

# Title

**mi estimate** — Estimation using multiple imputations

# Syntax

Compute MI estimates of coefficients by fitting estimation command to mi data

```
mi estimate [, options] : estimation_command ...
```

Compute MI estimates of transformed coefficients by fitting estimation command to mi data

```
mi estimate [spec] [, options] : estimation_command ...
```

where *spec* may be one or more terms of the form (*[name:] exp*). *exp* is any function of the parameter estimates allowed by `nlcom`; see [R] **nlcom**.

<i>options</i>	description
Options	
<code><b>nimputations</b>(#)</code>	specify number of imputations to use; default is to use all existing imputations
<code><b>imputations</b>(<i>numlist</i>)</code>	specify which imputations to use
<code><b>ufmitest</b></code>	perform unrestricted FMI model test
<code><b>nosmall</b></code>	do not apply small-sample adjustment to degrees of freedom
<code><b>saving</b>(<i>miestfile</i> [<i>, replace</i>])</code>	save individual estimation results to <i>miestfile.ster</i>
Tables	
<code><b>[no]citable</b></code>	suppress/display standard estimation table containing parameter-specific confidence intervals; default is <code>citable</code>
<code><b>dftable</b></code>	display degrees-of-freedom table; <code>dftable</code> implies <code>nocitable</code>
<code><b>vartable</b></code>	display variance information about estimates; <code>vartable</code> implies <code>citable</code>
<code><i>table_options</i></code>	control table output
<code><i>display_options</i></code>	control spacing and display of omitted variables and base and empty cells
Reporting	
<code><b>level</b>(#)</code>	set confidence level; default is <code>level(95)</code>
<code><b>dots</b></code>	display dots as estimations performed
<code><b>noisily</b></code>	display any output from <i>estimation_command</i> (and from <code>nlcom</code> if transformations specified)
<code><b>trace</b></code>	trace <i>estimation_command</i> (and <code>nlcom</code> if transformations specified); implies <code>noisily</code>
<code><b>nogroup</b></code>	suppress summary about groups displayed for <code>xt</code> commands
<code><i>xtme_options</i></code>	control output from mixed-effects commands

Advanced

<code>esample(<i>newvar</i>)</code>	store estimation sample in variable <i>newvar</i> ; available only in the <code>flong</code> and <code>flongsep</code> styles
<code>errorok</code>	allow estimation even when <i>estimation_command</i> (or <code>nlcom</code> ) errors out; such imputations are discarded from the analysis
<code>esampvaryok</code>	allow estimation when estimation sample varies across imputations
<code>cmdok</code>	allow estimation when <i>estimation_command</i> is not one of the officially supported estimation commands
† <code>coeflegend</code>	display coefficients' legend instead of coefficient table; implies <code>nocitable</code> and cannot be specified with <code>citable</code> or <code>dftable</code>
† <code>nowarning</code>	suppress the warning about varying estimation samples
† <code>eform_option</code>	display coefficient table in exponentiated form
† <code>post</code>	post estimated coefficients and VCE to <code>e(b)</code> and <code>e(V)</code>
† <code>noupdate</code>	do not perform <code>mi update</code> ; see [MI] <b>noupdate option</b>

---

† `coeflegend`, `nowarning`, `eform_option`, `post`, and `noupdate` do not appear in the dialog box.

You must `mi set` your data before using `mi estimate`; see [MI] **mi set**.

<i>table_options</i>	description
<code>noheader</code>	suppress table header(s)
<code>notable</code>	suppress table(s)
<code>nocoeff</code>	suppress output related to coefficients
<code>nocmdlegend</code>	suppress command legend that appears in the presence of transformed coefficients when <code>nocoeff</code> is used
<code>notrcoef</code>	suppress output related to transformed coefficients
<code>nolegend</code>	suppress table legend(s)
<code>nocnsreport</code>	do not display constraints

---

See [MI] **mi estimate postestimation** for features available after estimation. `mi estimate` is its own estimation command. The postestimation features for `mi estimate` do not include by default the postestimation features for *estimation\_command*. To replay results, type `mi estimate` without arguments.

## Menu

Statistics > Multiple imputation

## Description

`mi estimate`: *estimation\_command* runs *estimation\_command* on the imputed `mi` data, and adjusts coefficients and standard errors for the variability between imputations according to the combination rules by Rubin (1987).

## Options

### Options

- `nimputations(#)` specifies that the first `#` imputations be used; `#` must be  $2 \leq \# \leq M$ . The default is to use all imputations, `M`. Only one of `nimputations()` or `imputations()` may be specified.
- `imputations(numlist)` specifies which imputations to use. The default is to use all of them. `numlist` must contain at least two numbers. Only one of `nimputations()` or `imputations()` may be specified.
- `ufmitest` specifies that the unrestricted fraction missing information (FMI) model test be used. The default test performed assumes equal fractions of information missing due to nonresponse for all coefficients. This is equivalent to the assumption that the between-imputation and within-imputation variances are proportional. The unrestricted test may be preferable when this assumption is suspect provided the number of imputations is large relative to the number of estimated coefficients.
- `nosmall` specifies that no small-sample adjustment be made to the degrees of freedom. The small-sample adjustment is made by default to estimation commands that account for small samples. If the command saves residual degrees of freedom in `e(df_r)`, individual tests of coefficients (and transformed coefficients) use the small-sample adjustment of Barnard and Rubin (1999) and the overall model test uses the small-sample adjustment of Reiter (2007). If the command does not save residual degrees of freedom, the large-sample test is used and the `nosmall` option has no effect.
- `saving(miestfile[, replace])` saves estimation results from each model fit in `miestfile.ster`. The `replace` suboption specifies to overwrite `miestfile.ster` if it exists. `miestfile.ster` can later be used by `mi estimate using` (see [MI] **mi estimate using**) to obtain MI estimates of coefficients or of transformed coefficients without refitting the completed-data models. This file is written in the format used by `estimates use`; see [R] **estimates save**.

### Tables

- All table options below may be specified at estimation time or when redisplaying previously estimated results. Table options must be specified as options to `mi estimate`, not to `estimation_command`.
- `citable` and `nocitable` specify whether the standard estimation table containing parameter-specific confidence intervals is displayed. The default is `citable`. `nocitable` can be used with `varitable` to suppress the confidence-interval table.
- `dftable` displays a table containing parameter-specific degrees of freedom and percentages of increase in standard errors due to nonresponse. `dftable` implies `nocitable`.
- `varitable` displays a table reporting variance information about MI estimates. The table contains estimates of within-imputation variances, between-imputation variances, total variances, relative increases in variance due to nonresponse, fractions of information about parameter estimates missing due to nonresponse, and relative efficiencies for using finite `M` rather than a hypothetically infinite number of imputations. `varitable` implies `citable`.
- `table_options` control the appearance of all displayed table output:
- `noheader` suppresses all header information from the output. The table output is still displayed.
  - `notable` suppresses all tables from the output. The header information is still displayed.
  - `nocoef` suppresses the display of tables containing coefficient estimates. This option affects the output produced by `citable`, `dftable`, and `varitable`.
  - `nocmdlegend` suppresses the table legend showing the specified command line, `estimation_command`, from the output. This legend appears above the tables containing transformed

coefficients (or above the variance-information table if `vartable` is used) when `nocoeff` is specified.

`notrcoef` suppresses the display of tables containing estimates of transformed coefficients (if specified). This option affects the output produced by `citable`, `dftable`, and `vartable`.

`nolegend` suppresses all table legends from the output.

`nocnsreport`; see [R] **estimation options**.

*display\_options*: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`; see [R] **estimation options**.

#### Reporting

Reporting options must be specified as options to `mi estimate` and not as options to *estimation\_command*.

`level(#)`; see [R] **estimation options**.

`dots` specifies that dots be displayed as estimations are successfully completed. An `x` is displayed if the *estimation\_command* returns an error, if the model fails to converge, or if `nlcom` fails to estimate one of the transformed coefficients specified in *spec*.

`noisily` specifies that any output from *estimation\_command* and `nlcom`, used to obtain the estimates of transformed coefficients if transformations are specified, be displayed.

`trace` traces the execution of *estimation\_command* and traces `nlcom` if transformations are specified. `trace` implies `noisily`.

`nogroup` suppresses the display of group summary information (number of groups, average group size, minimum, and maximum) as well as other command-specific information displayed for `xt` commands.

*xtme\_options*: `variance`, `norettable`, `nofetable`, `estmetric`; see, for example, [XT] **xtmixed**. These options are relevant only with the mixed-effects commands such as `xtmixed`, `xtmelogit` (see [XT] **xtmelogit**), and `xtmepoisson` (see [XT] **xtmepoisson**). The `estmetric` option is implied when `vartable` or `dftable` is used.

#### Advanced

`esample(newvar)` creates *newvar* containing `e(sample)`. This option is useful to identify which observations were used in the estimation, especially when the estimation sample varies across imputations (see *Potential problems that can arise when using mi estimate* for details). *newvar* is zero in the original data ( $m = 0$ ) and in any imputations ( $m > 0$ ) in which the estimation failed or that were not used in the computation. `esample()` may be specified only if the data are `flong` or `flongsep`; see [MI] **mi convert** to convert to one of those styles. The variable created will be super varying and therefore must not be registered; see [MI] **mi varying** for more explanation. The saved estimation sample *newvar* may be used later with `mi extract` (see [MI] **mi extract**) to set the estimation sample.

`errorok` specifies that estimations that fail be skipped and the combined results be based on the successful individual estimation results. The default is that `mi estimate` stops if an individual estimation fails. If `errorok` is specified with `saving()`, all estimation results, including failed, are saved to a file.

`esampvaryok` allows estimation to continue even if the estimation sample varies across imputations. `mi estimate` stops if the estimation sample varies. If `esampvaryok` is specified, results from all imputations are used to compute MI estimates and a warning message is displayed at the bottom

of the table. Also see the `esample()` option. See *Potential problems that can arise when using mi estimate* for more information.

`cmdok` allows unsupported estimation commands to be used with `mi estimate`; see [MI] **estimation** for a list of supported estimation commands. Alternatively, if you want `mi estimate` to work with your estimation command, add the property `mi` to the program properties; see [P] **program properties**.

The following options are available with `mi estimate` but are not shown in the dialog box:

`coeflegend`; see [R] **estimation options**. `coeflegend` implies `nocitable` and cannot be combined with `citable` or `dftable`.

`nowarning` suppresses the warning message at the bottom of table output that occurs if the estimation sample varies and `esampvaryok` is specified. See *Potential problems that can arise when using mi estimate* for details.

*eform\_option*; see [R] **eform\_option**. Regardless of the *estimation\_command* specified, `mi estimate` reports results in the coefficient metric under which the combination rules are applied. You may use the appropriate *eform\_option* to redisplay results in exponentiated form, if desired. If `dftable` is also specified, the reported degrees of freedom and percentage increases in standard errors are not adjusted and correspond to the original coefficient metric.

`post` requests MI estimates of coefficients and their respective VCEs be posted in the usual way. This allows the use of *estimation\_command*-specific postestimation tools with MI estimates. There are issues; see *Using the command-specific postestimation tools* in [MI] **mi estimate postestimation**. `post` may be specified at estimation time or when redisplaying previously estimated results.

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

## Remarks

`mi estimate` requires that imputations be already formed; see [MI] **mi impute**. To import existing multiply imputed data, see [MI] **mi import**.

Remarks are presented under the following headings:

*Using mi estimate*

*Example 1: Completed-data logistic analysis*

*Example 2: Completed-data regression analysis*

*Example 3: Completed-data survival analysis*

*Example 4: Estimating transformations*

*Potential problems that can arise when using mi estimate*

## Using mi estimate

`mi estimate` estimates model parameters from multiply imputed data and adjusts coefficients and standard errors for the variability between imputations. It runs the specified *estimation\_command* on each of the  $M$  imputed datasets to obtain the  $M$  completed-data estimates of coefficients and their VCEs. It then computes MI estimates of coefficients and standard errors by applying combination rules (Rubin 1987, 77) to the  $M$  completed-data estimates. See [MI] **intro substantive** for a discussion of MI analysis and see *Methods and formulas* for computational details.

To use `mi estimate`, your data must contain at least two imputations. The basic syntax of `mi estimate` is

```
. mi estimate: estimation_command ...
```

*estimation\_command* is any estimation command from the list of supported estimation commands; see [MI] **estimation**.

If you wish to estimate on survey data, type

```
. mi estimate: svy: estimation_command ...
```

If you want to vary the number of imputations or select which imputations to use in the computations, use the `nimputations()` or the `imputations()` option, respectively.

```
. mi estimate, nimputations(9): estimation_command ...
```

Doing so is useful to evaluate the stability of MI results.

You can obtain more-detailed information about imputation results by specifying the `dftable` and `vartable` options.

You can additionally obtain estimates of transformed coefficients by specifying expressions with `mi estimate`; see *Example 4: Estimating transformations* for details.

When using `mi estimate`, keep in mind that

1. `mi estimate` is its own estimation command.
2. `mi estimate` uses different degrees of freedom for each estimated parameter when computing its significance level and confidence interval.
3. `mi estimate` reports results in the coefficient metric under which combination rules are applied regardless of the default reporting metric of the specified *estimation\_command*. Use *eform\_option* with `mi estimate` to report results in the exponentiated metric, if you wish.
4. `mi estimate` has its own reporting options and does not respect command-specific reporting options. The reporting options specified with *estimation\_command* affect only the output of the command that is displayed when `mi estimate`'s `noisily` option is specified. Specify `mi estimate`'s options immediately after the `mi estimate` command:

```
. mi estimate, options: estimation_command ...
```

## Example 1: Completed-data logistic analysis

Recall the logistic analysis of the heart attack data from [MI] **intro substantive**. The goal of the analysis was to explore the relationship between heart attacks and smoking adjusted for other factors such as age, body mass index, gender, and educational status. The original data contain missing values of BMI. The listwise-deletion analysis on the original data determined that smoking and BMI have significant impact on a heart attack. After imputing missing values of BMI, age was determined to be a significant factor as well. See *A brief introduction to MI using Stata* in [MI] **intro substantive** for details. The data we used are stored in `mheart1s20.dta`.

Below we refit the logistic model using the imputed data. We also specify the `dots` option so that dots will be displayed as estimations are completed.

(Continued on next page)

```

. use http://www.stata-press.com/data/r11/mheart1s20
(Fictional heart attack data; bmi missing)
. mi estimate, dots: logit attack smokes age bmi hsgrad female
Imputations (20):
.....10.....20 done
Multiple-imputation estimates          Imputations    =          20
Logistic regression                   Number of obs   =          154
                                       Average RVI     =          0.0312
                                       DF:            min    =        1060.38
                                       avg            =        223362.56
                                       max            =        493335.88
DF adjustment:  Large sample
Model F test:      Equal FMI          F( 5,71379.3) =          3.59
Within VCE type:  OIM                Prob > F       =          0.0030

```

	attack	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
	smokes	1.198595	.3578195	3.35	0.001	.4972789 1.899911
	age	.0360159	.0154399	2.33	0.020	.0057541 .0662776
	bmi	.1039416	.0476136	2.18	0.029	.010514 .1973692
	hsgrad	.1578992	.4049257	0.39	0.697	-.6357464 .9515449
	female	-.1067433	.4164735	-0.26	0.798	-.9230191 .7095326
	_cons	-5.478143	1.685075	-3.25	0.001	-8.782394 -2.173892

The left header column reports information about the fitted MI model. The right header column reports the number of imputations and the number of observations used, the average relative variance increase (RVI) due to nonresponse, a summary about parameter-specific degrees of freedom (DF), and the overall model test that all coefficients, excluding the constant, are equal to zero.

Notice first that `mi estimate` reports Student's  $t$  and  $F$  statistics for inference although `logit` would usually report  $Z$  and  $\chi^2$  statistics.

`mi estimate: logit` is not `logit`. `mi estimate` uses Rubin's combination rules to obtain the estimates from multiply imputed data. The variability of the MI estimates consists of two components: variability within imputations and variability between imputations. Therefore, the precision of the MI estimates is governed not only by the number of observations in the sample but also by the number of imputations. As such, even if the number of observations is large, if the number of imputations is small, the reference distribution used for inference will deviate from the normal distribution. Because in practice the number of imputations tends to be small, `mi estimate` uses a reference  $t$  distribution.

Returning to the output, average RVI reports the average relative increase (averaged over all coefficients) in variance of the estimates because of the missing `bmi` values. A relative variance increase is an increase in the variance of the estimate because of the loss of information about the parameter due to nonresponse relative to the variance of the estimate with no information lost. The closer this number is to zero, the less effect missing data have on the variance of the estimate. Note that the reported RVI will be zero if you use `mi estimate` with the complete data or with missing data that have not been imputed. In our case, average RVI is small: 0.03.

The coefficient-specific degrees of freedom (DF) averaging 223,363 are large. They are large because the MI degrees of freedom depends not only on the number of imputations but also on the RVI due to nonresponse. Specifically, the degrees of freedom is inversely related to RVI. The closer RVI is to zero, the larger the degrees of freedom regardless of the number of imputations.

To the left of the DF, we see that the degrees of freedom are obtained under a large-sample assumption. The alternative is to use a small-sample adjustment. Whether the small-sample adjustment is applied is determined by the type of the reference distribution used for inference by the specified estimation command. For the commands that use a large-sample (normal) approximation for inference, a large-sample approximation is used when computing the MI degrees of freedom. For the commands that

use a small-sample (Student's  $t$ ) approximation for inference, a small-sample approximation is used when computing the MI degrees of freedom. See *Methods and formulas* for details. As we already mentioned, `logit` assumes large samples for inference, and thus the MI degrees of freedom is computed assuming a large sample.

The model  $F$  test rejects the hypothesis that all coefficients are equal to zero and thus rules out a constant-only model for heart attacks. By default, the model test uses the assumption that the fractions of missing information of all coefficients are equal (as noted by `Equal FMI` to the left). Although this assumption may not be supported by the data, it is used to circumvent the difficulties arising with the estimation of the between-imputation variance matrix based on a small number of imputations. See *Methods and formulas* and `[MI] mi estimate postestimation` for details.

`mi estimate` also reports the type of variance estimation used by the estimation command to compute variance estimates in the individual completed-data analysis. These completed-data variance estimates are then used to compute the within-imputation variance. In our example, the observed-information-matrix (OIM) method, the default variance-estimation method used by maximum likelihood estimation, is used to compute completed-data VCEs. This is labeled as `Within VCE type: OIM` in the output.

Finally, `mi estimate` reports a coefficient table containing the combined estimates. Unlike all other Stata estimation commands, the reported significance levels and confidence intervals in this table are based on degrees of freedom that is specific to each coefficient. Remember that the degrees of freedom depends on the relative variance increases and thus on how much information is lost about the estimated parameter due to missing data. How much information is lost is specific to each parameter and so is the degrees of freedom.

As we already saw, a summary of the coefficient-specific degrees of freedom (minimum, average, and maximum) was reported in the header. We can obtain a table containing coefficient-specific degrees of freedom by replaying the results with the `dftable` option:

```
. mi estimate, dftable
Multiple-imputation estimates      Imputations      =      20
Logistic regression              Number of obs    =     154
                                  Average RVI      =     0.0312
DF adjustment: Large sample      DF:      min     =    1060.38
                                  avg           =  223362.56
                                  max           =  493335.88
Model F test:      Equal FMI      F( 5,71379.3)   =     3.59
Within VCE type:  OIM            Prob > F        =     0.0030
```

attack	Coef.	Std. Err.	t	P> t	DF	% Increase Std. Err.
smokes	1.198595	.3578195	3.35	0.001	320019.4	0.39
age	.0360159	.0154399	2.33	0.020	493335.9	0.31
bmi	.1039416	.0476136	2.18	0.029	1060.4	7.45
hsgrad	.1578992	.4049257	0.39	0.697	165126.7	0.54
female	-.1067433	.4164735	-0.26	0.798	358078.3	0.37
_cons	-5.478143	1.685075	-3.25	0.001	2554.8	4.61

Notice that we type `mi estimate` to replay the results, not `logit`.

The header information remains the same. In particular, degrees of freedom ranges from 1,060 to 493,336 and averages 223,363. In the table output, the columns for the confidence intervals are replaced with the `DF` and `% Increase Std. Err.` columns. We now see that the smallest degrees of freedom corresponds to the coefficient for `bmi`. We should have anticipated this because `bmi` is the only variable containing missing values in this example. The largest degrees of freedom is observed



for the coefficient for `age`, which suggests that the loss of information due to nonresponse is the smallest for the estimation of this coefficient.

The last column displays as percentages the increase in standard errors of the parameters due to nonresponse. We observe a 7% increase in the standard error for the coefficient of `bmi` and a 4% increase in the standard error for the constant. Increases in standard errors of other coefficients are negligible.

In this example, we displayed a degrees-of-freedom table on replay by specifying the `dftable` option. We could also obtain this table if we specified this option at estimation time. Alternatively, if desired, we could display both tables by specifying the `citable` and `dftable` options together.

We can obtain more detail about imputation results by specifying the `var` option. We specify this option on replay and also use the `nocitable` option to suppress the default confidence-interval table:

```
. mi estimate, vartable nocitable
Multiple-imputation estimates          Imputations      =      20
Variance information
```

	Imputation variance			RVI	FMI	Relative efficiency
	Within	Between	Total			
<code>smokes</code>	.127048	.00094	.128035	.007765	.007711	.999615
<code>age</code>	.000237	1.4e-06	.000238	.006245	.00621	.99969
<code>bmi</code>	.001964	.000289	.002267	.154545	.135487	.993271
<code>hsgrad</code>	.162206	.001675	.163965	.010843	.010739	.999463
<code>female</code>	.172187	.001203	.17345	.007338	.00729	.999636
<code>_cons</code>	2.5946	.233211	2.83948	.094377	.086953	.995671

The first three columns of the table provide the variance information specific to each parameter. As we already discussed, MI variance contains two sources of variation: within imputation and between imputation. The first two columns provide estimates for the within-imputation and between-imputation variances. The third column is a total variance that is the sum of the two variances plus an adjustment for using a finite number of imputations. The next two columns are individual RVIs and fractions of missing information (FMIs) due to nonresponse. The last column records relative efficiencies for using a finite number of imputations (20 in our example) versus the theoretically optimal infinite number of imputations.

We notice that the coefficient for `age` has the smallest within-imputation and between-imputation variances. The between-imputation variability is very small relative to the within-imputation variability, which is why `age` had such a large estimate of the degrees of freedom we observed earlier. Correspondingly, this coefficient has the smallest values for RVI and FMI. As expected, the coefficient for `bmi` has the highest RVI and FMI.

The reported relative efficiencies are high for all coefficient estimates, with the smallest relative efficiency, again, corresponding to `bmi`. These estimates, however, are only approximations and thus should not be used exclusively to determine the required number of imputations. See Royston, Carlin, and White (2009) for other ways of determining a suitable number of imputations.

## Example 2: Completed-data regression analysis

Recall the data on house resale prices from example 3 of [MI] `mi impute mvn`. We use the imputed data stored in `mhouses1993s30.dta` to examine the relationship of various predictors on price via linear regression:

```

. use http://www.stata-press.com/data/r11/mhouses1993s30
(Albuquerque Home Prices Feb15-Apr30, 1993)
. mi estimate, ni(5): regress price tax sqft age nfeatures ne custom corner

Multiple-imputation estimates      Imputations      =      5
Linear regression                  Number of obs    =     117
                                   Average RVI      =     0.2939
                                   Complete DF     =     109
DF adjustment: Small sample       DF:      min    =     7.51
                                   avg          =     61.46
                                   max          =     94.51
Model F test:      Equal FMI      F( 7, 79.3)    =     53.00
Within VCE type:  OLS              Prob > F       =     0.0000

```

	price	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
	tax	.5375429	.1399677	3.84	0.001	.2491601 .8259257
	sqft	.2940729	.0890706	3.30	0.004	.1066042 .4815415
	age	-.7325252	.7605651	-0.96	0.365	-2.506304 1.041254
	nfeatures	3.696554	13.69565	0.27	0.788	-23.62366 31.01677
	ne	8.303979	34.64444	0.24	0.811	-60.47857 77.08653
	custom	129.2014	44.29287	2.92	0.004	41.25603 217.1468
	corner	-63.77848	41.48302	-1.54	0.128	-146.1749 18.61793
	_cons	135.4265	64.57713	2.10	0.039	7.184164 263.6688

By default, `mi estimate` uses all available imputations in the analysis. For the purpose of illustration, we use only the first 5 imputations out of the available 30 by specifying the `nimputations(5)` option, which we abbreviated as `ni(5)`.

Compared with the output from the previous example, an additional result, `Complete DF`, is reported. Also notice that the adjustment for the degrees of freedom is now labeled as `Small sample`. Remember that `mi estimate` determines what adjustment to use based on the reference distribution used for inference by the specified estimation command.

`regress` uses a reference  $t$  distribution with  $117 - 8 = 109$  residual degrees of freedom. Thus a small-sample adjustment is used by `mi estimate` for the MI degrees of freedom.

`Complete DF` contains the degrees of freedom used for inference in each completed dataset (imputation). It corresponds to the completed-data residual degrees of freedom saved by the command in `e(df_r)`. In most applications, the completed-data residual degrees of freedom will be the same, and so `Complete DF` will correspond to the *complete degrees of freedom*, the degrees of freedom that would have been used for inference if the data were complete. In the case when the completed-data residual degrees of freedom varies across imputations (as may happen when the estimation sample varies; see *Potential problems that can arise when using mi estimate*), `Complete DF` reports the smallest of them.

In our example, all completed-data residual degrees of freedom are equal, and `Complete DF` is equal to 109, the completed-data residual degrees of freedom obtained from `regress`. `mi estimate` uses the complete degrees of freedom to adjust the MI degrees of freedom for a small sample (Barnard and Rubin 1999).

### Example 3: Completed-data survival analysis

Consider survival data on 48 participants in a cancer drug trial. The dataset contains information about participants' ages, treatments received (drug or placebo), times to death measured in months, and a censoring indicator. The data are described in more detail in *Cox regression with censored data*

of [ST] `stcox`. We consider a version of these data containing missing values for age. The imputed data are saved in `mdrugtrs25.dta`:

```
. use http://www.stata-press.com/data/r11/mdrugtrs25
(Patient Survival in Drug Trial)

. mi describe
Style: mlong
Obs.:  complete      40
      incomplete     8  (M = 25 imputations)
      -----
      total          48
Vars.: imputed:  1; age(8)
      passive:  0
      regular:  3; studytime died drug
      system:   3; _mi_m _mi_id _mi_miss
      (there are no unregistered variables)
```

The dataset contains 25 imputations for 8 missing values of age.

We analyze these data using `stcox` with `mi estimate`. These data have not yet been `stset`, so we use `mi stset` (see [MI] `mi XXXset`) to set them and then perform the analysis using `mi estimate: stcox`:

```
. mi stset studytime, failure(died)
      failure event:  died != 0 & died < .
obs. time interval:  (0, studytime]
exit on or before:   failure
```

---

```
48 total obs.
0  exclusions
```

---

```
48 obs. remaining, representing
31 failures in single record/single failure data
744 total analysis time at risk, at risk from t =      0
      earliest observed entry t =      0
      last observed exit t =      39
```

```
. mi estimate, dots: stcox drug age
Imputations (25):
.....10.....20..... done
Multiple-imputation estimates          Imputations      =      25
Cox regression: Breslow method for ties Number of obs    =      48
Average RVI                          =      0.1059
DF adjustment: Large sample           DF: min         =      998.63
                                       avg         = 11621.53
                                       max         = 22244.42
Model F test: Equal FMI               F( 2, 4448.6)   =      13.43
Within VCE type: OIM                  Prob > F        =      0.0000
```

---

_t	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
drug	-2.204572	.4589047	-4.80	0.000	-3.104057	-1.305086
age	.1242711	.040261	3.09	0.002	.0452652	.2032771

Notice that `mi estimate` displays the results in the coefficient metric and not in the hazard-ratio metric. By default, `mi estimate` reports results in the metric under which the combination rules were applied. To obtain the results as hazard ratios, we can use the `hr` option with `mi estimate`:

```

. mi estimate, hr
Multiple-imputation estimates      Imputations      =      25
Cox regression: Breslow method for ties  Number of obs    =      48
                                      Average RVI       =      0.1059
DF adjustment: Large sample         DF: min         =     998.63
                                      avg             =    11621.53
                                      max             =    22244.42
Model F test: Equal FMI           F( 2, 4448.6)   =     13.43
Within VCE type: OIM              Prob > F        =     0.0000

```

_t	Haz. Ratio	Std. Err.	t	P> t	[95% Conf. Interval]	
drug	.1102977	.0506161	-4.80	0.000	.0448668	.2711491
age	1.132323	.0455885	3.09	0.002	1.046305	1.225412

We obtain results similar to those from the corresponding example in [ST] **stcox**.

We specified the `hr` option above on replay. We can also specify it at estimation time:

```

. mi estimate, hr: stcox drug age
(output omitted)

```

Notice that the `hr` option must be specified with `mi estimate` to obtain hazard ratios. Specifying it with the command itself,

```

. mi estimate: stcox drug age, hr
(output omitted)

```

will not affect the output from `mi estimate` but only that of the `stcox` command. You see `stcox`'s output only if you specify `mi estimate`'s `noisily` option.

## Example 4: Estimating transformations

Stata estimation commands usually support `lincom` and `nlcom` (see [R] **lincom** and [R] **nlcom**) to obtain estimates of the transformed coefficients after estimation by using the delta method. Because MI estimates based on a small number of imputations may not yield a valid VCE, this approach is not generally viable. Also, transformations applied to the combined coefficients are only asymptotically equivalent to the combined transformed coefficients. With a small number of imputations, these two ways of obtaining transformed coefficients can differ significantly.

Thus `mi estimate` provides its own way of combining transformed coefficients. You need to use `mi estimate`'s method for both linear and nonlinear combinations of coefficients. We are about to demonstrate how using the ratio of coefficients as an example, but what we are about to do would be equally necessary if we wanted to obtain the difference in two coefficients.

For the purpose of illustration, suppose that we want to estimate the ratio of the coefficients, say, `age` and `sqft` from example 2. We can do this by typing

(Continued on next page)

```

. use http://www.stata-press.com/data/r11/mhouses1993s30, clear
(Albuquerque Home Prices Feb15-Apr30, 1993)
. mi estimate (ratio: _b[age]/_b[sqft]):
> regress price tax sqft age nfeatures ne custom corner

Multiple-imputation estimates      Imputations      =      30
Linear regression                 Number of obs    =     117
                                   Average RVI      =     0.5415
                                   Complete DF     =     109
DF adjustment: Small sample      DF: min         =     16.42
                                   avg            =     72.83
                                   max            =     101.18
Model F test: Equal FMI          F( 7, 96.3)    =     45.63
Within VCE type: OLS             Prob > F        =     0.0000

```

	price	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
	tax	.5516444	.1443319	3.82	0.000	.2612817 .842007
	sqft	.2900879	.0904748	3.21	0.003	.1073624 .4728134
	age	-.7524605	1.097246	-0.69	0.502	-3.073675 1.568754
	nfeatures	4.361055	13.59917	0.32	0.749	-22.67719 31.3993
	ne	5.495913	34.97562	0.16	0.875	-63.95148 74.94331
	custom	132.3453	43.26507	3.06	0.003	46.52087 218.1697
	corner	-66.95606	40.55801	-1.65	0.102	-147.4264 13.51429
	_cons	130.3491	64.03837	2.04	0.044	3.277868 257.4203

```

Transformations                    Average RVI      =     2.0039
                                   Complete DF     =     109
DF adjustment: Small sample      DF: min         =     23.04
                                   avg            =     23.04
                                   max            =     23.04
Within VCE type: OLS
    ratio: _b[age]/_b[sqft]

```

	price	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
	ratio	-2.167045	3.460205	-0.63	0.537	-9.324358 4.990268

We use the `nlcom` syntax to specify the transformation: `(ratio: _b[age]/_b[sqft])` defines the transformation and its name is `ratio`. All transformations must be specified following `mi estimate` and before the colon, and must be bound in parentheses.

A separate table containing the estimate of the ratio is displayed following the estimates of coefficients. If desired, we can suppress the table containing the estimates of coefficients by specifying the `nocoeff` option. The header reports the average RVI due to nonresponse and the summaries of the degrees of freedom specific to the estimated transformations. Because we specified only one transformation, the minimum, average, and maximum degrees of freedom are the same. They correspond to the individual degrees of freedom for `ratio`.

See [MI] `mi estimate postestimation` for an example of linear transformation.

## Potential problems that can arise when using `mi estimate`

There are two problems that can arise when using `mi estimate`:

1. The estimation sample varies across imputations.
2. Different covariates are omitted across the imputations.

`mi estimate` watches for and issues an error if either of these problems occur. Below we explain how each can arise and what to do about it. If you see one of these messages, be glad that `mi estimate` mentioned the problem, because otherwise, it might have gone undetected. A varying-estimation sample may result in biased or inefficient estimates. Different covariates being omitted always results in the combined results being biased.

If the first problem arises, `mi estimate` issues the error “estimation sample varies between  $m = \#$  and  $m = \#$ ”. `mi estimate` expects that when it runs the estimation command on the first imputation, on the second, and so on, the estimation command will use the same observations in each imputation. `mi estimate` does not just count, it watches which observations are used.

Perhaps the difference is due to a past mistake, such as not having imputed all the missing values. Perhaps you even corrupted your `mi` data so that the imputed variable is missing in some imputations and not in others.

Another reason the error can arise is because you specified an `if` condition based on imputed or passive variables. `mi estimate` considers this a mistake but, if this is your intent, you can reissue the `mi estimate` command and include the `esampvaryok` option.

Finally, it is possible that the varying observations are merely a characteristic of the estimator when combined with the two different imputed datasets. In this case, just as in the previous one, you can reissue `mi estimate` with the `esampvaryok` option.

The easy way to diagnose why you got this error is to use `mi xeq` (see [MI] **mi xeq**) to run the estimation command separately on the two imputations mentioned in the error message. Alternatively, you can rerun the `mi estimate` command immediately with the `esampvaryok` option and with the `esample(varname)` option, which will create in new variable `varname` the `e(sample)` from each of the individual estimations. If you use the second approach, you must first `mi convert` your data to `flong` or `flongsep` if they are not recorded in that style already; see [MI] **mi convert** for details.

The second problem we mentioned concerns omitted variables as opposed to omitted observations. `mi estimate` reports “omitted variables vary” and goes on to mention the two imputations between which the variation was detected.

This can be caused when you include factor variables but did not specify base categories. It was the base categories that differed in the two imputations. That could happen if you specified `i.group`. By default, Stata chooses to omit the most frequent category. If `group` were imputed or passive, then the most frequent category could vary between two imputations. The solution is to specify the base category for yourself by typing, for instance, `b2.group`; see [U] **11.4.3 Factor variables**.

There are other possible causes. Varying omitted variables includes 1) different variables being omitted in the two imputations and 2) includes no variables being omitted in one imputation and, in the other, one or more variables being omitted.

When different variables are being omitted, it is usually caused by collinearity, and one of the variables needs to be dropped from the model. Variables `x1` and `x2` are collinear; sometimes the estimation command is choosing to omit `x1` and other times, `x2`. The solution is that you choose which to omit by removing it from your model.

If no variables were omitted in one of the imputations, the problem is more difficult to explain. Say that you included `i.group` in your model, the base category remained the same for the two imputations, but in one of the imputations, no one is observed in group 3, and thus no coefficient for group 3 could be estimated. Your choices are to accept that you cannot estimate a group 3 coefficient and combine group 3 with, say, group 4, or to drop all imputations in which there is no one in group 3. If you want to drop imputations 3, 9, and 12, you type `mi set m -= (3,9,12)`; see [MI] **mi set**.

## □ Technical note

As we already mentioned, `mi estimate` obtains MI estimates by using the combination rules to pool results from the specified command executed separately on each imputation. As such, certain concepts (for example, likelihood function) and most postestimation tools specific to the command may not be applicable to the MI estimates; see *Analysis of multiply imputed data* in [MI] **intro substantive**. MI estimates may not even have a valid variance–covariance matrix associated with them when the number of imputations is smaller than the number of estimated parameters. For these reasons, the system matrices `e(b)` and `e(V)` are not set by `mi estimate`. If desired, you can save the MI estimates and its variance–covariance estimate in `e(b)` and `e(V)` by specifying the `post` option. See [MI] **mi estimate postestimation** for postestimation tools available after `mi estimate`. □

## Saved results

`mi estimate` saves the following in `e()`:

### Scalars

<code>e(df_avg[_Q]_mi)</code>	average degrees of freedom
<code>e(df_c_mi)</code>	complete degrees of freedom (if originally saved by <i>estimation_command</i> in <code>e(df_r)</code> )
<code>e(df_max[_Q]_mi)</code>	maximum degrees of freedom
<code>e(df_min[_Q]_mi)</code>	minimum degrees of freedom
<code>e(df_m_mi)</code>	MI model test denominator (residual) degrees of freedom
<code>e(df_r_mi)</code>	MI model test numerator (model) degrees of freedom
<code>e(esampvary_mi)</code>	varying-estimation sample flag (0 or 1)
<code>e(F_mi)</code>	model test <i>F</i> statistic
<code>e(k_exp_mi)</code>	number of expressions (transformed coefficients)
<code>e(M_mi)</code>	number of imputations
<code>e(N_mi)</code>	number of observations (minimum, if varies)
<code>e(N_min_mi)</code>	minimum number of observations
<code>e(N_max_mi)</code>	maximum number of observations
<code>e(N_g_mi)</code>	number of groups
<code>e(g_min_mi)</code>	smallest group size
<code>e(g_avg_mi)</code>	average group size
<code>e(g_max_mi)</code>	largest group size
<code>e(p_mi)</code>	MI model test <i>p</i> -value
<code>e(rvi_avg[_Q]_mi)</code>	average RVI
<code>e(rvi_avg_F_mi)</code>	average RVI associated with the residual degrees of freedom for model test
<code>e(ufmi_mi)</code>	1 if unrestricted FMI model test is performed, 0 if equal FMI model test is performed

## Macros

<code>e(mi)</code>	mi
<code>e(cmdline_mi)</code>	command as typed
<code>e(prefix_mi)</code>	mi estimate
<code>e(cmd_mi)</code>	name of <i>estimation_command</i>
<code>e(cmd)</code>	mi estimate (equals <code>e(cmd_mi)</code> when <code>post</code> is used)
<code>e(title_mi)</code>	“Multiple-imputation estimates”
<code>e(wvce_mi)</code>	title used to label within-imputation variance in the table header
<code>e(modeltest_mi)</code>	title used to label the model test in the table header
<code>e(dfadjust_mi)</code>	title used to label the degrees-of-freedom adjustment in the table header
<code>e(exprnames_mi)</code>	names of expressions specified in <i>spec</i>
<code>e(exp#_mi)</code>	expressions of the transformed coefficients specified in <i>spec</i>
<code>e(rc_mi)</code>	return codes for each imputation
<code>e(m_mi)</code>	specified imputation numbers
<code>e(m_est_mi)</code>	imputation numbers used in the computation
<code>e(names_vv1_mi)</code>	command-specific <code>e()</code> macro names that contents varied across imputations
<code>e(names_vvm_mi)</code>	command-specific <code>e()</code> matrix names that values varied across imputations (excluding <code>b</code> , <code>V</code> , and <code>Cns</code> )
<code>e(names_vvs_mi)</code>	command-specific <code>e()</code> scalar names that values varied across imputations

## Matrices

<code>e(b)</code>	MI estimates of coefficients (equals <code>e(b_mi)</code> , saved only if <code>post</code> is used)
<code>e(V)</code>	variance–covariance matrix (equals <code>e(V_mi)</code> , saved only if <code>post</code> is used)
<code>e(Cns)</code>	constraint matrix (for constrained estimation only; equals <code>e(Cns_mi)</code> , saved only if <code>post</code> is used)
<code>e(N_g_mi)</code>	group counts
<code>e(g_min_mi)</code>	group-size minimums
<code>e(g_avg_mi)</code>	group-size averages
<code>e(g_max_mi)</code>	group-size maximums
<code>e(b[_Q]_mi)</code>	MI estimates of coefficients (or transformed coefficients)
<code>e(V[_Q]_mi)</code>	variance–covariance matrix (total variance)
<code>e(Cns_mi)</code>	constraint matrix (for constrained estimation only)
<code>e(W[_Q]_mi)</code>	within-imputation variance matrix
<code>e(B[_Q]_mi)</code>	between-imputation variance matrix
<code>e(re[_Q]_mi)</code>	parameter-specific relative efficiencies
<code>e(rvi[_Q]_mi)</code>	parameter-specific RVIs
<code>e(fmi[_Q]_mi)</code>	parameter-specific FMIs
<code>e(df[_Q]_mi)</code>	parameter-specific degrees of freedom
<code>e(pise[_Q]_mi)</code>	parameter-specific percentages increase in standard errors
<code>e(vs_names_vs_mi)</code>	values of command-specific <code>e()</code> scalar <i>vs_names</i> that varied across imputations

*vs\_names* include (but are not restricted to) `df_r`, `N`, `N_strata`, `N_psu`, `N_pop`, `N_sub`, `N_postrata`, `N_stdize`, `N_subpop`, `N_over`, and `converged`.

Results `N_g_mi`, `g_min_mi`, `g_avg_mi`, and `g_max_mi` are saved for panel-data models only. The results are saved as matrices for mixed-effects models and as scalars for other panel-data models.

If transformations are specified, the corresponding estimation results are saved with the `_Q_mi` suffix, as described above.

Command-specific `e()` results that remain constant across imputations are also saved. Command-specific results that vary from imputation to imputation are posted as missing, and their names are saved in the corresponding macros `e(names_vv1_mi)`, `e(names_vvm_mi)`, and `e(names_vvs_mi)`. For some command-specific `e()` scalars (see *vs\_names* above), their values from each imputation are saved in a corresponding matrix with the `_vs_mi` suffix.

## Methods and formulas

Let  $\mathbf{q}$  define a column vector of parameters of interest. For example,  $\mathbf{q}$  may be a vector of coefficients (or functions of coefficients) from a regression model. Let  $\{(\hat{\mathbf{q}}_i, \hat{\mathbf{U}}_i) : i = 1, 2, \dots, M\}$



be the completed-data estimates of  $\mathbf{q}$  and the respective variance–covariance estimates from  $M$  imputed datasets.

The MI estimate of  $\mathbf{q}$  is

$$\bar{\mathbf{q}}_M = \frac{1}{M} \sum_{i=1}^M \hat{\mathbf{q}}_i$$

The variance–covariance estimate (VCE) of  $\bar{\mathbf{q}}_M$  (*total variance*) is

$$\mathbf{T} = \bar{\mathbf{U}} + \left(1 + \frac{1}{M}\right)\mathbf{B}$$

where  $\bar{\mathbf{U}} = \sum_{i=1}^M \hat{\mathbf{U}}_i/M$  is the within-imputation variance–covariance matrix and  $\mathbf{B} = \sum_{i=1}^M (\mathbf{q}_i - \bar{\mathbf{q}}_M)(\mathbf{q}_i - \bar{\mathbf{q}}_M)'/(M-1)$  is the between-imputation variance–covariance matrix.

Methods and formulas are presented under the following headings:

*Univariate case*  
*Multivariate case*

## Univariate case

Let  $Q$ ,  $\bar{Q}_M$ ,  $B$ ,  $\bar{U}$ , and  $T$  correspond to the scalar analogues of the above formulas. Univariate inferences are based on the approximation

$$T^{-1/2}(Q - \bar{Q}_M)^2 \sim t_\nu \quad (1)$$

where  $t_\nu$  denotes a Student's  $t$  distribution with  $\nu$  degrees of freedom, which depends on the number of imputations,  $M$ , and the increase in variance of estimates due to missing data. Under the large-sample assumption with respect to complete data, the degrees of freedom is

$$\nu_{\text{large}} = (M-1) \left(1 + \frac{1}{r}\right)^2 \quad (2)$$

where

$$r = \frac{(1 + M^{-1})B}{\bar{U}} \quad (3)$$

is an RVI due to missing data. Under the small-sample assumption, the degrees of freedom is

$$\nu_{\text{small}} = \left(\frac{1}{\nu_{\text{large}}} + \frac{1}{\hat{\nu}_{\text{obs}}}\right)^{-1} \quad (4)$$

where  $\hat{\nu}_{\text{obs}} = \nu_c(\nu_c + 1)(1 - \gamma)/(\nu_c + 3)$ ,  $\gamma = (1 + 1/M)B/T$ , and  $\nu_c$  are the complete degrees of freedom, the degrees of freedom used for inference when data are complete (Barnard and Rubin 1999).

The sample-sample adjustment (4) is applied to the degrees of freedom  $\nu$  when the specified command saves the residual degrees of freedom in `e(df_r)`. This number of degrees of freedom is used as the complete degrees of freedom,  $\nu_c$ , in the computation. (If `e(df_r)` varies across imputations, the smallest is used in the computation, resulting in conservative inference.) If `e(df_r)` is not set by the specified command or if the `nosmall` option is specified, then (2) is used to compute the degrees of freedom,  $\nu$ .

Parameter-specific significance levels, confidence intervals, and degrees of freedom as reported by `mi estimate` are computed using the formulas above.

The percentage of standard-error increase due to missing data, as reported by `mi estimate`, `dftable`, is computed as  $\{(T/\bar{U})^{1/2} - 1\} \times 100\%$ .

The FMIs due to missing data and relative efficiencies reported by `mi estimate`, `variable` are computed as follows.

The fraction of information about  $Q$  missing due to nonresponse is

$$\lambda = \frac{r + 2/(\nu_{\text{large}} + 3)}{r + 1}$$

where the RVI,  $r$ , is defined in (3).

The relative (variance) efficiency of using  $M$  imputations versus the infinite number of imputations is  $RE = (1 + \lambda/M)^{-1}$  (Rubin 1987, 114).

Also see Rubin (1987, 76–77) and Schafer (1997, 109–111) for details.

## Multivariate case

The approximation (1) can be generalized to the multivariate case:

$$(\mathbf{q} - \bar{\mathbf{q}}_M)\mathbf{T}^{-1}(\mathbf{q} - \bar{\mathbf{q}}_M)' / k \sim F_{k,\nu} \quad (5)$$

where  $F_{k,\nu}$  denotes an  $F$  distribution with  $k = \text{rank}(T)$  numerator degrees of freedom and  $\nu$  denominator degrees of freedom defined as in (2), where the RVI,  $r$ , is replaced with the average RVI,  $r_{\text{ave}}$ :

$$r_{\text{ave}} = (1 + 1/M)\text{tr}(\mathbf{B}\bar{\mathbf{U}}^{-1})/k$$

The approximation (5) is inadequate with a small number of imputations because the between-imputation variance,  $\mathbf{B}$ , cannot be estimated reliably based on small  $M$ . Moreover, when  $M$  is smaller than the number of estimated parameters,  $\mathbf{B}$  does not have a full rank. As such, the total variance,  $\mathbf{T}$ , may not be a valid variance–covariance matrix for  $\bar{\mathbf{q}}_M$ .

One solution is to assume that the between-imputation and within-imputation matrices are proportional, that is  $B = \bar{\lambda} \times \bar{\mathbf{U}}$  (Rubin 1987, 78). This assumption implies that FMIs of all estimated parameters are equal. Under this assumption, approximation (5) becomes

$$(1 + r_{\text{ave}})^{-1}(\mathbf{q} - \bar{\mathbf{q}}_M)\bar{\mathbf{U}}^{-1}(\mathbf{q} - \bar{\mathbf{q}}_M)' / k \sim F_{k,\nu_*} \quad (6)$$

where  $k = \text{rank}(U)$  and  $\nu_*$  is computed as described in Li, Meng, Raghunathan, and Rubin (1991, 1067).

Also see Rubin (1987, 77–78) and Schafer (1997, 112–114) for details.

We refer to (6) as an equal FMI test and to (5) as the unrestricted FMI test. By default, `mi estimate` uses the approximation (6) for the model test. If the `ufmitest` option is specified, it uses the approximation (5) for the model test.

Similar to the univariate case, the degrees of freedom  $\nu_*$  and  $\nu$  are adjusted for small samples when the command saves the completed-data residual degrees of freedom in `e(df_r)`.

In the small-sample case, the degrees of freedom  $\nu_*$  is computed as described in Reiter (2007) (in the rare case, when  $k(M - 1) \leq 4$ ,  $\nu_* = (k + 1)\nu_1/2$ , where  $\nu_1$  is the degrees of freedom from Barnard and Rubin [1999]). In the small-sample case, the degrees of freedom  $\nu$  is computed as described in Barnard and Rubin (1999) and Marchenko and Reiter (2009).

## References

- Barnard, J., and D. B. Rubin. 1999. Small-sample degrees of freedom with multiple imputation. *Biometrika* 86: 948–955.
- Li, K.-H., X.-L. Meng, T. E. Raghunathan, and D. B. Rubin. 1991. Significance levels from repeated  $p$ -values with multiply-imputed data. *Statistica Sinica* 1: 65–92.
- Marchenko, Y. V., and J. P. Reiter. 2009. Improved degrees of freedom for multivariate significance tests obtained from multiply imputed, small-sample data. *Stata Journal* 9: 388–397.
- Reiter, J. P. 2007. Small-sample degrees of freedom for multi-component significance tests with multiple imputation for missing data. *Biometrika* 94: 502–508.
- Royston, P., J. B. Carlin, and I. R. White. 2009. Multiple imputation of missing values: New features for `mim`. *Stata Journal* 9: 252–264.
- Rubin, D. B. 1987. *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley.
- Schafer, J. L. 1997. *Analysis of Incomplete Multivariate Data*. Boca Raton, FL: Chapman & Hall/CRC.

## Also see

[MI] **mi estimate using** — Estimation using previously saved estimation results

[MI] **mi estimate postestimation** — Postestimation tools for `mi estimate`

[MI] **intro substantive** — Introduction to multiple-imputation analysis

[MI] **intro** — Introduction to `mi`

[MI] **Glossary**