Title

stata.com

mi varying — Identify variables that vary across imputations

Description Menu Syntax Options
Remarks and examples Stored results Also see

Description

`mi varying` lists the names of variables that are unexpectedly varying and super varying; see [MI] **Glossary** for a definition of varying and super-varying variables.

Menu

Statistics > Multiple imputation

Syntax

```
mi varying [varlist] [, noupdate]

mi varying, unregistered [noupdate]
```

`collect` is allowed; see [U] **11.1.10 Prefix commands**.

Options

`unregistered` specifies that the listing be made only for unregistered variables. Specifying this option saves time, especially when the data are flongsep.

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

Remarks and examples

stata.com

A variable is said to be varying if it varies over m in the complete observations. A variable is said to be super varying if it varies over m in the incomplete observations.

Remarks are presented under the following headings:

Detecting problems
Fixing problems

Detecting problems

`mi varying` looks for five potential problems:

1. *Imputed nonvarying*. Variables that are registered as imputed and are nonvarying either
   a. do not have their missing values in $m > 0$ filled in yet, in which case you should use `mi impute` (see [MI] **mi impute**) to impute them, or
   b. have no missing values in $m = 0$, in which case you should `mi unregister` the variables and perhaps use `mi register` to register the variables as regular (see [MI] **mi set**).
2. *Passive nonvarying*. Variables that are registered as passive and are nonvarying either
   a. have missing values in the incomplete observations in $m > 0$, in which case after you have filled in the missing values of your imputed variables, you should use `mi passive` (see [MI] **mi passive**) to update the values of these variables, or
   b. have no missing values in $m = 0$, in which case you should `mi unregister` the variables and perhaps use `mi register` to register the variables as regular (see [MI] **mi set**).
3. *Unregistered varying*.
   a. It is most likely that such variables should be registered as imputed or as passive.
   b. If the variables are varying but should not be, use `mi register` to register them as regular. That will fix the problem; values from $m = 0$ will be copied to $m > 0$.
   c. It is possible that this is just like potential problem 5, below, and it just randomly turned out that the only observations in which variation occurred were the incomplete observations. In that case, leave the variable unregistered.
4. *Unregistered super/varying*. These are variables that are super varying but would have been categorized as varying if they were registered as imputed. This is to say that while they have varying values in the complete observations as complete is defined this instant—which is based on the variables currently registered as imputed—these variables merely vary in observations for which they themselves contain missing in $m = 0$, and thus they could be registered as imputed without loss of information. Such variables should be registered as imputed.
5. *Unregistered super varying*. These variables really do super vary and could not be registered as imputed without loss of information. These variables either contain true errors or they are passive variables that are functions of groups of observations. Fix the errors by registering the variables as regular and leave unregistered those intended to be super varying. If you intentionally have super-varying variables in your data, remember never to convert to the wide or mlong styles. Super-varying variables can appear only in the flong and flongsep styles.

`mi varying` output looks like this:

```
            Possible problem   Variable names
------------------------------------------------------------------------------
         imputed nonvarying: (none)
         passive nonvarying: (none)
       unregistered varying: (none)
*unregistered super/varying: (none)
 unregistered super varying: (none)
------------------------------------------------------------------------------
* super/varying means super varying but would be varying if registered as
  imputed; variables vary only where equal to soft missing in m=0.
```

If there are possible problems, variable names are listed in the table.

Super-varying variables can arise only in flong and flongsep data, so the last two categories are omitted when `mi varying` is run on wide or mlong data. If there are no imputed variables, or no passive variables, or no unregistered variables, the corresponding categories are omitted from the table.

Fixing problems

If `mi varying` detects problems, register all imputed variables before registering passive variables. Rerun `mi varying` as you register new imputed variables. Registering new variables as imputed can change which observations are classified as complete and incomplete, and that classification in turn can change the categories to which the other variables are assigned. After registering a variable as imputed, another variable previously listed as super varying might now be merely varying.

Stored results

`mi varying` stores the following in `r()`:

Macros

`r(ivars)`	nonvarying imputed variables
`r(pvars)`	nonvarying passive variables
`r(uvars_v)`	varying unregistered variables
`r(uvars_s_v)`	(super) varying unregistered variables
`r(uvars_s_s)`	super-varying unregistered variables

Also see

[MI] **Intro** — Introduction to mi

[MI] **mi misstable** — Tabulate pattern of missing values

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.