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*]

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.

1

options	Description
Main	
<u>off</u>set(*varname*)	include *varname* in model with coefficient constrained to 1
augment	perform augmented regression in the presence of perfect prediction
<u>cond</u>itional(*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 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**.

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

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.

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.

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.

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/r19/mheart4
(Fictional heart attack data; alcohol missing)

. tabulate alcohol, missing
```

Alcohol consumption: none, <2 drinks/day, >=2 drinks/day	Freq.	Percent	Cum.
Do not drink	18	11.69	11.69
Less than 3 drinks/day	83	53.90	65.58
Three or more drinks/day	44	28.57	94.16
.	9	5.84	100.00
Total	154	100.00	

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 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

	Observations per m			
Variable	Complete	Incomplete	Imputed	Total
alcohol	145	9	9	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 with logistic regression via mi estimate:

```
. mi estimate: logit attack smokes age bmi female hsgrad i.alcohol
```

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 $\mathbf{x} = (x_1, x_2, \ldots, x_n)'$ that contains K ordered categories and follows an ordered logistic model

$$
\begin{aligned}
\Pr(x_i = k | \mathbf{z}_i) &= \Pr(\gamma_{k-1} < \mathbf{z}_i'\boldsymbol{\beta} + u \le \gamma_k) \\
&= \frac{1}{1 + \exp(-\gamma_k + \mathbf{z}_i'\boldsymbol{\beta})} - \frac{1}{1 + \exp(-\gamma_{k-1} + \mathbf{z}_i'\boldsymbol{\beta})}
\end{aligned} \tag{1}
$$

where $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iq})'$ records values of predictors of $\mathbf{x}$ for observation i, $\boldsymbol{\beta}$ is the $q \times 1$ vector of unknown regression coefficients, and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_{K-1})'$ 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.)

$\mathbf{x}$ 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 ologit follows the steps below to fill in $\mathbf{x}_m$:

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

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

3. Obtain one set of imputed values, $\mathbf{x}_m^1$, 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 $\boldsymbol{\beta} = \boldsymbol{\beta}_\star$, $\boldsymbol{\gamma} = \boldsymbol{\gamma}_\star$, 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 $\boldsymbol{\theta}_\star$ 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

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