Title

> **mi misstable** — Tabulate pattern of missing values

Description

mi misstable runs misstable on $m = 0$ or on $m = \#$ if the m(#) option is specified. misstable makes tables to help in understanding the pattern of missing values in your data; see [R] **misstable**.

Menu

Statistics > Multiple imputation

Syntax

mi <u>misstable</u> <u>summ</u>arize [*varlist*] [*if*] [, *options*]

mi <u>misstable</u> <u>patterns</u> [*varlist*] [*if*] [, *options*]

mi <u>misstable</u> <u>tree</u> [*varlist*] [*if*] [, *options*]

mi <u>misstable</u> <u>nested</u> [*varlist*] [*if*] [, *options*]

options	Description
Main	
exmiss	treat .a, .b, ..., .z as missing
m(#)	run misstable on $m = \#$; default $m = 0$
other_options	see [R] **misstable** (generate() is not allowed; exok is assumed)
nopreserve	programmer's option; see [P] **nopreserve option**

Options

> Main

exmiss specifies that the extended missing values, .a, .b, ..., .z, are to be treated as missing. misstable treats them as missing by default and has the exok option to treat them as nonmissing. mi misstable turns that around and has the exmiss option.

In the mi system, extended missing values that are recorded in imputed variables indicate values not to be imputed and thus are, in a sense, not missing, or more accurately, missing for a good and valid reason.

The exmiss option is intended for use with the patterns, tree, and nested subcommands. You may specify exmiss with the summarize subcommand, but the option is ignored because summarize reports both extended and system missing in separate columns.

m(#) specifies the imputation dataset on which misstable is to be run. The default is $m = 0$, the original data.

other_options are allowed; see [R] **misstable**.

Remarks and examples

See [R] **misstable**.

Stored results

See [R] **misstable**.

Also see

[MI] **Intro** — Introduction to mi

[R] **misstable** — Tabulate missing values

[MI] **mi varying** — Identify variables that vary across imputations