation commands support **svy** results.

References

- [1] Archer, K. J. and S. Lemeshow. 2006. Goodness-of-fit test for a logistic regression model fitted using survey sample data. *Stata Journal* 6: 97–105.
- [2] Fay, R. E., and G. Train. 1995. Aspects of survey and model-based postcensal estimation of income and poverty characteristics for states and counties. *Proceedings of the Government Statistics Section, American Statistical Association*, 154–159.
- [3] Levy, P. S., and S. Lemeshow. 1999. *Sampling of Populations*. 3rd ed. New York: Wiley.
- [4] StataCorp. 2011. *Survey Data Reference Manual: Release 12*. College Station, TX: StataCorp LP.