**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**

```stata
mi impute mlogit ivar [ indepvars ] [ if ] [ weight ] [ , impute_options options ]
```

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

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

**Main**

- **noconstant**: suppress constant term
- **baseoutcome(#)**: specify value of `ivar` that will be the base outcome
- **augment**: perform augmented regression in the presence of perfect prediction
- **conditional(if)**: perform conditional imputation
- **bootstrap**: 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.
`fweights`, `iweights`, and `pweights` are allowed; see [U] 11.1.6 weight.
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.

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

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

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 nominal 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/r16/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
                     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)
```
mi impute mlogit — Impute using multinomial logistic regression

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

\[
\Pr(x_i = k|z_i) = \begin{cases} 
1 & \text{if } k = 1 \\
\frac{\exp(z_i'\beta_k)}{1 + \sum_{l=2}^{K} \exp(z_i'\beta_l)} & \text{if } k > 1
\end{cases}
\]

where \( z_i = (z_{i1}, z_{i2}, \ldots, z_{iq})' \) records values of predictors of \( x \) for observation \( i \) and \( \beta_l \) 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—\( z_{i1} = 1, i = 1, \ldots, n \).)

\( x \) contains missing values that are to be filled in. Consider the partition of \( x = (x_o', x_m') \) into \( n_0 \times 1 \) and \( n_1 \times 1 \) vectors containing the complete and the incomplete observations. Consider a similar partition of \( Z = (Z_o, Z_m) \) into \( n_0 \times q \) and \( n_1 \times q \) submatrices.

mi impute mlogit follows the steps below to fill in \( x_m \):

1. Fit a multinomial logistic model (1) to the observed data \((x_o, Z_o)\) to obtain the maximum likelihood estimates, \( \hat{\beta} = (\hat{\beta}_2, \ldots, \hat{\beta}_K)' \), and their asymptotic sampling variance, \( \hat{U} \).
2. Simulate new parameters, \( \beta_* \), from the large-sample normal approximation, \( N(\hat{\beta}, \hat{U}) \), to its posterior distribution assuming the noninformative prior \( \Pr(\beta) \propto \text{const} \).
3. Obtain one set of imputed values, \( x_1^m \), 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 \( \beta_l = \beta_{*l} \) and \( z_i = z_{im} \).
4. Repeat steps 2 and 3 to obtain \( M \) sets of imputed values, \( x_1^m, x_2^m, \ldots, x_M^m \).

Steps 2 and 3 above correspond to only approximate draws from the posterior predictive distribution of the missing data \( \Pr(x_m|x_o, Z_o) \) because \( \beta_* \) 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


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