Title

stata.com

mi impute intreg — Impute using interval regression

Description

`mi impute intreg` fills in missing values of a continuous partially observed (censored) variable using an interval regression imputation method. You can perform separate imputations on different subsets of the data by using the `by()` option. You can also account for analytic, frequency, importance, and sampling weights.

Menu

Statistics > Multiple imputation

Syntax

`mi impute intreg` *newivar* [*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 interval censoring
*ul(*varname*)	upper limit for interval censoring
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

*ll() and ul() are required.

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

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

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

Options

Main

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

ll(*varname*) and ul(*varname*) specify variables containing the lower and upper limits for interval censoring. You must specify both. Nonmissing observations with equal values in ll() and ul() are fully observed observations with missing values in both ll() and ul() are unobserved (missing), and the remaining observations are partially observed (censored). Partially observed cases are left-censored when ll() contains missing, right-censored when ul() contains missing, and interval-censored when ll() < ul(). Fully observed cases are also known as point data; also see *Description* in [R] **intreg**. In addition to *newivar*, mi impute intreg fills in unobserved (missing) values of variables supplied in ll() and ul(); censored values remain unchanged.

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

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 interval 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] **intreg**. 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:

Univariate imputation using interval regression
Using mi impute intreg
Example

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 interval regression

The interval regression imputation method can be used to fill in missing values of a continuous partially observed (censored) variable (Royston 2007). It is a parametric method that assumes an underlying normal model for the partially observed imputed variable (given other predictors). This method is based on the asymptotic approximation of the posterior predictive distribution of the missing data.

Partially observed data arise when instead of observing an actual value, we observe the range where that value can lie. Such data include interval-censored, left-censored, and right-censored data; see [R] **intreg** for a more detailed discussion of censored data.

Do not confuse censoring with truncation. Truncated data are observed and are known to be in a certain range. Censored data come from a mixture of a continuous distribution and point masses at censoring limits. Truncated data come from a continuous truncated distribution. See the technical note in *Remarks and examples* of [R] **truncreg** for details. Use `mi impute truncreg` (see [MI] **mi impute truncreg**) to impute truncated data.

The imputation of censored data has certain unique characteristics. First, censored data are recorded in two variables containing the lower and the upper interval-censoring limits. So technically, there are two imputation variables. Second, in addition to complete observations (point data) and incomplete observations ("truly" missing data), there are partially complete (censored) observations for which only the lower and upper limits are known, not the values themselves. We can treat partially observed cases as "missing" and impute them along with other completely unobserved data, provided we respect

their observed limits during imputation. As a result, we will end up with a single imputed variable where missing and partially observed cases are replaced with plausible values consistent with the observed censoring limits. See *Methods and formulas* for technical details.

In what follows, when referring to missing data (or missing observations) we will mean completely unobserved, truly missing data and when referring to incomplete data (or incomplete observations) we will mean both censored and truly missing data.

Using mi impute intreg

To accommodate the above characteristics, `mi impute intreg` requires modifications to the standard syntax of univariate imputation methods. First, `mi impute intreg` requires that variables containing interval-censoring limits be specified in the `ll()` and `ul()` options; see the description of `ll()` and `ul()` in *Options*. Second, `mi impute intreg` requires you to specify a new variable name *newivar* to store the resulting imputed values. `mi impute intreg` creates a new variable, *newivar*, and registers it as imputed.

The values of *newivar* are determined by `ll()` and `ul()`. Observations of *newivar* for which `ll()` and `ul()` are different or for which both contain soft missing are set to soft missing (`.`) and considered incomplete. Observations for which either `ll()` or `ul()` contains hard missing are set to the extended missing value `.a` and, as usual, are omitted from imputation. The remaining observations, corresponding to the observed point data, are complete.

After imputation, `mi impute intreg` stores imputed values in *newivar*. It also registers variables in `ll()` and `ul()` as passive (see `mi register` in [MI] **mi set**), if they are not already registered as passive, and replaces observations for which `ll()` and `ul()` both contain soft missing with the corresponding imputed values. That is, only missing data are replaced in these variables; censored data are not changed.

Later, you may decide to add more imputations or to revise your imputation model and replace existing imputations with new ones. In such cases, you do not need to provide a new variable name. You can reuse the name of the variable created previously by `mi impute intreg`. `mi impute intreg` will check that the variable is registered as imputed and that it is consistent in the observed data with the variables supplied in `ll()` and `ul()`. That is, the variable must have the same values as `ll()` and `ul()` in the observations where `ll()` and `ul()` are equal, and soft missing values in the remaining observations. If `ll()` or `ul()` contain hard missing values, the variable must contain hard missing values in the corresponding observations as well.

Example

We continue the example of imputing missing values of variable `bmi` from [MI] **mi impute pmm**. The primary analysis of interest is the logistic model investigating the effect of smoking adjusted for other predictors (including `bmi`) on heart attacks; see [MI] **intro substantive** for details.

The bmi variable is not censored in the original data. For the purpose of illustration, we use a version of the dataset in which the first three observations are censored:

```
. use http://www.stata-press.com/data/r15/mheartintreg
(Fictional heart attack data; BMI censored and missing)
. list lbmi ubmi in 1/10
```

	lbmi	ubmi
1.	.	22
2.	20	.
3.	30	31
4.	24.62917	24.62917
5.	22.52744	22.52744
6.	21.87975	21.87975
7.	17.77057	17.77057
8.	.	.
9.	23.47249	23.47249
10.	24.48916	24.48916

Rather than a single bmi variable, we have lbmi and ubmi variables containing lower and upper interval-censoring limits of BMI. The first observation is left-censored with an upper limit of 22, the second observation is right-censored with a lower limit of 20, and the third observation is interval-censored with the range [30, 31]. Observation 8, for which both lbmi and ubmi are missing, is missing.

Let's impute censored BMI values:

```
. mi set mlong
. mi impute intreg newbmi attack smokes age hsgrad female, add(20)
> ll(lbmi) ul(ubmi)
Univariate imputation                       Imputations =       20
Interval regression                               added =       20
Imputed: m=1 through m=20                       updated =        0
Limit: lower =         lbmi              Number missing =       22
       upper =         ubmi            Number censored =        3
                                                interval =        1
                                                    left =        1
                                                   right =        1
```

Variable	Observations per m			
	Complete	Incomplete	Imputed	Total
newbmi	129	25	25	154

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

Following mi impute intreg, we provided a new variable name, newbmi, to contain imputed values. Because newbmi did not exist we did not need to register it before using mi impute intreg. We also specified the lower and upper interval-censoring limits in the ll() and ul() options. These options are required with mi impute intreg.

mi impute intreg reported that 25 incomplete BMI values were imputed. Among these incomplete observations, there are 22 missing observations and 3 censored observations (one interval-censored, one left-censored, and one right-censored).

Let's describe our mi data:

```
. mi describe, detail
  Style:  mlong
          last mi update 21jan2017 12:53:02, 0 seconds ago
  Obs.:   complete          129
          incomplete         25  (M = 20 imputations)
          ------------------------
          total             154
  Vars.:  imputed:  1; newbmi(25; 20*0)
          passive:  2; lbmi(23; 20*1) ubmi(23; 20*1)
          regular:  0
          system:   3; _mi_m _mi_id _mi_miss
         (there are 5 unregistered variables)
```

We used the `detail` option to also see missing-value counts in the imputed data.

According to `mi describe`, the new variable `newbmi` is registered as imputed and contains 25 incomplete observations in the original data. It does not contain incomplete values in any of the 20 imputations. `lbmi` and `ubmi` are registered as passive. Each of `lbmi` and `ubmi` contains 23 incomplete values in the original data and one incomplete value in each imputation. The 22 missing values for `lbmi` and `ubmi` are imputed. The incomplete value for each of these variables that is not imputed corresponds to a censored observation (left-censored observation 1 for `lbmi` and right-censored observation 2 for `ubmi`). `mi impute intreg` replaces only missing observations of `lbmi` and `ubmi` with imputed data and leaves censored observations unchanged.

As described in *Methods and formulas*, missing observations are simulated from an unrestricted normal distribution. So, the 22 imputed values may contain any value on the whole real line. This may not be desirable because the BMI measure is positive and, in fact, has a limited range.

To restrict imputed values to a certain range, we may replace `lbmi` and `ubmi` with lower and upper limits in observations for which these variables are missing. For example, let's restrict imputed values to be between 17 and 39, consistent with the observed range of BMI.

```
. use http://www.stata-press.com/data/r15/mheartintreg, clear
(Fictional heart attack data; BMI censored and missing)
. replace lbmi = 17 if lbmi==.
(23 real changes made)
. replace ubmi = 39 if ubmi==.
(23 real changes made)
. list lbmi ubmi in 1/10
```

	lbmi	ubmi
1.	17	22
2.	20	39
3.	30	31
4.	24.62917	24.62917
5.	22.52744	22.52744
6.	21.87975	21.87975
7.	17.77057	17.77057
8.	17	39
9.	23.47249	23.47249
10.	24.48916	24.48916

We replace missing lower limits with 17 and missing upper limits with 39 and proceed with imputation:

```
. mi set mlong

. mi impute intreg newbmi attack smokes age hsgrad female, add(20)
> ll(lbmi) ul(ubmi)

Univariate imputation                       Imputations =       20
Interval regression                                added =       20
Imputed: m=1 through m=20                        updated =        0

Limit: lower =         lbmi              Number missing =        0
       upper =         ubmi            Number censored =       25
                                              interval =       25
                                                  left =        0
                                                 right =        0
```

Variable	Observations per m			
	Complete	Incomplete	Imputed	Total
newbmi	129	25	25	154

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

All the incomplete observations are now interval-censored on [17, 39].

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

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

In [MI] **mi impute truncreg**, we used `mi impute truncreg` to accommodate a restricted range of BMI during imputation. In the code above, we showed how to use `mi impute intreg` to ensure that imputed values are within a specified range. Which one should be used?

The answer to this question depends on our belief about the distribution of the imputation variable. If we believe that the underlying distribution of BMI is a normal distribution and we happened to only observe values within a certain range, then `mi impute intreg` should be used to impute BMI. We know, however, that BMI is positive and has an upper limit. As such, the assumption of a truncated distribution for BMI is more plausible, in which case `mi impute truncreg` should be used to impute its missing values.

Stored results

`mi impute intreg` 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(N_miss)`	number of missing observations
`r(N_cens)`	number of censored observations
`r(N_lcens)`	number of left-censored observations
`r(N_rcens)`	number of right-censored observations
`r(N_intcens)`	number of interval-censored observations
`r(k_ivars)`	number of imputed variables (always `1`)
`r(N_g)`	number of imputed groups (`1` if `by()` is not specified)

Macros

`r(method)`	name of imputation method (`intreg`)
`r(ivars)`	names of imputation variables
`r(llname)`	name of variable containing lower interval-censoring limits
`r(ulname)`	name of variable containing upper interval-censoring limits
`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 latent univariate variable $\mathbf{x}^u = (x_1^u, x_2^u, \ldots, x_n^u)'$ that follows a normal linear regression

$$x_i^u|\mathbf{z}_i \sim N(\mathbf{z}_i'\boldsymbol{\beta}, \sigma^2) \tag{1}$$

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

Instead of $\mathbf{x}^u$, we observe $(\mathbf{x}^{\rm ll}, \mathbf{x}^{\rm ul})$, where $x_j^{\rm ll} = x_j^{\rm ul} = x_j^u$ for point (observed) data $j \in \mathcal{C}$; $x_j^{\rm ll} = -\infty$ and $x_j^{\rm ul} < +\infty$ for left-censored data $j \in \mathcal{L}$; $x_j^{\rm ll} > -\infty$ and $x_j^{\rm ul} = +\infty$ for right-censored data $j \in \mathcal{R}$; $x_j^{\rm ll} = -\infty$ and $x_j^{\rm ul} = +\infty$ for missing data $j \in \mathcal{M}$. Observations from subset $\mathcal{C}$ are considered complete and the remaining observations are considered incomplete.

Let $\mathbf{x} = \mathbf{x}^u$ for observations in subset $\mathcal{C}$, and let $\mathbf{x}$ contain missing values in the remaining observations. We want to fill in missing values in $\mathbf{x}$. 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 intreg` follows the steps below to fill in $\mathbf{x}_m$:

1. Fit an interval regression to the interval-censored data $(\mathbf{x}^{\rm ll}, \mathbf{x}^{\rm ul})$ to obtain the maximum likelihood estimates of parameters in (1), $\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\beta}}', \ln\widehat{\sigma})'$, and their asymptotic sampling variance, $\widehat{\mathbf{U}}$. See [R] **intreg** for details.

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. Let $\mu_{\star i} = \mathbf{z}_i'\boldsymbol{\beta}_\star$. Obtain one set of imputed values, $\mathbf{x}_m^1$, by simulating from a truncated normal model with the density

$$f_{(x_i^{\rm ll},x_i^{\rm ul})}(x|\mathbf{z}_i) = \frac{1}{\sigma_\star}\phi\left(\frac{x-\mu_{\star i}}{\sigma_\star}\right) \times \left\{\Phi\left(\frac{x_i^{\rm ul}-\mu_{\star i}}{\sigma_\star}\right) - \Phi\left(\frac{x_i^{\rm ll}-\mu_{\star i}}{\sigma_\star}\right)\right\}^{1}, \quad x_i^{\rm ll} < x < x_i^{\rm ul}$$

for every incomplete observation $i \notin \mathcal{C}$. For missing observations $i \in \mathcal{M}$, when $x_i^{\rm ll} = -\infty$ and $x_i^{\rm ul} = +\infty$, the above density reduces to a normal density. Thus missing observations are simulated from the corresponding unrestricted normal distribution.

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] **intreg** 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.

Reference

Royston, P. 2007. Multiple imputation of missing values: Further update of ice, with an emphasis on interval censoring. *Stata Journal* 7: 445–464.

Also see

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

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

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

[MI] **mi impute truncreg** — Impute using truncated regression

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

[MI] **intro** — Introduction to mi

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