mi impute ologit — Impute using ordered logistic regression

Description

mi impute ologit fills in missing values of an ordinal variable using an ordered 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 ologit  ivar  [ indepvars ]  [ if ]  [ weight ]  [,  impute_options  options ]

<table>
<thead>
<tr>
<th>impute_options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main</strong></td>
<td></td>
</tr>
<tr>
<td>*add(#)</td>
<td>specify number of imputations to add; required when no imputations exist</td>
</tr>
<tr>
<td>*replace</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>noupdate</td>
<td>do not perform mi update; see [MI] noupdate option</td>
</tr>
</tbody>
</table>

*add(#) is required when no imputations exist; add(#) or replace is required if imputations exist. noupdate does not appear in the dialog box.
**mi impute ologit — Impute using ordered logistic regression**

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main</strong></td>
<td></td>
</tr>
<tr>
<td>offset(varname)</td>
<td>include varname in model with coefficient constrained to 1</td>
</tr>
<tr>
<td>augment</td>
<td>perform augmented regression in the presence of perfect prediction</td>
</tr>
<tr>
<td>conditional(if)</td>
<td>perform conditional imputation</td>
</tr>
<tr>
<td>bootstrap</td>
<td>estimate model parameters using sampling with replacement</td>
</tr>
<tr>
<td><strong>Maximization</strong></td>
<td>maximize_options control the maximization process; seldom used</td>
</tr>
</tbody>
</table>

You must mi set your data before using mi impute ologit; see [MI] mi set.
You must mi register ivar as imputed before using mi impute ologit; 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.

## Options

### Main

- add(), replace, rseed(), double, by(); see [MI] mi impute.
- offset(varname); see [R] Estimation options.
- 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 ordered 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.
Impute using ordered logistic regression

**Maximization**

`maximize_options;` see [R] `ologit`. These options are seldom used. `difficult`, `technique()`, `gradient`, `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

Remarks are presented under the following headings:

- Univariate imputation using ordered logistic regression
- Using `mi impute ologit`

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 ordered logistic regression

The ordered logistic regression imputation method can be used to fill in missing values of an ordinal variable (for example, Raghunathan et al. [2001] and van Buuren [2007]). It is a parametric method that assumes an underlying 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 ologit`

Following the example from [MI] `mi impute mlogit`, we consider the heart attack data (for example, [MI] `Intro substantive`, [MI] `mi impute`), where a logistic model of interest now includes information about alcohol consumption, variable `alcohol`—`logit attack smokes age bmi female hsgrad i.alcohol`.

```
use https://www.stata-press.com/data/r16/mheart4
(Fictional heart attack data; alcohol missing)
tabulate alcohol, missing
```

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>none, &lt;2 drinks/day,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>do not drink</td>
<td>18</td>
<td>11.69</td>
<td>11.69</td>
</tr>
<tr>
<td>less than 3 drinks/</td>
<td>83</td>
<td>53.90</td>
<td>65.58</td>
</tr>
<tr>
<td>day</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>three or more drinks/</td>
<td>44</td>
<td>28.57</td>
<td>94.16</td>
</tr>
<tr>
<td>day</td>
<td>9</td>
<td>5.84</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>154</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

From the output, the `alcohol` variable has three unique ordered categories and nine missing observations. We use the ordered logistic imputation method to impute missing values of `alcohol`. We create 10 imputations by specifying the `add(10)` option:
mi impute ologit — Impute using ordered logistic regression

. mi set mlong
. mi register imputed alcohol
(9 m=0 obs. now marked as incomplete)
. mi impute ologit alcohol attack smokes age bmi female hsgrad, add(10)

Univariate imputation
Imputations = 10
Ordered logistic regression
added = 10
Imputed: m=1 through m=10
updated = 0

<table>
<thead>
<tr>
<th>Variable</th>
<th>Complete</th>
<th>Incomplete</th>
<th>Imputed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>alcohol</td>
<td>145</td>
<td>9</td>
<td>9</td>
<td>154</td>
</tr>
</tbody>
</table>

(complete + incomplete = total; imputed is the minimum across m of the number of filled-in observations.)

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

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

Stored results

mi impute ologit 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 (ologit)
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 \( K \) ordered categories and follows an ordered logistic model

\[
\Pr(x_i = k|z_i) = \Pr(\gamma_{k-1} < z'_i\beta + u \leq \gamma_k) = \frac{1}{1 + \exp(-\gamma_k + z'_i\beta)} - \frac{1}{1 + \exp(-\gamma_{k-1} + z'_i\beta)}
\] (1)
$z_i = (z_{i1}, z_{i2}, \ldots, z_{iq})'$ records values of predictors of $x$ for observation $i$, $\beta$ is the $q \times 1$ vector of unknown regression coefficients, and $\gamma = (\gamma_1, \ldots, \gamma_{K-1})'$ are the unknown cutpoints with $\gamma_0 = -\infty$ and $\gamma_K = \infty$. (There is no constant in this model because its effect is absorbed into the cutpoints; see [R] ologit for details.)

$x$ 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 ologit follows the steps below to fill in $x_m$:

1. Fit an ordered logistic model (1) to the observed data $(x_o, Z_o)$ to obtain the maximum likelihood estimates, $\hat{\theta} = (\hat{\beta}', \hat{\gamma}')'$, and their asymptotic sampling variance, $\hat{U}$.

2. Simulate new parameters, $\theta_*$, from the large-sample normal approximation, $N(\hat{\theta}, \hat{U})$, to its posterior distribution assuming the noninformative prior $Pr(\theta) \propto \text{const}$.

3. Obtain one set of imputed values, $x_1^m$, by simulating from an ordered logistic distribution as defined by (1): one of $K$ categories is randomly assigned to a missing category, $i_m$, using the cumulative probabilities computed from (1) with $\beta = \beta_*$, $\gamma = \gamma_*$, and $z_i = z_{i_m}$.

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 $\theta_*$ is drawn from the asymptotic approximation to its posterior distribution.

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

References


Also see
[MI] mi impute — Impute missing values
[MI] mi impute mlogit — Impute using multinomial logistic regression
[MI] mi estimate — Estimation using multiple imputations
[MI] Intro — Introduction to mi
[MI] Intro substantive — Introduction to multiple-imputation analysis