

**stataversion()** — Version of Stata being used

Syntax Diagnostics	Description Also see	Remarks and examples	Conformability
-----------------------	-------------------------	----------------------	----------------

## Syntax

*real scalar* `stataversion()`

*real scalar* `statasetaersion()`

*void* `statasetaersion(real scalar version)`

Note: the version number is multiplied by 100: Stata 2.0 is 200, Stata 5.1 is 510, and Stata 13.0 is 1300.

## Description

`stataversion()` returns the version of Stata/Mata that is running, multiplied by 100. For instance, if you have Stata 13 installed on your computer, `stataversion()` returns 1300.

`statasetaersion()` returns the version of Stata that has been set by the user—the version of Stata that Stata is currently emulating—multiplied by 100. Usually `stataversion() == statasetaersion()`. If the user has set a previous version—say, version 8 by typing `version 8` in Stata—`statasetaersion()` will return a number less than `stataversion()`.

`statasetaersion(version)` allows you to reset the version being emulated. Results are the same as using Stata's `version` command; see [P] [version](#). *version*, however, is specified as an integer equal to 100 times the version you want.

## Remarks and examples

stata.com

It is usually not necessary to reset `statasetaersion()`. If you do reset `statasetaersion()`, good form is to set it back when you are finished:

```
current_version = statasetaersion()
statasetaersion(desired_version)
...
statasetaersion(current_version)
```

## Conformability

`stataversion()`:  
*result:*  $1 \times 1$

`statasetaersion()`:  
*result:*  $1 \times 1$

`stataversion(version)`:

*version*:  $1 \times 1$

*result*: *void*

### Diagnostics

`stataversion(version)` aborts with error if *version* is less than 100 or greater than `stataversion()`.

### Also see

[M-5] [bufio\(\)](#) — Buffered (binary) I/O

[M-5] [byteorder\(\)](#) — Byte order used by computer

[M-4] [programming](#) — Programming functions