The Stata Journal

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata and Mata are registered trademarks of StataCorp LP.

The Stata Journal (2007)
7, Number 4, pp. 445–464

Multiple imputation of missing values: further update of ice, with an emphasis on interval censoring

Patrick Royston
Cancer and Statistical Methodology Groups
MRC Clinical Trials Unit
London UK

Abstract. Multiple imputation of missing data continues to be a topic of considerable interest and importance to applied researchers. In this article, the `ice` package for multiple imputation is further updated. Special attention in this article is paid to imputing interval-censored observations, and a suggestion to use imputation of right-censored survival data to elucidate covariate effects graphically.

Keywords: st0067_3, ice, uvis, micombine, ice_reformat, multiple imputation, interval censoring, visualization, censored survival data

1 Introduction

Royston (2004) introduced `mvis`, an implementation for Stata of MICE, a method of multiple multivariate imputation of missing values under missing-at-random (MAR) assumptions. In a second article, Royston (2005a) described `ice`, an upgrade incorporating various improvements and changes to the software based on personal experience, discussion with colleagues, and user requests. An update of `ice` was described by Royston (2005b), and this article presents a further update. The changes are less substantial than before but nevertheless, I feel, are important enough to warrant a paper. I will focus particularly on the new `interval()` option for imputing interval-censored observations. This option may be used with covariates recorded only in categories (such as stated income in surveys or a different application) to impute the *missing* part of left-, interval-, or right-censored time-to-event observations.

The current `ice` system consists of three ado-files: `ice`, `uvis`, and `micombine`. Previous components `mijoin` and `misplit` are out of date and have been removed. This is the final release of `micombine`, since a related article (Carlin, Galati, and Royston 2008) describes a new ado-file, `mim`, which replaces `micombine` and has more facilities.

Finally, another ado-file, `ice_reformat`, is included in the present release for backward compatibility of data files. It converts `.dta` files created by earlier releases of `ice` to the format required by `mim`.

2 Syntax

`ice` *mainvarlist* [*if*] [*in*] [*weight*] [`, boot`[`(`*varlist*`)`] `cc(`*ccvarlist*`)`
`cmd(`*cmdlist*`)` `cycles(`#`)` `dropmissing` `dryrun` `eq(`*eqlist*`)` `genmiss(`*string*`)`
`id(`*string*`)` `m(`#`)` `interval(`*intlist*`)` `match`[`(`*varlist*`)`] `noconstant` `nopp`
`noshoweq` `nowarning` `on(`*varlist*`)` `orderasis` `passive(`*passivelist*`)` `replace`
`saving(`*filename* [`, replace`]`)` `seed(`#`)` `substitute(`*sublist*`)`
`trace(`*filename*`)`]

`uvis` *regression_cmd* { *yvar* | *llvar ulvar* } *xvarlisti* [*if*] [*in*] [*weight*]`,`
`gen(`*newvarname*`)` [`boot` `match` `noconstant` `nopp` `replace` `seed(`#`)`]

where *regression_cmd* may be `intreg`, `logistic`, `logit`, `mlogit`, `ologit`, or `regress`. All weight types supported by *regression_cmd* are allowed. *llvar ulvar* are required with `uvis intreg`.

`micombine` *regression_cmd* [*yvar*] [*covarlist*] [*if*] [*in*] [*weight*] [`, br` `detail`
`eform(`*string*`)` `genxb(`*newvarname*`)` `impid(`*varname*`)` `lrr` `noconstant`
`obsid(`*varname*`)` `svy`[`(`*svy_options*`)`] *regression_cmd_options*]

where *regression_cmd* includes `clogit`, `cnreg`, `glm`, `logistic`, `logit`, `mlogit`, `nbreg`, `ologit`, `oprobit`, `poisson`, `probit`, `qreg`, `regress`, `rreg`, `stcox`, `streg`, or `xtgee`. Other *regression_cmd*s will work but not all have been tested by the author. All weight types supported by *regression_cmd* are allowed.

`ice_reformat` *filename*`, replace`

3 What is new?

The principal changes to `ice` (version 1.4.4), `uvis` (version 1.2.7), and `micombine` (version 1.1.6) compared with the November 2005 release (Royston 2005b) (versions 1.1.1, 1.1.0, and 1.1.0, respectively) are as follows:

1. `ice` now checks for perfect prediction of the outcome when logistic regression (`logit`, `logistic`, `ologit`, `mlogit`) is used to impute a binary, ordered or unordered categorical variable. If perfect prediction is found, `ice` and `uvis` work with a modified type of logistic regression command. The dataset is extended by several pseudo-observations in such a way that nonperfect prediction results and the estimated β regression coefficient and its SE are finite. This approach guarantees sensible imputations in such cases. Treatment of the perfect prediction bug can be suppressed by using the `nopp` option of `ice` or `uvis`.

2. An interval() option has been added to ice. This option is the key change and its functionality is the main topic of the present article.
3. The imputation and observation indicator variables have been changed from _j and _i to _mj and _mi.
4. The original data, including missing values, are output by ice to the file of imputations, indexed by _mj = 0.
5. ice's substitute() option has been improved by making it imply passive() for the relevant variables. This saves typing and reduces the chance of making a mistake in the specification.
6. dropmissing, orderasis, and nowarning options have been added to ice.
7. A nopp option has been added to ice and uvis.
8. The using *filename* syntax has been replaced with a saving(*filename*[, replace]) option. The old syntax still works but is undocumented.
9. The help file for ice/uvis has been modernized.
10. svy commands for Stata 8 and 9 are now supported by micombine.
11. uvis supports imputation of interval-censored variables with the uvis intreg syntax.
12. ice_reformat replaces *filename* with a new version of the data, with the following changes:
    a. _i and _j are renamed to _mi and _mj, respectively.
    b. The contents of characteristic char _dta[mi_id] are changed from _i to _mi.

4 Options

Only new or changed options are described.

Options for ice

dropmissing is a feature designed to save memory when using the file of imputed data created by ice. It omits from *filename* all observations that are not in the estimation sample, that is, for which either (i) they are filtered out by if or in, or a nonpositive weight, or (ii) the values of all variables in *mainvarlist* are missing. This option provides a clean analysis file of imputations, with no missing values. The observations not in the estimation sample are also omitted from the original data, stored as the imputation indexed by _mj==0 in *filename*.

`interval(`*intlist*`)` imputes interval-censored variables. An interval-censored value is one that is known to lie in an interval $[a, b]$, where a and b are finite and $a \leq b$; in $(-\infty, b]$; or in $[a, \infty)$. When either terminal is infinite, we have left or right censoring, respectively. *intlist* has the syntax *varname*`:` *llvar ulvar* $\big[$ `,` *varname*`:` *llvar ulvar* ...$\big]$, where each *varname* is an interval-censored variable, each *llvar* contains the lower bound (a) for *varname* and each *ulvar* contains the upper bound (b) for *varname* (or a missing value to represent $\pm\infty$). The supplied values of *varname* are irrelevant because they will be replaced anyway; it is only required that *varname* exist. Observations with *llvar* missing and *ulvar* present are left-censored for *varname*. Observations with *llvar* present and *ulvar* missing are right-censored for *varname*. Observations with *llvar* = *ulvar* are complete, and no imputation is done for them. Observations with both *llvar* and *ulvar* missing are imputed assuming an uncensored normal distribution.

`nopp` suppresses treatment of the perfect prediction bug.

`nowarning` suppresses warning messages.

`orderasis` enters the variables in *mainvarlist* into the MICE algorithm in the order given. The default is to order them according to the number of missing values; the variable with the least missingness gets imputed first and so on.

`saving(`*filename* $\big[$ `, replace` $\big]$`)` saves the imputations to *filename*. `replace` allows *filename* to be overwritten with new data. Unless `dryrun` has been specified, `saving()` is required.

4.1 Options for uvis

`nopp` suppresses treatment of the perfect prediction bug.

4.2 Options for micombine

`svy`$\big[$ `(`*svy_options*`)` $\big]$ (Stata 9) performs survey regression. The prefix `svy:` is placed before *regression_cmd*. If *svy_options* are supplied then `,` *svy_options* is placed between `svy` and the colon. The data must be `svyset` before this option is used and before `ice` is used to impute missing values. That the data have been `svyset` is inherited by the file of imputations created by `ice`.

`svy` (Stata 8) performs survey regression. The prefix `svy` is placed before *regression_cmd*, so that for example `micombine regress` ... `, svy` is interpreted as `svy regress` .... Options for survey regression are included as options to `micombine`. The data must be `svyset` before the `svy` option is used. This must be done before `ice` is used to impute missing values. That the data have been `svyset` is inherited by the file of imputations created by `ice`.

5 Interval censoring

5.1 Introduction

A value x is said to be *interval-censored* on $[a, b]$ if x is known to lie between a and b but its exact value is not known. An example is a sample survey in which respondents are asked to indicate an income range (e.g., \$0–\$5,000, \$5,001–\$10,000) but not their precise income. In clinical medicine it is not uncommon for continuous or ordinal values to be recorded only in categories. In node-positive breast cancer, for example, the most important prognostic variable, the number of positive lymph nodes (say, `nodes`), is sometimes converted to the lymph node stage (`nstage`), coded as 0 for node negative (`nodes` = 0), 1 for 1–3, 2 for 4–9, and 3 for 10+ nodes. A dataset compiled from different centers could even contain a mixture of `nodes` and `nstage` values, depending on local practice.

Interval-censored data includes some important special cases. For example, with right censoring (e.g., *time-to-event* data), a datum x may be completely observed, in which case, $a = b = x$, or known to be at least x_0, in which case, $a = x_0$ and $b = +\infty$ and x is right-censored. A datum left-censored at x_0 has $a = -\infty$ and $b = x_0$. In the `nodes` example, observations in `nstage` category 0 are exact, whereas those in categories 1 and 2 are interval-censored and those in category 3 are right-censored.

Sometimes, for example, for modeling or descriptive purposes, the continuous values underlying an interval-censored variable need to be imputed. For example, `nodes` is the most powerful predictor of outcome in primary breast cancer. If `nstage` is recorded for some patients and `nodes` for others, the most informative analysis of the dataset may require imputation of exact value of `nodes` for cases with only `nstage` known. One may also be faced with imputing missing values of `nodes`/`nstage`.

An interesting application of imputing interval-censored observations is with time-to-event (e.g., survival) data. Visualization of survival times and other explorations of the data may be more easily achieved with the censored observations replaced with imputed values. I will illustrate this scenario in some detail in section 6.

5.2 The model

In `ice`, imputation of interval-censored observations is based on the assumption that the underlying (*latent*) continuous variable is normally distributed. The Stata command `intreg` is used to estimate the mean and variance of this distribution, based on the interval-censored (doubly truncated) values and on covariates comprising the imputation model. It is assumed that the underlying continuous variable follows a truncated normal distribution in the observed categories. To help make the modeling more realistic, the software allows the imposition of an absolute lower and/or upper limit on the imputed values. This is implemented by truncation of the normal distribution at the specified value(s).

Figure 1 shows the principle of imputation sampling here. For example, an observation of x is known to lie in $[1, 3]$ and a continuous value is sampled from the shaded density. This density takes into account the mean and SD of the underlying normal distribution (bell-shaped curve). These parameters are estimated by `intreg` from the covariates comprising the imputation model. To ensure that the imputations are proper, the parameter values actually used are drawn from their estimated posterior predictive distribution, as is routinely done by `ice`.

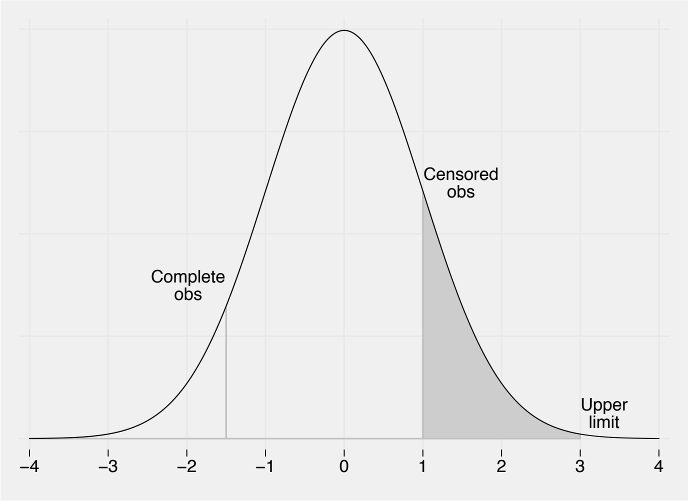

Figure 1: Interval censoring and the normal density function. The gray area indicates an observation that lies somewhere between 1 and 3. `ice` with the `interval()` option would sample from the density corresponding to the gray area.

5.3 In practice

To perform imputation with the `interval()` option, `ice` requires two variables: *ll* containing the lower boundary a for each observation of a variable x, and *ul*, containing the upper boundary, b. Each value of *ll* and *ul* may be missing or nonmissing, but *ll* must never exceed *ul*. Missing values of *ll* indicate left-censored observations; of *ul*, right-censored observations; and of both variables, truly missing observations.

The normality assumption must be plausible for the procedure to be successful in the sense of generating imputations with a realistic distribution. When the variable in question is intrinsically positive and positively skewed, a log transformation is often advantageous since the imputed values are guaranteed to be positive after back-transformation (exponentiation). If a subset of exactly observed values is available, an approximate transformation to normality can often be found by power transformation followed by a normal plot of the transformed variable (`qnorm` command). One is looking for approximate linearity of the normal plot. If the variable has zeros, a common practice is to add 1 before seeking such a transformation.

Variables that are integer-valued (e.g., `nodes`) and interval-censored (e.g., `nstage`) present a further challenge. Clearly the distribution of the underlying latent variable is not really continuous, but such an assumption is a convenient fiction. The case can be handled by judicious rounding. Consider `nstage`. Recalling that the categories 1–3 of `nstage` represent `nodes` values of 1–3, 4–9, and 10+, one might assign the values 1, 4, and 10 to *ll* and 3, 9, and "." (missing) to *ul*. However, with this scheme the imputed continuous values will have gaps in the intervals $(3, 4)$ and $(9, 10)$. A better scheme is to pretend that an observation of k nodes is really an underlying continuous value in the range $(k - 0.5, k + 0.5)$ and specify *ll* as 0.5, 3.5, and 9.5, and *ul* as 3.5, 9.5, and missing. The final step in such a scheme is to round the continuous imputed values to the nearest integer. By making the lower limit of the lowest group 0.5, we are guaranteed that imputed values will not be less than 1 after rounding.

If the variable requires a preliminary transformation to achieve approximate normality, the extra steps of pretransforming *ll* and *ul* and posttransforming them back to the original scale after imputation must be performed. In the `nodes` example, rounding to integers would be the final step.

5.4 Example

Preliminaries

I will illustrate the `nodes` example with real data. Consider the variable `x5` (`nodes`, number of positive lymph nodes) in the breast cancer dataset `brcaex.dta` analyzed by Royston (2004). The distribution of `x5` takes the integers 1, 2, ..., and has coefficient of skewness $\sqrt{b_1} = 2.9$, which is large. More than 25% of the values are 1.

Figure 2 shows two normal plots of `x5`.

(*Continued on next page*)

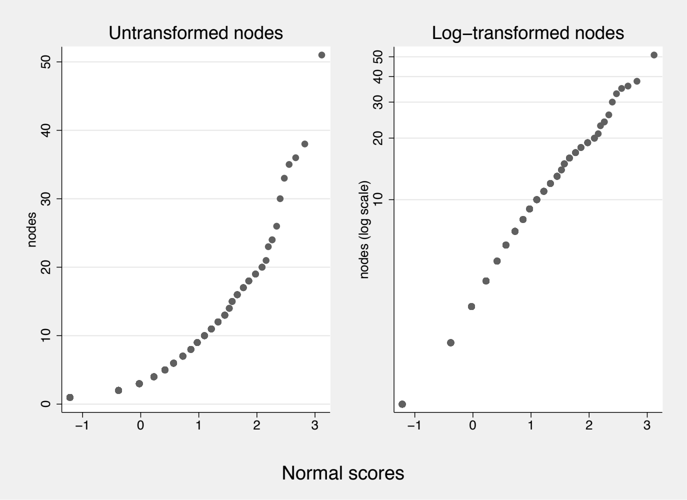

Figure 2: Normal Q–Q plots of untransformed (left panel) and log-transformed (right panel) `nodes`

However, these are not the usual normal plots. Instead, I have created the normal scores variable, `z`, corresponding to `x5` by using the ado-file `nscore` provided with this article. The syntax used is simply `nscore z = x5`. The difference between `nscore` and the factory-supplied program `qnorm` is that `nscore` averages normal scores corresponding to ties in the source variable. This greatly facilitates a visual assessment of linearity, because each horizontal sequence of markers representing tied values is removed from the normal plot (i.e., scatterplot of `x5` against `z`) and replaced with one point. If desired the multiplicity of these points can be indicated by weighting the plot by the number of values at each point.

Clearly untransformed `x5` is far from normal, but log `x5` is reasonably normal (it has $\sqrt{b_1} = 0.3$). Further refinement could be achieved, for example, by adding a constant to `x5` before transformation and tuning the constant to make the normal plot as linear as possible, but this is not really necessary.

Suppose that we did not have the raw values of `x5` but have only the `nstage` categorization. Assessing normality is obviously more difficult now. However, provided we have at least a reasonable idea of the mean of `x5` in each category, perhaps from other datasets, we can get some idea of whether a log transformation makes the data more normal. We replace each category value (1, 2, or 3) with our estimate of the category mean. Here we know the category means: 1.7, 5.9, and 15.2. In reality we might estimate them as the category midpoints (2 and 6.5 for categories 1 and 2) and make an informed guess, say, 14 for category 3. A simple measure of normality (equivalent, in fact, to the Shapiro–Francia statistic) is the correlation coefficient between the mean (or log mean) category values and the category normal equivalent deviates (NEDs). As before, the NEDs are computed by nscore and averaged over tied values, here giving just three distinct values: −0.72, 0.54, and 1.55. The resulting correlations are 0.9768 for the untransformed and 0.9985 for the log-transformed means. The log transformation is therefore favored.

Imputation

I will now illustrate how to use ice to impute plausible values of x5 from nstage as discussed above. A preliminary multivariable analysis showed that nstage is associated with x3 (tumor size) and x4a/x4b (dummy variables for tumor grade 1/2/3), so these two variables are included in the imputation model. I added one minor modification: instead of allowing imputed numbers of nodes to be unlimited, I restricted them to a maximum of 55 (the maximum in the original data being 51). Limiting the range of imputed values is often sensible. Stata code to create $m = 10$ imputations is as follows:

```
. gen llnodes = log(0.5*(nstage==1) + 3.5*(nstage==2) + 9.5*(nstage==3))
. gen lunodes = log(3.5*(nstage==1) + 9.5*(nstage==2) +  55*(nstage==3))
. gen lnodes = .
(686 missing values generated)
. ice lnodes llnodes lunodes x3 x4a x4b, saving(nodesimp)
 > m(10) interval(lnodes: llnodes lunodes)
   #missing |
     values |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |        686      100.00      100.00
------------+-----------------------------------
      Total |        686      100.00
   Variable | Command | Prediction equation
------------+---------+-------------------------------------------------------
     lnodes | intreg  | x3 x4a x4b
    llnodes |         | [Lower bound for lnodes]
    lunodes |         | [Upper bound for lnodes]
         x3 |         | [No missing data in estimation sample]
        x4a |         | [No missing data in estimation sample]
        x4b |         | [No missing data in estimation sample]
------------------------------------------------------------------------------

Imputing
[Only 1 variable to be imputed, therefore no cycling needed.]
1..2..3..4..5..6..7..8..9..10..file nodesimp.dta saved)
. use nodesimp, clear
(German breast cancer data)
. gen int nodes = round(exp(lnodes), 1)
(686 missing values generated)
```

ice reports 686 occurrences of 1 missing value because we initially assigned all values of lnodes to missing.

Figure 3 compares the imputed nodes values with the known values of x5 in the first imputation (_mj==1).

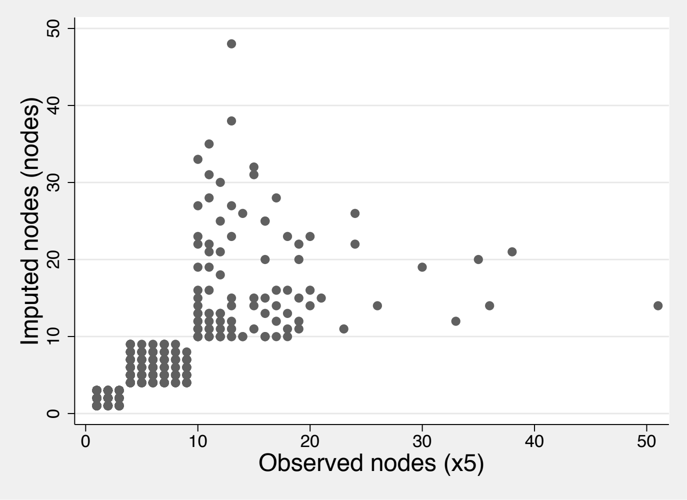

Figure 3: Imputation of interval-censored values of x5; comparison of original with imputed values in imputation 1

Because the imputation model does not explain much of the variation, there is considerable uncertainty in the imputations and hence scatter. The Spearman rank correlation between nodes and x5 is between 0.82 and 0.84 across the imputations. Nevertheless, the imputation seems to have done a good job. Using Rubin's rules for combining estimates across imputations, the mean (SE) of nodes is 5.18(0.23) and of x5 (the gold standard) is 5.01(0.21). The bias in the mean is negligible. The mean (SE) of the regression coefficient in a univariate Cox model on log(nodes) is 0.556(0.068), compared with 0.543(0.063) for log(x5).

6 Imputing right-censored survival data

6.1 Why bother?

As Royston, Parmar, and Altman (2008) discuss and illustrate, with censored survival data it is difficult to visualize and therefore to understand the distribution of the time-to-event outcome variable in relation to covariates. The Cox model, for example, is conceived in terms of hazard ratios, but these are rather indirectly related to differences in survival times. In a clinical trial, the experimental treatment may exhibit a substantial reduction in risk of an event compared with the control arm, as evidenced by a hazard ratio of, say, 0.7. The corresponding Kaplan−Meier survival curves for the two arms may look impressively separated in a plot. However, the actual distributions of time to event may overlap considerably. A scatterplot of these times will go a long way to correcting an overoptimistic impression of the effectiveness of the treatment. Judicious imputation of the right-censored times to event can provide the analyst with a tool that greatly assists inspection of such distributions and hence allows a more realistic assessment of the effect of a treatment at the level of individual survival times. Similar comments apply to the effects of prognostic variables.

6.2 Quantile–quantile plot of censored survival times

In primary breast cancer, time-to-disease recurrence is approximately lognormally distributed (Royston 2001). The marginal distribution of time to event may be assessed in a modified version of a normal quantile–quantile plot. Let $t_{(1)} \leq \cdots \leq t_{(n)}$ be the ordered survival or censoring times of n individuals with estimated survival probabilities (obtained by the Kaplan–Meier or some other suitable method) $S_1 \geq \cdots \geq S_n$. Write $z_j = -\Phi^{-1}(S_j)$ [in Stata, use the `invnormal()` function for $\Phi^{-1}(\cdot)$]. A scatterplot of the $t_{(j)}$ against the z_j is a normal quantile–quantile plot for censored data. Figure 4 shows such a plot for the breast cancer example dataset.

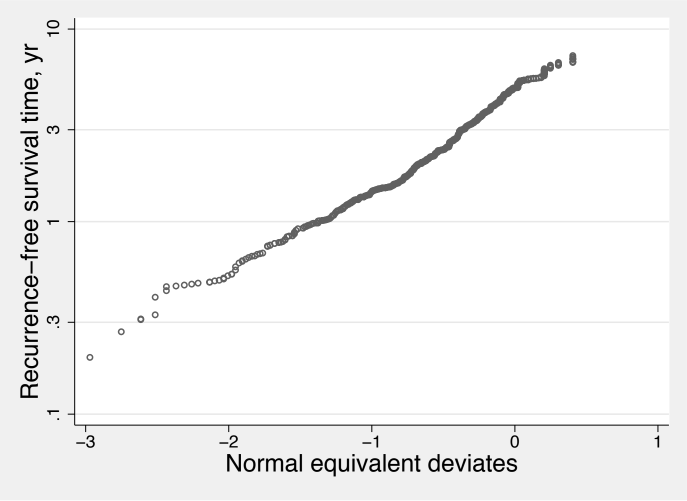

Figure 4: Normal quantile–quantile plot of censored recurrence-free survival time (RFS) data. Vertical axis is a log scale. Linearity suggests that the time-to-event is approximately lognormally distributed.

The times have been plotted on a log scale. The relationship is roughly linear, supporting a lognormal distribution as a reasonable approximation.

6.3 Doing the imputations

The right-censored times can be imputed by using `ice` with the `interval()` option. First, an imputation model is needed to allow for the possible effects of covariates. Because we are working with a lognormal distribution, a sensible approach is to build a multivariable model by using some type of censored-normal regression of the log times on prognostic factors in the dataset. First, let us consider what may be a reasonable upper limit for the imputed survival times. The lognormal distribution is longtailed. Unless we are careful, we may find ourselves creating implausible imputed times (e.g., a recurrence-free survival time of 300 years). We get around this problem by specifying the upper limit of time to be something realistic, for example, 90 minus the age of the patient (`x1`) at entry to the study. (All patients were well under 90 years of age at entry.) We can then use Stata's `mfp` command with `intreg` to find a predictive model based on fractional polynomial transformation of the influential continuous predictors, where needed:

```
. stset rectime censrec, scale(365.25) // time in years
. gen lnt = ln(_t)
. gen ll = lnt
. gen ul = cond(_d==0, ln(90-x1), lnt)
. mfp intreg ll ul x1 x2 x3 x4a x4b x5 x6 x7 hormon, select(.05) df(x5:2)
  (output omitted)
```

The selected model has the following variables (with power(s) in parentheses, when transformed—power 0 meaning log): `x1` ($-1, -1$), `x4a`, `x5` (0), `x6` (0), and `hormon`. The residual SD (parameter `sigma`) is reported as 0.842. The variance explained by the model may be estimated as $R^2 = 1 - \mathrm{var}(y|\mathbf{x})/\mathrm{var}(y)$ and here is $1 - 0.842^2/0.976^2$ or about 26%. The value of $\mathrm{var}(y) = 0.976^2$ was found by running `intreg` with no covariates (i.e., `intreg ll ul`). The reported value of `sigma` is 0.976.

We now use this imputation model with `ice` to create 10 imputed datasets. The variables `ll` and `ul` are needed again:

```
. gen lnt = ln(_t)
. gen ll = lnt
. gen ul = cond(_d==0, ln(90-x1), lnt)
. fracgen x1 -1 -1
-> gen double x1_1 = X^-1
-> gen double x1_2 = X^-1*ln(X)
   (where: X = x1/10)
. fracgen x5 0
-> gen double x5_1 = ln(X)
   (where: X = x5/10)
. fracgen x6 0
-> gen double x6_1 = ln(X)
   (where: X = (x6+1)/1000)
```

```
. ice lnt ll ul x1_1 x1_2 x4a x5_1 x6_1 hormon, saving(brcaexi, replace) m(10)
> interval(lnt:ll ul)

   #missing |
     values |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        686      100.00      100.00
------------+-----------------------------------
      Total |        686      100.00

   Variable | Command | Prediction equation
------------+---------+-------------------------------------------------------
        lnt | intreg  | x1_1 x1_2 x4a x5_1 x6_1 hormon
         ll |         | [Lower bound for lnt]
         ul |         | [Upper bound for lnt]
       x1_1 |         | [No missing data in estimation sample]
       x1_2 |         | [No missing data in estimation sample]
        x4a |         | [No missing data in estimation sample]
       x5_1 |         | [No missing data in estimation sample]
       x6_1 |         | [No missing data in estimation sample]
     hormon |         | [No missing data in estimation sample]
------------------------------------------------------------------------------

Imputing
[Only 1 variable to be imputed, therefore no cycling needed.]
1..2..3..4..5..6..7..8..9..10..file brcaexi.dta saved
```

6.4 Plots using the imputed data

Let us now consider visualizing the effect of hormonal treatment (hormon) on recurrence-free survival time. Figure 5 shows a Kaplan–Meier plot of the original, censored time variable according to hormon treatment status.

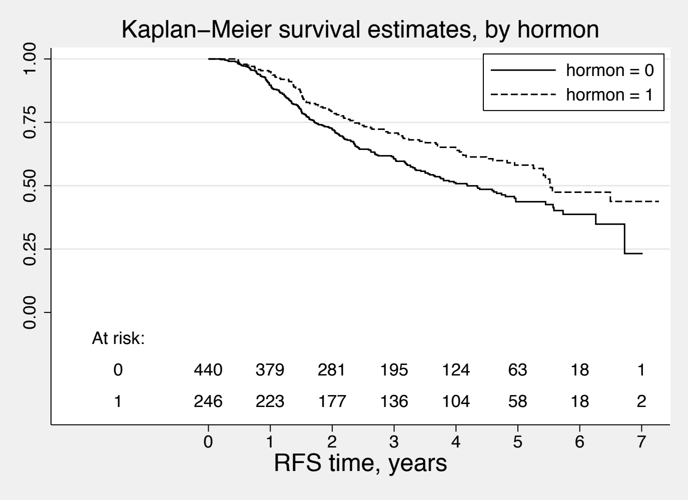

Figure 5: Kaplan–Meier plot of recurrence-free survival time according to hormonal treatment status (hormon)

There is visible white space between the curves, suggesting a large difference in survival. The parameter estimate for `hormon` in the original `intreg` model (adjusted for other predictors) is 0.27 (SE 0.08), suggesting that the treatment increases log RFS time on average by 0.27 or RFS time by about 31%.

Figure 6 is a dot plot of the observed and imputed RFS time in the first imputation (results for the other 9 imputations are similar).

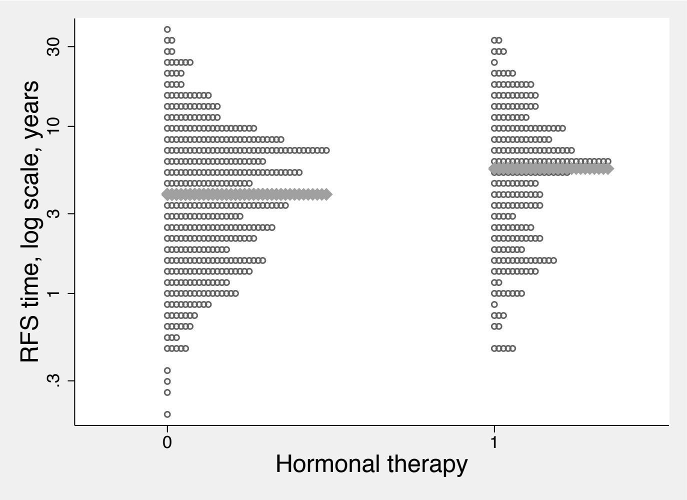

Figure 6: Comparison of time to RFS event for the patients untreated or treated with hormonal therapy (`hormon`) for the first imputation of the RFS time. Horizontal lines show the medians. The vertical scale of the dot plot is logarithmic.

The large degree of overlap between the two survival time distributions is now obvious. The therapy certainly has some effect but is not a miracle cure.

Figure 7 shows a smoothed scatterplot of the relationship between log RFS time and the strongest predictor (number of positive lymph nodes, `x5`) in the first imputation.

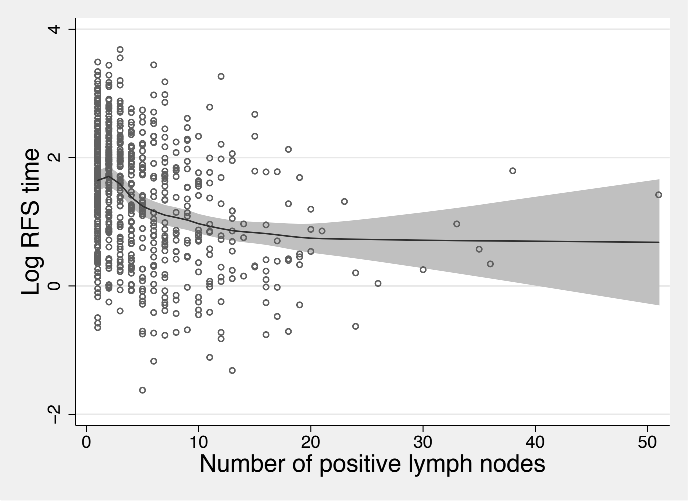

Figure 7: Relation between log RFS time and number of positive lymph nodes (x5) in the first imputation, with running-line smooth and 95% pointwise confidence interval

The smoothing was done by using a running-line smoother (Sasieni, Royston, and Cox 2005). A clear nonlinear relationship is present, but there is considerable random variation around the regression line. The Spearman correlation between time and x5 is −0.31.

6.5 To model or not to model?

Having obtained m complete imputed datasets and having seen the advantages of really getting to grips with the times to event at the individual patient level, it is tempting to try to build new models with the imputed data. First, the parameters of the imputation model are faithfully reproduced (apart from minor random variation) in the multiply imputed dataset. Because intreg assumes a truncated normal distribution on the log survival times, it is appropriate to use regress on log t followed by application of Rubin's rules (Rubin 1987) to estimate the parameters of the original imputation model in the imputed data. The original (intreg) and reestimated (regress) parameters are shown in table 1.

(*Continued on next page*)

Table 1: Comparison of regression coefficients and their standard errors for the `intreg` model on the original data and the `regress` model on the imputed data

Predictor	Original (`intreg`)			Imputed (`regress`)		
	$\widehat{\beta}$	SE	$\widehat{\beta}$/SE	$\widehat{\beta}$	SE	$\widehat{\beta}$/SE
`x1`$^{-2}$	−9.65	2.38	−4.05	−9.51	2.50	−3.81
`x1`$^{-0.5}$	17.53	4.62	3.79	17.15	4.70	3.65
`x4a`	−0.294	0.127	−2.31	−0.284	0.124	−2.28
ln `x5`	−0.328	0.039	−8.41	−0.328	0.040	−8.24
ln (`x6`+1)	0.126	0.020	6.32	0.125	0.021	6.02
`hormon`	0.270	0.079	3.41	0.260	0.077	3.36
`_cons`	−1.99	1.04	−1.91	−1.90	1.06	−1.80

Apart from a small amount of random variation, the parameter estimates from the two models are identical; the SEs are usually slightly larger and the $\widehat{\beta}$/SE values slighter smaller in the imputed data. This is as expected, because the imputation involves the injection of random variation, and with only $m = 10$ imputations, a little information is inevitably lost. As m is increased, the similarity of the $\widehat{\beta}$s and of the SEs increases (data not shown).

The imputed dataset faithfully reproduces the characteristics assumed in the original model on which the imputations are based. We assumed a truncated lognormal distribution for the log survival times with certain parameters and functional forms for the effects of covariates, which is what we got.

Going beyond the imputation model may cause problems, however. For example, it is known that there is an interaction between hormonal treatment (`hormon`) and estrogen receptor status (`x7`). Royston and Sauerbrei (2004) showed that the interaction can be adequately modeled by the product term `hormon`×(`x7`+1)$^{-0.5}$. Let us call this interaction variable `x7h`. Suppose that we extended the original `intreg` model by including the terms `hormon` and (`x7`+1)$^{-0.5}$ (i.e., the main effects for the interaction) and `x7h`, estimated the parameters, and then reestimated them by using `micombine regress` or `mim: regress` in the imputed dataset. Table 2 shows the resulting parameter estimates.

Table 2: Comparison of regression coefficients and their standard errors for the `intreg` model on the original data and the `regress` model on the imputed data. The interaction between `x7` and `hormon` is examined.

Predictor	Original (`intreg`)			Imputed (`regress`)		
	$\widehat{\beta}$	SE	$\widehat{\beta}$/SE	$\widehat{\beta}$	SE	$\widehat{\beta}$/SE
`hormon`	0.433	0.108	3.99	0.370	0.113	3.29
$(\texttt{x7}+1)^{-0.5}$	0.113	0.174	0.65	0.089	0.182	0.49
`x7h`	−0.554	0.252	−2.20	−0.386	0.257	−1.50

In the original `intreg` model, `x7h` is significant at the $P = 0.02$ level, whereas in the `regress` model in the imputed dataset, we have $P = 0.13$; the corresponding $\widehat{\beta}$ is reduced in magnitude from −0.55 to −0.39. Imputing using a wrong (or rather, incomplete) model has introduced a nontrivial amount of bias into the estimated interaction between `hormon` and `x7`.

Of course, such a finding is neither surprising nor specific to this situation. An inadequate imputation model can always induce bias of this sort; hence, the generally accepted advice is to use a large imputation model rather than a parsimonious one and to include interactions when necessary. We went against such advice here by building the imputation model with selection of variables and functions at the 5% significance level and not considering interactions at all.

Nevertheless, there is certainly a question as to whether one should include interactions or other higher-order terms in the imputation model. Generally, the issue is how to strike a satisfactory balance between a sufficiently comprehensive imputation model and the possibility of instability due to a grossly overfitted model. In the current example, we already knew from earlier work that an interaction existed, but usually such prior information will not be available. Developing a satisfactory imputation model is still an open issue in the practical analysis of multiple imputed datasets.

With right-censored survival times, a pragmatic approach may be to use imputation simply as a tool to explore the implications of a model fitted to the original data in more detail, as we have done here with the `intreg` approach. For example, the availability of scatterplot smoothers for the imputed data makes it easier to get a feel for the relationships within the data and to look for lack of fit. Nevertheless, to make this process safer and more informative, it is probably sensible to start with a rather larger imputation model. Here we could have included all the available predictors in the `intreg` model and perhaps allowed `mfp` to detect and model nonlinearity at a more relaxed significance level, such as 0.2. We could have also included in the model the interaction between $(\texttt{x7}+1)^{-0.5}$ and `hormon`.

6.6 Incompatibility between imputation and substantive models

Suppose that, having obtained m imputations as described above, we had contemplated doing not ordinary regression but Cox regression on the imputed dataset. Let us compare the regression coefficients of a Cox model estimated on the original and imputed datasets. The imputation model assumes one type of error structure (linear regression on log time) whereas the Cox model assumes another (a proportional hazards model). What effect does this incompatibility have on the $\widehat{\beta}$s?

Table 3 compares the $\widehat{\beta}$s and shows the percentage bias between the two ways of fitting the Cox model.

Table 3: Parameter estimates for a Cox model on the original data and imputed data assuming an incompatible imputation model

Predictor	$\widehat{\beta}$ in Cox model for Original data	 Imputed data	% bias
`x1`$^{-2}$	16.6	15.1	−9
`x1`$^{-0.5}$	−30.1	−29.0	−3
`x4a`	0.497	0.231	−54
ln `x5`	0.508	0.366	−28
ln (`x6`+1)	−0.179	−0.130	−27
`hormon`	−0.390	−0.273	−30

The results show that the incompatibility between the imputation and substantive models induces major bias in most of the estimated βs. The bias is always toward the null (i.e., brings the $\widehat{\beta}$s closer to zero than they should be).

Clearly, there are pitfalls that the user should beware of when contemplating imputing a censored outcome variable. These will also apply (but to a lesser extent, because extrapolation is less likely to be involved) to imputing missing values of a noncensored outcome variable.

7 Final comment

Here I have focused on multiple imputation of interval- or right-censored observations using `ice` and illustrated how judicious use of the `interval()` option may be helpful. I have also pointed out serious pitfalls when the method is used without care to complete a right-censored time-to-event variable. I believe that the user should be wary of literature claims of robustness to misspecification when such a type of imputation is used. For example, Hsu et al. (2007) use proportional hazards models to derive risk scores that help impute interval-censored outcome variables in a nonparametric way. The authors state “In addition to its robustness in this application, the general approach of multiple imputation methods has features that make it attractive. One such feature is that after imputation the data analyst is now free to choose and can easily perform any analysis appropriate for the goals of their study. Conditions for the appropriateness of this philosophy are discussed in Reference [23]". This advice appears to me dangerous—unless the reader carefully consults (and is sufficiently equipped to understand the implications of) Hsu et al. (2007)'s Reference 23 (Meng 1994). I would not, for example, advocate applying linear regression methods to such a multiply imputed dataset, because as far as I understand it, the imputation method implicitly assumes proportional-hazards effects of covariates. The result would be seriously biased regression estimates, as in table 3.

8 Acknowledgment

Ian White suggested and programmed the method used by `ice` and `uvis` to avoid the problem of perfect prediction with the `logit`, `ologit`, and `mlogit` commands.

9 References

Carlin, J. B., J. C. Galati, and P. Royston. 2008. A new framework for managing and analyzing multiply imputed data in Stata. *Stata Journal.* Forthcoming.

Hsu, C., J. M. G. Taylor, S. Murray, and D. Commenges. 2007. Multiple imputation for interval censored data with auxiliary variables. *Statistics in Medicine* 26: 769–781.

Meng, X. L. 1994. Multiple-imputation inferences with uncongenial sources of input (with discussion). *Statistical Science* 9: 538–573.

Royston, P. 2001. The lognormal distribution as a model for survival time in cancer, with an emphasis on prognostic factors. *Statistica Neerlandica* 55: 89–104.

———. 2004. Multiple imputation of missing values. *Stata Journal* 4: 227–241.

———. 2005a. Multiple imputation of missing values: update. *Stata Journal* 5: 188–201.

———. 2005b. Multiple imputation of missing values: update of ice. *Stata Journal* 5: 527–536.

Royston, P., M. K. B. Parmar, and D. G. Altman. 2008. Visualizing length of survival in time-to-event studies: a complement to Kaplan–Meier plots. *Journal of the National Cancer Institute* 100: 92–97.

Royston, P., and W. Sauerbrei. 2004. A new approach to modelling interactions between treatment and continuous covariates in clinical trials by using fractional polynomials. *Statistics in Medicine* 23: 2509–2525.

Rubin, D. B. 1987. *Multiple imputation for non-response in surveys.* New York: Wiley.

Sasieni, P., P. Royston, and N. J. Cox. 2005. Symmetric nearest neighbor linear smoothers. *Stata Journal* 5: 285.

About the author

Patrick Royston is a medical statistician with 30 years of experience, with a strong interest in biostatistical methodology and in statistical computing and algorithms. He works in clinical trials and related research issues in kidney cancer and other cancers. Currently, he is focusing on problems of model building and validation with survival data, including prognostic factors studies; on complex sample size problems in clinical trials with a survival-time endpoint; on writing a book on multivariable regression modeling; and on new trial designs.