Survey Data Analysis with Stata

Jeff Pitblado
Associate Director, Statistical Software
StataCorp LP

Bamberg, Germany 2011

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

http://www.stata.com/users/jpitblado/2011germany/

Table of contents

1 Introduction to Stata

1.1 About StataCorp

Founded in 1982 in Santa Monica, CA, under the name CRC

- Bill Gould and Finis Welch, UCLA
- Sold time on a mainframe
- Stata 1.0 released January 1985
- Gave up mainframe business in 1986

Relocated to College Station, TX in 1993

- Changed name to Stata Corporation at that time
- Later created Stata Press, a division of StataCorp

1.2 Stata software

- Pronounce it anyway you like. We say it should rhyme with "data".
- Stata is a name, not an acronym (we do not use STATA)

Available on many platforms

- Mac
- Windows
- Unix
  - Linux
  - IBM AIX
  - Oracle Solaris
- Stata 12 has been announced and will be shipping soon
- Major versions (e.g., Stata 9, Stata 10, Stata 11) are sold
- Minor versions (e.g., Stata 10.1, Stata 11.2) are free updates
- Other additions/fixes are also free updates
- Updates are done over the web

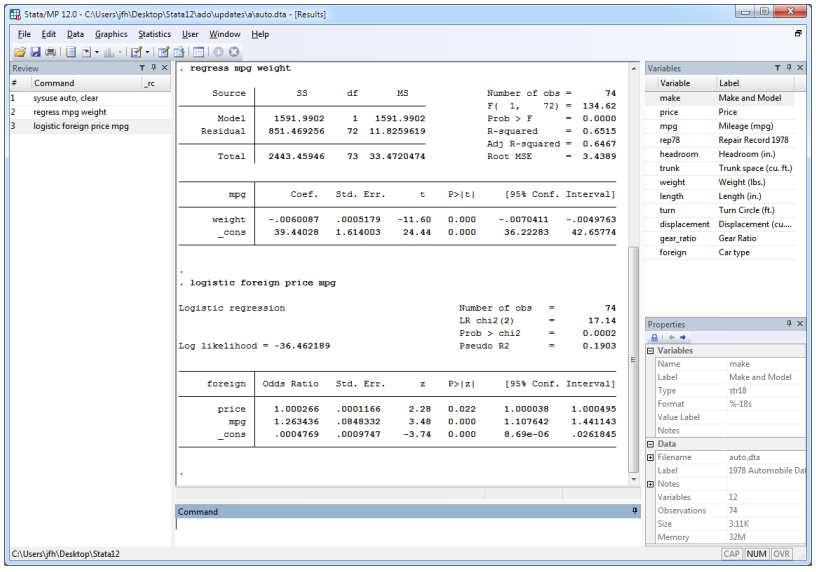
1.3 Documentation

Over 9,500 pages of documentation in Stata 12

Organized in fifteen volumes

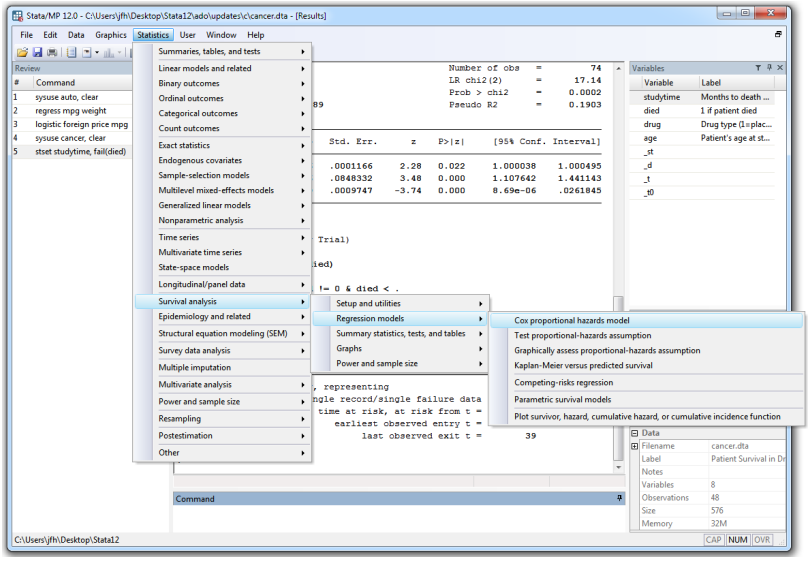
[GS] Getting Started–Mac, Unix, Windows
[U] User's Guide
[D] Data Management
[G] Graphics
[MI] Multiple Imputation
[MV] Multivariate Statistics
[R] Base Reference (4 volumes)
[ST] Survival Analysis/Epidemiological Tables
[SEM] Structural equations modeling
[SVY] Survey Data
[TS] Time Series
[XT] Panel/Longitudinal Data
[P] Programming

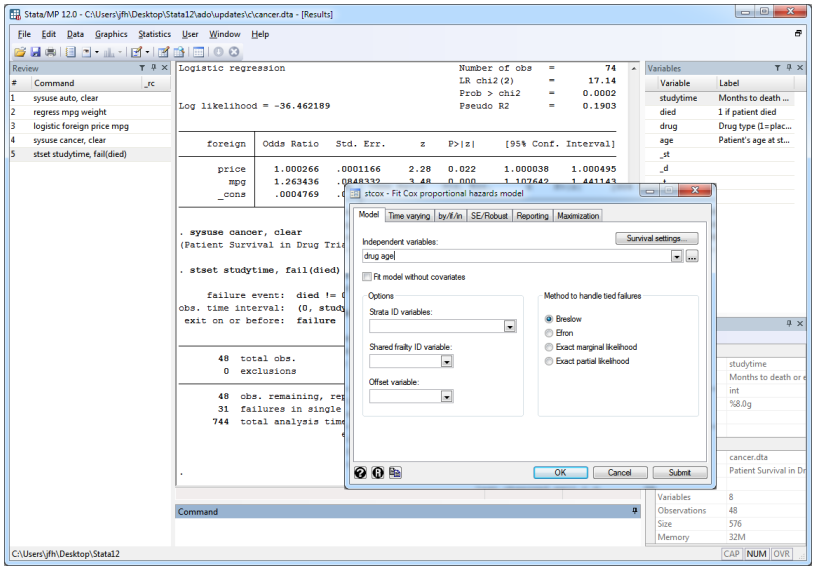
1.4 This is Stata 12



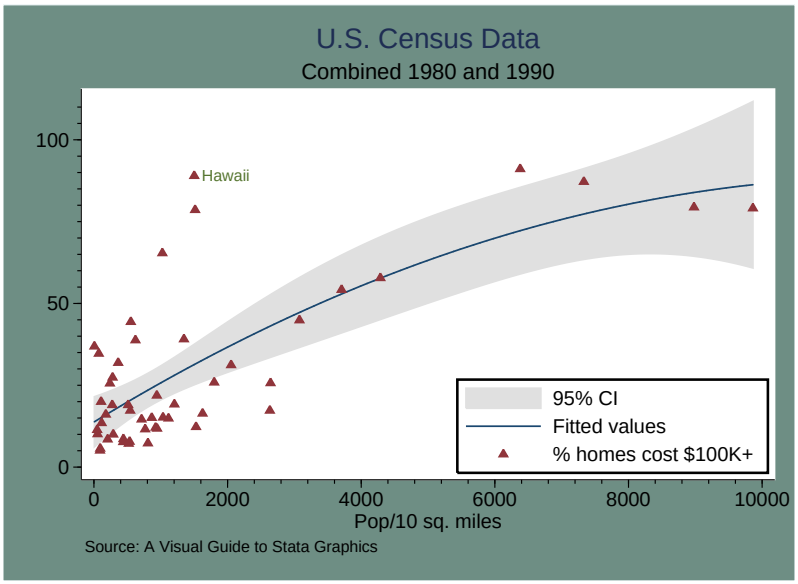
1.5 Graphical User Interface (GUI)

- You can point and click all the way
- Main menus are Data, Graphics, and Statistics
- Filling out a dialog box generates the needed command
- As such, it is a great way to learn Stata
- You can choose to work interactively by typing commands and/or using the menus
- You can also work through editable scripts of commands, known as *do-files*





1.6 Publication-quality graphics



1.7 Why use Stata for survey data?

- Stata 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

1.8 Let's get started

- First of all, let's make sure our Stata is up-to-date:

```
. update query
```

  and follow the instructions. You want to get in the habit of doing this.

- Now let's begin a log of what we are doing

```
. log using lesson1
```

  At the end of this lesson, we will close the log by typing

```
. log close
```

1.9 Stata datasets

- Stata datasets carry the `.dta` extension, and these are binary files
- The format is the same across all platforms
- To work with a dataset, it must be loaded into active memory from disk

```
. use "C:\program files\stata11\auto"
```

  or, better yet, from the web

```
. use http://www.stata-press.com/data/r11/cancer
(Patient Survival in Drug Trial)
```

- The above is so useful, we even have a shorthand for it, which we will use quite often

```
. webuse cancer
(Patient Survival in Drug Trial)
```

1.10 Getting your data into Stata

- You can use **`infile`** and **`insheet`** to import ASCII files of any complexity
- **`fdause`** and **`fdasave`** will load and save SAS transport files (`.xpt`)
- **`xmluse`** and **`xmlsave`** for `.xml` files
- Stat/Transfer by Circle Systems, Inc., is a very handy commercial program
- By far the easiest is to Copy and Paste your Excel spreadsheets into the Data Editor

1.11 Getting the help you need

- Help for any Stata command, e.g.

```
. help logit
```

- Manuals are available in PDF files, and can be accessed from the online help
- Help from the top level

```
. help contents
```

- Help from Stata and the web

```
. findit survey data

. findit violin plots
```

- **`findit`** is like Google for Stata. You can even install new software from it

We are now finished with this lesson.

```
. log close
. view lesson1.smcl
```

2 Fitting models

Start a new log for this lesson.

```
. log using lesson2
```

2.1 Overview

- We call commands that fit models *estimation commands*
- Uniformity across all models is critical:
  - syntax
  - displayed results
  - returned results
  - predictions and diagnostics
  - testing and inference
- Learn to use one, learn them all
- Survey-design aspects layer on top of the principles behind estimation

2.2 Syntax

- Estimation commands follow a standard syntax

  command *varlist* **`if`** *expression* **`in`** *range* , *options*

- *varlist* specifies the model, usually dependent variable followed by a set of regressors
- **`if`** and **`in`** determine the sample used to fit the model
- *options* are general options controlling the estimation and displaying of results, or, model-specific options such as how to handle ties in a Cox regression

2.3 Logistic Regression

Example 1. The Tower of London (Rabe-Hesketh et al. 2001)

- Study of cognitive abilities of patients with schizophrenia
- Cognitive ability was measured by successful completion of the Tower of London, a computerized task (binary variable `dtlm`)
- 226 subjects, all but one tested at three difficulty levels (variable `difficulty`)
- Subjects were not only patients (`group==3`), but relatives (`group==2`) and nonrelated controls (`group==1`)
- We can thus propose a logistic regression model for `dtlm` as, initially, a function of `difficulty`

2.4 Using logit

- Let's have a look at the dataset:

```
. webuse towerlondon, clear
(Tower of London data)
. describe
Contains data from http://localpress.stata.com/data/r12/towerlondon.dta
  obs:           677                          Tower of London data
 vars:             5                          31 May 2011 10:41
 size:         4,739                          (_dta has notes)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
family          int    %8.0g                  Family ID
subject         int    %9.0g                  Subject ID
dtlm            byte   %9.0g                  1 = task completed
difficulty      byte   %9.0g                  Level of difficulty: -1, 0, or 1
group           byte   %8.0g                  1: controls; 2: relatives; 3:
                                                schizophrenics
-------------------------------------------------------------------------------
Sorted by:  family  subject
. notes
_dta:
  1.  Source:  Rabe-Hesketh, S., R. Touloupulou, and R. M. Murray.  2001.
      Multilevel modeling of cognitive function in schizophrenics and their
      first degree relatives.  Multivariate Behavioral Research 36:  279-298.
. codebook
-------------------------------------------------------------------------------
family                                                              Family ID
-------------------------------------------------------------------------------
                  type:  numeric (int)

                 range:  [1,125]                      units:  1
         unique values:  118                  missing .:  0/677

                  mean:   42.5258
             std. dev:   36.1426

           percentiles:        10%       25%       50%       75%       90%
                                 4        12        30        68       103
-------------------------------------------------------------------------------
```

```
subject                                                         Subject ID
-------------------------------------------------------------------------
                  type:  numeric (int)
                 range:  [1,251]                      units:  1
         unique values:  226                      missing .:  0/677
                  mean:   125.941
             std. dev:   74.2733
           percentiles:        10%       25%       50%       75%       90%
                                23        60       123       194       229
-------------------------------------------------------------------------

dtlm                                                    1 = task completed
-------------------------------------------------------------------------
                  type:  numeric (byte)
                 range:  [0,1]                        units:  1
         unique values:  2                        missing .:  0/677
            tabulation:  Freq.  Value
                           514  0
                           163  1
-------------------------------------------------------------------------

difficulty                                  Level of difficulty: -1, 0, or 1
-------------------------------------------------------------------------
                  type:  numeric (byte)
                 range:  [-1,1]                       units:  1
         unique values:  3                        missing .:  0/677
            tabulation:  Freq.  Value
                           225  -1
                           226  0
                           226  1
-------------------------------------------------------------------------

group                          1: controls; 2: relatives; 3: schizophrenics
-------------------------------------------------------------------------
                  type:  numeric (byte)
                 range:  [1,3]                        units:  1
         unique values:  3                        missing .:  0/677
            tabulation:  Freq.  Value
                           194  1
                           294  2
                           189  3
```

- Let's now fit the logistic model:

```
. logit dtlm difficulty
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -322.62356
Iteration 2:   log likelihood = -319.37224
Iteration 3:   log likelihood = -319.36323
Iteration 4:   log likelihood = -319.36323
Logistic regression                               Number of obs   =        677
                                                  LR chi2(1)      =     108.63
                                                  Prob > chi2     =     0.0000
Log likelihood = -319.36323                       Pseudo R2       =     0.1454
------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |  -1.290393   .1391505    -9.27   0.000    -1.563123   -1.017663
       _cons |  -1.422938   .1131947   -12.57   0.000    -1.644795    -1.20108
------------------------------------------------------------------------------
```

- How about odds ratios and 90% CIs?

```
. logit, or level(90)
Logistic regression                                     Number of obs   =        677
                                                        LR chi2(1)      =     108.63
                                                        Prob > chi2     =     0.0000
Log likelihood = -319.36323                             Pseudo R2       =     0.1454

------------------------------------------------------------------------------
        dtlm | Odds Ratio   Std. Err.      z    P>|z|     [90% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |   .2751627    .038289    -9.27   0.000     .2188705    .3459328
       _cons |   .2410049   .0272805   -12.57   0.000     .2000623    .2903265
------------------------------------------------------------------------------
```

2.5 Categorical variables

- Variable `group` has three levels, and the whole point of the study is to assess the affect of this factor

```
. logit dtlm difficulty i.group
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -317.84501
Iteration 2:   log likelihood = -313.90332
Iteration 3:   log likelihood = -313.89079
Iteration 4:   log likelihood = -313.89079
Logistic regression                                     Number of obs   =        677
                                                        LR chi2(3)      =     119.58
                                                        Prob > chi2     =     0.0000
Log likelihood = -313.89079                             Pseudo R2       =     0.1600

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |  -1.313382   .1409487    -9.32   0.000    -1.589636   -1.037127
             |
       group |
          2  |  -.1396641   .2282452    -0.61   0.541    -.5870164    .3076883
          3  |  -.8313329   .2742339    -3.03   0.002    -1.368822   -.2938443
             |
       _cons |  -1.160498   .1824503    -6.36   0.000    -1.518094   -.8029023
------------------------------------------------------------------------------
```

- or the interaction effects of `group` and `difficulty`

```
. gen diff2 = difficulty + 1
. logit dtlm group#diff2, or
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -312.65962
Iteration 2:   log likelihood =  -306.9542
Iteration 3:   log likelihood = -306.76917
Iteration 4:   log likelihood = -306.76827
Iteration 5:   log likelihood = -306.76827
```

(*Continued on next page*)

```
Logistic regression                                     Number of obs   =        677
                                                        LR chi2(8)      =     133.82
                                                        Prob > chi2     =     0.0000
Log likelihood = -306.76827                             Pseudo R2       =     0.1791
```

dtlm	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
group#diff2						
1 1	.191358	.0793591	-3.99	0.000	.0848869	.4313729
1 2	.191358	.0793591	-3.99	0.000	.0848869	.4313729
2 0	1.061924	.341687	0.19	0.852	.5652085	1.995161
2 1	.2113636	.0763888	-4.30	0.000	.1040874	.4292026
2 2	.0722611	.0336156	-5.65	0.000	.0290355	.1798373
3 0	.6601147	.2362156	-1.16	0.246	.3273597	1.331109
3 1	.0469697	.0301691	-4.76	0.000	.0133378	.1654064
3 2	.0307998	.0234353	-4.57	0.000	.0069323	.1368419
_cons	1.064516	.2662591	0.25	0.803	.6520004	1.738027

```
. logit dtlm group##diff2, or
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -312.65962
Iteration 2:   log likelihood =  -306.9542
Iteration 3:   log likelihood = -306.76917
Iteration 4:   log likelihood = -306.76827
Iteration 5:   log likelihood = -306.76827
Logistic regression                                     Number of obs   =        677
                                                        LR chi2(8)      =     133.82
                                                        Prob > chi2     =     0.0000
Log likelihood = -306.76827                             Pseudo R2       =     0.1791
```

dtlm	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
group						
2	1.061924	.341687	0.19	0.852	.5652085	1.995161
3	.6601147	.2362156	-1.16	0.246	.3273597	1.331109
diff2						
1	.191358	.0793591	-3.99	0.000	.0848869	.4313729
2	.191358	.0793591	-3.99	0.000	.0848869	.4313729
group#diff2						
2 1	1.040136	.5513843	0.07	0.941	.3680119	2.939807
2 2	.3556022	.2153684	-1.71	0.088	.1085024	1.165439
3 1	.3718362	.2850016	-1.29	0.197	.082781	1.670216
3 2	.243827	.2117046	-1.63	0.104	.0444652	1.337038
_cons	1.064516	.2662591	0.25	0.803	.6520004	1.738027

(*Continued on next page*)

```
. logit dtlm group#c.difficulty, or
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood =     -321.8
Iteration 2:   log likelihood = -318.46598
Iteration 3:   log likelihood = -318.45691
Iteration 4:   log likelihood = -318.45691
Logistic regression                               Number of obs   =        677
                                                  LR chi2(3)      =     110.45
                                                  Prob > chi2     =     0.0000
Log likelihood = -318.45691                       Pseudo R2       =     0.1478
```

dtlm	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
group# c.difficulty						
1	.3330375	.0772957	-4.74	0.000	.2113173	.5248693
2	.2294294	.045795	-7.38	0.000	.1551478	.3392753
3	.2965616	.0701317	-5.14	0.000	.1865608	.4714214
_cons	.2401366	.0272172	-12.59	0.000	.1923015	.2998707

- For models with a single factor with a small number of levels, you might prefer dummy variables to abstract codings

```
. describe
Contains data from http://localpress.stata.com/data/r12/towerlondon.dta
  obs:           677                          Tower of London data
 vars:             6                          31 May 2011 10:41
 size:         7,447                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
family	int	%8.0g		Family ID
subject	int	%9.0g		Subject ID
dtlm	byte	%9.0g		1 = task completed
difficulty	byte	%9.0g		Level of difficulty: -1, 0, or 1
group	byte	%8.0g		1: controls; 2: relatives; 3: schizophrenics
diff2	float	%9.0g		

```
Sorted by:  family  subject
     Note:  dataset has changed since last saved
. gen relative = group == 2
. gen sphrenic = group == 3
. logit dtlm difficulty relative sphrenic, or
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -317.84501
Iteration 2:   log likelihood = -313.90332
Iteration 3:   log likelihood = -313.89079
Iteration 4:   log likelihood = -313.89079
```

(*Continued on next page*)

```
Logistic regression                               Number of obs   =        677
                                                  LR chi2(3)      =     119.58
                                                  Prob > chi2     =     0.0000
Log likelihood = -313.89079                       Pseudo R2       =     0.1600
```

dtlm	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
difficulty	.2689092	.0379024	-9.32	0.000	.2039998	.3544716
relative	.8696503	.1984935	-0.61	0.541	.5559836	1.360277
sphrenic	.4354685	.1194202	-3.03	0.002	.2544066	.7453925
_cons	.31333	.0571671	-6.36	0.000	.2191291	.4480268

2.6 Weights

- Stata supports four kinds of weights
  - **`fweight`** – Frequency
  - **`aweight`** – Analytic
  - **`pweight`** – Probability/Sampling
  - **`iweight`** – Importance
- Use **`pweight`** for survey data
- In our example, suppose we wanted to weight according to family size:

```
. bysort family: gen fsize = _N
. logit dtlm difficulty relative sphrenic [pw=fsize], or
Iteration 0:   log pseudolikelihood = -3561.5164
Iteration 1:   log pseudolikelihood = -2939.7205
Iteration 2:   log pseudolikelihood = -2890.7725
Iteration 3:   log pseudolikelihood = -2890.4865
Iteration 4:   log pseudolikelihood = -2890.4865
Logistic regression                               Number of obs   =        677
                                                  Wald chi2(3)    =      68.95
                                                  Prob > chi2     =     0.0000
Log pseudolikelihood = -2890.4865                 Pseudo R2       =     0.1884
```

dtlm	Odds Ratio	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
difficulty	.2287205	.0420496	-8.02	0.000	.1595197	.3279412
relative	.9626698	.2404593	-0.15	0.879	.5900122	1.570702
sphrenic	.4969365	.1507077	-2.31	0.021	.2742547	.9004256
_cons	.2962582	.0615008	-5.86	0.000	.1972277	.4450133

- Note that Stata uses Taylor linearization to estimate standard errors when **`pweight`**s are specified

2.7 Clustering

- In this example, each subject takes the Tower of London test three times, at three levels of difficulty
- There is likely intra-subject correlation to account for. You can either model it directly, or control for it the "survey" way

```
. logit dtlm difficulty relative sphrenic [pw=fsize], or ///
>       vce(cluster subject)
Iteration 0:   log pseudolikelihood = -3561.5164
Iteration 1:   log pseudolikelihood = -2939.7205
Iteration 2:   log pseudolikelihood = -2890.7725
Iteration 3:   log pseudolikelihood = -2890.4865
Iteration 4:   log pseudolikelihood = -2890.4865
Logistic regression                             Number of obs   =        677
                                                Wald chi2(3)    =      87.22
                                                Prob > chi2     =     0.0000
Log pseudolikelihood = -2890.4865               Pseudo R2       =     0.1884
                              (Std. Err. adjusted for 226 clusters in subject)
------------------------------------------------------------------------------
             |               Robust
        dtlm | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |   .2287205   .0373523    -9.03   0.000     .1660718    .3150029
    relative |   .9626698   .2780869    -0.13   0.895     .5465003     1.69576
    sphrenic |   .4969365   .1578948    -2.20   0.028     .2665896     .926315
       _cons |   .2962582   .0716047    -5.03   0.000     .1844751    .4757766
------------------------------------------------------------------------------
```

- It would probably be more appropriate to cluster on `family`, since the patients and their relatives are likely to be correlated

```
. logit dtlm difficulty relative sphrenic [pw=fsize], or ///
>       vce(cluster family)
Iteration 0:   log pseudolikelihood = -3561.5164
Iteration 1:   log pseudolikelihood = -2939.7205
Iteration 2:   log pseudolikelihood = -2890.7725
Iteration 3:   log pseudolikelihood = -2890.4865
Iteration 4:   log pseudolikelihood = -2890.4865
Logistic regression                             Number of obs   =        677
                                                Wald chi2(3)    =      60.43
                                                Prob > chi2     =     0.0000
Log pseudolikelihood = -2890.4865               Pseudo R2       =     0.1884
                              (Std. Err. adjusted for 118 clusters in family)
------------------------------------------------------------------------------
             |               Robust
        dtlm | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |   .2287205   .0442463    -7.63   0.000      .156545    .3341729
    relative |   .9626698   .2661693    -0.14   0.891     .5599226    1.655109
    sphrenic |   .4969365   .1677144    -2.07   0.038     .2564622    .9628942
       _cons |   .2962582   .0727037    -4.96   0.000     .1831387    .4792485
------------------------------------------------------------------------------
```

- Clustering on family subsumes clustering on subject.

2.8 Cataloging model fits

- Within a Stata session, you can store estimates for later use and for comparison with other models

```
. logit dtlm difficulty relative sphrenic
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -317.84501
Iteration 2:   log likelihood = -313.90332
Iteration 3:   log likelihood = -313.89079
Iteration 4:   log likelihood = -313.89079
Logistic regression                               Number of obs   =        677
                                                  LR chi2(3)      =     119.58
                                                  Prob > chi2     =     0.0000
Log likelihood = -313.89079                       Pseudo R2       =     0.1600

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |  -1.313382   .1409487    -9.32   0.000    -1.589636   -1.037127
    relative |  -.1396641   .2282452    -0.61   0.541    -.5870164    .3076883
    sphrenic |  -.8313329   .2742339    -3.03   0.002    -1.368822   -.2938443
       _cons |  -1.160498   .1824503    -6.36   0.000    -1.518094   -.8029023
------------------------------------------------------------------------------

. est store logit
. probit dtlm difficulty relative sphrenic
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -315.39848
Iteration 2:   log likelihood = -314.50212
Iteration 3:   log likelihood = -314.50121
Iteration 4:   log likelihood = -314.50121
Probit regression                                 Number of obs   =        677
                                                  LR chi2(3)      =     118.36
                                                  Prob > chi2     =     0.0000
Log likelihood = -314.50121                       Pseudo R2       =     0.1584

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |  -.7360316   .0751099    -9.80   0.000    -.8832443   -.5888189
    relative |  -.1036621   .1323102    -0.78   0.433    -.3629853    .1556612
    sphrenic |   -.515422   .1576395    -3.27   0.001    -.8243898   -.2064543
       _cons |  -.6536079   .1021229    -6.40   0.000    -.8537652   -.4534506
------------------------------------------------------------------------------

. est store probit
. est dir

-------------------------------------------------------------------------------
        name | command      depvar       npar  title
-------------+-----------------------------------------------------------------
       logit | logit        dtlm            4
      probit | probit       dtlm            4
-------------------------------------------------------------------------------

. est table _all

----------------------------------------
    Variable |    logit        probit
-------------+--------------------------
  difficulty |  -1.3133816   -.73603162
    relative |  -.13966408   -.10366205
    sphrenic |   -.8313329   -.51542205
       _cons |  -1.1604983   -.65360788
----------------------------------------
```

```
. est stats _all
```

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
logit	677	-373.6794	-313.8908	4	635.7816	653.8523
probit	677	-373.6794	-314.5012	4	637.0024	655.0731

Note: N=Obs used in calculating BIC; see **[R] BIC note**

- Stata 10 introduced the ability to also save results to disk for retrieval during a later session

```
. probit
Probit regression                                 Number of obs   =        677
                                                  LR chi2(3)      =     118.36
                                                  Prob > chi2     =     0.0000
Log likelihood = -314.50121                       Pseudo R2       =     0.1584
```

dtlm	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
difficulty	-.7360316	.0751099	-9.80	0.000	-.8832443	-.5888189
relative	-.1036621	.1323102	-0.78	0.433	-.3629853	.1556612
sphrenic	-.515422	.1576395	-3.27	0.001	-.8243898	-.2064543
_cons	-.6536079	.1021229	-6.40	0.000	-.8537652	-.4534506

```
. est save probit
file probit.ster saved
```

- The next time you start Stata, you could type

```
. est use probit
. probit
Probit regression                                 Number of obs   =        677
                                                  LR chi2(3)      =     118.36
                                                  Prob > chi2     =     0.0000
Log likelihood = -314.50121                       Pseudo R2       =     0.1584
```

dtlm	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
difficulty	-.7360316	.0751099	-9.80	0.000	-.8832443	-.5888189
relative	-.1036621	.1323102	-0.78	0.433	-.3629853	.1556612
sphrenic	-.515422	.1576395	-3.27	0.001	-.8243898	-.2064543
_cons	-.6536079	.1021229	-6.40	0.000	-.8537652	-.4534506

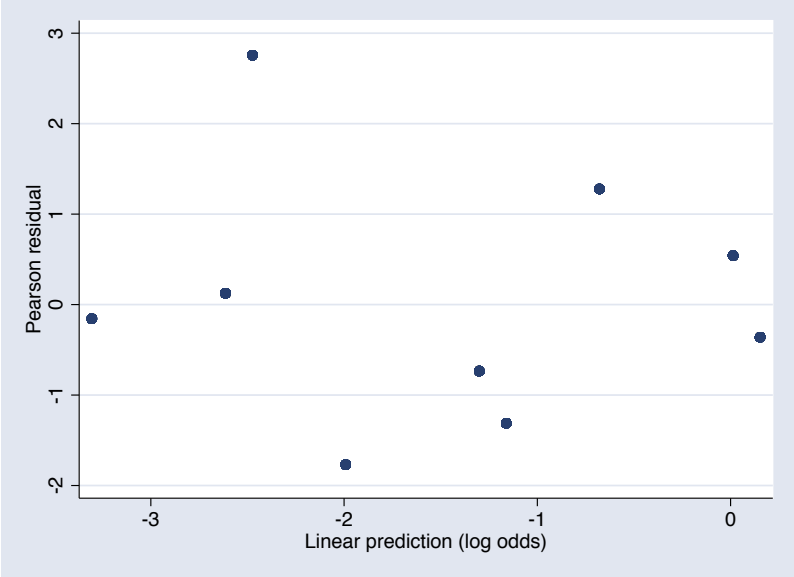
2.9 Predictions and Diagnostics

- The **`predict`** command will generate a new variable containing the prediction or diagnostic of your choice (or the default)

```
. logit dtlm difficulty relative sphrenic
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -317.84501
Iteration 2:   log likelihood = -313.90332
Iteration 3:   log likelihood = -313.89079
Iteration 4:   log likelihood = -313.89079
Logistic regression                               Number of obs   =        677
                                                  LR chi2(3)      =     119.58
                                                  Prob > chi2     =     0.0000
Log likelihood = -313.89079                       Pseudo R2       =     0.1600

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |  -1.313382   .1409487    -9.32   0.000    -1.589636   -1.037127
    relative |  -.1396641   .2282452    -0.61   0.541    -.5870164    .3076883
    sphrenic |  -.8313329   .2742339    -3.03   0.002    -1.368822   -.2938443
       _cons |  -1.160498   .1824503    -6.36   0.000    -1.518094   -.8029023
------------------------------------------------------------------------------

. predict phat
(option pr assumed; Pr(dtlm))
. list dtlm difficulty relative sphrenic phat in 1/20

     +-------------------------------------------------------+
     | dtlm   diffic~y   relative   sphrenic       phat |
     |-------------------------------------------------------|
  1. |    1         -1          0          1   .3366074 |
  2. |    0          0          0          1   .1200633 |
  3. |    0          1          0          1   .0353928 |
  4. |    0         -1          0          1   .3366074 |
  5. |    1          0          0          1   .1200633 |
     |-------------------------------------------------------|
  6. |    0          1          0          1   .0353928 |
  7. |    1         -1          0          1   .3366074 |
  8. |    0          0          0          1   .1200633 |
  9. |    0          1          0          1   .0353928 |
 10. |    0         -1          1          0   .5033048 |
     |-------------------------------------------------------|
 11. |    0          0          1          0   .2141377 |
 12. |    0          1          1          0   .0682718 |
 13. |    0         -1          1          0   .5033048 |
 14. |    0          0          1          0   .2141377 |
 15. |    0          1          1          0   .0682718 |
     |-------------------------------------------------------|
 16. |    1         -1          1          0   .5033048 |
 17. |    1          0          1          0   .2141377 |
 18. |    0          1          1          0   .0682718 |
 19. |    1         -1          1          0   .5033048 |
 20. |    0          0          1          0   .2141377 |
     +-------------------------------------------------------+
```

- For logistic regression a useful diagnostic plot is one of Pearson residuals vs. the linear predictor (the log relative odds)

```
. predict pearson, residuals
. predict xbeta, xb
. scatter pearson xbeta
```



- Predictions can be made on the estimation data, or on other data as long as the variable names are the same

2.10 Tests of hypotheses

- As part of any estimation results, you get an omnibus test for all model coefficients. This omnibus test is either a likelihood-ratio test, or a Wald test

```
. logit
Logistic regression                                     Number of obs   =        677
                                                        LR chi2(3)      =     119.58
                                                        Prob > chi2     =     0.0000
Log likelihood = -313.89079                             Pseudo R2       =     0.1600
```

dtlm	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
difficulty	-1.313382	.1409487	-9.32	0.000	-1.589636	-1.037127
relative	-.1396641	.2282452	-0.61	0.541	-.5870164	.3076883
sphrenic	-.8313329	.2742339	-3.03	0.002	-1.368822	-.2938443
_cons	-1.160498	.1824503	-6.36	0.000	-1.518094	-.8029023

- You can use **`test`** to get other Wald tests

```
. test sphrenic = -1
 ( 1)  [dtlm]sphrenic = -1
           chi2(  1) =    0.38
         Prob > chi2 =    0.5385
. test relative sphrenic
 ( 1)  [dtlm]relative = 0
 ( 2)  [dtlm]sphrenic = 0
           chi2(  2) =   10.24
         Prob > chi2 =    0.0060
```

- You can use **`lrtest`** to get other LR tests, but this requires refitting the model

```
. est store full
. logit dtlm difficulty
Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -322.62356
Iteration 2:   log likelihood = -319.37224
Iteration 3:   log likelihood = -319.36323
Iteration 4:   log likelihood = -319.36323
Logistic regression                               Number of obs   =        677
                                                  LR chi2(1)      =     108.63
                                                  Prob > chi2     =     0.0000
Log likelihood = -319.36323                       Pseudo R2       =     0.1454
------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  difficulty |  -1.290393   .1391505    -9.27   0.000    -1.563123   -1.017663
       _cons |  -1.422938   .1131947   -12.57   0.000    -1.644795    -1.20108
------------------------------------------------------------------------------
. lrtest . full
Likelihood-ratio test                                 LR chi2(2)  =     10.94
(Assumption: . nested in full)                        Prob > chi2 =    0.0042
```

- LR tests not appropriate for survey data; don't worry, Stata will tell you so

We are now finished with this lesson.

```
. log close
```

3 Types of data

3.1 Why survey data?

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

3.2 Types of data

Simple random sample (SRS) data
Observations are ”independently” sampled from a data generating process.

- Typical assumption: 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'll use this term to distinguish this data from survey data.

Correlated data
Individuals are assumed not independent.

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

What do you do about it?

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

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

Distinguishes itself from other forms of data:

- Complex nature under which individuals are sampled
- Make inferences about the fixed population
- Sample variability is attributed to the survey design

4 Aspects of survey data

Start a new log for this lesson.

```
. log using lesson4
```

4.1 Survey data characteristics

Standard data

- Estimation commands for standard data:
  - **proportion**
  - **regress**
- We'll refer to these as *standard estimation commands*.

Survey data

- Survey estimation commands are governed by the **svy** prefix.
  - **svy: proportion**
  - **svy: regress**
- **svy** requires that the data is **svyset**.

4.2 Survey in Stata is easy

- Once you get the design aspects and other preferences declared, estimation is quite easy.
- For example, to estimate proportions:

```
. webuse nhanes2, clear
. 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

```
. svyset
      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: strata
         SU 1: psu
        FPC 1: <zero>
. 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

- So really, this workshop is about declaring your design to Stata, and for that we have **`svyset`**

4.3 Some definitions

Single-stage syntax

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

- PSU – primary sampling units
- **`pweight`** – sampling weights
- Strata
- FPC – finite population correction

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.

Sampling weight
The reciprocal of the probability for 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.

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

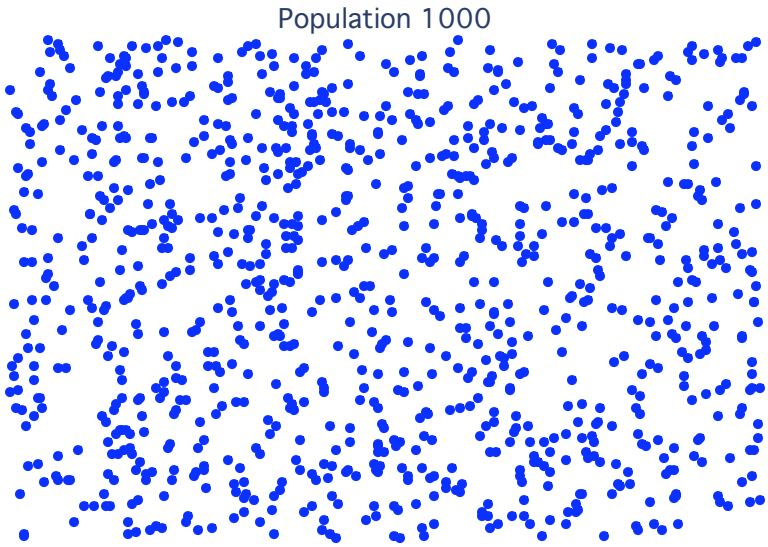
- Sampling units are independently sampled from within each stratum.
- Stratification usually results in smaller variance estimates compared to SRS.
- Although there is potential for improving efficiency by reducing sampling variability, it is usually not very practical to stratify on demographic information.

Finite population correction (FPC)
An adjustment applied to the variance due to sampling without replacement.

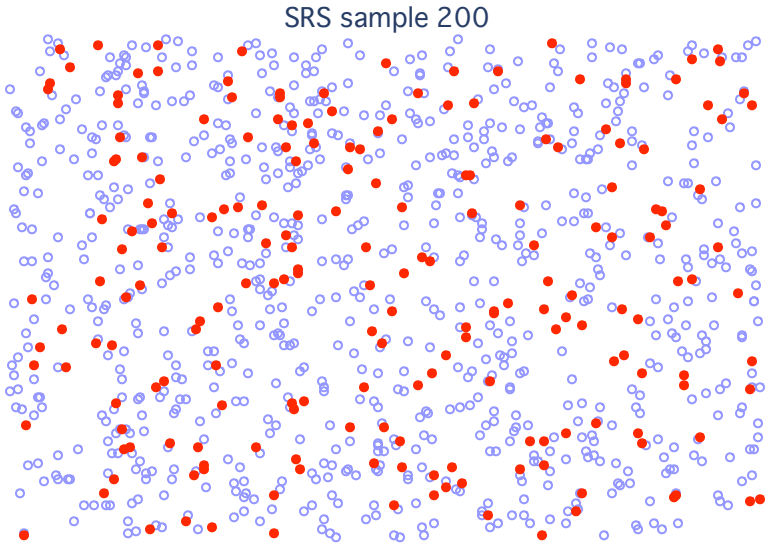
- Sampling without replacement from a finite population reduces sampling variability.

4.4 What does sampling look like?

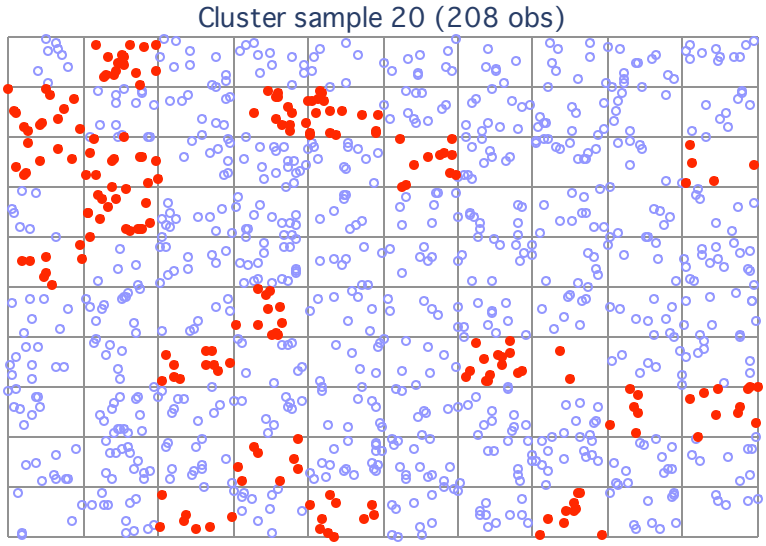
Below is a visual representation of a hypothetical population. Suppose each blue dot represents an individual.



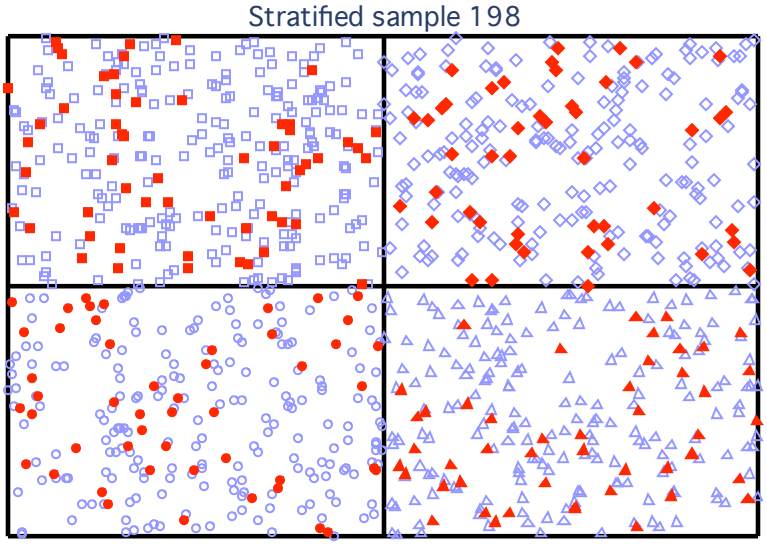
The following shows a 20% simple-random-sample. The solid symbols identify sampled individuals.



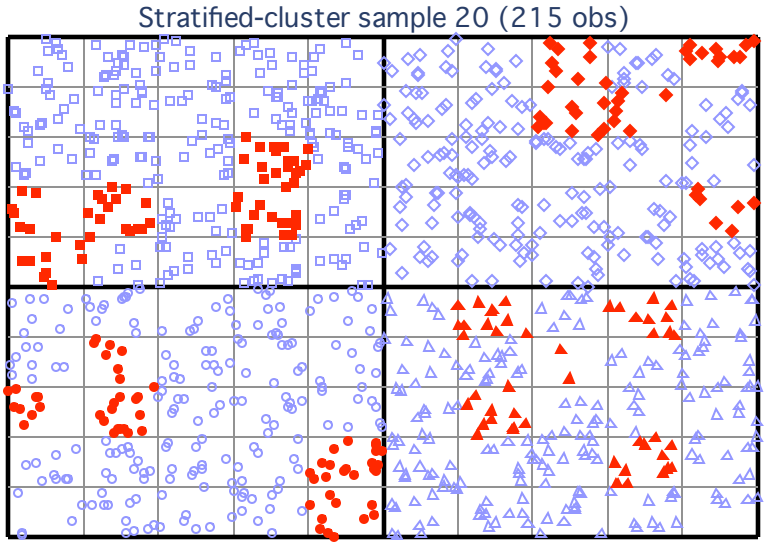
Here we partition the population into small blocks, then sample 20% of the blocks. Not all blocks contain the same number of individuals, so the sample size is a random quantity.



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



Here we re-establish the smaller blocks within the four regions, then sample 20% of the blocks within each region.



We are now finished with this lesson.

```
. log close
```

5 Using svyset

Start a new log for this lesson.

```
. log using lesson5
```

5.1 Recall the single-stage syntax

Single-stage syntax

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

- PSU – primary sampling units
- **`pweight`** – sampling weights
- Strata
- FPC – finite population correction

5.2 Some examples

- Simple Random Sample:

```
. sysuse auto, clear
(1978 Automobile Data)
. svyset _n
      pweight: <none>
          VCE: linearized
  Single unit: missing
     Strata 1: <one>
         SU 1: <observations>
        FPC 1: <zero>
. svy: regress mpg weight
(running regress on estimation sample)
Survey: Linear regression
Number of strata   =         1                  Number of obs      =        74
Number of PSUs     =        74                  Population size    =        74
                                                Design df          =        73
                                                F(   1,     73)    =    107.30
                                                Prob > F           =    0.0000
                                                R-squared          =    0.6515
------------------------------------------------------------------------------
             |             Linearized
         mpg |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |  -.0060087   .0005801   -10.36   0.000    -.0071648   -.0048526
       _cons |   39.44028   1.974654    19.97   0.000      35.5048    43.37576
------------------------------------------------------------------------------
```

- Stratified design (National Maternal and Infant Health Survey):

```
. webuse nmihs, clear
. describe
Contains data from http://localpress.stata.com/data/r12/nmihs.dta
  obs:         9,953
 vars:            23                          30 Jan 2011 11:28
 size:       418,026
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
idnum           long   %10.0f                 ID number
stratan         byte   %8.0g                  Strata indicator 1-6
finwgt          double %10.0g                 Adjusted sampling weight
scale           double %10.0g                 Scaled sampling weight
wgtstr          int    %8.0g                  ?
fsr             byte   %8.0g                  Answer questionnaire
marital         byte   %8.0g       marital    0=single, 1=married
age             byte   %8.0g                  Mother´s age in years
race            byte   %8.0g       race       Race: 1=black, 0=white/other
highbp          byte   %8.0g       hibp       High blood pressure: 1=yes,0=no
vagbleed        byte   %8.0g       vaghal     Vaginal bleeding: 1=yes, 0=no
miscar          byte   %8.0g       miscar     Previous miscarriage:1=yes,0=no
childsex        byte   %8.0g                  Sex of child
multiple        byte   %8.0g       multi      Multiple birth: 1=yes, 0=no
birthwgt        int    %8.0g                  Birthweight in grams
bwgrp           byte   %8.0g       gbw        Birth weight groups
```

(*Continued on next page*)

```
vlowbw          byte   %8.0g       lt1500     Birth weight < 1500 g
lowbw           byte   %8.0g       lbw        Birth weight < 2500 g
agegrp          byte   %8.0g       gage       Age groups 1-5
age20           byte   %8.0g       lblage20   Age < 20
age25           byte   %8.0g                  Age < 25
age25_34        byte   %8.0g                  Age 25-34
age35           byte   %8.0g                  Age 35+
-------------------------------------------------------------------------------
Sorted by:
. svyset [pw=finwgt], strata(stratan)
      pweight: finwgt
          VCE: linearized
  Single unit: missing
     Strata 1: stratan
         SU 1: <observations>
        FPC 1: <zero>
. svy: tabulate agegrp lowbw
(running tabulate on estimation sample)
Number of strata   =         6                  Number of obs      =       9946
Number of PSUs     =      9946                  Population size    = 3895561.7
                                                 Design df          =       9940

------------------------------------
Age         | Birth weight < 2500
groups      |          g
1-5         | bw>25  bw<25  Total
------------+-----------------------
 age15-19   | .1114  .0113  .1227
 age20-24   | .2539  .0196  .2736
 age25-29   | .2987   .019  .3177
 age30-34   | .1927  .0131  .2058
   age35+   | .0743   .006  .0803

    Total   |  .931   .069      1
------------------------------------
  Key:  cell proportions
  Pearson:
    Uncorrected   chi2(4)         =   15.7727
    Design-based  F(3.99, 39650.11)=    7.6700    P = 0.0000
```

- Single-stage design with all the characteristics:

```
. webuse fpc, clear
. describe
Contains data from http://localpress.stata.com/data/r12/fpc.dta
  obs:             8
 vars:             6                          30 Jan 2011 11:28
 size:           192
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
stratid         float  %9.0g
psuid           float  %9.0g
weight          float  %9.0g
nh              float  %9.0g
Nh              float  %9.0g
x               float  %9.0g
-------------------------------------------------------------------------------
Sorted by:
```

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

5.3 A multistage design

High school data

Purpose: Study the smoking habits of teenagers in the US.

Multistage 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 – secondary/subsequent sampling units
- FPC is required at stage s for stage $s + 1$ to play a role in the linearized variance estimator

Multiple stages of cluster sampling

1. PSUs are independently selected within each Stratum.
2. SSUs are independently selected within each sampled PSU.
3. ...

- Sampling units are independently selected within each sampled SSU.
- Stratification is also allowed at each sampling stage.

High school data

Smoking habits of teenagers in the US.

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.

FPC variables

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

```
. 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
. save myseniors
file myseniors.dta saved
. svy: logit smoked i.gender i.race, or
(running logit on estimation sample)
Survey: Logistic regression
Number of strata   =        50                  Number of obs      =     10559
Number of PSUs     =       100                  Population size    =  20992929
                                                Design df          =        50
                                                F(   3,     48)    =      0.17
                                                Prob > F           =    0.9140
```

smoked	Odds Ratio	Linearized Std. Err.	t	P>|t|	[95% Conf.	Interval]
2.gender	1.049387	.1004749	0.50	0.617	.8657963	1.271908
race						
2	1.036061	.1330842	0.28	0.784	.800453	1.341019
3	1.159822	.4114422	0.42	0.678	.5687799	2.36504
_cons	.6393395	.0506846	-5.64	0.000	.5452281	.7496954

```
. test 2.race 3.race
Adjusted Wald test
 ( 1)  [smoked]2.race = 0
 ( 2)  [smoked]3.race = 0
       F(  2,    49) =    0.12
            Prob > F =    0.8862
```

5.4 Poststratification

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

- Adjusts weights to sum to the poststratum sizes in the population
- Reduces bias due to nonresponse and underrepresented groups
- Can result in smaller variance estimates
- Recall that I said it is usually not vey 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.

Syntax

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

Cats and dogs

- Source: Levy and Lemeshow (1999)
- Veterinarian has 1300 clients, 450 cats and 850 dogs
- He wishes to estimate average annual expenses, but only has time to randomly select 50 clients (i.e., each client represents 26)
- Problem: As we shall see, dogs are about twice as expensive as cats

Let's estimate total expenses:

```
. webuse poststrata
. describe
Contains data from http://localpress.stata.com/data/r12/poststrata.dta
  obs:            50
 vars:             7                          30 Jan 2011 11:28
 size:         1,500
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
id              byte   %8.0g
visits          byte   %8.0g
totexp          double %10.0g                 total expenses
type            long   %8.0g       type       animal type; 1 = dog, 2 = cat
fpc             float  %9.0g
weight          double %10.0g
postwgt         float  %9.0g                  poststratification weight
-------------------------------------------------------------------------------
Sorted by:  type
. 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


. codebook weight
-------------------------------------------------------------------------------
weight                                                          (unlabeled)
-------------------------------------------------------------------------------

                  type:  numeric (double)
                 range:  [26,26]                      units:  1
         unique values:  1                        missing .:  0/50
            tabulation:  Freq.  Value
                           50  26

. svyset [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
-------------------------------------------------------------------------------
             |             Linearized
             |       Mean   Std. Err.     [95% Conf. Interval]
-------------+-----------------------------------------------------------------
      totexp |    39.7254   2.265746      35.17221    44.27859
-------------------------------------------------------------------------------
```

```
. codebook postwgt fpc
```

```
postwgt                                                 poststratification weight
```

```
                 type:  numeric (float)
                range:  [450,850]                    units:  10
        unique values:  2                        missing .:  0/50
           tabulation:  Freq.  Value
                           18  450
                           32  850
```

```
fpc                                                                   (unlabeled)
```

```
                 type:  numeric (float)
                range:  [1300,1300]                  units:  1
        unique values:  1                        missing .:  0/50
           tabulation:  Freq.  Value
                           50  1300
```

```
. 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

```
. svy: total totexp
(running total on estimation sample)
Survey: Total estimation
Number of strata =       1          Number of obs    =      50
Number of PSUs   =      50          Population size  =    1300
N. of poststrata =       2          Design df        =      49
```

	Total	Linearized Std. Err.	[95% Conf.	Interval]
totexp	52149.67	1512.548	49110.09	55189.25

5.5 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.
- Certainty units
- Bad design

Problem

- This is a big issue for variance estimation:
  - Consider a sample with only 1 observation
  - **`svy`** reports missing standard error estimates by default

Solution

- Use **`svydescribe`**:
  - Describes the strata and sampling units
  - Helps find strata with a single sampling unit
- Drop them from the estimation sample.
- **`svyset`** one of the ad-hoc adjustments in the **`singleunit()`** option.
- Somehow combine them with other strata.

- Example: Second National Health and Nutrition Examination Survey

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

			#Obs per Unit		
Stratum	#Units	#Obs	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

- Everything looks fine, but what if we are examining high density lipids?

```
. codebook hdresult
```

```
hdresult                                              high density lipids (mg/dL)
```

```
                  type:  numeric (int)
                 range:  [15,187]                     units:  1
         unique values:  108                      missing .:  1631/10351
                  mean:   49.6427
             std. dev:   14.3118
           percentiles:        10%       25%       50%       75%       90%
                               34        39        48        57        68
```

```
. svy: mean hdresult
(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]
hdresult	49.67141	.	.	.

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

. 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				

- Even better would be

```
. svydescribe hdresult, 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>
```

(*Continued on next page*)

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						

5.6 Certainty units

Certainty units

Sampling units that are guaranteed to be chosen by the design.

- Treat each certainty unit as a stratum with an FPC of 1.
- No contribution to the variance.
- Certainty PSUs are not counted in the design degrees of freedom.

```
. svyset
      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: strata
         SU 1: psu
        FPC 1: <zero>
. svyset psu [pw=finalwgt], strata(strata) singleunit(certainty)
      pweight: finalwgt
          VCE: linearized
  Single unit: certainty
     Strata 1: strata
         SU 1: psu
        FPC 1: <zero>
. svy: mean hdresult
(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]
hdresult	49.67141	.3829811	48.88813	50.4547

```
Note: strata with single sampling unit treated as certainty
      units.
```

We are now finished with this lesson.

```
. log close
```

6 Variance estimation

Start a new log for this lesson.

```
. log using lesson6
```

Stata has five variance estimation methods for survey data:

- Linearization
- Balanced repeated replication (BRR)
- Survey jackknife
- Survey bootstrap
- Successive difference replication (SDR)

6.1 Overview

- 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.
    - Due to 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, computing replicates from each resample, then using the replicates to estimate the variance.

6.2 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

6.3 The total estimator

Total estimator – Stratified two-stage design

- y_{hijk} – observed value from a sampled individual
- Strata: $h = 1, \ldots, L$
- PSU: $i = 1, \ldots, n_h$
- SSU: $j = 1, \ldots, m_{hi}$
- Individual: $k = 1, \ldots, m_{hij}$

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

- 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

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

- Returning to our seniors data, we can estimate the total number of seniors who have smoked

```
. use myseniors, clear
. 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
. 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
--------------------------------------------------------------
             |             Linearized
             |      Total   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
      smoked |    8347260   331155.1       7682115     9012404
--------------------------------------------------------------
```

- Prior to Stata 9, you could only incorporate the first stage of the sample design

```
. svyset county [pw=sampwgt], strata(state) fpc(ncounties)
      pweight: sampwgt
          VCE: linearized
  Single unit: missing
     Strata 1: state
         SU 1: county
        FPC 1: ncounties
. 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
--------------------------------------------------------------
             |             Linearized
             |      Total   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
      smoked |    8347260   312585.7       7719413     8975107
--------------------------------------------------------------
```

6.4 Regression models

Linearized variance for regression models

- Model is fit using estimating equations.
- $\widehat{G}()$ is a total estimator, use Taylor expansion to get $\widehat{V}(\widehat{\boldsymbol{\beta}})$.

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

$$\widehat{V}(\widehat{\boldsymbol{\beta}}) = D\widehat{V}\{\widehat{G}(\boldsymbol{\beta})\}|_{\boldsymbol{\beta}=\widehat{\boldsymbol{\beta}}}D'$$

ML models

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

Least squares regression

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

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

6.5 Example

- For this, we return to NHANES2

```
. webuse nhanes2, clear
. svyset
      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: strata
         SU 1: psu
        FPC 1: <zero>
```

- and fit a logit model for high blood pressure:

```
. describe highbp height weight age female race
              storage  display     value
variable name   type   format      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
. discard
. local model highbp height weight c.age##c.age i.female i.race
. svy: logit `model´, 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
------------------------------------------------------------------------------
             |             Linearized
      highbp |      Coef.   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
------------------------------------------------------------------------------
. est store taylor
```

- We can also estimate the odds ratio for a 5-year age increase and a 10 kg weight increase, and its survey-adjusted standard error

```
. lincom 5*age + 25*c.age#c.age + 10*weight, or
 ( 1)  10*[highbp]weight + 5*[highbp]age + 25*[highbp]c.age#c.age = 0
```

highbp	Odds Ratio	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	3.43827	.3345431	12.69	0.000	2.8194	4.192985

6.6 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, 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.

❑

BRR replicate weights

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

The adjusted sampling weight for the ith 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 a copy of the point estimates (replicate). 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

- $\widehat{\boldsymbol{\theta}}$ – point estimates
- $\widehat{\boldsymbol{\theta}}_{(i)}$ – ith replicate of the point estimates
- $\overline{\boldsymbol{\theta}}_{(.)}$ – average of the replicates

Default variance formula:

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

Mean squared error (MSE) formula:

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

❑ 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, taking you to a short help file informing you that you used the MSE formula for BRR variance estimation.

❑

6.7 Example

- We can use a version of NHANES2 that already has a set of replicate-weight variables in it

```
. webuse nhanes2brr, clear
. svyset, vce(brr, mse) noclear
      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 `model´, baselevel
(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
female						
0	0	(base)				
1	-.3502998	.0876341	-4.00	0.000	-.5290307	-.1715689
race						
1	0	(base)				
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

- We can also compare with the previous results that used Taylor linearization

```
. est store brr
. est table _all, se eform
```

Variable	taylor	brr
height	.96830051	.96830051
	.00561371	.005707
weight	1.0503741	1.0503741
	.00335345	.00337165
age	1.166568	1.166568
	.02428977	.02421677
c.age#c.age	.99892752	.99892752
	.00020076	.00019977
female		
1	.70447688	.70447688
	.06071702	.06173619
race		
2	1.4135946	1.4135946
	.20000428	.20532892
3	1.1626309	1.1626309
	.50570438	.64664467
_cons	.00690943	.00690943
	.00807545	.00808666

legend: b/se

6.8 Jackknife

The jackknife
A replication method for variance estimation. 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`**.

❑

Delete-1 jackknife replicate weights

- w_{hij} – sampling weight for individual j in PSU i of stratum h.
- Dropping PSU i^* from stratum h^*.
- n_{h^*} replicates from stratum h^*.

The adjusted sampling weight is

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

Delete-k jackknife replicate weights

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

The adjusted sampling weight is

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

Jackknife variance formulas

- $\widehat{\boldsymbol{\theta}}_{(h,i)}$ – replicate of the point estimates from stratum h, PSU i
- $\overline{\boldsymbol{\theta}}_h$ – average of the replicates from stratum h
- $m_h = (n_h - 1)/n_h$ – delete-1 multiplier for stratum h
- $m_h = (n_h - k)/c_h k$ – delete-k

Default variance formula:

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

Mean squared error (MSE) formula:

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

❑ 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.
- **`Jknife *`** is clickable, taking 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.

❑

6.9 Example

- You can specify jackknife standard errors when you fit the model:

```
. webuse nhanes2, clear
. svyset
      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: strata
         SU 1: psu
        FPC 1: <zero>
. svy jackknife, mse: logit `model´, baselevel
(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
female						
0	0	(base)				
1	-.3502998	.0862108	-4.06	0.000	-.5261279	-.1744716
race						
1	0	(base)				
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

- or you can, equivalently, specify jackknife estimation as your preferred method

```
. svyset psu [pw=finalwgt], strata(strata) vce(jackknife, mse)
      pweight: finalwgt
          VCE: jackknife
          MSE: on
  Single unit: missing
     Strata 1: strata
         SU 1: psu
        FPC 1: <zero>
. svy: logit `model´, baselevel
(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
female						
0	0	(base)				
1	-.3502998	.0862108	-4.06	0.000	-.5261279	-.1744716
race						
1	0	(base)				
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

6.10 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.
- Requires replicate weight variables.

Syntax

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

Bootstrap variance formulas

- $\widehat{\boldsymbol{\theta}}$ – point estimates
- $\widehat{\boldsymbol{\theta}}_{(i)}$ – ith replicate of the point estimates
- $\overline{\boldsymbol{\theta}}_{(.)}$ – average of the replicates
- b – number of bootstrap samples used to generate each replicate weight variable, default is **`bsn(1)`**

Default variance formula:

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

Mean squared error (MSE) formula:

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

6.11 Successive difference replication

Successive difference replication – SDR
Replication method designed for systematic samples where the observed sampling units are ordered.

- Resample the observed data by adjusting the sampling weights.
- Requires replicate weight variables.

Syntax

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

SDR variance formulas

- $\widehat{\boldsymbol{\theta}}$ – point estimates
- $\widehat{\boldsymbol{\theta}}_{(i)}$ – ith replicate of the point estimates
- $\overline{\boldsymbol{\theta}}_{(.)}$ – average of the replicates
- f – sampling fraction from **`fpc()`** option

Default variance formula:

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

Mean squared error (MSE) formula:

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

6.12 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.
- Eliminate 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)
```

6.13 Example

- Consider a privacy-conscious version of NHANES:

```
. webuse nhanes2jknife, clear
. svyset
      pweight: finalwgt
          VCE: jackknife
          MSE: off
    jkrweight: jkw_1 jkw_2 jkw_3 jkw_4 jkw_5 jkw_6 jkw_7 jkw_8 jkw_9 jkw_10
               jkw_11 jkw_12 jkw_13 jkw_14 jkw_15 jkw_16 jkw_17 jkw_18 jkw_19
               jkw_20 jkw_21 jkw_22 jkw_23 jkw_24 jkw_25 jkw_26 jkw_27 jkw_28
               jkw_29 jkw_30 jkw_31 jkw_32 jkw_33 jkw_34 jkw_35 jkw_36 jkw_37
               jkw_38 jkw_39 jkw_40 jkw_41 jkw_42 jkw_43 jkw_44 jkw_45 jkw_46
               jkw_47 jkw_48 jkw_49 jkw_50 jkw_51 jkw_52 jkw_53 jkw_54 jkw_55
               jkw_56 jkw_57 jkw_58 jkw_59 jkw_60 jkw_61 jkw_62
  Single unit: missing
     Strata 1: <one>
         SU 1: <observations>
        FPC 1: <zero>

. webuse nhanes2jknife, clear

. describe

. svyset [pw=finalwgt], vce(jackknife) jkrweight(jkw_*)
```

- We can thus "replicate" the jackknife calculation, without having to know anything about strata and PSU membership

```
. svy, mse: logit `model´, baselevel
(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
                                                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
female						
0	0	(base)				
1	-.3502998	.0862108	-4.06	0.000	-.5261279	-.1744716
race						
1	0	(base)				
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

We are now finished with this lesson.

```
. log close
```

7 Subpopulations

Start a new log for this lesson.

```
. log using lesson7
```

7.1 Two perspectives sampling subsets

Focus on a subset of the population

- Subpopulation variance estimation:
  - Assumes the same survey design for subsequent data collection.
  - 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.
  - The **`if`** and **`in`** restrictions.

7.2 The subpopulation total

Total from *SRS* data

- Data is $y_1, \ldots, y_n$ 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:

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

- Sampling weight and subpopulation size:

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

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.

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

Variance of a restricted-sample total
Sample n_S without replacement from the subpopulation of N_S values.

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

7.3 Example

- Returning the the National Maternal and Infant Health Survey

```
. webuse nmihs, clear
. svyset
      pweight: finwgt
          VCE: linearized
  Single unit: missing
     Strata 1: stratan
         SU 1: <observations>
        FPC 1: <zero>
. des birthwgt highbp
              storage  display     value
variable name   type   format      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
```

- We can estimate the mean birth weight for the high-blood-pressure subpopulation

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

- How does that compare to the restricted-sample estimate?

```
. 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

7.4 Getting fancy

- How about the other subpopulation, those *without* high blood pressure?

```
. svy, subpop(if !highbp): mean birthwgt
(running mean on estimation sample)
Survey: Mean estimation
Number of strata =       6          Number of obs    =     9946
Number of PSUs   =    9946          Population size  =  3895562
                                    Subpop. no. obs  =     9351
                                    Subpop. size     =  3709365
                                    Design df        =     9940
```

	Mean	Linearized Std. Err.	[95% Conf.	Interval]
birthwgt	3363.131	6.605511	3350.183	3376.079

- How about across values of a categorical variable?

```
. codebook agegrp
```

```
agegrp                                                        Age groups 1-5
```

```
                 type:  numeric (byte)
                label:  gage

                range:  [1,5]                        units:  1
        unique values:  5                    missing .:  0/9953

           tabulation:  Freq.   Numeric  Label
                         1653         1  age15-19
                         2866         2  age20-24
                         2848         3  age25-29
                         1851         4  age30-34
                          735         5  age35+
```

```
. svy: mean birthwgt, over(agegrp)
(running mean on estimation sample)

Survey: Mean estimation

Number of strata =       6          Number of obs    =     9946
Number of PSUs   =    9946          Population size  =  3895562
                                    Design df        =     9940

    _subpop_1: agegrp = age15-19
    _subpop_2: agegrp = age20-24
    _subpop_3: agegrp = age25-29
    _subpop_4: agegrp = age30-34
    _subpop_5: agegrp = age35+
```

Over	Mean	Linearized Std. Err.	[95% Conf.	Interval]
birthwgt				
_subpop_1	3205.137	18.59472	3168.688	3241.587
_subpop_2	3314.949	12.12495	3291.182	3338.717
_subpop_3	3399.925	12.07594	3376.254	3423.597
_subpop_4	3413.299	15.28501	3383.338	3443.261
_subpop_5	3398.944	28.25415	3343.56	3454.328

We are now finished with this lesson.

```
. log close
```

8 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**

Peform ANOVA-style tests on the effects of factor variables

- **contrast**

Perform pairwise comparisons of marginal means and slopes

- **pwcompare**

8.1 Goodness of fit

- Recall the logistic model we fit using the NHANES2 data.

```
. webuse nhanes2, clear
. local model highbp height weight c.age##c.age i.female i.race
. svy: logit `model´, 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

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

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

8.2 Predictive margins

Predictive margins

- The variable `race` is a categorical (factor) variable with three coded levels.

```
. describe race
              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
race            byte   %9.0g       race       1=white, 2=black, 3=other
. label list race
race:
           1 White
           2 Black
           3 Other
```

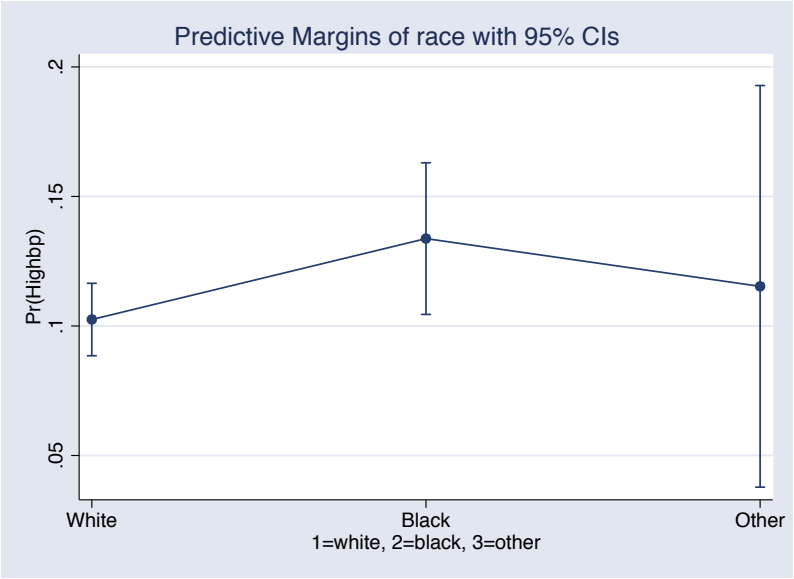
From this we can see that the odds ratio comparing blacks with whites is clearly large and statistically significant. One might ask if this represents a sizeable change in the predicted probabilities.

`margins` computes all sorts of marginal statistics using predicted values from the currently fitted model. 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.

- Let's use **`margins`** to look at the predicted probablities of high blood pressure for each level of `race`.

```
. margins race, vce(unconditional)
Predictive margins                              Number of obs     =      10351
Expression   : Pr(highbp), predict()
------------------------------------------------------------------------------
             |            Linearized
             |     Margin   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
------------------------------------------------------------------------------
. marginsplot
  Variables that uniquely identify margins: race
```

- Here is a profile plot of the predictive margins.



8.3 Marginal effects

It is tempting to look at the overlapping confidence intervals (CI) and conclude that the difference in the marginal probabilities between the first two levels of `race` is not significant at the 5% level. The problem with this comparison is that is does not account for the covariance between these two point estimates.

Marginal effects

- We can use the **`dydx()`** option to get **`margins`** to compute the marginal effects of `race`.

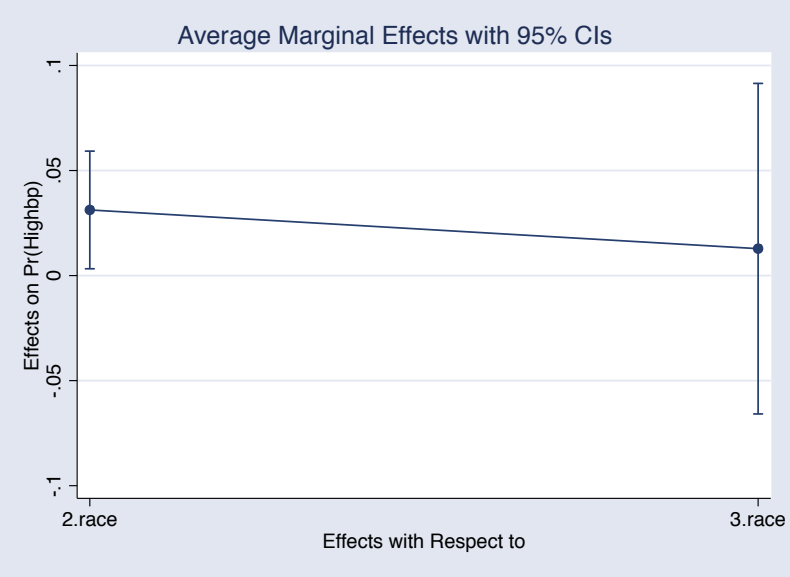
```
. margins, vce(unconditional) dydx(race)
Average marginal effects                        Number of obs      =     10351
Expression   : Pr(highbp), predict()
dy/dx w.r.t. : 2.race 3.race
```

	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.
. marginsplot
  Variables that uniquely identify margins: _deriv
```

Now we can conclude that there is a significant difference between the two marginal propabilities at 5% level.

- Here is a profile plot of the marginal effects.



8.4 Discrete marginal effects

Contrasts

- The **`contrast`** command and contrast operators are new in Stata 12.
- **`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*.

```
. 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.
```

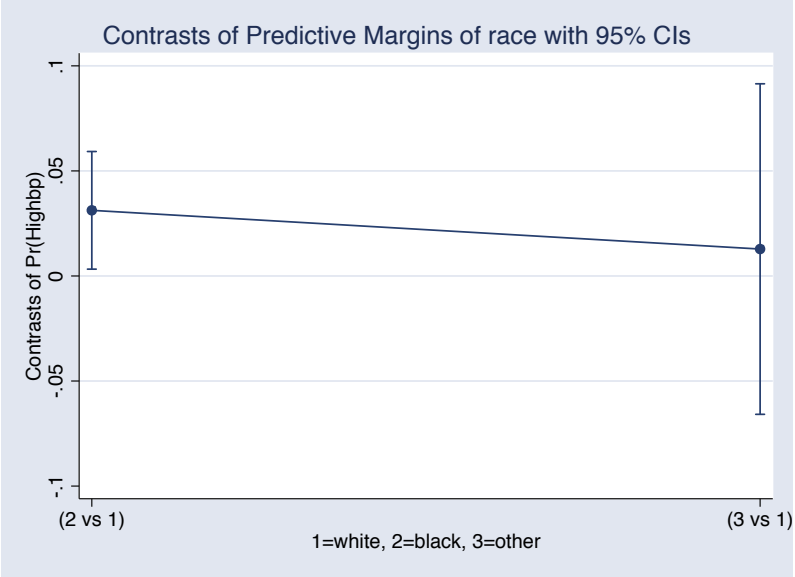
(*Continued on next page*)

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

```
. marginsplot
  Variables that uniquely identify margins: race
```

Notice that 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.



9 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. 2010. 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*. 154–159. American Statistical Association.

[3] Levy, P. 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.