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
```

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)
```

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
bsample — Sampling with replacement

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.

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strata(varlist)</td>
<td>variables identifying strata</td>
</tr>
<tr>
<td>cluster(varlist)</td>
<td>variables identifying resampling clusters</td>
</tr>
<tr>
<td>idcluster(newvar)</td>
<td>create new cluster ID variable</td>
</tr>
<tr>
<td>weight(varname)</td>
<td>replace varname with frequency weights</td>
</tr>
</tbody>
</table>

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/r16/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/r16/bsample1, clear
  . bsample 200, strata(female)
  . tabulate female
  
<table>
<thead>
<tr>
<th>female</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>male</td>
<td>200</td>
<td>50.00</td>
<td>50.00</td>
</tr>
<tr>
<td>female</td>
<td>200</td>
<td>50.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>
  
  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/r16/bsample1, clear
  . generate nsamp = cond(female,300,200)
  . bsample nsamp, strata(female)
  . tabulate female
  
<table>
<thead>
<tr>
<th>female</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>male</td>
<td>200</td>
<td>40.00</td>
<td>40.00</td>
</tr>
<tr>
<td>female</td>
<td>300</td>
<td>60.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>
  
  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/r16/bsample1, clear
. tabulate female
female | Freq.  Percent Cum.  
-------------|---------|---------|
   male      |  2392   |  41.17  |  41.17 |
 female     |  3418   |  58.83  | 100.00 |
-------------|---------|---------|
       Total |   5810  |  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 $0.1 \times 2392$ and $0.1 \times 3418$. Our sample now has 239 males and 342 females:

```
. tabulate female
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/r16/bsample1, clear
. bsample 200 if female
. tabulate female
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:

```stata
. use https://www.stata-press.com/data/r16/bsample1, clear
. set seed 1234
. generate fw = .
(5,810 missing values generated)
. bsample 200 if female, weight(fw)
. tabulate fw female
```

<table>
<thead>
<tr>
<th>fw</th>
<th>male</th>
<th>female</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2,392</td>
<td>3,222</td>
<td>5,614</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>192</td>
<td>192</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>2,392</td>
<td>3,418</td>
<td>5,810</td>
</tr>
</tbody>
</table>

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,

```stata
. use https://www.stata-press.com/data/r16/bsample1, clear
. set seed 1234
. expand 2
(5,810 observations created)
. bsample, strata(female)
. tabulate female
```

<table>
<thead>
<tr>
<th>female</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>male</td>
<td>4,784</td>
<td>41.17</td>
<td>41.17</td>
</tr>
<tr>
<td>female</td>
<td>6,836</td>
<td>58.83</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>11,620</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

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.
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/r16/bsample2, clear
tabstat x, stat(n mean) by(group)
```

Summary for variables: x
by categories of: group

<table>
<thead>
<tr>
<th>group</th>
<th>N</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>15</td>
<td>-.3073028</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>-.00984</td>
</tr>
<tr>
<td>C</td>
<td>11</td>
<td>.0810985</td>
</tr>
<tr>
<td>D</td>
<td>11</td>
<td>-.1989179</td>
</tr>
<tr>
<td>E</td>
<td>29</td>
<td>-.095203</td>
</tr>
<tr>
<td>Total</td>
<td>76</td>
<td>-.1153269</td>
</tr>
</tbody>
</table>

`bsample` will complain if we simply expand the dataset.

```
use https://www.stata-press.com/data/r16/bsample2
expand 3
(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/r16/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
by categories of: expgroup

<table>
<thead>
<tr>
<th>expgroup</th>
<th>N</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>-.3073028</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>-.3073028</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>-.00984</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>-.00984</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>.0810985</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>.0810985</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>-.1989179</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>-.1989179</td>
</tr>
<tr>
<td>9</td>
<td>29</td>
<td>-.095203</td>
</tr>
<tr>
<td>10</td>
<td>29</td>
<td>-.095203</td>
</tr>
</tbody>
</table>

Total | 152 | -.1153269 |

. generate fw = .
   (152 missing values generated)
. bsample 8, cluster(expgroup) weight(fw)
. tabulate fw group

<table>
<thead>
<tr>
<th>fw</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
<td>10</td>
<td>22</td>
<td>11</td>
<td>58</td>
<td>116</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

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/r16/bsample2, clear
. set seed 1234
. tabulate group strid

<table>
<thead>
<tr>
<th>group</th>
<th>strid 1</th>
<th>strid 2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>E</td>
<td>14</td>
<td>15</td>
<td>29</td>
</tr>
<tr>
<td>Total</td>
<td>36</td>
<td>40</td>
<td>76</td>
</tr>
</tbody>
</table>

. 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

<table>
<thead>
<tr>
<th>fw</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>14</td>
<td>36</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>14</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>14</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>28</td>
<td>72</td>
</tr>
</tbody>
</table>

-> strid = 2

<table>
<thead>
<tr>
<th>fw</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>15</td>
<td>27</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>5</td>
<td>0</td>
<td>12</td>
<td>15</td>
<td>48</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>16</td>
<td>10</td>
<td>12</td>
<td>12</td>
<td>30</td>
<td>80</td>
</tr>
</tbody>
</table>

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 \((1 + 0 + 3 + 1 + 3 = 8)\).
bsample — Sampling with replacement

References


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