

**by** — Repeat Stata command on subsets of the data

<a href="#">Description</a> <a href="#">Remarks and examples</a>	<a href="#">Quick start</a> <a href="#">References</a>	<a href="#">Syntax</a> <a href="#">Also see</a>	<a href="#">Options</a>
---	---	--	-------------------------

## Description

Most Stata commands allow the `by` prefix, which repeats the command for each group of observations for which the values of the variables in *varlist* are the same. `by` without the `sort` option requires that the data be sorted by *varlist*; see [D] [sort](#).

Stata commands that work with the `by` prefix indicate this immediately following their syntax diagram by reporting, for example, “`by` is allowed; see [D] [by](#)” or “`bootstrap`, `by`, etc., are allowed; see [U] [11.1.10 Prefix commands](#)”.

`by` and `bysort` are really the same command; `bysort` is just `by` with the `sort` option.

The `varlist1 (varlist2)` syntax is of special use to programmers. It verifies that the data are sorted by `varlist1 varlist2` and then performs a `by` as if only `varlist1` were specified. For instance,

```
by pid (time): generate growth = (bp - bp[_n-1])/bp
```

performs the `generate` by values of `pid` but first verifies that the data are sorted by `pid` and `time` within `pid`.

## Quick start

Generate `newv` as an observation number within each level of `catvar`

```
by catvar: generate newv = _n
```

As above, but sort data by `catvar` first

```
by catvar, sort: generate newv = _n
```

Same as above

```
bysort catvar: generate newv = _n
```

As above, but sort by `v` within values of `catvar`

```
bysort catvar (v): generate newv = _n
```

Generate `newv` as an observation number for each observation in levels of `catvar` and `v`

```
bysort catvar v: generate newv = _n
```

Note: Any command that accepts the `by` prefix may be substituted for `generate` above.

## Syntax

by *varlist* : *stata\_cmd*

bysort *varlist* : *stata\_cmd*

The above diagrams show `by` and `bysort` as they are typically used. The full syntax of the commands is

by *varlist*<sub>1</sub> [(*varlist*<sub>2</sub>)] [, sort *rc0*] : *stata\_cmd*

bysort *varlist*<sub>1</sub> [(*varlist*<sub>2</sub>)] [, *rc0*] : *stata\_cmd*

## Options

`sort` specifies that if the data are not already sorted by *varlist*, `by` should sort them.

`rc0` specifies that even if the *stata\_cmd* produces an error in one of the `by`-groups, then `by` is still to run the *stata\_cmd* on the remaining `by`-groups. The default action is to stop when an error occurs. `rc0` is especially useful when *stata\_cmd* is an estimation command and some `by`-groups have insufficient observations.

## Remarks and examples

[stata.com](https://www.stata.com)

### ▷ Example 1

```
. use https://www.stata-press.com/data/r17/autornd
(1978 automobile data)
. keep in 1/20
(54 observations deleted)
. by mpg: egen mean_w = mean(weight)
not sorted
r(5);
. sort mpg
. by mpg: egen mean_w = mean(weight)
```

```
. list
```

	make	weight	mpg	mean_w
1.	AMC Pacer	3500	15	3916.667
2.	Buick Electra	4000	15	3916.667
3.	Cad. Eldorado	4000	15	3916.667
4.	Cad. Deville	4500	15	3916.667
5.	Buick Riviera	4000	15	3916.667
6.	Chev. Impala	3500	15	3916.667
7.	Chev. Nova	3500	20	3350
8.	Buick Century	3500	20	3350
9.	AMC Concord	3000	20	3350
10.	Chev. Malibu	3000	20	3350
11.	Buick Regal	3500	20	3350
12.	Chev. Monte Carlo	3000	20	3350
13.	Buick Skylark	3500	20	3350
14.	Cad. Seville	4500	20	3350
15.	AMC Spirit	2500	20	3350
16.	Buick LeSabre	3500	20	3350
17.	Buick Opel	2000	25	2500
18.	Chev. Monza	3000	25	2500
19.	Chev. Chevette	2000	30	2000
20.	Dodge Colt	2000	30	2000

by requires that the data be sorted. In the above example, we could have typed `by mpg, sort: egen mean_w = mean(weight)` or `bysort mpg: egen mean_w = mean(weight)` rather than the separate `sort`; all would yield the same results.

◀

For more examples, see [U] 11.1.2 by varlist:, [U] 11.5 by varlist: construct, and [U] 13.7 Explicit subscripting. For extended introductions with detailed examples, see Cox (2002) and Mitchell (2020, chap. 8).

## □ Technical note

by repeats the `stata_cmd` for each group defined by `varlist`. If `stata_cmd` stores results, only the results from the last group on which `stata_cmd` executes will be stored.

□

## References

- Cox, N. J. 2002. Speaking Stata: How to move step by: step. *Stata Journal* 2: 86–102.
- . 2020. Speaking Stata: Concatenating values over observations. *Stata Journal* 20: 236–243.
- Huber, C. 2014. How to simulate multilevel/longitudinal data. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2014/07/18/how-to-simulate-multilevel-longitudinal-data/>.
- Mitchell, M. N. 2020. *Data Management Using Stata: A Practical Handbook*. 2nd ed. College Station, TX: Stata Press.

## Also see

[D] **sort** — Sort data

[D] **statsby** — Collect statistics for a command across a by list

[P] **byable** — Make programs byable

[P] **foreach** — Loop over items

[P] **forvalues** — Loop over consecutive values

[P] **while** — Looping

[U] **11.1.2 by varlist:**

[U] **11.1.10 Prefix commands**

[U] **11.4 varname and varlists**

[U] **11.5 by varlist: construct**