bsample — Sampling with replacement

Description

`bsample` replaces the data in memory with a bootstrap sample (random sample with replacement) drawn from the current dataset. Clusters can be optionally sampled during each replication in place of observations. Bootstrap samples can also be selected within strata.

Quick start

Bootstrap sample with the same number of observations as the current dataset

```
bsample
```

Same as above, but restrict to just 100 observations

```
bsample 100
```

Stratified bootstrap sample of 100 observations at each level of `svar`

```
bsample 100, strata(svar)
```

Bootstrap sample of 10 clusters identified by values of `cvar`

```
bsample 10, cluster(cvar)
```

Same as above, but create a new unique ID code for sampled clusters and store it in `cvar2`

```
bsample 10, cluster(cvar) idcluster(cvar2)
```

Menu

Statistics > Resampling > Draw bootstrap sample

Syntax

`bsample` [*exp*] [*if*] [*in*] [, *options*]

where *exp* is a standard Stata expression specifying the size of the sample; see **[U] 13 Functions and expressions**.

exp must be less than or equal to `_N` (the number of observations; **[U] 13.4 System variables (_variables)**) when neither the `cluster()` nor the `strata()` option is specified. `_N` is the default when *exp* is not specified.

Observations that do not meet the optional `if` and `in` criteria are dropped from the resulting dataset.

options	Description
`strata(`*varlist*`)`	variables identifying strata
`cluster(`*varlist*`)`	variables identifying resampling clusters
`idcluster(`*newvar*`)`	create new cluster ID variable
`weight(`*varname*`)`	replace *varname* with frequency weights

Options

`strata(`*varlist*`)` specifies the variables identifying strata. If `strata()` is specified, bootstrap samples are selected within each stratum, and *exp* must be less than or equal to `_N` within the defined strata.

`cluster(`*varlist*`)` specifies the variables identifying resampling clusters. If `cluster()` is specified, the sample drawn during each replication is a bootstrap sample of clusters, and *exp* must be less than or equal to N_c (the number of clusters identified by the `cluster()` option). If `strata()` is also specified, *exp* must be less than or equal to the number of within-strata clusters.

`idcluster(`*newvar*`)` creates a new variable containing a unique identifier for each resampled cluster.

`weight(`*varname*`)` specifies a variable in which the sampling frequencies will be placed. *varname* must be an existing variable, which will be replaced. After `bsample`, *varname* can be used as an `fweight` in any Stata command that accepts `fweights`, which can speed up resampling for commands like `regress` and `summarize`. This option cannot be combined with `idcluster()`.

By default, `bsample` replaces the data in memory with the sampled observations; however, specifying the `weight()` option causes only the specified *varname* to be changed.

Remarks and examples

Below is a series of examples illustrating how `bsample` is used with various sampling schemes.

▷ Example 1: Bootstrap sampling

We have data on the characteristics of hospital patients and wish to draw a bootstrap sample of 200 patients. We type

```
. use https://www.stata-press.com/data/r19/bsample1
. bsample 200
. count
  200
```

◁

▷ Example 2: Stratified samples with equal sizes

Among the variables in our dataset is `female`, an indicator for the female patients. To get a bootstrap sample of 200 female patients and 200 male patients, we type

```
. use https://www.stata-press.com/data/r19/bsample1, clear
. bsample 200, strata(female)
. tabulate female

   Indicator |
  for female |      Freq.     Percent        Cum.
-------------+-----------------------------------
        Male |        200       50.00       50.00
      Female |        200       50.00      100.00
-------------+-----------------------------------
       Total |        400      100.00
```

◁

▷ Example 3: Stratified samples with unequal sizes

To sample 300 females and 200 males, we must generate a variable that is 300 for females and 200 for males and then use this variable in *exp* when we call `bsample`.

```
. use https://www.stata-press.com/data/r19/bsample1, clear
. generate nsamp = cond(female,300,200)
. bsample nsamp, strata(female)
. tabulate female

   Indicator |
  for female |      Freq.     Percent        Cum.
-------------+-----------------------------------
        Male |        200       40.00       40.00
      Female |        300       60.00      100.00
-------------+-----------------------------------
       Total |        500      100.00
```

◁

Example 4: Stratified samples with proportional sizes

Our original dataset has 2,392 males and 3,418 females.

```
. use https://www.stata-press.com/data/r19/bsample1, clear

. tabulate female

   Indicator |
  for female |      Freq.     Percent        Cum.
-------------+-----------------------------------
        Male |      2,392       41.17       41.17
      Female |      3,418       58.83      100.00
-------------+-----------------------------------
       Total |      5,810      100.00
```

To sample 10% from females and males, we type

```
. bsample round(0.1*_N), strata(female)
```

`bsample` requires that the specified size of the sample be an integer, so we use the `round()` function to obtain the nearest integer to 0.1 × 2392 and 0.1 × 3418. Our sample now has 239 males and 342 females:

```
. tabulate female

   Indicator |
  for female |      Freq.     Percent        Cum.
-------------+-----------------------------------
        Male |        239       41.14       41.14
      Female |        342       58.86      100.00
-------------+-----------------------------------
       Total |        581      100.00
```

Example 5: Samples satisfying a condition

For a bootstrap sample of 200 female patients, we type

```
. use https://www.stata-press.com/data/r19/bsample1, clear

. bsample 200 if female

. tabulate female

   Indicator |
  for female |      Freq.     Percent        Cum.
-------------+-----------------------------------
      Female |        200      100.00      100.00
-------------+-----------------------------------
       Total |        200      100.00
```

▷ Example 6: Generating frequency weights

To identify the sampled observations using frequency weights instead of dropping unsampled observations, we use the `weight()` option (we will need to supply it an existing variable name) and type

```
. use https://www.stata-press.com/data/r19/bsample1, clear

. set seed 1234

. generate fw = .
(5,810 missing values generated)

. bsample 200 if female, weight(fw)

. tabulate fw female

                 Indicator for female
        fw        Male      Female         Total
                                                 
         0       2,392       3,222         5,614 
         1           0         192           192 
         2           0           4             4 
                                                 
     Total       2,392       3,418         5,810 
```

Note that $(192 \times 1) + (4 \times 2) = 200$.

◁

▷ Example 7: Oversampling observations

`bsample` requires the expression in *exp* to evaluate to a number that is less than or equal to the number of observations. To sample twice as many male and female patients as there are already in memory, we must expand the data before using `bsample`. For example,

```
. use https://www.stata-press.com/data/r19/bsample1, clear

. set seed 1234

. expand 2
(5,810 observations created)

. bsample, strata(female)

. tabulate female

  Indicator
 for female        Freq.     Percent        Cum.
                                                
       Male        4,784       41.17       41.17
     Female        6,836       58.83      100.00
                                                
      Total       11,620      100.00
```

◁

▷ Example 8: Stratified oversampling with unequal sizes

To sample twice as many female patients as male patients, we must expand the records for the female patients because there are less than twice as many of them as there are male patients, but first put the number of observed male patients in a local macro. After expanding the female records, we generate a variable that contains the number of observations to sample within the two groups.

```
. use https://www.stata-press.com/data/r19/bsample1, clear
. set seed 1234
. count if !female
  2,392
. local nmale = r(N)
. expand 2 if female
(3,418 observations created)
. generate nsamp = cond(female,2*‘nmale’,‘nmale’)
. bsample nsamp, strata(female)
. tabulate female

  Indicator │
 for female │      Freq.     Percent        Cum.
────────────┼───────────────────────────────────
       Male │      2,392       33.33       33.33
     Female │      4,784       66.67      100.00
────────────┼───────────────────────────────────
      Total │      7,176      100.00
```

Example 9: Oversampling of clusters

For clustered data, sampling more clusters than are present in the original dataset requires more than just expanding the data. To illustrate, suppose we wanted a bootstrap sample of eight clusters from a dataset consisting of five clusters of observations.

```
. use https://www.stata-press.com/data/r19/bsample2, clear
. tabstat x, stat(n mean) by(group)
Summary for variables: x
Group variable: group

 group │         N      Mean
───────┼────────────────────
     A │        15 -.3073028
     B │        10   -.00984
     C │        11  .0810985
     D │        11 -.1989179
     E │        29  -.095203
───────┼────────────────────
 Total │        76 -.1153269
────────────────────────────
```

bsample will complain if we simply expand the dataset.

```
. use https://www.stata-press.com/data/r19/bsample2
. expand 3
(152 observations created)
. bsample 8, cluster(group)
resample size must not be greater than number of clusters
r(498);
```

Expanding the data will only partly solve the problem. We also need a new variable that uniquely identifies the copied clusters. We use the expandcl command to accomplish both these tasks; see [D] **expandcl**.

```
. use https://www.stata-press.com/data/r19/bsample2, clear

. set seed 1234

. expandcl 2, generate(expgroup) cluster(group)
(76 observations created)

. tabstat x, stat(n mean) by(expgroup)

Summary for variables: x
Group variable: expgroup (New cluster ID from expandcl)

expgroup |         N      Mean
---------+--------------------
       1 |        15 -.3073028
       2 |        15 -.3073028
       3 |        10   -.00984
       4 |        10   -.00984
       5 |        11  .0810985
       6 |        11  .0810985
       7 |        11 -.1989179
       8 |        11 -.1989179
       9 |        29  -.095203
      10 |        29  -.095203
---------+--------------------
   Total |       152 -.1153269
------------------------------

. generate fw = .
(152 missing values generated)

. bsample 8, cluster(expgroup) weight(fw)

. tabulate fw group

           |                          group
        fw |         A          B          C          D          E |     Total
-----------+-------------------------------------------------------+----------
         0 |        15         10         22         11         58 |       116 
         1 |         0          0          0         11          0 |        11 
         2 |        15          0          0          0          0 |        15 
         5 |         0         10          0          0          0 |        10 
-----------+-------------------------------------------------------+----------
     Total |        30         20         22         22         58 |       152 
```

The results from `tabulate` on the generated frequency weight variable versus the original cluster ID (`group`) show us that the bootstrap sample contains one copy of cluster `A`, one copy of cluster `B`, two copies of cluster `C`, two copies of cluster `D`, and two copies of cluster `E` ($1 + 1 + 2 + 2 + 2 = 8$).

◁

▷ Example 10: Stratified oversampling of clusters

Suppose that we have a dataset containing two strata with five clusters in each stratum, but the cluster identifiers are not unique between the strata. To get a stratified bootstrap sample with eight clusters in each stratum, we first use `expandcl` to expand the data and get a new cluster ID variable. We use `cluster(strid group)` in the call to `expandcl`; this action will uniquely identify the $2 * 5 = 10$ clusters across the strata.

```
. use https://www.stata-press.com/data/r19/bsample2, clear

. set seed 1234

. tabulate group strid

           |         strid
     group |         1          2 |     Total
-----------+----------------------+----------
         A |         7          8 |        15
         B |         5          5 |        10
         C |         5          6 |        11
         D |         5          6 |        11
         E |        14         15 |        29
-----------+----------------------+----------
     Total |        36         40 |        76

. expandcl 2, generate(expgroup) cluster(strid group)
(76 observations created)
```

Now, we can use bsample with the expanded data, stratum ID variable, and new cluster ID variable.

```
. generate fw = .
(152 missing values generated)

. bsample 8, cluster(expgroup) str(strid) weight(fw)

. by strid, sort: tabulate fw group

-------------------------------------------------------------------------------
-> strid = 1

           |                          group
        fw |         A          B          C          D          E |     Total
-----------+-------------------------------------------------------+----------
         0 |         7          0         10          5         14 |        36
         1 |         0          5          0          5         14 |        24
         2 |         7          0          0          0          0 |         7
         3 |         0          5          0          0          0 |         5
-----------+-------------------------------------------------------+----------
     Total |        14         10         10         10         28 |        72

-------------------------------------------------------------------------------
-> strid = 2

           |                          group
        fw |         A          B          C          D          E |     Total
-----------+-------------------------------------------------------+----------
         0 |         0          0         12          0         15 |        27
         1 |        16          5          0         12         15 |        48
         2 |         0          5          0          0          0 |         5
-----------+-------------------------------------------------------+----------
     Total |        16         10         12         12         30 |        80
```

The results from by strid: tabulate on the generated frequency weight variable versus the original cluster ID (group) show us how many times each cluster was sampled for each stratum. For stratum 1, the bootstrap sample contains two copies of cluster A, one copy of cluster B, two copies of cluster C, one copy of cluster D, and two copies of cluster E ($2+1+2+1+2=8$). For stratum 2, the bootstrap sample contains one copy of cluster A, zero copies of cluster B, three copies of cluster C, one copy of cluster D, and three copies of cluster E ($1+0+3+1+3=8$).

◁

References

Gould, W. W. 2012a. Using Stata's random-number generators, part 2: Drawing without replacement. *The Stata Blog: Not Elsewhere Classified*. https://blog.stata.com/2012/08/03/using-statas-random-number-generators-part-2-drawing-without-replacement/.

———. 2012b. Using Stata's random-number generators, part 3: Drawing with replacement. *The Stata Blog: Not Elsewhere Classified*. https://blog.stata.com/2012/08/29/using-statas-random-number-generators-part-3-drawing-with-replacement/.

Also see

[R] **bootstrap** — Bootstrap sampling and estimation

[R] **bstat** — Report bootstrap results

[R] **simulate** — Monte Carlo simulations

[D] **sample** — Draw random sample

[D] **splitsample** — Split data into random samples

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on citing Stata documentation.