

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

[Description](#)    [Syntax](#)    [Remarks and examples](#)    [Conformability](#)  
[Diagnostics](#)    [Also see](#)

## Description

`stataversion()` returns the version of Stata/Mata that is running, multiplied by 100. For instance, if you have Stata 17 installed on your computer, `stataversion()` returns 1700.

`statasetaversion()` 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() == statasetaversion()`. If the user has set a previous version—say, version 8 by typing `version 8` in Stata—`statasetaversion()` will return a number less than `stataversion()`.

`statasetaversion(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.

## Syntax

*real scalar* `stataversion()`

*real scalar* `statasetaversion()`

*void*            `statasetaversion(real scalar version)`

Note: The version number is multiplied by 100: Stata 2.0 is 200, Stata 5.1 is 510, and Stata 17.0 is 1700.

## Remarks and examples

[stata.com](#)

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

```
current_version = statasetaversion()
statasetaversion(desired_version)
...
statasetaversion(current_version)
```

## Conformability

```
stataversion():  
    result:    1 × 1  
  
statasetversion():  
    result:    1 × 1  
  
statasetversion(version):  
    version:   1 × 1  
    result:    void
```

## Diagnostics

`statasetversion(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