Title

stata.com

mi estimate — Estimation using multiple imputations

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).

Menu

Statistics > Multiple imputation

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**.

options	Description
Options	
nimputations(#)	specify number of imputations to use; default is to use all existing imputations
imputations(*numlist*)	specify which imputations to use
mcerror	compute Monte Carlo error estimates
ufmitest	perform unrestricted FMI model test
nosmall	do not apply small-sample correction to degrees of freedom
saving(*miestfile*[, replace])	save individual estimation results to *miestfile*.ster
Tables	
[no]citable	suppress/display standard estimation table containing parameter-specific confidence intervals; default is citable
dftable	display degrees-of-freedom table; dftable implies nocitable
vartable	display variance information about estimates; vartable implies citable
table_options	control table output
display_options	control columns and column formats, row spacing, display of omitted variables and base and empty cells, and factor-variable labeling
Reporting	
level(#)	set confidence level; default is level(95)
dots	display dots as estimations are performed
noisily	display any output from *estimation_command* (and from nlcom if transformations specified)
trace	trace *estimation_command* (and nlcom if transformations specified); implies noisily
nogroup	suppress summary about groups displayed for xt commands
me_options	control output from mixed-effects commands
Advanced	
esample(*newvar*)	store estimation sample in variable *newvar*; available only in the flong and flongsep styles
errorok	allow estimation even when *estimation_command* (or nlcom) errors out; such imputations are discarded from the analysis
esampvaryok	allow estimation when estimation sample varies across imputations
cmdok	allow estimation when *estimation_command* is not one of the supported estimation commands
coeflegend	display legend instead of statistics
nowarning	suppress the warning about varying estimation samples
eform_option	display coefficient table in exponentiated form
post	post estimated coefficients and VCE to e(b) and e(V)
noupdate	do not perform mi update; see [MI] **noupdate option**

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

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

table_options	Description
noheader	suppress table header(s)
notable	suppress table(s)
nocoef	suppress table output related to coefficients
nocmdlegend	suppress command legend that appears in the presence of transformed coefficients when nocoef is used
notrcoef	suppress table output related to transformed coefficients
nolegend	suppress table legend(s)
nocnsreport	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.

Options

Options

nimputations(#) specifies that the first # imputations be used; # must be $M_{\min} \leq \# \leq M$, where $M_{\min} = 3$ if mcerror is specified and $M_{\min} = 2$, otherwise. 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. If mcerror is specified, *numlist* must contain at least three numbers. Only one of nimputations() or imputations() may be specified.

mcerror specifies to compute Monte Carlo error (MCE) estimates for the results displayed in the estimation, degrees-of-freedom, and variance-information tables. MCE estimates reflect variability of MI results across repeated uses of the same imputation procedure and are useful for determining an adequate number of imputations to obtain stable MI results; see White, Royston, and Wood (2011) for details and guidelines.

MCE estimates are obtained by applying the jackknife procedure to multiple-imputation results. That is, the jackknife pseudovalues of MI results are obtained by omitting one imputation at a time; see [R] **jackknife** for details about the jackknife procedure. As such, the MCE computation requires at least three imputations.

If level() is specified during estimation, MCE estimates are obtained for confidence intervals using the specified confidence level instead of using the default 95% confidence level. If any of the options described in [R] ***eform_option*** is specified during estimation, MCE estimates for the coefficients, standard errors, and confidence intervals in the exponentiated form are also computed. mcerror can also be used upon replay to display MCE estimates. Otherwise, MCE estimates are not reported upon replay even if they were previously computed.

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 correction be made to the degrees of freedom. The small-sample correction is made by default to estimation commands that account for small samples. If the command stores residual degrees of freedom in `e(df_r)`, individual tests of coefficients (and transformed coefficients) use the small-sample correction of Barnard and Rubin (1999) and the overall model test uses the small-sample correction of Reiter (2007). If the command does not store 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 `vartable` 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`.

`vartable` 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. `vartable` 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 table output produced by `citable`, `dftable`, and `vartable`.

`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 `nocoef` is specified.

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

`nolegend` suppresses all table legends from the output.

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

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, and `sformat(%`*fmt*`)`; 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; see the list of commands under *Panel-data models* in [MI] **Estimation**.

me_options: `stddeviations`, `variance`, `noretable`, `nofetable`, and `estmetric`. These options are relevant only with the mixed-effects command `mixed` (see [ME] **mixed**). See the corresponding mixed-effects commands for more information. The `stddeviations` option is the default with `mi estimate`. 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 that 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 and examples

stata.com

`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

`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. MCE estimates of the parameters are also useful for determining the stability of MI results. You can use the `mcerror` option to obtain these estimates. Your data must contain at least three imputations to use `mcerror`.

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 5: 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. For example, `mi estimate: logistic` reports coefficients and not odds ratios as `logistic`. To obtain odds ratios, you must specify the `or` option with `mi estimate`:

```
. mi estimate, or: logistic ...
```

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 (BMI), 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.

```
. use https://www.stata-press.com/data/r16/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
                                                Largest FMI       =     0.1355
DF adjustment:   Large sample                   DF:     min       =   1,060.38
                                                        avg       = 223,362.56
                                                        max       = 493,335.88
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|	[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, the largest fraction of missing information (FMI), 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 χ^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 and the FMI are not low, 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: 0.0312.

`Largest FMI` reports the largest of all the FMI about coefficient estimates due to nonresponse. This number can be used to get an idea of whether the specified number of imputations is sufficient for the analysis. A rule of thumb is that $M \geq 100 \times \text{FMI}$ provides an adequate level of reproducibility of MI analysis. In our example, the largest FMI is 0.14 and the number of imputations, 20, exceeds the required number of imputations: 14 ($= 100 \times 0.14$) according to this rule.

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 is 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 test** 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 because of 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
                                                Largest FMI       =     0.1355
DF adjustment:   Large sample                   DF:     min       =   1,060.38
                                                        avg       = 223,362.56
                                                        max       = 493,335.88
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 a percentage 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 5% 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 `vartable` 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
Logistic regression
Variance information
```

	Imputation variance					Relative
	Within	Between	Total	RVI	FMI	efficiency
smokes	.127048	.00094	.128035	.007765	.007711	.999615
age	.000237	1.4e-06	.000238	.006245	.00621	.99969
bmi	.001964	.000289	.002267	.154545	.135487	.993271
hsgrad	.162206	.001675	.163965	.010843	.010739	.999463
female	.172187	.001203	.17345	.007338	.00729	.999636
_cons	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) and White, Royston, and Wood (2011) for other ways of determining a suitable number of imputations.

Example 2: Completed-data linear 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 https://www.stata-press.com/data/r16/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       =     0.0685
                                                Largest FMI       =     0.2075
                                                Complete DF       =        109
DF adjustment:   Small sample                   DF:     min       =      48.59
                                                        avg       =      85.22
                                                        max       =     104.79
Model F test:       Equal FMI                   F(   7,  103.9)   =      67.50
Within VCE type:          OLS                   Prob > F          =     0.0000
```

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
tax	.6631356	.122443	5.42	0.000	.4195447	.9067265
sqft	.2185884	.0670182	3.26	0.002	.0856051	.3515718
age	-.0395402	1.613185	-0.02	0.981	-3.28205	3.202969
nfeatures	8.735622	13.42251	0.65	0.517	-18.01198	35.48323
ne	4.069381	36.94491	0.11	0.913	-69.4355	77.57426
custom	130.4925	42.93286	3.04	0.003	45.36257	215.6225
corner	-71.25406	40.06697	-1.78	0.078	-150.7152	8.207084
_cons	130.2002	70.38012	1.85	0.068	-9.624642	270.025

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 with complete data. It corresponds to the completed-data residual degrees of freedom stored 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 https://www.stata-press.com/data/r16/mdrugtrs25
(Patient Survival in Drug Trial)
. mi describe
  Style:  mlong
          last mi update 19apr2019 14:00:11, 8 days ago
  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. Missing values were imputed following guidelines in White and Royston (2009).

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 observations
          0  exclusions
------------------------------------------------------------------------------
         48  observations remaining, representing
         31  failures in single-record/single-failure data
        744  total analysis time at risk and under observation
                                        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
                                                Largest FMI       =     0.1567
DF adjustment:   Large sample                   DF:     min       =     998.63
                                                        avg       =  11,621.53
                                                        max       =  22,244.42
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
                                                Largest FMI       =     0.1567
DF adjustment:   Large sample                   DF:     min       =     998.63
                                                        avg       =  11,621.53
                                                        max       =  22,244.42
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 command, stcox. You see stcox's output only if you specify mi estimate's noisily option.

See Cleves, Gould, and Marchenko (2016, sec. 9.6) for more information on Cox regression with multiply imputed data.

Example 4: Panel data and multilevel models

We have data on the math scores of students in their third and fifth years of education. There are 887 students from 48 schools in inner London; see Mortimore et al. (1988) for more information on the study. We would like to fit a random-effects model to the fifth-year score, math5, on the third-year score, math3, using a random effect at the school level.

We created a version of the data that contains missing values for math3 and then performed imputation following guidelines from the Stata FAQ "How can I account for clustering when creating imputations with mi impute?"; see https://www.stata.com/support/faqs/stat/impute_cluster.html. The resulting imputed data are saved in mjsps5.dta:

```
. use https://www.stata-press.com/data/r16/mjsps5, clear
(LEA Junior School Project data (Mortimore et al., 1988) with missing values)
. mi describe
  Style:  mlong
          last mi update 19apr2019 14:00:11, 8 days ago
  Obs.:   complete          705
          incomplete        182  (M = 5 imputations)
          ------------------------
          total             887
  Vars.:  imputed:  1; math3(182)
          passive:  0
          regular:  2; school math5
          system:   3; _mi_m _mi_id _mi_miss
         (there are no unregistered variables)
```

There are five imputations for 182 missing values of the third-year score, math3. Variable math3 is an imputed variable, whereas variable math5 and variable school, recording school identifiers, are complete and are registered as regular.

Our random-effects model includes only a random intercept, the school effect, so we can use the xtreg command, or more specifically mi estimate: xtreg, for our primary analysis.

Without imputed data, to use xtreg or any other panel-data command, we must first declare data to be panel (xt) data by using xtset. With imputed data, we should use the mi xtset command instead. We declare school as our panel variable:

```
. mi xtset school
       panel variable:  school (unbalanced)
```

Next we use `mi estimate: xtreg` to regress the fifth-year math score on the third-year score.

```
. mi estimate: xtreg math5 math3
Multiple-imputation estimates                   Imputations       =          5
Random-effects GLS regression                   Number of obs     =        887
Group variable: school                          Number of groups  =         48
                                                Obs per group:
                                                              min =          5
                                                              avg =       18.5
                                                              max =         62
                                                Average RVI       =     0.0595
                                                Largest FMI       =     0.1071
DF adjustment:   Large sample                   DF:     min       =     381.40
                                                        avg       =  85,771.71
                                                        max       = 171,162.01
Model F test:       Equal FMI                   F(   1,  381.4)   =     305.71
Within VCE type: Conventional                   Prob > F          =     0.0000
```

math5	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
math3	.6101277	.0348951	17.48	0.000	.5415168	.6787385
_cons	30.48295	.3576417	85.23	0.000	29.78198	31.18392
sigma_u	2.0684286					
sigma_e	5.3206673					
rho	.13128791	(fraction of variance due to u_i)				

Note: **sigma_u** and **sigma_e** are combined in the original metric.

Third-year math scores are positively associated with fifth-year math scores. Because we use a random-effects model, the coefficient on `math3` is for comparison of students from the same school or from different schools.

In the above results, multiple-imputation estimates of variance components `sigma_u` and `sigma_e` are obtained by applying Rubin's combination rules to the completed-data estimates in the original, standard deviation metric.

Alternatively, we can use the `mixed` command to fit our two-level random-effects model and to obtain variance-component estimates of the school effect. `mixed` can be used to fit more complicated multilevel models; see [ME] **mixed** for details.

We fit a two-level linear model with `mi estimate: mixed` and specify `school` as our second-level variable. `mixed` does not require prior declaration of the data, so we do not need to use `mi xtset` with `mi estimate: mixed`:

```
. mi estimate: mixed math5 math3 || school: , reml
Multiple-imputation estimates                   Imputations       =          5
Mixed-effects REML regression                   Number of obs     =        887
Group variable: school                          Number of groups  =         48
                                                Obs per group:
                                                              min =          5
                                                              avg =       18.5
                                                              max =         62
                                                Average RVI       =     0.0574
                                                Largest FMI       =     0.1079
DF adjustment:   Large sample                   DF:     min       =     376.05
                                                        avg       =  44,112.02
                                                        max       = 167,428.86
Model F test:       Equal FMI                   F(   1,  376.0)   =     305.41
                                                Prob > F          =     0.0000
```

math5	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
math3	.6100335	.0349069	17.48	0.000	.5413963	.6786708
_cons	30.48217	.3536049	86.20	0.000	29.78911	31.17522

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
school: Identity				
sd(_cons)	2.033826	.3069989	1.512894	2.734129
sd(Residual)	5.321503	.1355669	5.061821	5.594508

The estimated coefficients, random-effects standard deviations, and other statistics are similar to those from `mi estimate: xtreg`. Unlike `mi estimate: xtreg`, the `mi estimate: mixed` command combines variance components in the estimation metric described in [ME] **mixed** and then back-transforms the estimates to display results in the original metric. In our example, the reported standard deviations are exponentiated multiple-imputation estimates of the log standard-deviations.

The random-effects parameters are displayed as standard deviations. We can display variances instead by replaying the `mi estimate` command with the `variance` option:

```
. mi estimate, variance
Multiple-imputation estimates                   Imputations       =          5
Mixed-effects REML regression                   Number of obs     =        887
Group variable: school                          Number of groups  =         48
                                                Obs per group:
                                                              min =          5
                                                              avg =       18.5
                                                              max =         62
                                                Average RVI       =     0.0574
                                                Largest FMI       =     0.1079
DF adjustment:   Large sample                   DF:     min       =     376.05
                                                        avg       =  44,112.02
                                                        max       = 167,428.86
Model F test:       Equal FMI                   F(   1,  376.0)   =     305.41
                                                Prob > F          =     0.0000
```

math5	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
math3	.6100335	.0349069	17.48	0.000	.5413963	.6786708
_cons	30.48217	.3536049	86.20	0.000	29.78911	31.17522

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
school: Identity				
var(_cons)	4.136447	1.248765	2.288848	7.475462
var(Residual)	28.3184	1.442839	25.62204	31.29852

Although the random-effects parameters are now displayed as variances, they are still combined and stored in the log–standard-deviation metric.

Example 5: 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 to use the method 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

```
. use https://www.stata-press.com/data/r16/mhouses1993s30
(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       =     0.0648
                                                Largest FMI       =     0.2533
                                                Complete DF       =        109
DF adjustment:   Small sample                   DF:     min       =      69.12
                                                        avg       =      94.02
                                                        max       =     105.51
Model F test:       Equal FMI                   F(   7,  106.5)   =      67.18
Within VCE type:          OLS                   Prob > F          =     0.0000
```

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
tax	.6768015	.1241568	5.45	0.000	.4301777	.9234253
sqft	.2118129	.069177	3.06	0.003	.0745091	.3491168
age	.2471445	1.653669	0.15	0.882	-3.051732	3.546021
nfeatures	9.288033	13.30469	0.70	0.487	-17.12017	35.69623
ne	2.518996	36.99365	0.07	0.946	-70.90416	75.94215
custom	134.2193	43.29755	3.10	0.002	48.35674	220.0818
corner	-68.58686	39.9488	-1.72	0.089	-147.7934	10.61972
_cons	123.9118	71.05816	1.74	0.085	-17.19932	265.0229

```
Transformations                                 Average RVI       =     0.2899
                                                Largest FMI       =     0.2316
                                                Complete DF       =        109
DF adjustment:   Small sample                   DF:     min       =      72.51
                                                        avg       =      72.51
Within VCE type:          OLS                           max       =      72.51
       ratio: _b[age]/_b[sqft]
```

price	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
ratio	1.44401	8.217266	0.18	0.861	-14.93485	17.82287

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 `nocoef` option. The header reports the average RVI due to nonresponse, the largest FMI, 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 test** for an example of linear transformation.

Example 6: Monte Carlo error estimates

Multiple imputation is a stochastic procedure. Each time we reimpute our data, we get different sets of imputations because of the randomness of the imputation step, and therefore we get different multiple-imputation estimates. However, we want to be able to reproduce MI results. Of course, we can always set the random-number seed to ensure reproducibility by obtaining the same imputed values. However, what if we use a different seed? Would we not want our results to be similar regardless of what seed we use? This leads us to a notion we call statistical reproducibility—we want results to be similar across repeated uses of the same imputation procedure; that is, we want to minimize the simulation error associated with our results.

To assess the level of simulation error, White, Royston, and Wood (2011) propose to use a Monte Carlo error of the MI results, defined as the standard deviation of the results across repeated runs of the same imputation procedure using the same data. The authors suggest evaluating Monte Carlo error estimates not only for parameter estimates but also for other statistics, including p-values and confidence intervals, as well as MI statistics including RVI and FMI.

Clearly, as the number of imputations increases, the simulation error decreases. Consider the total MI variance $T = \overline{U} + B + B/M$ of a single parameter, where $\overline{U}$ is the within-imputation variance and B is the between-imputation variance; see *Methods and formulas* for details. The term B/M reflects the increase in variance due to using a finite number of imputations, and its square root defines the Monte Carlo error associated with a single parameter. In general, Monte Carlo error estimates are obtained by applying a jackknife procedure to MI results. That is, an MCE estimate of an MI statistic is the standard error of the mean of the pseudovalues for that statistic, computed by omitting one imputation at a time; see [R] **jackknife** for technical details.

Consider our heart attack data analysis from example 1. Let's compute Monte Carlo error estimates of MI results. To obtain MCE estimates, we specify the `mcerror` option during estimation:

```
. use https://www.stata-press.com/data/r16/mheart1s20
(Fictional heart attack data; bmi missing)
. mi estimate, dots mcerror: 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
                                                Largest FMI       =     0.1355
DF adjustment:   Large sample                   DF:     min       =   1,060.38
                                                        avg       = 223,362.56
                                                        max       = 493,335.88
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|	[95% Conf.	Interval]
smokes	1.198595	.3578195	3.35	0.001	.4972789	1.899911
	.0068541	.0008562	0.01	0.000	.0056572	.0082212
age	.0360159	.0154399	2.33	0.020	.0057541	.0662776
	.0002654	.0000351	0.01	0.001	.0002319	.0003108
bmi	.1039416	.0476136	2.18	0.029	.010514	.1973692
	.0038014	.0008904	0.09	0.006	.0039928	.0044049
hsgrad	.1578992	.4049257	0.39	0.697	-.6357464	.9515449
	.0091517	.0010209	0.02	0.016	.0086215	.0100602
female	-.1067433	.4164735	-0.26	0.798	-.9230191	.7095326
	.0077566	.0009279	0.02	0.015	.006985	.0088408
_cons	-5.478143	1.685075	-3.25	0.001	-8.782394	-2.173892
	.1079841	.0248274	0.07	0.000	.1310618	.1050817

```
Note: Values displayed beneath estimates are Monte Carlo error estimates.
```

As the note describes, MCE estimates are displayed beneath parameter estimates. Following practical guidelines from White, Royston, and Wood (2011), MCE estimates of coefficients should be less than 10% of the standard errors of the coefficients; MCE estimates of test statistics should be approximately 0.1; and MCE estimates of p-values should be approximately 0.01 when the true p-value is 0.05 and 0.02 when the true p-value is 0.1. Our results based on 20 imputations satisfy these conditions, so we can be reasonably sure about the statistical reproducibility of our results.

We can also see Monte Carlo error estimates for other MI statistics reported by the `vartable` option. To redisplay Monte Carlo error estimates, we use the `mcerror` option upon replay. We also suppress the coefficient table by using the `nocitable` option.

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

	Imputation variance					Relative
	Within	Between	Total	RVI	FMI	efficiency
smokes	.127048	.00094	.128035	.007765	.007711	.999615
	.000559	.000211	.000613	.001744	.00172	.00009
age	.000237	1.4e-06	.000238	.006245	.00621	.99969
	8.6e-07	4.6e-07	1.1e-06	.002054	.002033	.000107
bmi	.001964	.000289	.002267	.154545	.135487	.993271
	.000026	.000077	.000085	.04134	.031986	.00166
hsgrad	.162206	.001675	.163965	.010843	.010739	.999463
	.000521	.000552	.000827	.003579	.003516	.000185
female	.172187	.001203	.17345	.007338	.00729	.999636
	.000614	.000297	.000773	.001811	.001788	.000094
_cons	2.5946	.233211	2.83948	.094377	.086953	.995671
	.029651	.070081	.083436	.028332	.024216	.001263

```
Note: Values displayed beneath estimates are Monte Carlo error estimates.
```

MCE estimates of all statistics are small.

What if we want to see MCE estimates of odds ratios? We know that we can use the `or` option on replay to redisplay results as odds ratios. However, using this option in combination with `mcerror` upon replay will not display MCE estimates of odds ratios:

```
. mi estimate, or mcerror
Multiple-imputation estimates                   Imputations       =         20
Logistic regression                             Number of obs     =        154
                                                Average RVI       =     0.0312
                                                Largest FMI       =     0.1355
DF adjustment:   Large sample                   DF:     min       =   1,060.38
                                                        avg       = 223,362.56
                                                        max       = 493,335.88
Model F test:       Equal FMI                   F(   5,71379.3)   =       3.59
Within VCE type:          OIM                   Prob > F          =     0.0030
```

attack	Odds Ratio	Std. Err.	t	P>|t|	[95% Conf.	Interval]
smokes	3.315455	1.186334	3.35	0.001	1.644241	6.685298
age	1.036672	.0160061	2.33	0.020	1.005771	1.068523
bmi	1.109536	.052829	2.18	0.029	1.010569	1.218194
hsgrad	1.171048	.4741875	0.39	0.697	.5295401	2.589707
female	.8987564	.3743082	-0.26	0.798	.3973177	2.033041
_cons	.0041771	.0070387	-3.25	0.001	.0001534	.1137342

```
Note: _cons estimates baseline odds.
Note: Monte Carlo error estimates are not available for exponentiated
      coefficients.
```

The same applies to a combination of the `level()` and `mcerror` options specified on replay to try to display MCE estimates of confidence intervals for a confidence level other than the one used during estimation.

To compute MCE estimates for odds ratios in addition to coefficients, you need to specify the `or` option in combination with `mcerror` during estimation. Similarly, to compute MCE estimates for confidence intervals with a specific confidence level, you need to specify the `level()` option in combination with `mcerror` during estimation. Otherwise, MCE estimates of 95% confidence intervals are computed.

```
. mi estimate, mcerror or level(90): logit attack smokes age bmi hsgrad female
Multiple-imputation estimates                   Imputations       =         20
Logistic regression                             Number of obs     =        154
                                                Average RVI       =     0.0312
                                                Largest FMI       =     0.1355
DF adjustment:   Large sample                   DF:     min       =   1,060.38
                                                        avg       = 223,362.56
                                                        max       = 493,335.88
Model F test:       Equal FMI                   F(   5,71379.3)   =       3.59
Within VCE type:          OIM                   Prob > F          =     0.0030
```

attack	Odds Ratio	Std. Err.	t	P>|t|	[90% Conf.	Interval]
smokes	3.315455	1.186334	3.35	0.001	1.840491	5.97245
	.0227267	.0104806	0.01	0.000	.0107398	.0477351
age	1.036672	.0160061	2.33	0.020	1.010676	1.063337
	.0002752	.000039	0.01	0.001	.0002388	.0003221
bmi	1.109536	.052829	2.18	0.029	1.025885	1.200007
	.0042178	.001033	0.09	0.006	.0040064	.0051089
hsgrad	1.171048	.4741875	0.39	0.697	.6016087	2.279478
	.0107188	.0049031	0.02	0.016	.0052248	.02254
female	.8987564	.3743082	-0.26	0.798	.4530363	1.782998
	.0069686	.00341	0.02	0.015	.0032087	.0154128
_cons	.0041771	.0070387	-3.25	0.001	.000261	.0668412
	.0004519	.0007338	0.07	0.000	.0000336	.0068716

```
Note: _cons estimates baseline odds.
Note: Values displayed beneath estimates are Monte Carlo error estimates.
```

Similarly to the MCE estimates for coefficients, the MCE estimates for odds ratios are within acceptable limits.

If you wish to obtain Monte Carlo error estimates of confidence intervals for a number of different confidence levels, a more computationally efficient way of doing so is to use **`mi estimate using`** (see [MI] **mi estimate using**).

First, use **`mi estimate`** to save individual estimation results from a model to an estimation file:

```
. mi estimate, saving(miest): ...
```

Then use **`mi estimate using`** to obtain MCE estimates for different confidence intervals,

```
. mi estimate using miest, mcerror level(90) ...
. mi estimate using miest, mcerror level(80) ...
```

or for odds ratios,

```
. mi estimate using miest, mcerror or ...
```

without refitting the model.

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 message 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 message "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 that "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 1) includes 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. You 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 their variance–covariance estimates in `e(b)` and `e(V)` by specifying the `post` option. See [MI] **mi estimate postestimation** for postestimation tools available after `mi estimate`.

❑

Stored results

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

Scalars

`e(df_avg[_Q]_mi)`	average degrees of freedom
`e(df_c_mi)`	complete degrees of freedom (if originally stored by *estimation_command* in `e(df_r)`)
`e(df_max[_Q]_mi)`	maximum degrees of freedom
`e(df_min[_Q]_mi)`	minimum degrees of freedom
`e(df_m_mi)`	MI model test denominator (residual) degrees of freedom
`e(df_r_mi)`	MI model test numerator (model) degrees of freedom
`e(esampvary_mi)`	varying-estimation sample flag (`0` or `1`)
`e(F_mi)`	model test F statistic
`e(k_exp_mi)`	number of expressions (transformed coefficients)
`e(M_mi)`	number of imputations
`e(N_mi)`	number of observations (minimum, if varies)
`e(N_min_mi)`	minimum number of observations
`e(N_max_mi)`	maximum number of observations
`e(N_g_mi)`	number of groups
`e(g_min_mi)`	smallest group size
`e(g_avg_mi)`	average group size
`e(g_max_mi)`	largest group size
`e(p_mi)`	MI model test p-value
`e(cilevel_mi)`	confidence level used to compute Monte Carlo error estimates of confidence intervals
`e(fmi_max[_Q]_mi)`	largest FMI
`e(rvi_avg[_Q]_mi)`	average RVI
`e(rvi_avg_F_mi)`	average RVI associated with the residual degrees of freedom for model test
`e(ufmi_mi)`	`1` if unrestricted FMI model test is performed, `0` if equal FMI model test is performed

Macros

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

Matrices

e(b)	MI estimates of coefficients (equals e(b_mi); stored only if post is used)
e(V)	variance–covariance matrix (equals e(V_mi); stored only if post is used)
e(Cns)	constraint matrix, for constrained estimation only (equals e(Cns_mi); stored only if post is used)
e(N_g_mi)	group counts
e(g_min_mi)	group-size minimums
e(g_avg_mi)	group-size averages
e(g_max_mi)	group-size maximums
e(b[_Q]_mi)	MI estimates of coefficients (or transformed coefficients)
e(V[_Q]_mi)	variance–covariance matrix (total variance)
e(Cns_mi)	constraint matrix (for constrained estimation only)
e(W[_Q]_mi)	within-imputation variance matrix
e(B[_Q]_mi)	between-imputation variance matrix
e(re[_Q]_mi)	parameter-specific relative efficiencies
e(rvi[_Q]_mi)	parameter-specific RVIs
e(fmi[_Q]_mi)	parameter-specific FMIs
e(df[_Q]_mi)	parameter-specific degrees of freedom
e(pise[_Q]_mi)	parameter-specific percentages increase in standard errors
e(*vs_names*_vs_mi)	values of command-specific e() scalar *vs_names* 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 stored for panel-data models only. The results are stored as matrices for mixed-effects models and as scalars for other panel-data models.

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

Command-specific e() results that remain constant across imputations are also stored. Command-specific results that vary from imputation to imputation are posted as missing, and their names are stored in the corresponding macros e(names_vvl_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 stored in a corresponding matrix with the _vs_mi suffix.

In addition to the above, the following is stored in r():

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when

any r-class command is run after the estimation command.

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 $\{(\widehat{\mathbf{q}}_i, \widehat{\mathbf{U}}_i) : i = 1, 2, \ldots, M\}$ 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

$$\overline{\mathbf{q}}_M = \frac{1}{M}\sum_{i=1}^{M}\widehat{\mathbf{q}}_i$$

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

$$\mathbf{T} = \overline{\mathbf{U}} + (1 + \frac{1}{M})\mathbf{B}$$

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

Methods and formulas are presented under the following headings:

Univariate case
Multivariate case

Univariate case

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

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

where t_ν denotes a Student's t distribution with ν 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

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

where

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

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

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

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

The small-sample adjustment (4) is applied to the degrees of freedom ν when the specified command stores the residual degrees of freedom in `e(df_r)`. This number of degrees of freedom is used as the complete degrees of freedom, ν_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, ν.

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 $\{(T/\overline{U})^{1/2} - 1\} \times 100\%$.

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

In the large-sample case, the fraction of information about Q missing due to nonresponse (Rubin 1987, 77) is

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

where the RVI, r, is defined in (3). In the small-sample case, the fraction of information about Q missing due to nonresponse (Barnard and Rubin 1999, 953) is

$$\lambda = 1 - \frac{\lambda(\nu_{\text{small}})}{\lambda(\nu_c)}\frac{\overline{U}}{T}$$

where $\lambda(u) = (u + 1)/(u + 3)$.

The relative (variance) efficiency of using M imputations versus the infinite number of imputations is $\text{RE} = (1 + \lambda/M)^{-1}$ (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:

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

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

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

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 $\overline{\mathbf{q}}_M$.

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

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

where $k = \text{rank}(U)$ and $\nu_\star$ is computed as described in Li et al. (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 $\nu_\star$ and ν are adjusted for small samples when the command stores the completed-data residual degrees of freedom in `e(df_r)`.

In the small-sample case, the degrees of freedom $\nu_\star$ is computed as described in Reiter (2007) (in the rare case, when $k(M-1) \leq 4$, $\nu_\star = (k+1)\nu_1/2$, where ν_1 is the degrees of freedom from Barnard and Rubin [1999]). In the small-sample case, the degrees of freedom ν is computed as described in Barnard and Rubin (1999) and Marchenko and Reiter (2009).

Acknowledgments

The `mi estimate` command was inspired by the community-contributed command `mim` by John Carlin of the Murdoch Children's Research Institute and University of Melbourne; John Galati of Northside Christian College, Bundoora, Melbourne; Patrick Royston of the MRC Clinical Trials Unit, London, and coauthor of the Stata Press book *Flexible Parametric Survival Analysis Using Stata: Beyond the Cox Model*; and Ian White of the MRC Biostatistics Unit, London. We greatly appreciate the authors for their extensive body of work in Stata in the multiple-imputation area.

References

Aloisio, K. M., N. Micali, S. A. Swanson, A. Field, and N. J. Horton. 2014. Analysis of partially observed clustered data using generalized estimating equations and multiple imputation. *Stata Journal* 14: 863–883.

Barnard, J., and D. B. Rubin. 1999. Small-sample degrees of freedom with multiple imputation. *Biometrika* 86: 948–955.

Cleves, M. A., W. W. Gould, and Y. V. Marchenko. 2016. *An Introduction to Survival Analysis Using Stata*. Rev. 3rd ed. College Station, TX: Stata Press.

Comulada, W. S. 2015. Model specification and bootstrapping for multiply imputed data: An application to count models for the frequency of alcohol use. *Stata Journal* 15: 833–844.

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.

Mortimore, P., P. Sammons, L. Stoll, D. Lewis, and R. Ecob. 1988. *School Matters*. Berkeley, CA: University of California Press.

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.

Wagstaff, D. A., and O. Harel. 2011. A closer examination of three small-sample approximations to the multiple-imputation degrees of freedom. *Stata Journal* 11: 403–419.

——. 2019. A closer examination of three small-sample approximations to the multiple-imputation degrees of freedom, erratum. *Stata Journal* 19: 1021.

White, I. R., and P. Royston. 2009. Imputing missing covariate values for the Cox model. *Statistics in Medicine* 28: 1982–1998.

White, I. R., P. Royston, and A. M. Wood. 2011. Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine* 30: 377–399.

Also see

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

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

[MI] **Intro** — Introduction to mi

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

[MI] **Glossary**