Title

stata.com

mi reset — Reset imputed or passive variables

Description

`mi reset` resets the imputed or passive variables specified. Values are reset to the values in $m = 0$, which are typically missing, but if you specify `=` *exp*, they are reset to the value specified.

Menu

Statistics > Multiple imputation

Syntax

`mi reset` *varlist* [`=` *exp*] [*if*] [`,` *options*]

options	Description
Main	
`m(`*numlist*`)`	m to reset; default all
`noupdate`	see [MI] **noupdate option**

Options

Main

`m(`*numlist*`)` specifies the values of m that are to be reset; the default is to update all values of m. If M were equal to 3, the default would be equivalent to specifying `m(1/3)` or `m(1 2 3)`. If you wished to update the specified variable(s) in just $m = 2$, you could specify `m(2)`.

`noupdate` in some cases suppresses the automatic `mi update` this command might perform; see [MI] **noupdate option**.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Using mi reset
Technical notes and relation to mi update

Using mi reset

Resetting an imputed or passive variable means setting its values in $m > 0$ equal to the values recorded in $m = 0$. For instance, if variable `inc` were imputed, typing

```
. mi reset inc
(15 values reset)
```

would reset its incomplete values back to missing in all m. In the sample output shown, we happen to have $M = 5$ and reset back to missing the three previously imputed values in each imputation.

It is rare that you would want to reset an imputed variable, but one can imagine cases. Your coworker Joe sent you the data and just buzzed you on the telephone. "There is one value wrong in the data I sent you," he says. "There is an imputed value for `inc` that is 15,000, which is obviously absurd. Just reset it back to missing until I find out what happened." So you type

```
. mi reset inc if inc==15000
(1 value reset)
```

Later Joe calls back. "It is a long and very stupid story," he begins, and you can hear him settling into his chair to tell it. As you finish your second cup of coffee, he is wrapping up. "So the value of `inc` for `pid` 1433 should be 0.725." You type

```
. mi reset inc = .725 if pid=1433
(1 value reset)
```

It is common to need to reset passive variables if imputed values change. For instance, you have variables `age` and `lnage` in your data. You imputed `lnage`; `age` is passive. You recently updated the imputed values for `lnage`. One way to update the values for `age` would be to type

```
. mi passive: replace age = exp(lnage)
m=0:
m=1:
(10 real changes made)
m=2:
(10 real changes made)
m=3:
(8 real changes made)
```

Alternatively, you could type

```
. mi reset age = exp(lnage)
(28 values reset)
```

Technical notes and relation to mi update

`mi reset`, used with an imputed variable, changes only the values for which the variable contains hard missing (`.`) in $m = 0$. The other values are, by definition, already equal to their $m = 0$ values.

`mi reset`, used with a passive variable, changes only the values in incomplete observations, observations in which any imputed variable contains hard missing. The other values of the passive variable are, by definition, already equal to their $m = 0$ values.

`mi update` can be used to ensure that values that are supposed to be equal to their $m = 0$ values in fact are equal to them; see [MI] **mi update**.

Also see

[MI] **intro** — Introduction to mi

[MI] **mi update** — Ensure that mi data are consistent