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

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Remarks and examples</th>
<th>Conformability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Syntax

- `real scalar` `stataversion()`
- `real scalar` `statasetversion()`
- `void` `statasetversion(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.

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

`statasetversion(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

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

```stata
    current_version = statasetversion()
    statasetversion(desired_version)
    ...
    statasetversion(current_version)
```

### Conformability

- `stataversion()`:
  - `result:` `1 × 1`

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

  version:  1 × 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