Title

> **mi impute mlogit —** Impute using multinomial logistic regression

Description

mi impute mlogit fills in missing values of a nominal variable by using the multinomial (polytomous) logistic regression imputation method. You can perform separate imputations on different subsets of the data by specifying the by() option. You can also account for frequency, importance, and sampling weights.

Menu

Statistics > Multiple imputation

Syntax

> mi impute mlogit *ivar* [*indepvars*] [*if*] [*weight*] [, *impute_options options*]

impute_options	Description
Main	
* add(#)	specify number of imputations to add; required when no imputations exist
* replace	replace imputed values in existing imputations
rseed(#)	specify random-number seed
double	store imputed values in double precision; the default is to store them as float
by(*varlist*[, *byopts*])	impute separately on each group formed by *varlist*
Reporting	
dots	display dots as imputations are performed
noisily	display intermediate output
nolegend	suppress all table legends
Advanced	
force	proceed with imputation, even when missing imputed values are encountered
noupdate	do not perform mi update; see [MI] **noupdate option**

*add(#) is required when no imputations exist; add(#) or replace is required if imputations exist.
noupdate does not appear in the dialog box.

options	Description
[Main]	
<u>nocons</u>tant	suppress constant term
<u>baseo</u>utcome(*#*)	specify value of *ivar* that will be the base outcome
<u>augm</u>ent	perform augmented regression in the presence of perfect prediction
<u>conditional</u>(*if*)	perform conditional imputation
<u>boots</u>trap	estimate model parameters using sampling with replacement
[Maximization]	
maximize_options	control the maximization process; seldom used

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

You must mi register *ivar* as imputed before using mi impute mlogit; see [MI] **mi set**.

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

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

fweights, iweights, and pweights are allowed; see [U] **11.1.6 weight**.

Options

> [Main]

noconstant; see [R] **Estimation options**.

add(), replace, rseed(), double, by(); see [MI] **mi impute**.

baseoutcome(*#*) specifies the value of *ivar* to be treated as the base outcome. The default is to choose the most frequent outcome.

augment specifies that augmented regression be performed if perfect prediction is detected. By default, an error is issued when perfect prediction is detected. The idea behind the augmented-regression approach is to add a few observations with small weights to the data during estimation to avoid perfect prediction. See *The issue of perfect prediction during imputation of categorical data* under *Remarks and examples* in [MI] **mi impute** for more information. augment is not allowed with importance weights.

conditional(*if*) specifies that the imputation variable be imputed conditionally on observations satisfying *exp*; see [U] **11.1.3 if exp**. That is, missing values in a conditional sample, the sample identified by the *exp* expression, are imputed based only on data in that conditional sample. Missing values outside the conditional sample are replaced with a conditional constant, the value of the imputation variable in observations outside the conditional sample. As such, the imputation variable is required to be constant outside the conditional sample. Also, if any conditioning variables (variables involved in the conditional specification if *exp*) contain soft missing values (.), their missing values must be nested within missing values of the imputation variables. See *Conditional imputation* under *Remarks and examples* in [MI] **mi impute**.

bootstrap specifies that posterior estimates of model parameters be obtained using sampling with replacement; that is, posterior estimates are estimated from a bootstrap sample. The default is to sample the estimates from the posterior distribution of model parameters or from the large-sample normal approximation of the posterior distribution. This option is useful when asymptotic normality of parameter estimates is suspect.

⌐ Reporting ┐

dots, noisily, nolegend; see [MI] **mi impute**. noisily specifies that the output from the multino-
mial logistic regression fit to the observed data be displayed. nolegend suppresses all legends that
appear before the imputation table. Such legends include a legend about conditional imputation
that appears when the conditional() option is specified and group legends that may appear
when the by() option is specified.

⌐ Maximization ┐

maximize_options; see [R] **mlogit**. These options are seldom used. difficult, technique(), gra-
dient, showstep, hessian, and showtolerance are not allowed when the augment option is
used.

⌐ Advanced ┐

force; see [MI] **mi impute**.

The following option is available with mi impute but is not shown in the dialog box:

noupdate; see [MI] **noupdate option**.

Remarks and examples   stata.com

Remarks are presented under the following headings:

> Univariate imputation using multinomial logistic regression
> Using mi impute mlogit

See [MI] **mi impute** for a general description and details about options common to all imputation
methods, *impute_options*. Also see [MI] **Workflow** for general advice on working with mi.

Univariate imputation using multinomial logistic regression

The multinomial logistic regression imputation method can be used to fill in missing values of a
nomial variable (for example, Raghunathan et al. [2001] and van Buuren [2007]). It is a parametric
method that assumes an underlying multinomial logistic model for the imputed variable (given other
predictors). Similarly to the logistic imputation method, this method is based on the asymptotic
approximation of the posterior predictive distribution of the missing data.

Using mi impute mlogit

Consider the heart attack data introduced in [MI] **Intro substantive** and discussed in [MI] **mi impute**.
Suppose that we want our logistic model of interest to also include information about marital status
(categorical variable marstatus)—logit attack smokes age bmi female hsgrad i.marstatus.

We first tabulate values of `marstatus`:

```
. use https://www.stata-press.com/data/r18/mheart3
(Fictional heart attack data; marstatus missing)

. tabulate marstatus, missing
```

Marital status: single, married, divorced	Freq.	Percent	Cum.
Single	53	34.42	34.42
Married	48	31.17	65.58
Divorced	46	29.87	95.45
.	7	4.55	100.00
Total	154	100.00	

From the output, the `marstatus` variable has three unique categories and seven missing observations. Because `marstatus` is a categorical variable, we use the multinomial logistic imputation method to fill in its missing values.

We `mi set` the data, register `marstatus` as an imputed variable, and then create 10 imputations by specifying the `add(10)` option with `mi impute mlogit`:

```
. mi set mlong

. mi register imputed marstatus
(7 m=0 obs now marked as incomplete)

. mi impute mlogit marstatus attack smokes age bmi female hsgrad, add(10)
Univariate imputation                       Imputations =     10
Multinomial logistic regression                   added =     10
Imputed: m=1 through m=10                        updated =      0
```

	Observations per m			
Variable	Complete	Incomplete	Imputed	Total
marstatus	147	7	7	154

```
(Complete + Incomplete = Total; Imputed is the minimum across m
 of the number of filled-in observations.)
```

We can now analyze these multiply imputed data using logistic regression via `mi estimate`:

```
. mi estimate: logit attack smokes age bmi female hsgrad i.marstatus
  (output omitted)
```

Stored results

`mi impute mlogit` stores the following in `r()`:

Scalars
`r(M)`	total number of imputations
`r(M_add)`	number of added imputations
`r(M_update)`	number of updated imputations
`r(k_ivars)`	number of imputed variables (always 1)
`r(pp)`	1 if perfect prediction detected, 0 otherwise
`r(N_g)`	number of imputed groups (1 if by() is not specified)

Macros
`r(method)`	name of imputation method (mlogit)
`r(ivars)`	names of imputation variables
`r(rngstate)`	random-number state used
`r(by)`	names of variables specified within by()

Matrices
`r(N)`	number of observations in imputation sample in each group
`r(N_complete)`	number of complete observations in imputation sample in each group
`r(N_incomplete)`	number of incomplete observations in imputation sample in each group
`r(N_imputed)`	number of imputed observations in imputation sample in each group

Methods and formulas

Consider a univariate variable $\mathbf{x} = (x_1, x_2, \ldots, x_n)'$ that contains K categories (without loss of generality, let $k = 1$ be the base outcome) and follows a multinomial logistic model

$$\Pr(x_i = k | \mathbf{z}_i) = \begin{cases} \dfrac{1}{1 + \sum_{l=2}^{K} \exp(\mathbf{z}_i' \boldsymbol{\beta}_l)}, & \text{if } k = 1 \\[2ex] \dfrac{\exp(\mathbf{z}_i' \boldsymbol{\beta}_k)}{1 + \sum_{l=2}^{K} \exp(\mathbf{z}_i' \boldsymbol{\beta}_l)}, & \text{if } k > 1 \end{cases} \tag{1}$$

where $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iq})'$ records values of predictors of $\mathbf{x}$ for observation i and $\boldsymbol{\beta}_l$ is the $q \times 1$ vector of unknown regression coefficients for outcome $l = 2, \ldots, K$. (When a constant is included in the model—the default—$z_{i1} = 1$, $i = 1, \ldots, n$.)

$\mathbf{x}$ contains missing values that are to be filled in. Consider the partition of $\mathbf{x} = (\mathbf{x}_o', \mathbf{x}_m')$ into $n_0 \times 1$ and $n_1 \times 1$ vectors containing the complete and the incomplete observations. Consider a similar partition of $\mathbf{Z} = (\mathbf{Z}_o, \mathbf{Z}_m)$ into $n_0 \times q$ and $n_1 \times q$ submatrices.

`mi impute mlogit` follows the steps below to fill in $\mathbf{x}_m$:

1. Fit a multinomial logistic model (1) to the observed data $(\mathbf{x}_o, \mathbf{Z}_o)$ to obtain the maximum likelihood estimates, $\widehat{\boldsymbol{\beta}} = (\widehat{\boldsymbol{\beta}}_2', \ldots, \widehat{\boldsymbol{\beta}}_K')'$, and their asymptotic sampling variance, $\widehat{\mathbf{U}}$.

2. Simulate new parameters, $\boldsymbol{\beta}_\star$, from the large-sample normal approximation, $N(\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{U}})$, to its posterior distribution assuming the noninformative prior $\Pr(\boldsymbol{\beta}) \propto \text{const}$.

3. Obtain one set of imputed values, $\mathbf{x}_m^1$, by simulating from the multinomial logistic distribution: one of K categories is randomly assigned to a missing category, i_m, using the cumulative probabilities computed from (1) with $\boldsymbol{\beta}_l = \boldsymbol{\beta}_{\star l}$ and $\mathbf{z}_i = \mathbf{z}_{i_m}$.

4. Repeat steps 2 and 3 to obtain M sets of imputed values, $\mathbf{x}_m^1, \mathbf{x}_m^2, \ldots, \mathbf{x}_m^M$.

Steps 2 and 3 above correspond to only approximate draws from the posterior predictive distribution of the missing data $\Pr(\mathbf{x}_m | \mathbf{x}_o, \mathbf{Z}_o)$ because $\boldsymbol{\beta}_\star$ is drawn from the asymptotic approximation to its posterior distribution.

If weights are specified, a weighted multinomial logistic regression model is fit to the observed data in step 1 (see [R] **mlogit** for details).

References

Raghunathan, T. E., J. M. Lepkowski, J. Van Hoewyk, and P. Solenberger. 2001. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology* 27: 85–95.

van Buuren, S. 2007. Multiple imputation of discrete and continuous data by fully conditional specification. *Statistical Methods in Medical Research* 16: 219–242. https://doi.org/10.1177/0962280206074463.

Also see

[MI] **mi impute** — Impute missing values

[MI] **mi impute ologit** — Impute using ordered logistic regression

[MI] **mi estimate** — Estimation using multiple imputations

[MI] **Intro** — Introduction to mi

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