sample — Draw random sample

Description | Quick start | Menu | Syntax
Options | Remarks and examples | References | Also see

Description

`sample` draws random samples of the data in memory. "Sampling" here is defined as drawing observations without replacement; see [R] **bsample** for sampling with replacement.

The size of the sample to be drawn can be specified as a percentage or as a count:

- `sample` without the `count` option draws a #% pseudorandom sample of the data in memory, thus discarding (100 − #)% of the observations.
- `sample` with the `count` option draws a #-observation pseudorandom sample of the data in memory, thus discarding `_N` − # observations. # can be larger than `_N`, in which case all observations are kept.

In either case, observations not meeting the optional `if` and `in` criteria are kept (sampled at 100%).

If you are interested in reproducing results, you must first set the random-number seed; see [R] **set seed**.

Quick start

Draw 10% pseudorandom sample without replacement from data in memory

```
sample 10
```

Same as above, but perform sampling within strata identified by `svar`

```
sample 10, by(svar)
```

Sample 100 observations from data in memory

```
sample 100, count
```

Same as above, but only sample observations where `catvar` equals 5

```
sample 100 if catvar==5, count
```

Menu

Statistics > Resampling > Draw random sample

Syntax

```
sample # [ if ] [ in ] [ , count by(groupvars) ]
```

by is allowed; see [D] **by**.

Options

count specifies that # in sample # be interpreted as an observation count rather than as a percentage. Typing sample 5 without the count option means that a 5% sample be drawn; typing sample 5, count, however, would draw a sample of 5 observations.

Specifying # as greater than the number of observations in the dataset is not considered an error.

by(*groupvars*) specifies that a #% sample be drawn within each set of values of *groupvars*, thus maintaining the proportion of each group.

count may be combined with by(). For example, typing sample 50, count by(sex) would draw a sample of size 50 for men and 50 for women.

Specifying by *varlist*: sample # is equivalent to specifying sample #, by(*varlist*); use whichever syntax you prefer.

Remarks and examples

▷ Example 1

We have NLSY data on young women aged 14–24 years in 1968 and wish to draw a 10% sample of the data in memory.

```
. use https://www.stata-press.com/data/r19/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. describe, short
Contains data from https://www.stata-press.com/data/r19/nlswork.dta
 Observations:        28,534                  National Longitudinal Survey of
                                                Young Women, 14-24 years old in
                                                1968
    Variables:            21                  27 Nov 2024 08:14
Sorted by: idcode  year
. sample 10
(25,681 observations deleted)
. describe, short
Contains data from https://www.stata-press.com/data/r19/nlswork.dta
 Observations:         2,853                  National Longitudinal Survey of
                                                Young Women, 14-24 years old in
                                                1968
    Variables:            21                  27 Nov 2024 08:14
Sorted by:
     Note: Dataset has changed since last saved.
```

Our original dataset had 28,534 observations. The sample-10 dataset has 2,853 observations, which is the nearest number to 0.10×28534.

◁

Example 2

Among the variables in our data is race. By typing label list, we see that race = 1 denotes whites, race = 2 denotes blacks, and race = 3 denotes other races. We want to keep 100% of the nonwhite women but only 10% of the white women.

```
. use https://www.stata-press.com/data/r19/nlswork, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. tab race
```

Race	Freq.	Percent	Cum.
White	20,180	70.72	70.72
Black	8,051	28.22	98.94
Other	303	1.06	100.00
Total	28,534	100.00	

```
. sample 10 if race == 1
(18,162 observations deleted)
. describe, short
Contains data from https://www.stata-press.com/data/r19/nlswork.dta
 Observations:        10,372                  National Longitudinal Survey of
                                                Young Women, 14-24 years old in
                                                1968
    Variables:            21                  27 Nov 2024 08:14
Sorted by:
     Note: Dataset has changed since last saved.
. display .10*20180 + 8051 + 303
10372
```

Example 3

Now let's suppose that we want to keep 10% of each of the three categories of race.

```
. use https://www.stata-press.com/data/r19/nlswork, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. sample 10, by(race)
(25,681 observations deleted)
. tab race
```

Race	Freq.	Percent	Cum.
White	2,018	70.73	70.73
Black	805	28.22	98.95
Other	30	1.05	100.00
Total	2,853	100.00	

This differs from simply typing sample 10 in that with by(), sample holds constant the percentages of white, black, and other women.

❑ Technical note

We have a large dataset on disk containing 125,235 observations. We wish to draw a 10% sample of this dataset without loading the entire dataset (perhaps because the dataset will not fit in memory). `sample` will not solve this problem—the dataset must be loaded first—but it is rather easy to solve it ourselves. Say that `bigdata.dct` contains the dictionary for this dataset; see [D] **import**. One solution is to type

```
. infile using bigdata if runiform()<=.1
dictionary {
     etc.
}
(12,580 observations read)
```

The `if` qualifier on the end of `infile` drew uniformly distributed random numbers over the interval 0 and 1 and kept each observation if the random number was less than or equal to 0.1. This, however, did not draw an exact 10% sample—the sample was expected to contain only 10% of the observations, and here we obtained just more than 10%. This is probably a reasonable solution.

If the sample must contain precisely 12,524 observations, however, after getting too many observations, we could type

```
. generate u=runiform()
. sort u
. keep in 1/12524
(56 observations deleted)
```

That is, we put the resulting sample in random order and keep the first 12,524 observations. Now our only problem is making sure that, at the first step, we have more than 12,524 observations. Here we were lucky, but half the time we will not be so lucky—after typing `infile ... if runiform()<=.1`, we will have less than a 10% sample. The solution, of course, is to draw more than a 10% sample initially and then cut it back to 10%.

How much more than 10% do we need? That depends on the number of records in the original dataset, which in our example is 125,235.

A little experimentation with `bitesti` (see [R] **bitest**) provides the answer:

```
. bitesti 125235 12524 .102
Binomial probability test
            N   Observed k   Expected k   Assumed p   Observed p
------------------------------------------------------------------
      125,235       12,524    12,773.97     0.10200      0.10000
  Pr(k >= 12,524)                 = 0.990466  (one-sided test)
  Pr(k <= 12,524)                 = 0.009777  (one-sided test)
  Pr(k <= 12,524 or k >= 13,025) = 0.019584  (two-sided test)
```

Initially drawing a 10.2% sample will yield a sample larger than 10% 99 times of 100. If we draw a 10.4% sample, we are virtually assured of having enough observations (type `bitesti 125235 12524 .104` for yourself).

❑

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

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

[R] **bsample** — Sampling with replacement

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark 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.