Repeated half-sample bootstrap resampling

Philippe Van Kerm

CEPS/INSTEAD, Luxembourg

philippe.vankerm@ceps.lu

2013 London Stata Users Group meeting

September 12–13 2013, Cass Business School, London

Bootstrap variance estimation

- Bootstrap resampling provides convenient variance estimators (esp. for complicated statistics): resample, re-estimate, combine...
- Typical complex survey design features:
  - Stratification
  - Multi-stage sample selection
  - Unequal selection probabilities
  - Sampling without replacement
  - Imputation, non-response adjustments, post-stratification

$\Longrightarrow$ More elaborate bootstrap procedure required (to preserve the dependence structure of the data as much as possible)

Bootstrap variance estimation

- Bootstrap resampling provides convenient variance estimators (esp. for complicated statistics): resample, re-estimate, combine...
- Typical complex survey design features:
  - Stratification
  - Multi-stage sample selection
  - Unequal selection probabilities
  - Sampling without replacement
  - Imputation, non-response adjustments, post-stratification

$\Longrightarrow$ More elaborate bootstrap procedure required (to preserve the dependence structure of the data as much as possible)

Survey bootstrap

- Basic, naïve survey bootstrap: draw PSUs with replacement within each stratum
  - Variance estimates severely biased downwards if number of PSU/stratum is small
- Specialized survey bootstrap resampling plans (or other resampling methods such as jackknife or BRR; see Kolenikov, *Stata Journal* 2010), such as
  - *m* out of *n* bootstrap resampling with rescaling (Rao and Wu, *JASA* 1988; see `bsweights`)
  - repeated half-sample bootstrap (Saigo, Shao and Sitter, *Survey Methodology* 2001)

  $\Longrightarrow$ simple, consistent for any PSU/stratum, applicable to wide array of estimators and allows re-imputation of bootstrap samples

Survey bootstrap

- Basic, naïve survey bootstrap: draw PSUs with replacement within each stratum
  - Variance estimates severely biased downwards if number of PSU/stratum is small
- Specialized survey bootstrap resampling plans (or other resampling methods such as jackknife or BRR; see Kolenikov, *Stata Journal* 2010), such as
  - m out of n bootstrap resampling with rescaling (Rao and Wu, *JASA* 1988; see `bsweights`)
  - repeated half-sample bootstrap (Saigo, Shao and Sitter, *Survey Methodology* 2001)

  $\Longrightarrow$ simple, consistent for any PSU/stratum, applicable to wide array of estimators and allows re-imputation of bootstrap samples

Survey bootstrap

- Basic, naïve survey bootstrap: draw PSUs with replacement within each stratum
  - Variance estimates severely biased downwards if number of PSU/stratum is small
- Specialized survey bootstrap resampling plans (or other resampling methods such as jackknife or BRR; see Kolenikov, *Stata Journal* 2010), such as
  - *m* out of *n* bootstrap resampling with rescaling (Rao and Wu, *JASA* 1988; see **`bsweights`**)
  - repeated half-sample bootstrap (Saigo, Shao and Sitter, *Survey Methodology* 2001)

$\longrightarrow$ simple, consistent for any PSU/stratum, applicable to wide array of estimators and allows re-imputation of bootstrap samples

Survey bootstrap

- Basic, naïve survey bootstrap: draw PSUs with replacement within each stratum
  - Variance estimates severely biased downwards if number of PSU/stratum is small
- Specialized survey bootstrap resampling plans (or other resampling methods such as jackknife or BRR; see Kolenikov, *Stata Journal* 2010), such as
  - *m* out of *n* bootstrap resampling with rescaling (Rao and Wu, *JASA* 1988; see **`bsweights`**)
  - repeated half-sample bootstrap (Saigo, Shao and Sitter, *Survey Methodology* 2001)

$\Longrightarrow$ simple, consistent for any PSU/stratum, applicable to wide array of estimators and allows re-imputation of bootstrap samples

Repeated half-sample bootstrap mechanics

- When sample size N (PSUs per stratum) is even:
  - draw without replacement a sample of size $N/2$
  - duplicate each obs so the bootstrap sample has size N
- When N is odd, either (with probability $1/4$) ...
  - draw without replacement a sample of size $(N-1)/2$
  - duplicate each obs so the bootstrap sample has size $N-1$
  - triplicate one obs at random
- ... or (with probability $3/4$) ...
  - draw without replacement a sample of size $1+(N-1)/2$
  - duplicate each obs so the bootstrap sample has size $N+1$
  - remove one obs at random

Repeated half-sample bootstrap mechanics

- When sample size N (PSUs per stratum) is even:
  - draw without replacement a sample of size $N/2$
  - duplicate each obs so the bootstrap sample has size N
- When N is odd, either (with probability $1/4$) ...
  - draw without replacement a sample of size $(N-1)/2$
  - duplicate each obs so the bootstrap sample has size $N-1$
  - triplicate one obs at random
- ... or (with probability $3/4$) ...
  - draw without replacement a sample of size $1+(N-1)/2$
  - duplicate each obs so the bootstrap sample has size $N+1$
  - remove one obs at random

The **rhsbsample** command: Syntax

The command **rhsbsample** (available on SSC shortly) clones official **bsample** but samples on the basis of the repeated half-sample algorithm

Syntax

rhsbsample [*if*] [*in*]

[, strata(*varlist*) cluster(*varlist*) idcluster(*newvarname*)

weight(*varname*)]

The **rhsbsample** command: Usage example

Replace data in memory by a bootstrap sample:

Draw a (stratified clustered) bootstrap sample:

```
bsample    , strata(<strata>) cluster(<psu>)
rhsbsample , strata(<strata>) cluster(<psu>)
```

The **rhsbsample** command: Usage example

Generate 500 bootstrap replication weight variables:

Draw 500 (stratified clustered) bootstrap samples:

```
forvalues i=1/500 {
    qui gen brw‘i’ = .
    bsample    , strata(<strata>) cluster(<psu>) weight(nbrw‘i’)
    qui gen rhsbrw‘i’ = .
    rhsbsample , strata(<strata>) cluster(<psu>) weight(rhsbrw‘i’)
}
```

The **rhsbsample** command: Combination with **svy** prefix

Declare generated bootstrap replication weights in **svyset**:

Declare survey settings and use **svy** prefix:

```
svyset <psu> [pw=<wgt>], strata(<strata>)  ///
       vce(bootstrap) bsrweight(rhsbrw*)
svy : logistic ...
```

Illustrative example on nhanes2 data

Load data

```
. use nhanes2.dta

. * collapsed strata -- Kolenikov 2010
. egen upsu = group( strata psu )

. gen cstrata = floor( sqrt( 2*strata-1) )

.
. svyset upsu [pw=finalwgt], strata(cstrata)

      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: cstrata
         SU 1: upsu
        FPC 1: <zero>
```

Illustrative example on nhanes2 data (ctd.)

Survey design

```
. svydes

Survey: Describing stage 1 sampling units

      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: cstrata
         SU 1: upsu
        FPC 1: <zero>
```

Small stratum sizes

Stratum	#Units	#Obs	#Obs per Unit		
			min	mean	max
1	4	565	67	141.3	215
2	4	808	149	202.0	231
3	8	1364	105	170.5	270
4	8	1095	100	136.9	170
5	12	2215	144	184.6	215
6	10	1579	102	157.9	233
7	16	2725	116	170.3	288
7	62	10351	67	167.0	288

Illustrative example on nhanes2 data (ctd.)

Create replication weights

```
.
. * STEP 1: generate sets of bootstrap weights:
. loc R 500

. * 01: Naive bootstrap
. forvalues i=1/`R' {
  2.    qui gen nbrw`i' = .
  3.    bsample , strata(cstrata) cluster(upsu) weight(nbrw`i')
  4.    qui replace nbrw`i' = nbrw`i' * finalwgt
  5. }

. * 02: RHS bootstrap
. forvalues i=1/`R' {
  2.    qui gen rhsbrw`i' = .
  3.    rhsbsample , strata(cstrata) cluster(upsu) weight(rhsbrw`i')
  4.    qui replace rhsbrw`i' = rhsbrw`i' * finalwgt
  5. }

. * 03: simple rescaled bootstrap (bsweights reads information from svyset)
. bsweights rsbrw , reps(`R') n(-1)
```

multiply sample (design) weights

user-written command by Kolenikov

Illustrative example on nhanes2 data (ctd.)

Linearization-based variance

```
. svyset upsu [pw=finalwgt], clear strata(cstrata) vce(linearized)

      pweight: finalwgt
          VCE: linearized
  Single unit: missing
     Strata 1: cstrata
         SU 1: upsu
        FPC 1: <zero>

. svy: mean highbp height weight
(running mean on estimation sample)

Survey: Mean estimation

Number of strata =         7        Number of obs    =       10351
Number of PSUs   =        62        Population size  =   117157513
                                   Design df        =          55
```

	Mean	Linearized Std. Err.	[95% Conf.	Interval]
highbp	.1058141	.0065151	.0927576	.1188706
height	168.4599	.136994	168.1853	168.7344
weight	71.90064	.1675056	71.56495	72.23632

Illustrative example on nhanes2 data (ctd.)

Declare replication weights

```
. svyset upsu [pw=finalwgt], clear strata(cstrata) vce(bootstrap) bsrweight(nbrw*) mse

      pweight: finalwgt
          VCE: bootstrap
          MSE: on
    bsrweight: nbrw1 nbrw2 nbrw3 nbrw4 nbrw5 nbrw6 nbrw7 nbrw8 nbrw9 nbrw10 nbrw11 nbrw12
               nbrw13 nbrw14 nbrw15 nbrw16 nbrw17 nbrw18 nbrw19 nbrw20 nbrw21 nbrw22
               nbrw23 nbrw24 nbrw25 nbrw26 nbrw27 nbrw28 nbrw29 nbrw30 nbrw31 nbrw32
               nbrw33 nbrw34 nbrw35 nbrw36 nbrw37 nbrw38 nbrw39 nbrw40 nbrw41 nbrw42
               nbrw43 nbrw44 nbrw45 nbrw46 nbrw47 nbrw48 nbrw49 nbrw50 nbrw51 nbrw52
               nbrw53 nbrw54 nbrw55 nbrw56 nbrw57 nbrw58 nbrw59 nbrw60 nbrw61 nbrw62
               nbrw63 nbrw64 nbrw65 nbrw66 nbrw67 nbrw68 nbrw69 nbrw70 nbrw71 nbrw72
               nbrw73 nbrw74 nbrw75 nbrw76 nbrw77 nbrw78 nbrw79 nbrw80 nbrw81 nbrw82
               nbrw83 nbrw84 nbrw85 nbrw86 nbrw87 nbrw88 nbrw89 nbrw90 nbrw91 nbrw92
               nbrw93 nbrw94 nbrw95 nbrw96 nbrw97 nbrw98 nbrw99 nbrw100 nbrw101 nbrw102
```

Illustrative example on nhanes2 data (ctd.)

Naive bootstrap variance

```
. svy , nodots : mean highbp height weight

Survey: Mean estimation              Number of obs      =     10351
                                     Population size    = 117157513
                                     Replications       =       500
```

	Observed Mean	Bstrap * Std. Err.	[95% Conf.	Interval]
highbp	.1058141	.0061895	.0936828	.1179453
height	168.4599	.1348131	168.1957	168.7241
weight	71.90064	.1567976	71.59332	72.20795

```
. estimates store mn_naive
```

works with non-svy-aware, user-written commands

```
. svy bootstrap (r(coeff)) , nodots : sgini height

Bootstrap results                    Number of obs      =     10351
                                     Population size    = 117157513
                                     Replications       =       500

      command:  sgini height
        _bs_1:  r(coeff)
```

	Observed Coef.	Bstrap * Std. Err.	z	P>|z|	[95% Conf.	Interval]

Illustrative example on nhanes2 data (ctd.)

RHS replication weights

```
. svyset upsu [pw=finalwgt], clear strata(cstrata) vce(bootstrap) bsrweight(rhsbrw*) mse

      pweight: finalwgt
          VCE: bootstrap
          MSE: on
   bsrweight: rhsbrw1 rhsbrw2 rhsbrw3 rhsbrw4 rhsbrw5 rhsbrw6 rhsbrw7 rhsbrw8 rhsbrw9
              rhsbrw10 rhsbrw11 rhsbrw12 rhsbrw13 rhsbrw14 rhsbrw15 rhsbrw16 rhsbrw17
              rhsbrw18 rhsbrw19 rhsbrw20 rhsbrw21 rhsbrw22 rhsbrw23 rhsbrw24 rhsbrw25
              rhsbrw26 rhsbrw27 rhsbrw28 rhsbrw29 rhsbrw30 rhsbrw31 rhsbrw32 rhsbrw33
              rhsbrw34 rhsbrw35 rhsbrw36 rhsbrw37 rhsbrw38 rhsbrw39 rhsbrw40 rhsbrw41
              rhsbrw42 rhsbrw43 rhsbrw44 rhsbrw45 rhsbrw46 rhsbrw47 rhsbrw48 rhsbrw49
```

Illustrative example on nhanes2 data (ctd.)

RHS bootstrap variance

```
. svy , nodots : mean highbp height weight

Survey: Mean estimation                 Number of obs    =     10351
                                        Population size  =  117157513
                                        Replications     =       500
```

	Observed Mean	Bstrap * Std. Err.	[95% Conf.	Interval]
highbp	.1058141	.0067235	.0926363	.1189918
height	168.4599	.1400124	168.1855	168.7343
weight	71.90064	.1677603	71.57183	72.22944

```
. estimates store mn_rhs

. svy bootstrap (r(coeff)) , nodots : sgini height

Bootstrap results                       Number of obs    =     10351
                                        Population size  =  117157513
                                        Replications     =       500

      command:  sgini height
        _bs_1:  r(coeff)
```

	Observed	Bstrap *

Illustrative example on nhanes2 data (ctd.)

SE estimates compared

```
.
. estimates tab mn_lin mn_naive mn_rescaled mn_rhs , se
```

Variable	mn_lin	mn_naive	mn_resca~d	mn_rhs
highbp	.10581408	.10581408	.10581408	.10581408
	.00651508	.00618952	.00628382	.00672347
height	168.45989	168.45989	168.45989	168.45989
	.13699399	.13481314	.1400032	.14001242
weight	71.900636	71.900636	71.900636	71.900636
	.16750564	.15679756	.17280145	.16776031

legend: b/se

```
. estimates tab        gini_naive gini_rescaled gini_rhs  , se
```

Variable	gini_naive	gini_res~d	gini_rhs
_bs_1	.03282964	.03282964	.03282964
	.00023417	.00026128	.00024105

legend: b/se

References

- Kolenikov S. (2010). Resampling variance estimation for complex survey data. Stata Journal, 10(2): 165–199.
- Rao, J.N.K. and Wu, C.F.J. (1988). Resampling inference with complex survey data. Journal of the American Statistical Association, 83: 231–241.
- Saigo, H., Shao, J. and Sitter, R.R. (2001). A Repeated Half-Sample Bootstrap and Balanced Repeated Replications for Randomly Imputed Data. Survey Methodology, 27(2): 189–196.

Acknowledgements

This work is part of the project *"Information and Wage Inequality"* supported by the Luxembourg Fonds National de la Recherche (contract C10/LM/785657).