Title

stata.com

st_updata( ) — Determine or set data-have-changed flag

Description

`st_updata()` returns 0 if the data in memory have not changed since they were last saved and returns 1 otherwise.

`st_updata(`*value*`)` sets the data-have-changed flag to 0 if *value* = 0 and 1 otherwise.

Syntax

real scalar	`st_updata()`
void	`st_updata(`*real scalar value*`)`

Remarks and examples

stata.com

Stata's `describe` command (see [D] **describe**) reports whether the data have changed since they were last saved. Stata's `use` command (see [D] **use**) refuses to load a new dataset if the data currently in memory have not been saved since they were last changed. Other components of Stata also react to the data-have-changed flag.

`st_updata()` allows you to respect that same flag.

Also, as a Mata programmer, you must set the flag if your function changes the data in memory. Mata attempts to set the flag for you (for instance, when you add a new variable using `st_addvar()` [see [M-5] **st_addvar( )**]), but there are other places where the flag ought to be set, and you must do so. For instance, Mata does not set the flag every time you change a value in the dataset. Setting the flag what may be many thousands of times would reduce performance too much.

Moreover, even when Mata does set the flag, it might do so inappropriately, because the logic of your program fooled Mata. For instance, perhaps you added a variable and later dropped it. In such cases, the appropriate code is

```
priorupdatavalue = st_updata()
...
st_updata(priorupdatavalue)
```

Conformability

`st_updata()`:
result: 1 × 1

`st_updata(`*value*`)`:
value: 1 × 1
result: *void*

Diagnostics

None.

Also see

[M-4] **stata** — Stata interface functions