Survey Data Analysis with Stata

Jeff Pitblado

Associate Director, Statistical Software

StataCorp LP

Bamberg, Germany 2011

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

- 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

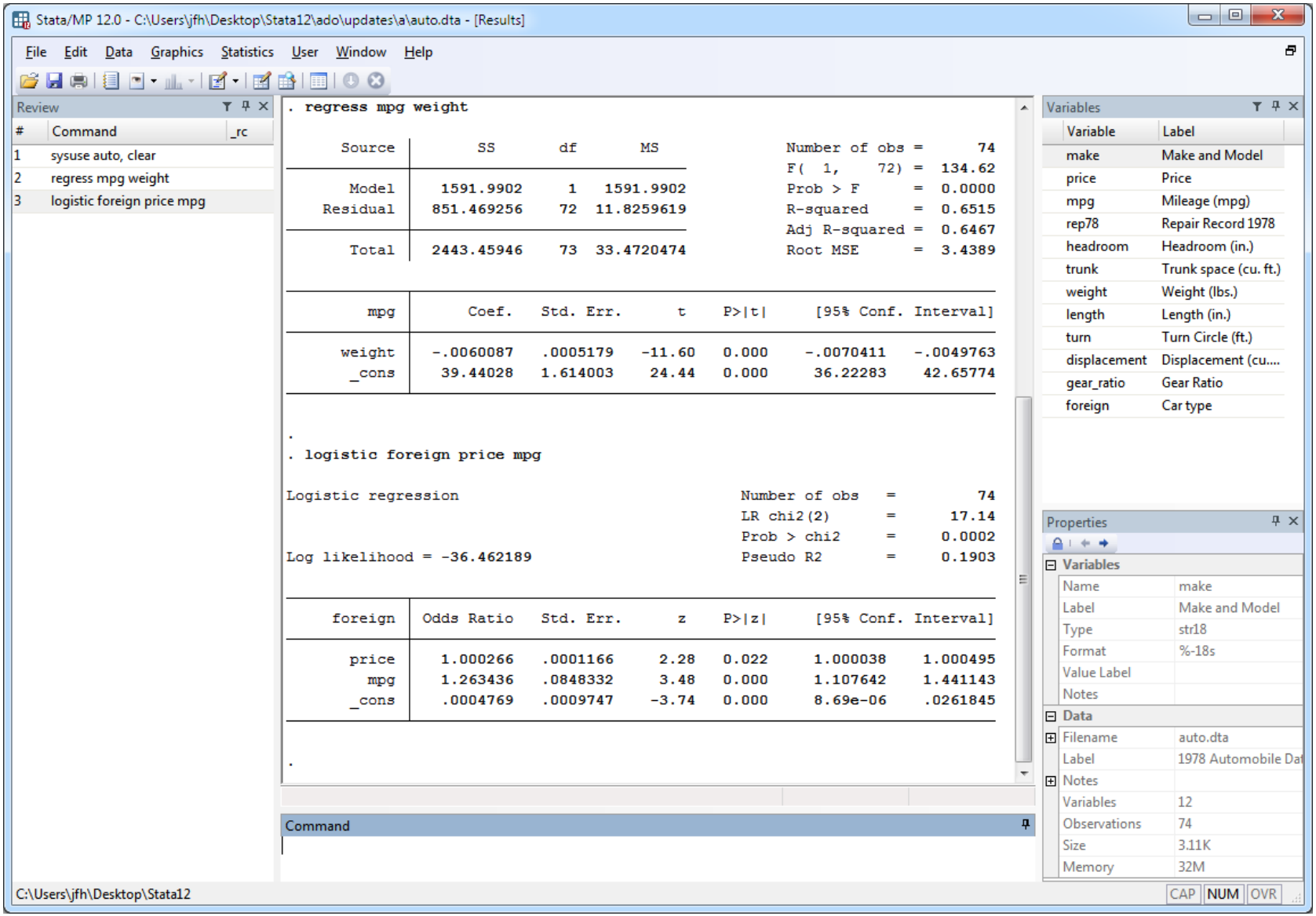
- Mac
- Windows
- Unix
  - Linux
  - IBM AIX
  - Oracle Solaris

- 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

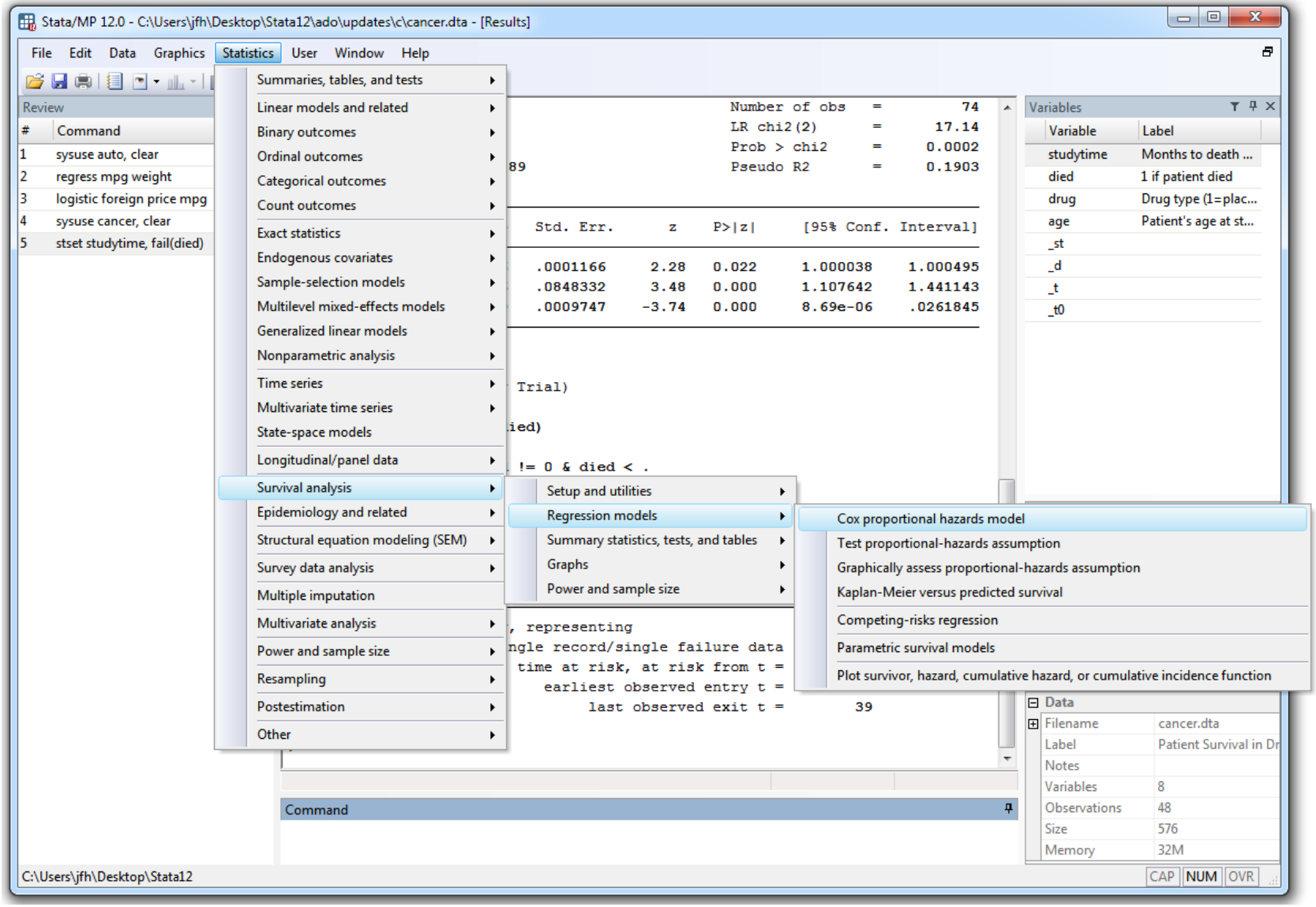
Over 9,500 pages of documentation in Stata 12

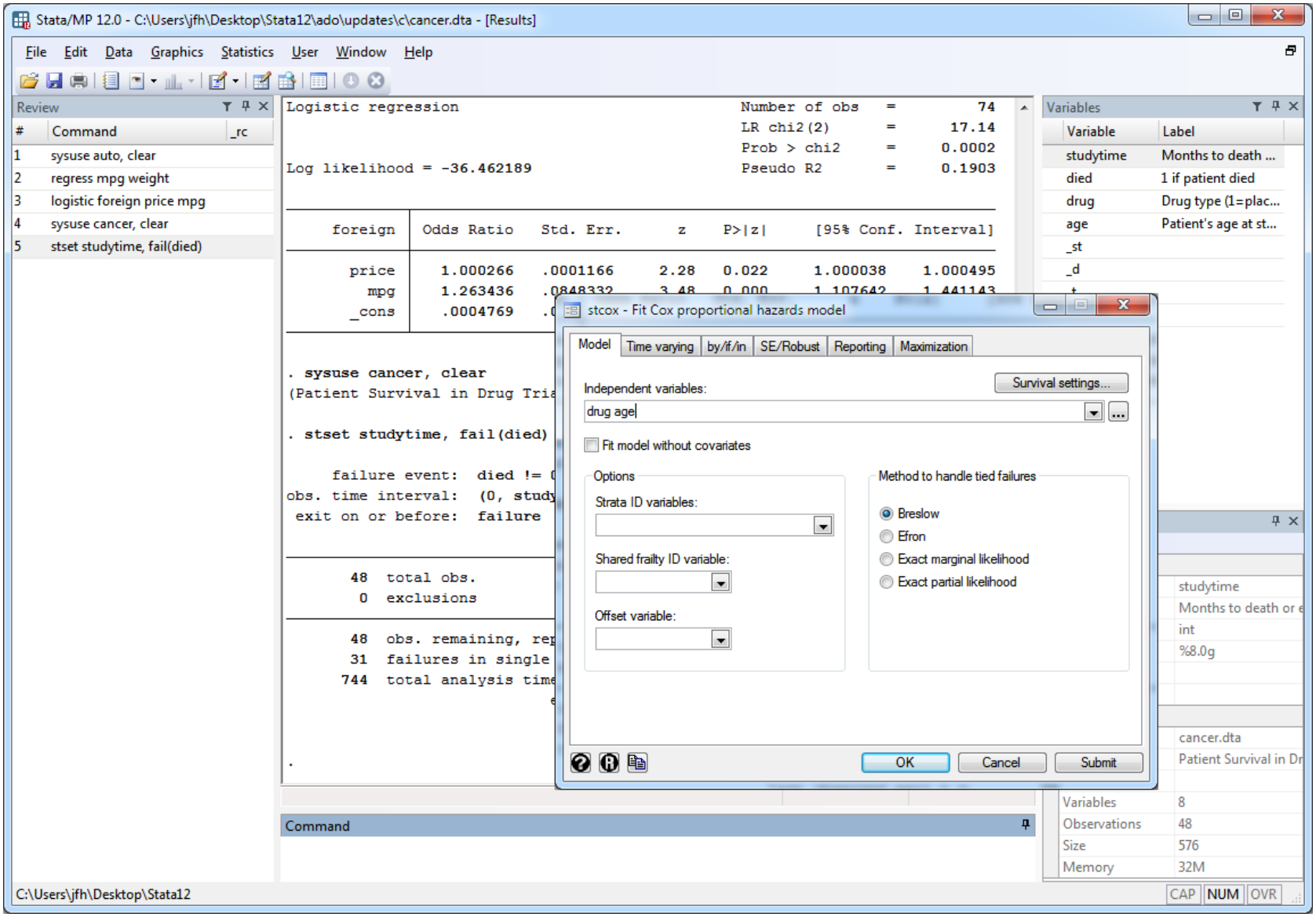
Organized in fifteen volumes

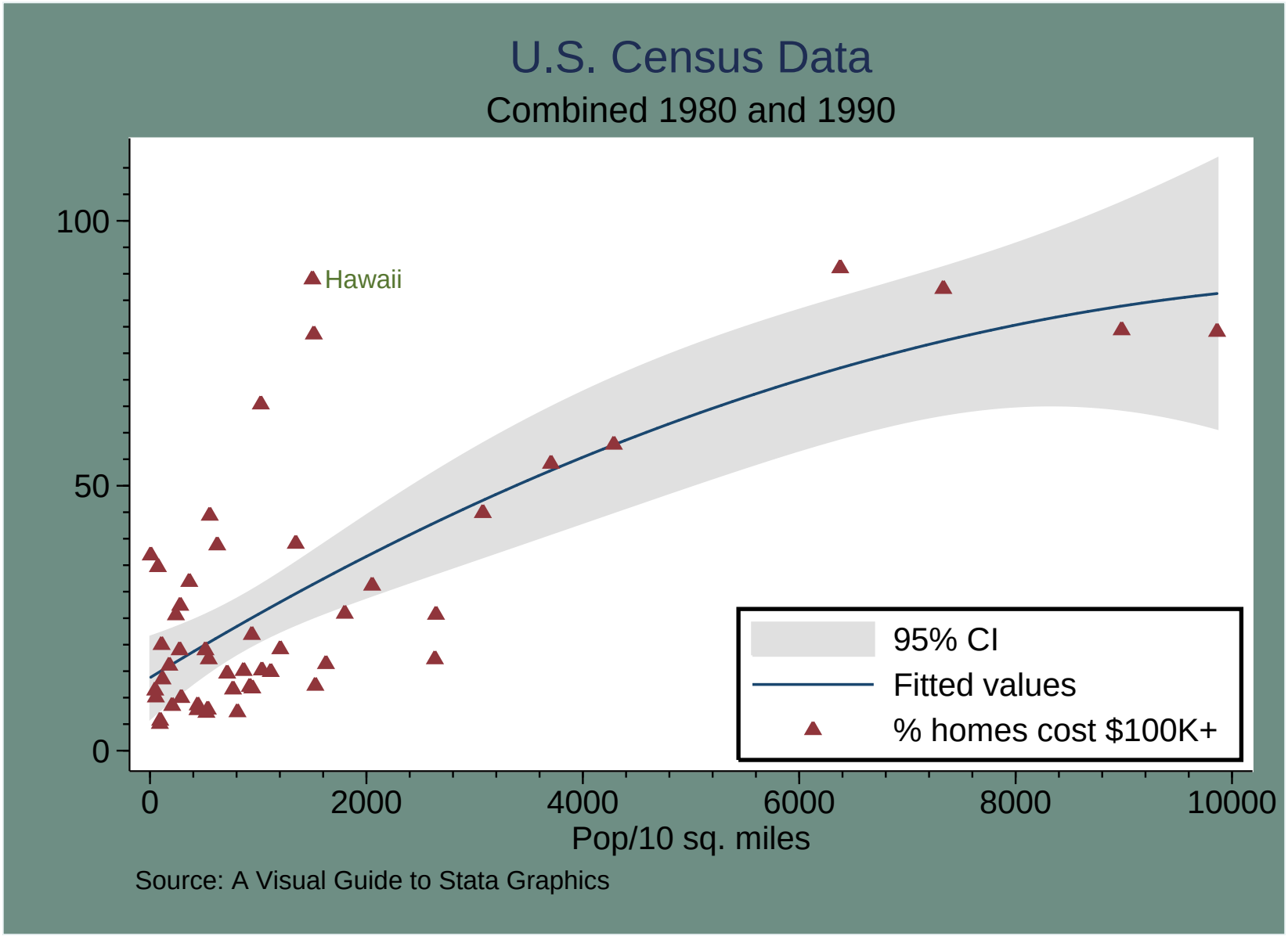
[GS]	Getting Started–Mac, Unix, Windows
[U]	User's Guide
[D]	Data Management
[G]	Graphics
[MI]	Multiple Imputation
[MV]	Multivariate Statistics
[R]	Base Reference (4 volumes)
[ST]	Survival Analysis/Epidemiological Tables
[SEM]	Structural equations modeling
[SVY]	Survey Data
[TS]	Time Series
[XT]	Panel/Longitudinal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


Stata/MP 12.0 - C:\Users\jfh\Desktop\Stata12\ado\updates\a\auto.dta - [Results]
File Edit Data Graphics Statistics User Window Help
Review
Command _rc
1 sysuse auto, clear
2 regress mpg weight
3 logistic foreign price mpg
. regress mpg weight
Source SS df MS
Model 1591.9902 1 1591.9902
Residual 851.469256 72 11.8259619
Total 2443.45946 73 33.4720474
Number of obs = 74
F(1, 72) = 134.62
Prob > F = 0.0000
R-squared = 0.6515
Adj R-squared = 0.6467
Root MSE = 3.4389
mpg Coef. Std. Err. t P>|t| [95% Conf. Interval]
weight -.0060087 .0005179 -11.60 0.000 -.0070411 -.0049763
_cons 39.44028 1.614003 24.44 0.000 36.22283 42.65774
. logistic foreign price mpg
Logistic regression
Number of obs = 74
LR chi2(2) = 17.14
Prob > chi2 = 0.0002
Log likelihood = -36.462189
Pseudo R2 = 0.1903
foreign Odds Ratio Std. Err. z P>|z| [95% Conf. Interval]
price 1.000266 .0001166 2.28 0.022 1.000038 1.000495
mpg 1.263436 .0848332 3.48 0.000 1.107642 1.441143
_cons .0004769 .0009747 -3.74 0.000 8.69e-06 .0261845
Command
Variables
Variable Label
make Make and Model
price Price
mpg Mileage (mpg)
rep78 Repair Record 1978
headroom Headroom (in.)
trunk Trunk space (cu. ft.)
weight Weight (lbs.)
length Length (in.)
turn Turn Circle (ft.)
displacement Displacement (cu....
gear_ratio Gear Ratio
foreign Car type
Properties
Variables
Name make
Label Make and Model
Type str18
Format %-18s
Value Label
Notes
Data
Filename auto.dta
Label 1978 Automobile Dat
Notes
Variables 12
Observations 74
Size 3.11K
Memory 32M
C:\Users\jfh\Desktop\Stata12
CAP NUM OVR

- 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*


Stata/MP 12.0 - C:\Users\jfh\Desktop\Stata12\ado\updates\c\cancer.dta - [Results]
File Edit Data Graphics Statistics User Window Help
Summaries, tables, and tests
Linear models and related
Binary outcomes
Ordinal outcomes
Categorical outcomes
Count outcomes
Exact statistics
Endogenous covariates
Sample-selection models
Multilevel mixed-effects models
Generalized linear models
Nonparametric analysis
Time series
Multivariate time series
State-space models
Longitudinal/panel data
Survival analysis
Epidemiology and related
Structural equation modeling (SEM)
Survey data analysis
Multiple imputation
Multivariate analysis
Power and sample size
Resampling
Postestimation
Other
Setup and utilities
Regression models
Summary statistics, tests, and tables
Graphs
Power and sample size
Cox proportional hazards model
Test proportional-hazards assumption
Graphically assess proportional-hazards assumption
Kaplan-Meier versus predicted survival
Competing-risks regression
Parametric survival models
Plot survivor, hazard, cumulative hazard, or cumulative incidence function
Review
Command
1 sysuse auto, clear
2 regress mpg weight
3 logistic foreign price mpg
4 sysuse cancer, clear
5 stset studytime, fail(died)
Number of obs = 74
LR chi2(2) = 17.14
Prob > chi2 = 0.0002
Pseudo R2 = 0.1903
Std. Err. z P>|z| [95% Conf. Interval]
.0001166 2.28 0.022 1.000038 1.000495
.0848332 3.48 0.000 1.107642 1.441143
.0009747 -3.74 0.000 8.69e-06 .0261845
Variables
Variable Label
studytime Months to death ...
died 1 if patient died
drug Drug type (1=plac...
age Patient's age at st...
_st
_d
_t
_t0
Data
Filename cancer.dta
Label Patient Survival in Dr
Notes
Variables 8
Observations 48
Size 576
Memory 32M
Command
C:\Users\jfh\Desktop\Stata12
CAP NUM OVR


Stata/MP 12.0 - C:\Users\jfh\Desktop\Stata12\ado\updates\c\cancer.dta - [Results]
File Edit Data Graphics Statistics User Window Help
Review
Command _rc
1 sysuse auto, clear
2 regress mpg weight
3 logistic foreign price mpg
4 sysuse cancer, clear
5 stset studytime, fail(died)
Logistic regression
Number of obs = 74
LR chi2(2) = 17.14
Prob > chi2 = 0.0002
Log likelihood = -36.462189
Pseudo R2 = 0.1903
foreign | Odds Ratio Std. Err. z P>|z| [95% Conf. Interval]
price 1.000266 .0001166 2.28 0.022 1.000038 1.000495
mpg 1.263436 .0848332 3.48 0.000 1.107642 1.441143
_cons .0004769
. sysuse cancer, clear
(Patient Survival in Drug Tria
. stset studytime, fail(died)
failure event: died != 0
obs. time interval: (0, study
exit on or before: failure
48 total obs.
0 exclusions
48 obs. remaining, rep
31 failures in single
744 total analysis time
Command
Variables
Variable Label
studytime Months to death ...
died 1 if patient died
drug Drug type (1=plac...
age Patient's age at st...
_st
_d
stcox - Fit Cox proportional hazards model
Model Time varying by/if/in SE/Robust Reporting Maximization
Independent variables:
drug age
Survival settings...
Fit model without covariates
Options
Strata ID variables:
Shared frailty ID variable:
Offset variable:
Method to handle tied failures
Breslow
Efron
Exact marginal likelihood
Exact partial likelihood
OK Cancel Submit
studytime
Months to death or e
int
%8.0g
cancer.dta
Patient Survival in Dr
Variables 8
Observations 48
Size 576
Memory 32M
C:\Users\jfh\Desktop\Stata12
CAP NUM OVR


U.S. Census Data
Combined 1980 and 1990
Hawaii
100
50
0
0
2000
4000
6000
8000
10000
Pop/10 sq. miles
95% CI
Fitted values
% homes cost $100K+
Source: A Visual Guide to Stata Graphics

- 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

- 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
```

- 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
```

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

```
. webuse cancer
```

- 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

- 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
```

Start a new log for this lesson.

```
. log using lesson2
```

- 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

- 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

Example

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`

- Let's have a look at the dataset:

```
. webuse towerlondon, clear
. describe
. notes
. codebook
```

- Let's now fit the logistic model:

```
. logit dtlm difficulty
```

- How about odds ratios and 90% CIs?

```
. logit, or level(90)
```

- 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
```

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

```
. gen diff2 = difficulty + 1
. logit dtlm group#diff2, or
. logit dtlm group##diff2, or
. logit dtlm group#c.difficulty, or
```

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

```
. describe
. gen relative = group == 2
. gen sphrenic = group == 3
. logit dtlm difficulty relative sphrenic, or
```

- 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
```

- Note that Stata uses Taylor linearization to estimate standard errors when **pweights** are specified

- 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)
```

- 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)
```

- Clustering on family subsumes clustering on subject.

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

```
. logit dtlm difficulty relative sphrenic
. est store logit
. probit dtlm difficulty relative sphrenic
. est store probit
. est dir
. est table _all
. est stats _all
```

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

```
. probit
. est save probit
```

- The next time you start Stata, you could type

```
. est use probit
. probit
```

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

```
. logit dtlm difficulty relative sphrenic
. predict phat
. list dtlm difficulty relative sphrenic phat in 1/20
```

- 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

- 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
```

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

```
. test sphrenic = -1
. test relative sphrenic
```

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

```
. est store full
. logit dtlm difficulty
. lrtest . full
```

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

We are now finished with this lesson.

```
. log close
```

- 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.

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

Start a new log for this lesson.

```
. log using lesson4
```

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`**.

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

```
. webuse nhanes2, clear
. proportion sex
. svyset
. svy: proportion sex
```

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

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

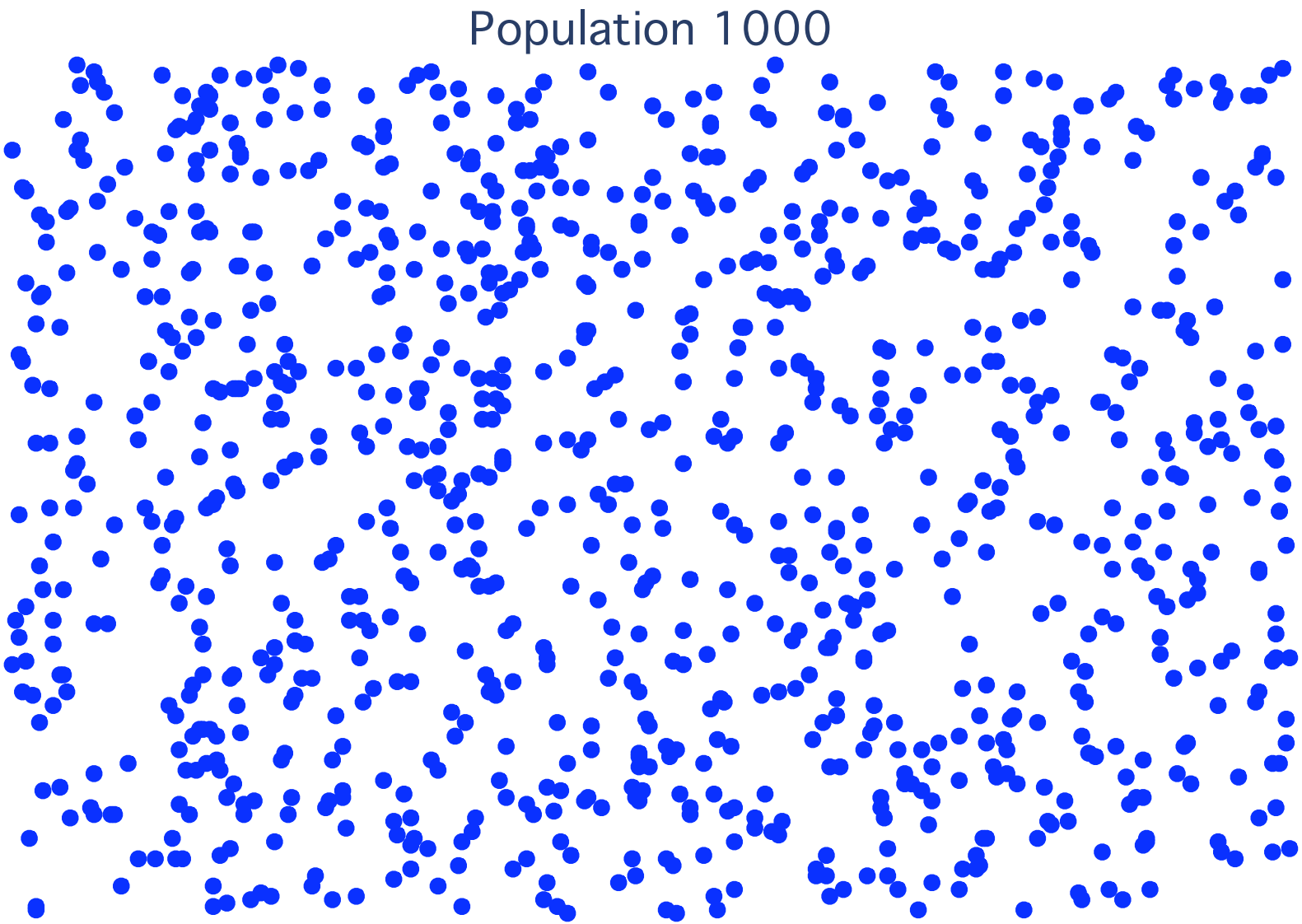
In stratified designs, the population is partitioned into well-defined groups, called strata.

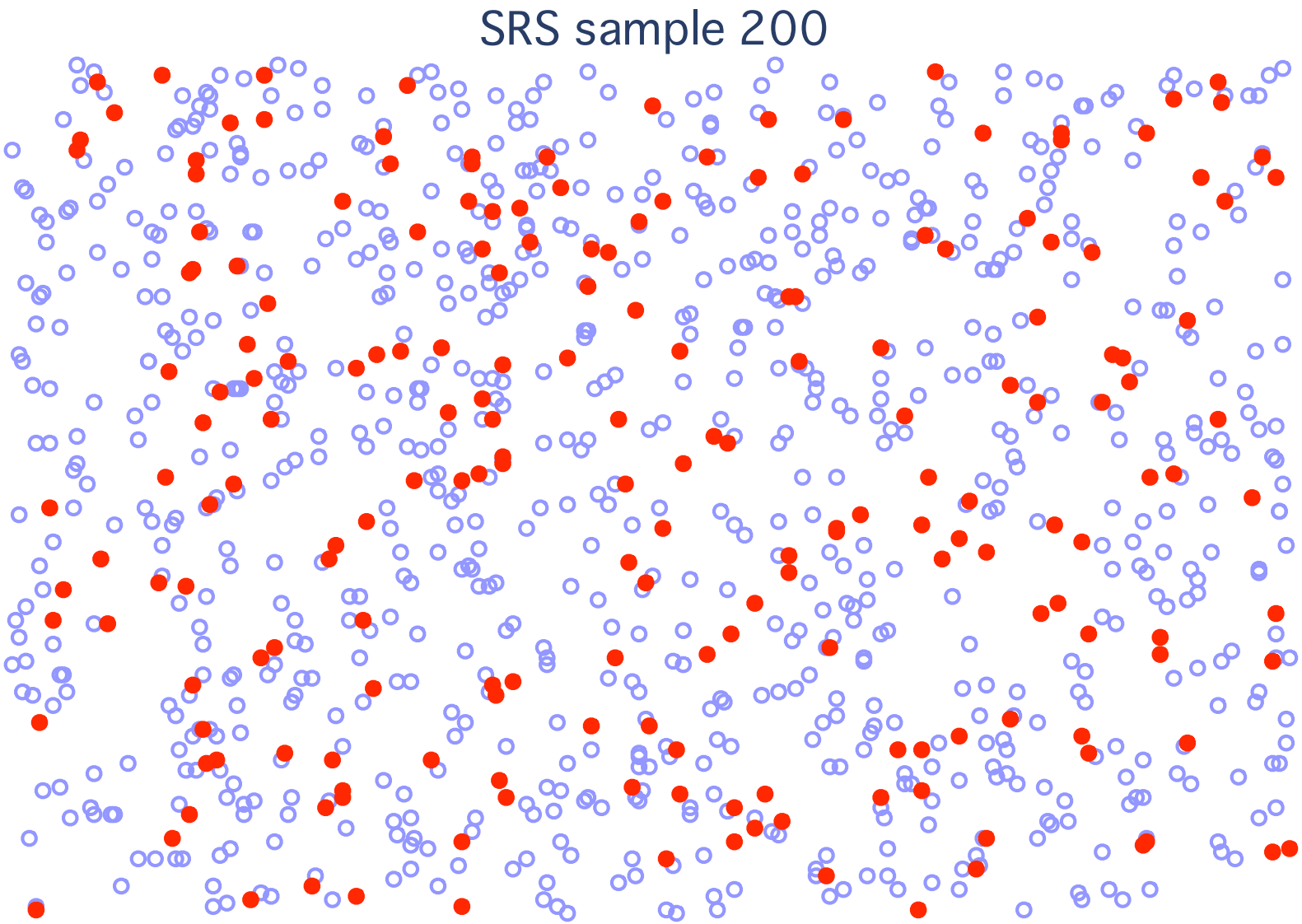
- Sampling units are independently sampled from within each stratum.
- Stratification usually results in smaller variance estimates compared to SRS.
- Although there is potential for improving efficiency by reducing sampling variability, it is usually not very practical to stratify on demographic information.

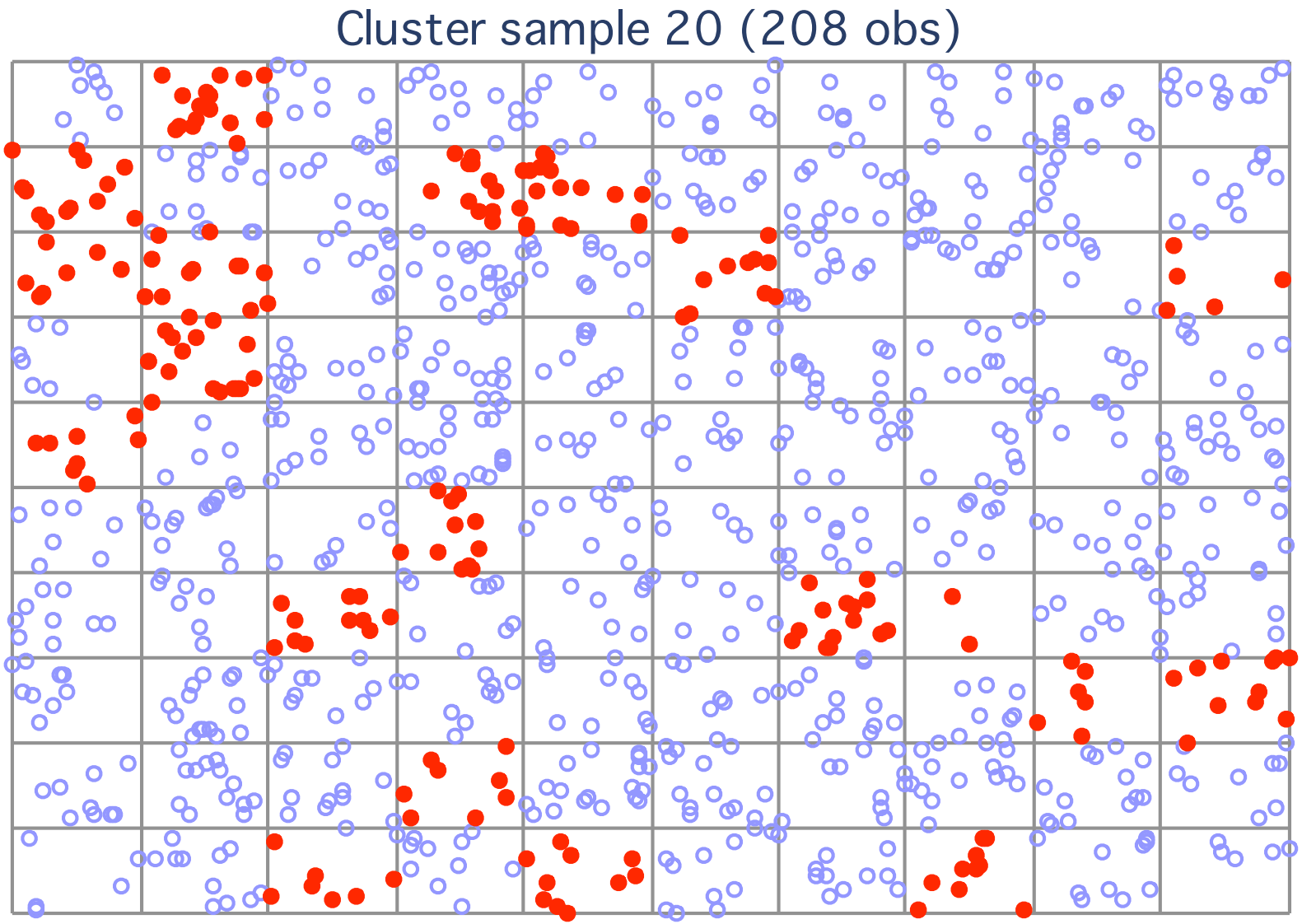
Finite population correction (FPC)

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

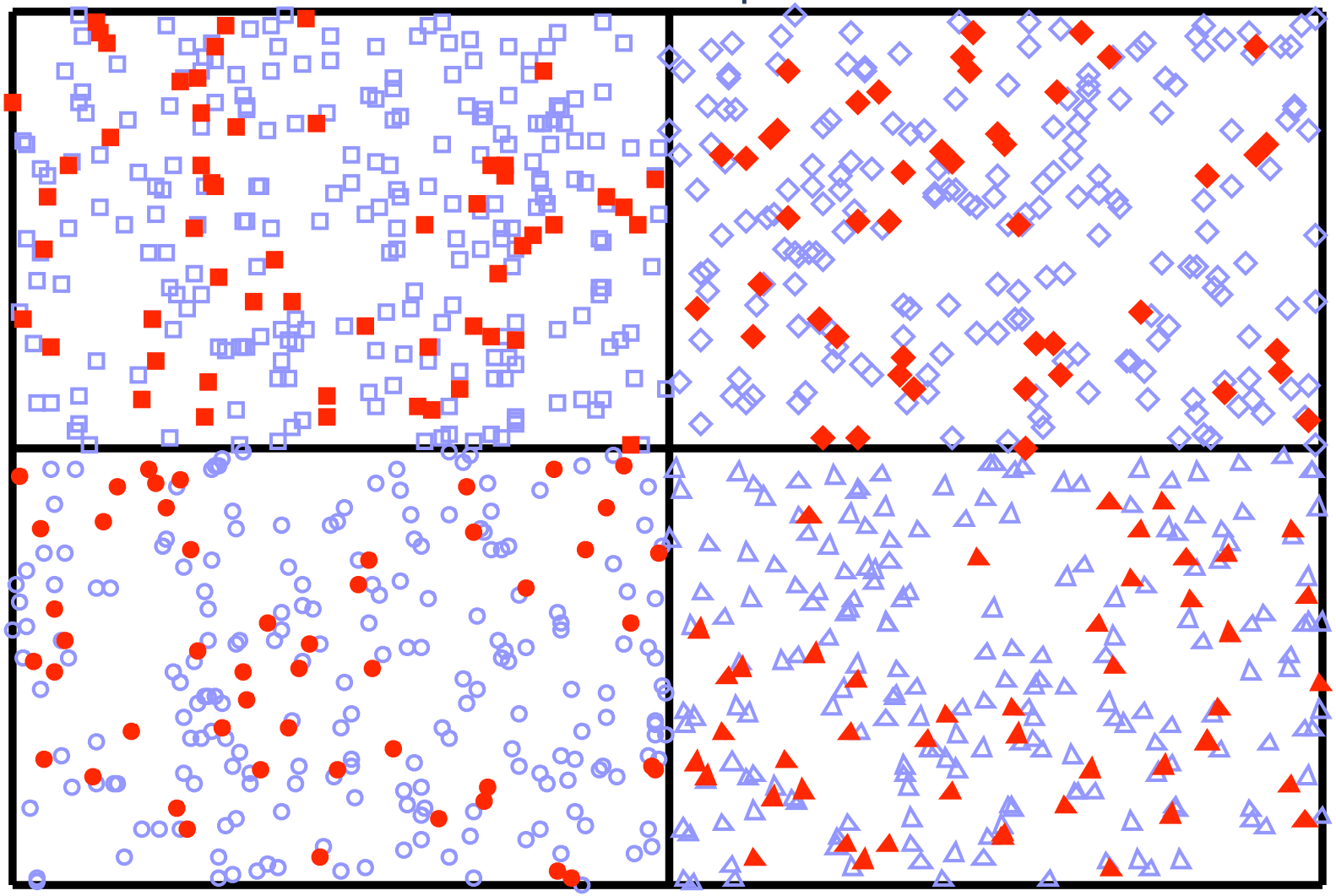
- Sampling without replacement from a finite population reduces sampling variability.

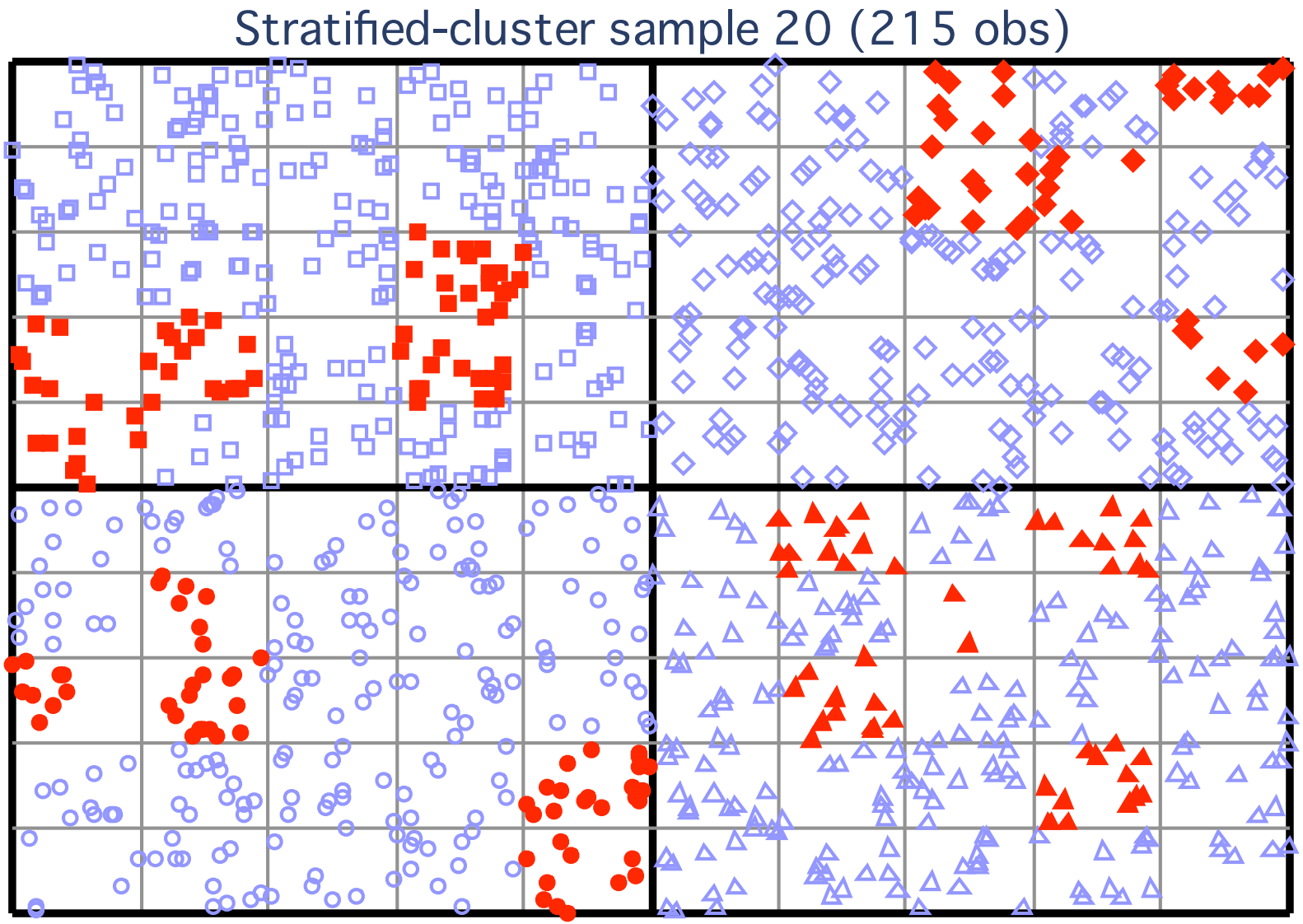

Population 1000


SRS sample 200


Cluster sample 20 (208 obs)

Stratified sample 198




Stratified-cluster sample 20 (215 obs)

We are now finished with this lesson.

```
. log close
```

Start a new log for this lesson.

```
. log using lesson5
```

Single-stage syntax

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

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

- Simple Random Sample:

```
. sysuse auto, clear
. svyset _n
. svy: regress mpg weight
```

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

```
. webuse nmihs, clear
. describe
. svyset [pw=finwgt], strata(stratan)
. svy: tabulate agegrp lowbw
```

- Single-stage design with all the characteristics:

```
. webuse fpc, clear
. describe
. svyset psuid [pw=weight], strata(stratid) fpc(Nh)
```

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)
. save myseniors
. svy: logit smoked i.gender i.race, or
. test 2.race 3.race
```

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

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
. bysort type: sum totexp

. codebook weight
. svyset [pw=weight]
. svy: mean totexp
. codebook postwgt fpc
. svyset [pw=weight], poststrata(type) postweight(postwgt) fpc(fpc)
. svy: mean totexp
. svy: total totexp
```

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
. svydescribe
```

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

```
. codebook hdresult
. svy: mean hdresult
. svydescribe if e(sample), single
```

- Even better would be

```
. svydescribe hdresult, single
```

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
. svyset psu [pw=finalwgt], strata(strata) singleunit(certainty)
. svy: mean hdresult
```

We are now finished with this lesson.

```
. log close
```

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)

Linearization

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

- Foundation: Variance of the total estimator

Syntax

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

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

Total estimator – Stratified two-stage design

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

$$\widehat{Y} = \sum w_{hijk} y_{hijk}$$

$$\widehat{V}(\widehat{Y}) = \sum_h (1 - f_h) \frac{n_h}{n_h - 1} \sum_i (y_{hi} - \overline{y}_h)^2 + \sum_h f_h \sum_i (1 - f_{hi}) \frac{m_{hi}}{m_{hi} - 1} \sum_j (y_{hij} - \overline{y}_{hi})^2$$

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

```
. use myseniors, clear
. svyset
. svy: total smoked
```

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

```
. svyset county [pw=sampwgt], strata(state) fpc(ncounties)
. svy: total smoked
```

Linearized variance for regression models

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

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

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

- For this, we return to NHANES2

```
. webuse nhanes2, clear
. svyset
```

- and fit a logit model for high blood pressure:

```
. describe highbp height weight age female race
. discard
. local model highbp height weight c.age##c.age i.female i.race
. svy: logit `model´, baselevel
. 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
```

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]
```

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

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

BRR variance formulas

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

Default variance formula:

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

Mean squared error (MSE) formula:

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

- 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
. svy: logit `model´, baselevel
```

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

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

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]
```

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

- $\widehat{\theta}_{(h,i)}$ – replicate of the point estimates from stratum h, PSU i
- $\overline{\theta}_h$ – 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:

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

Mean squared error (MSE) formula:

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

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

```
. webuse nhanes2, clear
. svyset
. svy jackknife, mse: logit `model´, baselevel
```

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

```
. svyset psu [pw=finalwgt], strata(strata) vce(jackknife, mse)
. svy: logit `model´, baselevel
```

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

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

Default variance formula:

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

Mean squared error (MSE) formula:

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

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

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

Default variance formula:

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

Mean squared error (MSE) formula:

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

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)
```

- Consider a privacy-conscious version of NHANES:

```
. webuse nhanes2jknife, clear
. svyset
. 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
```

We are now finished with this lesson.

```
. log close
```

Start a new log for this lesson.

```
. log using lesson7
```

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.

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.

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

- Returning the the National Maternal and Infant Health Survey

```
. webuse nmihs, clear
. svyset
. des birthwgt highbp
. label list hibp
```

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

```
. svy, subpop(highbp): mean birthwgt
```

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

```
. svy: mean birthwgt if highbp
```

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

```
. svy, subpop(if !highbp): mean birthwgt
```

- How about across values of a categorical variable?

```
. codebook agegrp
. svy: mean birthwgt, over(agegrp)
```

We are now finished with this lesson.

```
. log close
```

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`**

- 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
```

- 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
```

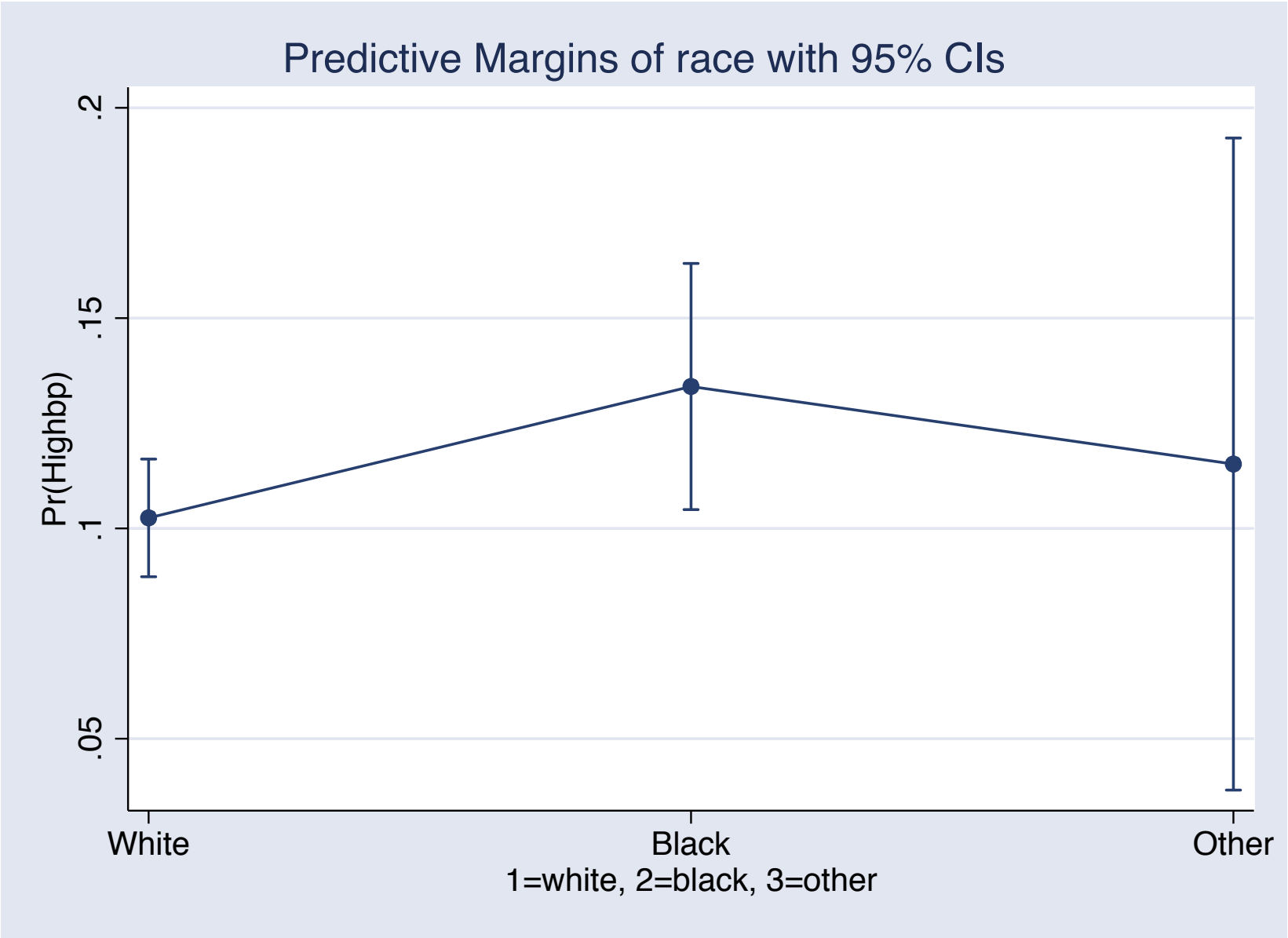
Predictive margins

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

```
. describe race
. label list race
```

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

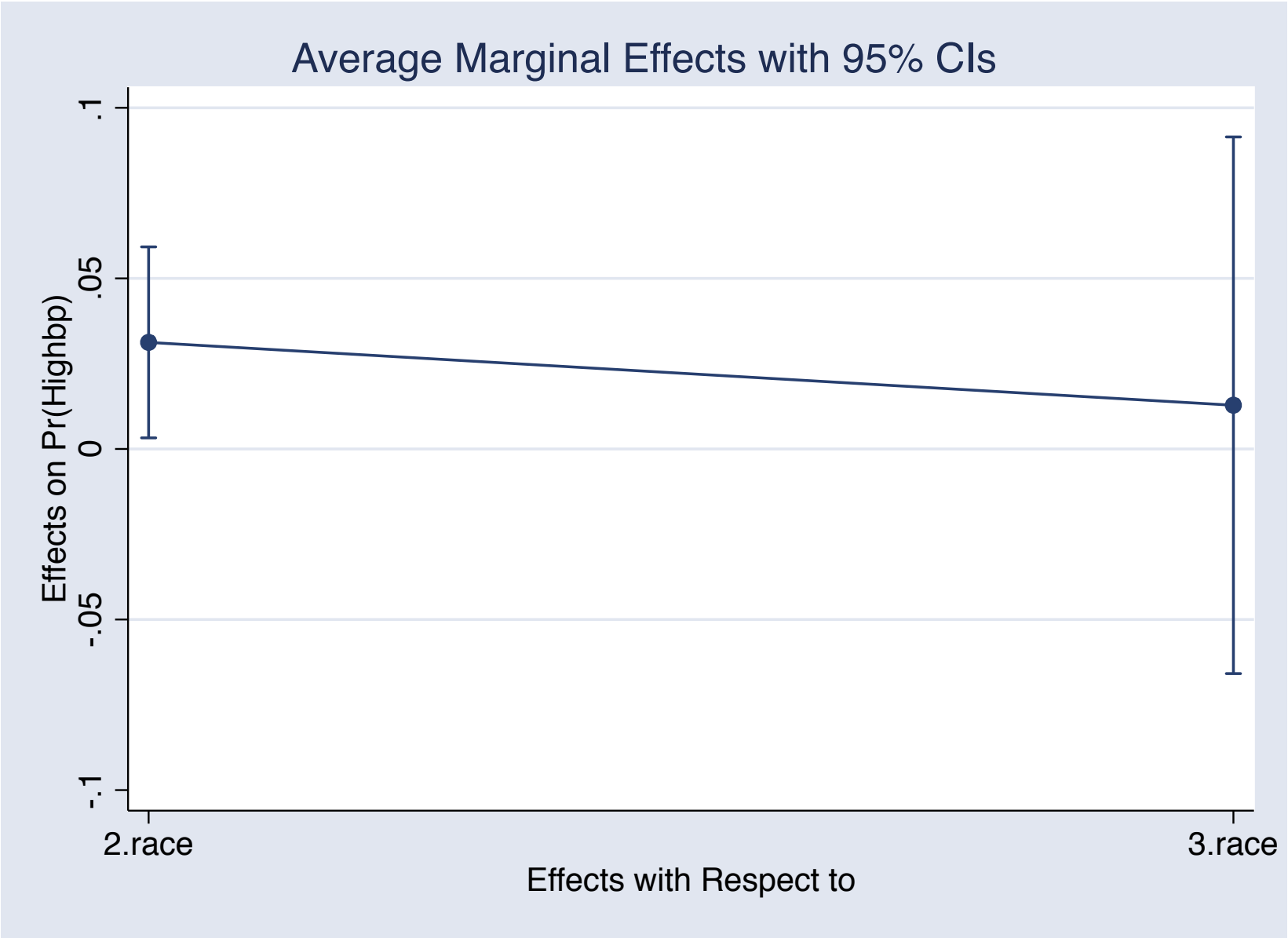
```
. margins race, vce(unconditional)
. marginsplot
```


Predictive Margins of race with 95% CIs
Pr(Highbp)
.05
.1
.15
.2
White
Black
Other
1=white, 2=black, 3=other

Marginal effects

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

```
. margins, vce(unconditional) dydx(race)
. marginsplot
```

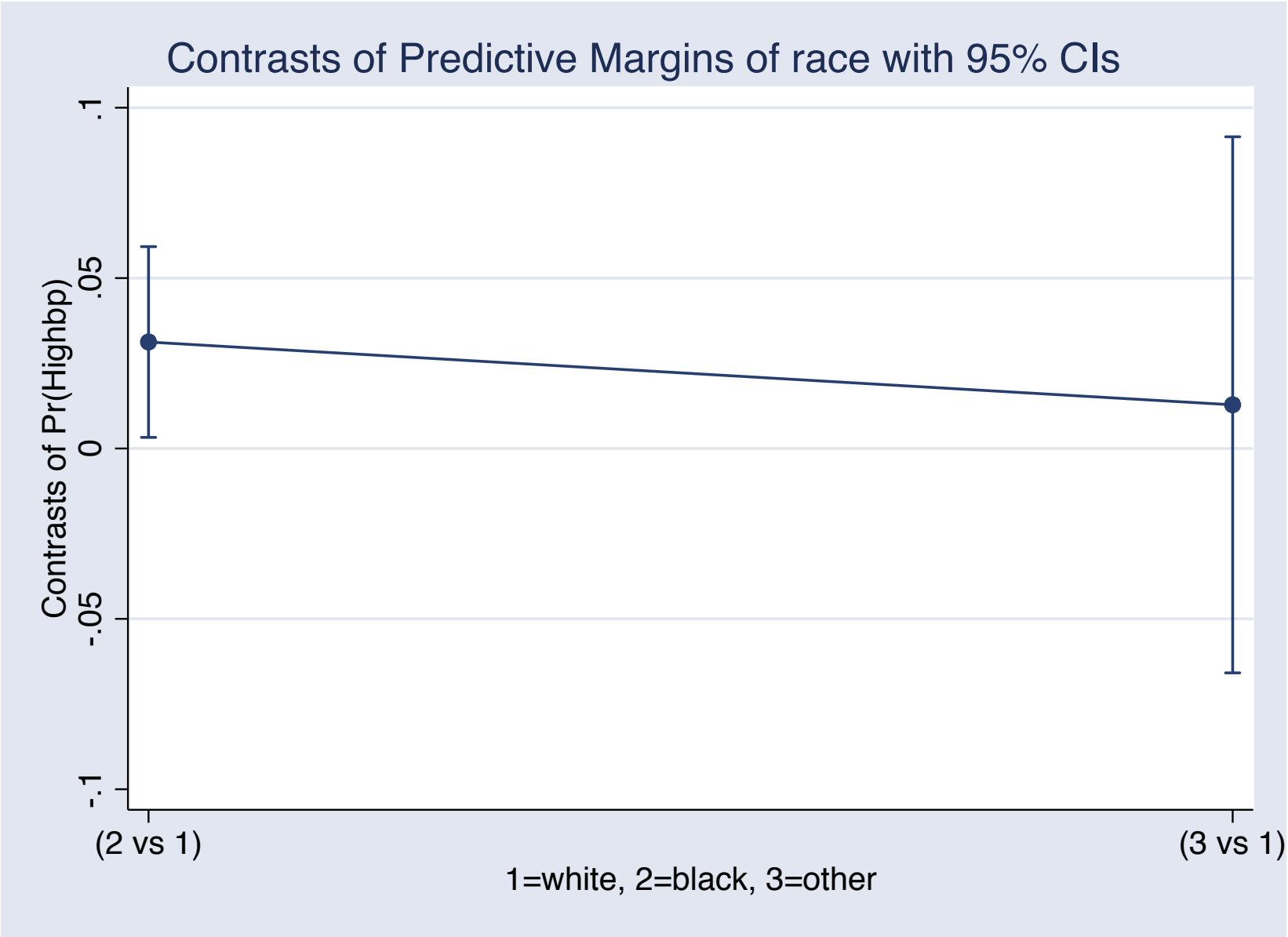

Average Marginal Effects with 95% CIs
Effects on Pr(Highbp)
.1
.05
0
-.05
-.1
2.race
3.race
Effects with Respect to

Contrasts

- The **`contrast`** command and contrast operators are new in Stata 12.
- **`margins`** has a richer set of operators for computing discrete marginal effects.

```
. margins r.race, vce(unconditional)
. marginsplot
```

- Here is a profile plot corresponding to these marginal effects.


Contrasts of Predictive Margins of race with 95% CIs
Contrasts of Pr(Highbp)
.1
.05
0
-.05
-.1
(2 vs 1)
(3 vs 1)
1=white, 2=black, 3=other

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.


Levy, P. and S. Lemeshow. 1999.
Sampling of Populations. 3rd ed.
New York: Wiley.


StataCorp. 2011.
Survey Data Reference Manual: Release 12.
College Station, TX: StataCorp LP.