Title

stata.com

mi impute truncreg — Impute using truncated regression

Description

`mi impute truncreg` fills in missing values of a continuous variable with a restricted range using a truncated regression imputation method. You can perform separate imputations on different subsets of the data by specifying the `by()` option. You can also account for analytic, frequency, importance, and sampling weights.

Menu

Statistics > Multiple imputation

Syntax

`mi impute truncreg` *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	
noconstant	suppress constant term
ll(*varname* | #)	lower limit for left-truncation
ul(*varname* | #)	upper limit for right-truncation
offset(*varname*$_o$)	include *varname*$_o$ in model with coefficient constrained to 1
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 truncreg`; see [MI] **mi set**.
You must `mi register` *ivar* as imputed before using `mi impute truncreg`; 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**.
`aweights`, `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**.

`ll(`*varname* | #`)` and `ul(`*varname* | #`)` indicate the lower and upper limits for truncation, respectively. You may specify one or both. Observations with *ivar* $\leq$ `ll()` are left-truncated, observations with *ivar* $\geq$ `ul()` are right-truncated, and the remaining observations are not truncated.

`offset(`*varname*$_o$`)`; see [R] **Estimation options**.

`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 truncated 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] **truncreg**. These options are seldom 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:

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 truncated regression

The truncated regression imputation method can be used to fill in missing values of a continuous variable with a restricted range (for example, Raghunathan et al. [2001] and Schafer [1997, 203]). It is a parametric method that assumes an underlying truncated normal model for the imputed variable (given other predictors). This method is based on the asymptotic approximation of the posterior predictive distribution of the missing data.

Similar to estimation, it is important to distinguish between truncation and censoring when imputing continuous variables with a limited range. Truncation arises when the distribution of a variable of interest is restricted to a certain range—a truncated distribution. The probability that the variable takes on values outside that range is zero. Truncated data may arise naturally (for example, SAT section scores may not exceed 800) or may be the result of a particular study design (for example, only subjects with income below a certain threshold are of interest in the study). See [R] **truncreg** for more details.

Use `mi impute intreg` (see [MI] **mi impute intreg**) to impute continuous partially observed (censored) variables.

Using mi impute truncreg

In [MI] **mi impute pmm**, we used predictive mean matching to impute missing values of `bmi` (used as a predictor in the logistic analysis of heart attacks as described in [MI] **Intro substantive**), restricting imputed values to be within the observed range of `bmi`.

`mi impute pmm` imputes missing values of `bmi`, replacing them only with values already observed in the data. Suppose that, instead, we want to allow imputed `bmi` values to take on any value within a certain range. We can achieve this by using `mi impute truncreg`.

```
. use https://www.stata-press.com/data/r17/mheart0
(Fictional heart attack data; BMI missing)
. summarize bmi
```

Variable	Obs	Mean	Std. dev.	Min	Max
bmi	132	25.24136	4.027137	17.22643	38.24214

The observed range of `bmi` in our data is between roughly 17 and 39.

We impute `bmi` from a normal distribution truncated at (17, 39):

```
. mi set mlong
. mi register imputed bmi
(22 m=0 obs now marked as incomplete)
. mi impute truncreg bmi attack smokes age hsgrad female, add(20) ll(17) ul(39)
Univariate imputation                       Imputations =       20
Truncated regression                              added =       20
Imputed: m=1 through m=20                       updated =        0
Limit: lower =           17           Number truncated =        0
       upper =           39                       left =        0
                                                  right =        0
```

	Observations per *m*			
Variable	Complete	Incomplete	Imputed	Total
bmi	132	22	22	154

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

`mi impute truncreg` reports in the output header the truncation limits used (17 and 39 in our example). If the `ll()` and `ul()` options are not specified, the truncation limits are displayed as `-inf` and `+inf`, respectively, and the imputation model becomes equivalent to that using (unrestricted) normal linear regression.

`mi impute truncreg` also reports numbers of truncated observations. In our example, all values of `bmi` lie between 17 and 39, so there are no truncated observations. Truncated observations are not used during estimation; see [R] **truncreg**.

Rather than restricting `bmi` to the observed range during imputation, it may be reasonable to assume a wider range that is still consistent with the observed dataset. It may also be reasonable to use different ranges for males and females. For example, considering the observed ages, suppose that we assume a normal distribution for `bmi` truncated at (14, 55) for females and at (17, 50) for males.

To accommodate varying ranges, we first create variables containing gender-specific truncation limits:

```
. quietly mi xeq: generate lbmi = cond(female==1, 14, 17)
. quietly mi xeq: generate ubmi = cond(female==1, 55, 50)
```

The declared style of our `mi` data is `mlong`, so it is not necessary to use the `mi xeq` prefix for generating new variables. It is good practice, however, to use `mi`-specific commands so that your data manipulation is appropriate no matter what the `mi` style is; see [MI] **mi xeq** and [MI] **Styles** for details.

We now replace the existing imputations with new ones, which account for varying ranges of bmi among males and females:

```
. mi impute truncreg bmi attack smokes age hsgrad female, replace ll(lbmi)
> ul(ubmi)
Univariate imputation                       Imputations =       20
Truncated regression                              added =        0
Imputed: m=1 through m=20                       updated =       20
Limit: lower =         lbmi            Number truncated =        0
       upper =         ubmi                        left =        0
                                                   right =        0
```

	Observations per m			
Variable	Complete	Incomplete	Imputed	Total
bmi	132	22	22	154

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

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

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

Stored results

mi impute truncreg 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(N_trunc)	number of truncated observations
r(N_ltrunc)	number of left-truncated observations
r(N_rtrunc)	number of right-truncated observations
r(ll)	lower truncation limit (if ll(#) is specified)
r(ul)	upper truncation limit (if ul(#) is specified)
r(N_g)	number of imputed groups (1 if by() is not specified)

Macros

r(method)	name of imputation method (truncreg)
r(ivars)	names of imputation variables
r(llopt)	contents of ll(), if specified
r(ulopt)	contents of ul(), if specified
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 follows a truncated normal model with the density

$$f_{(a,b)}(x|\mathbf{z}_i) = \frac{\frac{1}{\sigma}\phi\left(\frac{x-\mu_i}{\sigma}\right)}{\Phi\left(\frac{b-\mu_i}{\sigma}\right) - \Phi\left(\frac{a-\mu_i}{\sigma}\right)},\ a < x < b \tag{1}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the standard normal density and cumulative distribution functions, respectively, $\mu_i = \mathbf{z}_i'\boldsymbol{\beta}$, $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iq})'$ records values of predictors of $\mathbf{x}$ for observation i, $\boldsymbol{\beta}$ is the $q \times 1$ vector of unknown regression coefficients, σ^2 is the unknown scalar variance, and a and b are the respective known lower and upper truncation limits; also see [R] **truncreg**. (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 truncreg` follows the steps below to fill in $\mathbf{x}_m$:

1. Fit a truncated regression (1) to the observed data $(\mathbf{x}_o, \mathbf{Z}_o)$ to obtain the maximum likelihood estimates, $\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\beta}}', \ln\widehat{\sigma})'$, and their asymptotic sampling variance, $\widehat{\mathbf{U}}$.
2. Simulate new parameters, $\boldsymbol{\theta}_\star$, from the large-sample normal approximation, $N(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{U}})$, 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 a truncated normal model (1) with parameters set to their simulated values from step 2: $\boldsymbol{\beta} = \boldsymbol{\beta}_\star$ and $\sigma = \sigma_\star$.
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 regression model is fit to the observed data in step 1 (see [R] **truncreg** for details). Also, in the case of `aweights`, $\sigma_\star$ is replaced with $\sigma_\star w_i^{-1/2}$ in step 3, where w_i is the analytic weight for observation i.

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.

Schafer, J. L. 1997. *Analysis of Incomplete Multivariate Data.* Boca Raton, FL: Chapman & Hall/CRC.

Also see

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

[MI] **mi impute intreg** — Impute using interval regression

[MI] **mi impute pmm** — Impute using predictive mean matching

[MI] **mi impute regress** — Impute using linear regression

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

[MI] **Intro** — Introduction to mi

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