Journal of Statistical Software

December 2011, Volume 45, Issue 4. *http://www.jstatsoft.org/*

Multiple Imputation by Chained Equations (MICE): Implementation in Stata

Patrick Royston
Medical Research Council

Ian R. White
Medical Research Council

Abstract

Missing data are a common occurrence in real datasets. For epidemiological and prognostic factors studies in medicine, multiple imputation is becoming the standard route to estimating models with missing covariate data under a missing-at-random assumption. We describe **ice**, an implementation in Stata of the MICE approach to multiple imputation. Real data from an observational study in ovarian cancer are used to illustrate the most important of the many options available with **ice**. We remark briefly on the new database architecture and procedures for multiple imputation introduced in releases 11 and 12 of Stata.

Keywords: missing data, multiple imputation, chained equations, continuous variables, categorical variables.

1. Introduction

In large datasets, missing values commonly occur in several variables. Multiple imputation by chained equations (MICE), nicely motivated and described in the context of a medical application by van Buuren *et al.* (1999), is a practical approach to creating imputed datasets based on a set of imputation models, one model for each variable with missing values. MICE is an increasingly popular method of doing multiple imputation (MI, Sterne *et al.* 2009).

Useful literature on MI includes a primer (Schafer 1999), a critical review (Kenward and Carpenter 2007), and reviews of applications (Barnard and Meng 1999) and software (Horton and Kleinman 2007). White *et al.* (2011)'s recent tutorial article offers practical advice on the use of MICE. It includes several examples based on real and simulated datasets.

The aim of the present paper is to describe, with a practical focus, an implementation of MICE in Stata (StataCorp. 2009) called **ice** (Royston 2004, 2005a,b, 2007, 2009). Rather than formally describe the syntax and features of **ice**, we proceed by example with a real dataset in ovarian cancer. The structure of the paper is as follows. We first outline the MICE algorithm. We then review implementation of MI software in Stata, including some comments on Stata version 11, which had a major new MI component. Proceeding to a little more detail, we discuss imputation models available in **ice** for different types of variables with missing data. We next describe the example dataset, and go on to exemplify a simple imputation situation with just one incomplete variable. The example allows us to exhibit the basic operations of **ice**. We follow that up with three alternative ways of doing imputation of a continuous variable in a univariate setting. We go on to consider a more complex multivariate imputation, focusing on how **ice** handles categorical variables. Finally, we briefly describe other useful features of **ice** not covered in the examples, and conclude with a short discussion.

2. Multiple imputation by chained equations (MICE)

Here, we outline the MICE algorithm for a set of variables, $x_1, \ldots, x_k$, some or all of which have missing values. Initially, all missing values are filled in at random. The first variable with at least one missing value, x_1 say, is then regressed on the other variables, $x_2, \ldots, x_k$. The estimation is restricted to individuals with observed x_1. Missing values in x_1 are replaced by simulated draws from the posterior predictive distribution of x_1, an important step known as *proper imputation*. The next variable with missing values, say x_2, is regressed on all the other variables, $x_1, x_3, \ldots, x_k$. Estimation is restricted to individuals with observed x_2 and uses the imputed values of x_1. Again, missing values in x_2 are replaced by draws from the posterior predictive distribution of x_2. The process is repeated for all other variables with missing values in turn: one such round is called a *cycle*. To stabilize the results, the procedure (similar to a Gibbs sampler) is usually repeated for about ten cycles to produce a single imputed dataset. van Buuren *et al.* (1999) suggest 20 cycles but say that 10 or even 5 may be adequate. We have performed some simple experiments on the convergence of the sampling distribution of imputed variables (data not shown). We find that only if variables with missing values to be imputed are highly correlated (say, > 0.6) are more than 10 cycles needed for convergence. In most applications we encounter, such high correlations are rarely seen.

The entire procedure is repeated independently M times, yielding M imputed datasets. Standard texts on MI suggest that small numbers of imputed datasets ($M = 3$ to 5) are adequate. Recent opinion has shifted towards larger values of M. For example, White *et al.* (2011) suggest a rule of thumb that M should be at least equal to the percentage of incomplete cases in the dataset. If 80% of cases had complete data on all relevant variables, the rule would indicate $M = 20$. As opposed to what is appropriate in data analysis, to control Monte Carlo error in studies comparing methods, even larger values of M, perhaps in the range 100 to 1000, are needed.

Because each variable is imputed using its own imputation model, MICE can handle different variable types (for example, continuous, binary, unordered categorical, ordered categorical). Suitable choices of imputation models are discussed in a Stata context in Section 4.

3. An overview of MICE and MI estimation in Stata

3.1. MICE and the analysis model

Royston (2004) introduced **mvis**, the first implementation of MICE for Stata. The name of the main command was changed to **ice** (imputation by chained equations) in Royston (2005a). Three updates of **ice** have followed (Royston 2005b, 2007, 2009).

The current **ice** system comprises three ado-files: `ice`, `ice_` and `uvis`. An 'ado-file' is Stata-speak for a Stata add-on program. Such programs are placed where Stata can 'see' them, and thereby they become seamlessly integrated into the Stata environment. The `ice` command performs multiple, multivariate imputation. `ice` calls `ice_` which in turn repeatedly calls `uvis` to do proper imputation of a single incomplete variable on its own or on one or more complete variables.

Although `uvis` is a stand-alone program and can be used as such for certain tasks, in the present article we concentrate on `ice`. A user who learns how to use `ice` effectively need not care at all about the details of `uvis`.

MI is incomplete without considering also the *analysis model* (or models) that one plans to fit to the imputed data. The whole purpose of MI is to enable, under the missing-at-random assumption (Little and Rubin 2002), more efficient and less biased estimation of model parameters than by using complete cases. We do not rehearse the arguments for MI here. As is well known, the correct approach is to apply *Rubin's rules* to combine estimates of interest (e.g., regression coefficients) across the M imputed datasets. To obtain such overall estimates and their standard errors in Stata, a separate user-written program called `mim` is required. Although we make use of `mim` here, we do so with a minimum of explanation, since the command is quite transparent; readers interested in details should consult Carlin *et al.* (2008); Royston *et al.* (2009) and the online help for `mim`.

The 'official' releases of `ice` (version 1.7.3), `ice_` (version 1.1.3) and `uvis` (version 1.5.5), as described by Royston (2009), and of `mim` (version 1.2.5), as described by Royston *et al.* (2009), may be installed within Stata using

```
. net from http://www.stata-journal.com/software/sj9-3
. net install st0067_4.pkg, replace
. net from http://www.stata-journal.com/software/sj9-2
. net install st0139_1.pkg, replace
```

The `replace` option causes the existing installation (if any) of the software on the user's media to be overwritten. The latest version of the **ice** package (at the time of writing, versions 1.9.5, 1.3.1 and 1.7.1 for the three components, `ice`, `ice_` and `uvis`, respectively) and of `mim` (version 2.1.6) are available on the first author's webpage, and may be installed as follows:

```
. net from http://www.homepages.ucl.ac.uk/~ucakjpr/stata/
. net install ice.pkg, replace
. net install mim.pkg, replace
```

The programs are updated as needed from time to time. Descriptions of updated or new features are published in *The Stata Journal* when enough significant changes have accrued to make such publication worthwhile.

3.2. MI in Stata 11

The `ice` program was written for Stata version 9.2 and above. At the time this article was accepted, Stata version 11 was newly released (StataCorp. 2009). One of the major new

Method	Description	mi impute method	uvis method
regress	Linear regression for a continuous variable	yes	yes
pmm	Predictive mean matching for a continuous variable	yes	yes*
logit	Logistic regression for a binary variable	yes	yes
ologit	Ordinal logistic regression for an ordinal variable	yes	yes
mlogit	Multinomial logistic regression for a nominal variable	yes	yes
intreg	Interval censored regression for a continuous variable	no	yes
nbreg	Negative binomial regression for a count variable	no	yes
bootstrap	Estimates regression coefficients in a bootstrap sample	no	yes**

Table 1: Comparison of mi impute with uvis for univariate imputation of missing values. (* available via the match option. ** available via the boot option.)

features of Stata 11 was its MI system. The system comprised a new database architecture for imputed datasets, utilities for manipulating, checking and validating such datasets, a sequence of commands for doing imputation, and one command for combining results using Rubin's rules. Many of the imputation models available in uvis were replicated in new commands of the form mi impute *XXX*, where *XXX* is a keyword such as regress for linear regression. The main multivariate imputer in Stata 11 was mi impute mvn, which performed multivariate normal imputation along the lines of Schafer's **norm** program (Schafer 1997; Novo and Schafer 2010); a version under the name **inorm** has been ported to Stata by Galati and Carlin (2009), and may be downloaded in Stata using ssc install inorm. Also available in Stata 11 was mi impute monotone, a multivariate imputer which requires that the incomplete variables exhibit a monotone missingness pattern. Thus, ice was not replicated in Stata 11 and was still needed for performing MICE for data with arbitrary missingness patterns.

Table 1 compares the main features of mi impute and uvis for univariate imputation of missing values in Stata 11, the building blocks of the MICE algorithm.

Combining results using Rubin's rules may be done using the mi estimate command. The latter overlaps the feature set of mim, but mim has some facilities that were not provided by mi estimate or elsewhere in Stata 11 mi. A notable example is the mcerror option of mim, which quantifies the Monte Carlo error in estimates and in other statistics, and is available in Stata 12 mi.

The mi import ice and mi export ice commands make it easy to transport data into and out of the existing ice data format. Also available from the first author's webpage, under the heading mi_ice, is a Stata program mi ice, which is essentially an mi-aware wrapper for ice.

At the time of writing, Stata 12 has just been released (StataCorp. 2011). It includes extensions of the mi impute system, notably mi impute chained, which, in principle like ice, performs multiple imputation by chained equations. However, a description of the new facilities is beyond the scope of the present article.

4. Imputation models for different types of variable

As we have discussed in Section 2, the essence of the MICE algorithm is regression of an

Variable	Name	Type	Levels	% missing
Albumin (*outcome variable*)	`alb`	Continuous (rounded)	30	33.0
Grade of tumour	`grade`	Ordinal	3	11.6
Residual disease	`resdis`	Ordinal	3	6.8
Performance status	`ps`	Ordinal	4	42.7
Presence/absence of ascites	`ascites`	Binary	2	5.4
Age (exact years)	`age`	Continuous	–	0.0
FIGO stage	`figo`	Nominal	4	1.8
Histology	`histol`	Nominal	7	0.0
Chemotherapy regimen	`ctype`	Nominal	3	0.0
Surgery (yes/no)	`surg`	Binary	2	0.0
CA125 (a cancer antigen)	`ca125`	Continuous	–	36.7
Alkaline phosphatase	`alp`	Continuous	–	33.1
All variables	–	–	–	70.1

Table 2: Variables and their missingness in the ovarian cancer dataset.

incomplete variable on other variables. The types of incomplete variable and their associated regression commands (methods) are listed for univariate imputation with `uvis` in Table 1. They carry over directly to `ice`. The default method is logistic regression (`logit`) when there are two distinct values of the variable to be imputed, multinomial logistic regression (`mlogit`) when there are 3-5 values and linear regression (`regress`) otherwise. Methods for imputing different variable types may be left to the default, may be specified through the `cmd()` option of `ice`, or in the case of nominal or ordinal categorical variables, may conveniently be specified using a prefix syntax – see Section 7.2.

5. Data

We used data from the $1,189$ patients with primary epithelial ovarian cancer diagnosed at the Western General Hospital (Edinburgh, Scotland) between 1984-01-01 and 1999-12-31. Patients were aged between 15 and 90 years at the time of diagnosis. The original analysis of the dataset (Clark *et al.* 2001) was aimed at developing a prognostic model for survival with ovarian cancer based on patient and tumour characteristics. In the present paper, we ignore the time-to-event outcome and concentrate on imputing missing values of albumin, one of the prognostic factors. Clinical details of the available variables are given in Table 1 of Clark *et al.* (2001). We summarize the salient features in Table 2.

Albumin was reported to the nearest integer and has only 30 distinct values recorded. FIGO stage was treated as nominal for illustration purposes, but could also be viewed as ordinal. Strikingly, the percentage of cases complete for all variables is only about 30%.

6. Imputing a single variable with missing values

6.1. Preliminaries

We begin with a simple example: imputing missing values of a continuous variable from another continuous variable. We exemplify features of `ice` through the relationship between albumin (`alb`) and age (`age`) in the patients with ovarian cancer. We concentrate on estimating the linear regression coefficient, β, of `alb` on `age`. Later on, in Section 7, we address the more difficult task of estimating β adjusting for confounders which are a mix of binary, nominal, ordinal and continuous variables, most of which have missing values.

6.2. Imputation assuming normality

We first assume that `alb` is normally distributed given `age`. An `ice` command to impute $M = 100$ complete datasets and its output are as follows:

```
. ice alb age, m(100) seed(11) clear noverbose

   #missing |
     values |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        797       67.03       67.03
          1 |        392       32.97      100.00
------------+-----------------------------------
      Total |      1,189      100.00

   Variable | Command | Prediction equation
------------+---------+-------------------------------------------------------
        alb | regress | age
        age |         | [No missing data in estimation sample]
------------------------------------------------------------------------------
Imputing
[Only 1 variable to be imputed, therefore no cycling needed]
[note: imputed dataset now loaded in memory]
Warning: imputed dataset has not (yet) been saved to a file
```

In the `ice` syntax, all variables, whether complete or incomplete, involved in the imputation model(s) are listed before the first comma. Items after the comma are (in Stata-speak) ‘options’; here, `m(100)` sets M to 100, `seed(11)` sets Stata’s random number seed to the (arbitrary) value 11, `clear` permits the existing dataset to be augmented in the workspace with the imputed datasets, and `noverbose` suppresses messages about the progress of the imputations. Use of the `seed()` option ensures that if required, imputed values can be reproduced in a later, identical run.

The first table of output notes the missingness status of the dataset; of the 1189 observations, there are 797 complete cases and 392 cases with 1 missing value (in fact, of `alb`). The second table reports the imputation model applied to each incomplete variable; by default, all variables are included in the model for any variable with missing data. Here, `regress` (linear regression) with normally distributed errors is used to ‘predict’ missing values of `alb`. Missing values are imputed in ‘proper’ fashion, as mentioned in Section 2.

The message `Only 1 variable to be imputed, therefore no cycling needed` is posted because the imputation is univariate; there is no need to cycle iteratively among regressions for different incomplete variables.

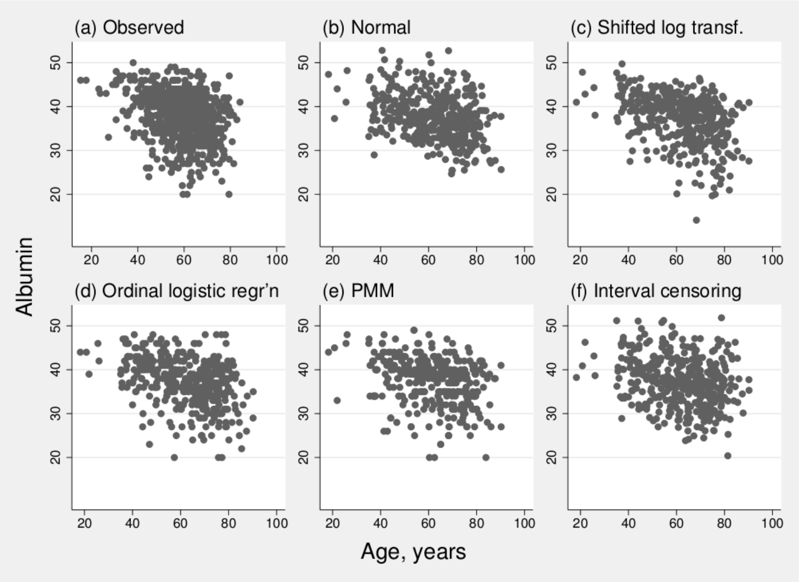

Figure 1: Relationship between `alb` and `age`. Panel (a): original data. Panels (b) to (f): imputed values of `alb` in the first imputed dataset, according to five different methods (see text for details).

Figure 1 shows the relationship between `alb` and `age` (a) in the original data and (b) in the missing data according to the conditionally normal imputation model.

The relation with `age` is nicely preserved. The only possible concern is that the distribution of imputed `alb` looks somewhat different from the distribution of the original observations. Shortly, we will consider three alternative imputation methods whose results are represented by panels (c) to (e) in Figure 1. The imputation method used in panel (f) differs from the other three and is described in Section 6.8. First, we discuss the format of a multiply imputed dataset and how to fit the analysis model.

6.3. Format of a multiply imputed dataset

If the original dataset had N observations (rows), a multiply imputed dataset produced by `ice` has $(M + 1)\,N$ observations. It is organized as shown below for the ovarian cancer data:

```
        +---------------------------------+
        |      alb        age   _mi   _mj |
        |---------------------------------|
     1. |        .    45.6783     1     0 |
     2. |       42   70.10815     2     0 |
     3. |        .   82.65572     3     0 |
     4. |       41   46.98426     4     0 |
     5. |       36   65.72485     5     0 |
...
  1190. | 36.92373    45.6783     1     1 |
  1191. |       42   70.10815     2     1 |
  1192. | 39.09165   82.65572     3     1 |
  1193. |       41   46.98426     4      1 |
  1194. |       36   65.72485     5      1 |
        +--------------------------------+
```

Of the first 5 observations in the data (indexed by `_mi` $= 1, \ldots, 5$ and `_mj` $= 0$, variables created automatically by `ice`), observations 1 and 3 of `alb` are missing, and imputations of them are seen in observations 1190 and 1192. The latter belong to the first imputation (`_mj` = 1). Complete observations are copied from the original to each of the imputed datasets.

The format used by `ice` is identical to the `flong` (full long) 'style' in Stata's `mi` system, except that the observation and imputation identifiers `_mi` and `_mj` are called `_mi_id` and `_mi_m`, respectively. The `mi` system has three other MI data formats (`flongsep`, `wide` and `mlong`). In addition, `mi` has a variable called `_mi_miss` which marks observations with any missing values.

6.4. Fitting the analysis model

We use `mim` to combine the estimated coefficients $\widehat{\beta}$ for linear regression of `alb` on `age` across the 100 imputed datasets according to Rubin's rules:

```
. mim: regress alb age

Multiple-imputation estimates (regress)                Imputations =     100
Linear regression                                      Minimum obs =    1189
                                                       Minimum dof =   252.7

----------------------------------------------------------------------------
       alb |     Coef.  Std. Err.      t    P>|t|    [95% Conf. Int.]     FMI
-----------+----------------------------------------------------------------
       age |  -.146285   .017189   -8.51   0.000   -.180137 -.112433   0.483
     _cons |   46.5035    1.0357   44.90   0.000    44.4643  48.5426   0.464
----------------------------------------------------------------------------
```

The combined $\widehat{\beta}$ is -0.146 (SE 0.017), close, as expected, to the complete-cases estimate of -0.147 (SE 0.017). To assess how much of the uncertainty in the reported quantities is due to Monte Carlo error, we request `mim`'s `mcerror` option:

```
. mim, mcerror

Multiple-imputation estimates (regress)                 Imputations =     100
Linear regression                                       Minimum obs =    1189
                                                        Minimum dof =   252.7

[Values displayed beneath estimates are Monte Carlo jackknife standard errors]
-----------------------------------------------------------------------------
        alb |     Coef.  Std. Err.      t    P>|t|    [95% Conf. Int.]     FMI
------------+----------------------------------------------------------------
        age |  -.146285   .017189   -8.51   0.000   -.180137 -.112433   0.483
```

```
            |   .001186   .000473    0.24   2.7e-13    .00156  .001499   0.029
            |
      _cons |   46.5035    1.0357   44.90     0.000   44.4643  48.5426   0.464
            |   .070062   .027479    1.20   5.1e-69   .089819  .089738   0.029
------------------------------------------------------------------------------
```

The Monte Carlo SE of $\widehat{\beta}$ and its SE are only 0.001 and 0.0005, respectively. Their small magnitude comes from creating a relatively large number of imputations. See White *et al.* (2011) for further discussion of Monte Carlo error.

6.5. Transformation toward normality

As described by Royston (2005b), to satisfy the normality assumption for a continuous variable, a transformation toward normality may be effective. Although, strictly speaking, *conditional* normality is required, in practice ensuring (approximate) *marginal* normality is often sufficient. A *shifted log transformation* of a variable z is one approach that we recommend; a positive or negative number γ is estimated such that $\ln(\pm z - \gamma)$ has zero sample skewness. If z is negatively skewed, the appropriate transformation is $\ln(-z - \gamma)$, otherwise it is $\ln(z - \gamma)$. We illustrate the approach with `alb`. The parameter γ may be estimated and the resulting shifted log transformation applied by using Stata's `lnskew0` command:

```
. lnskew0 lalb = alb

       Transform |         k     [95% Conf. Interval]        Skewness
-----------------+---------------------------------------------------
      ln(-alb-k) |   -64.1545      (not calculated)        -.0001185
(392 missing values generated)
```

Since `lnskew0` has reported the transformation as `ln(-alb-k)`, we deduce that `alb` is negatively skewed (it actually has $\sqrt{b_1} = -0.52$). The estimate of γ which approximately removes the negative skewness (the remaining skewness is -0.0001185) is `k` $= -64.1545$. Thus the created variable `lalb` equals `ln(-alb+64.1545)`. The inverse transformation, needed to back-transform imputed values of `lalb` to the original scale (`alb`), is `64.1545 - exp(lalb)`. The `ice` command which does the imputation is simply

```
. ice lalb age, m(100) seed(11) clear
```

[*output omitted*]

```
   Variable | Command | Prediction equation
------------+---------+-------------------------------------------------------
       lalb | regress | age
        age |         | [No missing data in estimation sample]
------------------------------------------------------------------------------
```

We then need to apply the inverse transformation to recover imputed values of `alb` on the original scale. We first use the `genmiss` subcommand of `mim` to create an indicator variable (`_mim_lalb`) which marks originally missing values of `lalb` in all the imputed datasets, and then replace the appropriate values of `alb`:

```
. mim: genmiss lalb
. replace alb = 64.1545 - exp(lalb) if (_mim_lalb == 1) & (_mj > 0)
```

The distribution of alb is more similar between the observed and imputed subsets than for the normal model (compare Figure 1 (c) with 1 (a) and 1 (b)). The combined estimate of $\widehat{\beta}$ is slightly larger than before, at -0.151 (SE 0.018).

6.6. Ordinal logistic regression

Provided it is appropriate to impute missing values of z from among the observed values of z, a convenient and often effective imputation model for z is ordinal logistic regression. In the present example, the approach is appealing since alb has been reported rounded to the nearest integer, giving only 30 distinct values. The ice command is as follows:

```
. ice o.alb age, m(100) seed(11) clear
```

A special feature of ice syntax is used here: o.alb. Details of the o. prefix are given in Section 7.2. Briefly, o.alb tells ice to impute alb using ordinal logistic regression. The results are illustrated in Figure 1 (d). The combined estimate of $\widehat{\beta}$ is similar to that with the normal model for alb given age, namely -0.149 (SE 0.016).

6.7. Predictive mean matching

Predictive mean matching (PMM) imputes missing values of a continuous variable z such that imputed values are sampled only from the observed values of z by matching predicted values as closely as possible. The resulting distribution of imputed z often closely matches that of observed z. PMM should be avoided when imputation appropriately involves extrapolation beyond the observed range of z or when the sample size is small. Mathematical details of how PMM is done are given by White *et al.* (2011, Section 4.2). In the terminology of the latter description, the value q of the match pool-size, i.e., the number of observations potentially available for matching predictions, is by default set to 3 in ice and uvis. It can be altered by using the matchpool() option. Older versions of the software did not have a matchpool() option and implicitly used matchpool(1).

PMM is implemented in ice as the match() option, for example:

```
. ice alb age, m(100) seed(11) clear match(alb)
```

The distribution of imputed values (see Figure 1 (e)) resembles that from ordinal logistic regression (Figure 1 (d)). Using PMM as above, the combined estimate of $\widehat{\beta}$ is -0.145 (SE 0.016).

6.8. Imputing an interval-censored variable

We illustrate here a slightly unusual but worthwhile feature of ice: the ability to impute on a continuous scale a metric variable that is recorded only in categories. Commonly encountered examples are age (recorded as age-groups) and salary (recorded as income brackets).

Suppose that age in the ovarian cancer data had been recorded only in the 5 age groups < 40, $[40, 50)$, $[50, 60)$, $[60, 70)$, ≥ 70. Let us assume that the lowest and highest possible ages for contracting ovarian cancer are 10 and 100 years, respectively. We create two categorical variables, say `age_ll` and `age_ul`, to store the lower and upper limits of each age group:

```
. recode age 0/39.999=10 40/49.999=40 50/59.999=50 60/69.999=60 *=70,
> generate(age_ll)
. recode age_ll 10=40 40=50 50=60 60=70 70=100, generate(age_ul)
. generate age2 = .
```

The character '`>`' denotes a long line that has been wrapped. The first `recode` creates the new variable `age_ll` storing the lower boundaries of the age groups. The second creates `age_ul`, the corresponding upper boundaries. (If we had preferred not to impose limits, we could have specified the lower boundary of the lowest age group and the upper boundary of the highest age group as missing.) We generate a third new variable, `age2`, initially as missing values, later to hold the newly created imputed continuous values of `age`. The model for `age2` assumes an underlying normal distribution, conditional on `alb`.

The `ice` command and the first part of its output are as follows:

```
. ice alb age2 age_ll age_ul, m(100) seed(11) clear
> interval(age2: age_ll age_ul)

   #missing |
     values |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |        797       67.03       67.03
          2 |        392       32.97      100.00
------------+-----------------------------------
      Total |      1,189      100.00

   Variable | Command | Prediction equation
------------+---------+-------------------------------------------------------
        alb | regress | age2
       age2 | intreg  | alb
     age_ll |         | [Lower bound for age2]
     age_ul |         | [Upper bound for age2]
-----------------------------------------------------------------------------
```

`ice` uses Stata's `intreg` command to obtain maximum likelihood estimates of the parameters of the normal model for `age2` as a function of `alb`. The distribution of imputed values in one imputed dataset is shown in Figure 1 (f). With the imputed data, the combined estimate of $\widehat{\beta}$ for regressing `alb` on `age2` is -0.142 (SE 0.016).

Note that here we have a truly multivariate (in fact, bivariate) imputation process. Although the variables `age_ll` and `age_ul` are both complete, nevertheless `ice` treats `age2` as 100% missing. It uses the MICE algorithm to switch between linear regression of `alb` on (imputed) `age2` and interval-censored regression of (`age_ll`, `age_ul`) on (observed and imputed) `alb`. Remarkably, despite the considerable loss of information caused by categorizing `age`, the MI regression of `alb` on `age2` gives almost exactly the same result as for the original `age` variable (see Figure 3).

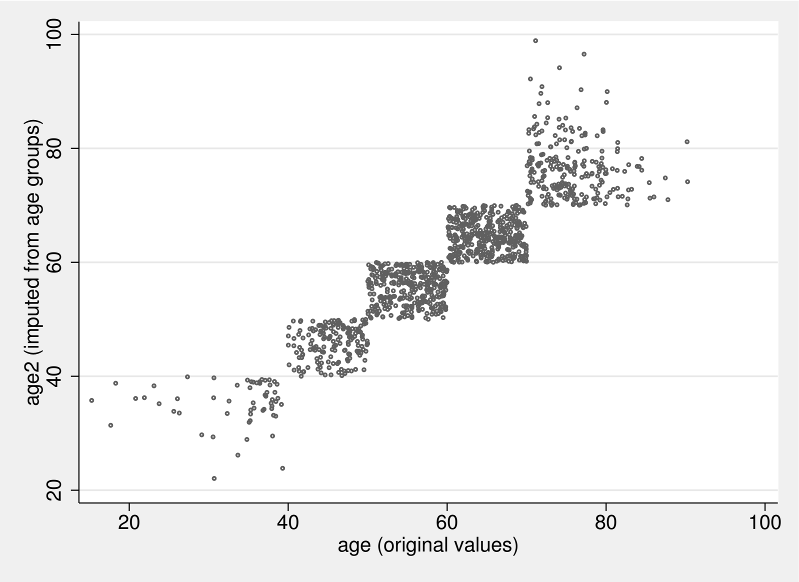

Figure 2: Relationship between `age2` and `age` in the first imputed dataset, based on interval-censored regression.

The relationship between `age2` and `age` in the first imputed dataset (`_mj = 1`) is shown in Figure 2.

While the overall Pearson correlation is quite high (0.91), there is by construction no true correlation within each age group. The correlations of `age` and `age2` with `alb` are about the same (−0.29 and −0.32, respectively).

6.9. Summary

We have presented five different solutions with `ice` to the problem of imputing a continuous variable with missing data (`alb`) from a complete continuous variable (`age`). A sixth possibility is complete cases analysis. Figure 3 shows the resulting estimates of the coefficient, $\widehat{\beta}$, and its 95% CI in the analysis model, which is linear regression of albumin on age.

Beta (i.e., $\widehat{\beta}$) is the regression coefficient in the complete cases analysis and the combined value (using Rubin's rules) in the other analyses.

The Monte Carlo error in each $\widehat{\beta}$ is of the order of 0.001. Although, therefore, there are real differences among the estimates, they are of no practical significance whatsoever. We can see this clearly in Figure 3.

7. Multivariate imputation

7.1. Preliminaries

We now turn to imputing missing values for use in a multivariable model in which `alb` is linearly regressed on `age` adjusting for several confounding variables. We use all ten available variables as potential confounders. The latter variables are `grade`, `resdis`, `ps`, `ascites`,

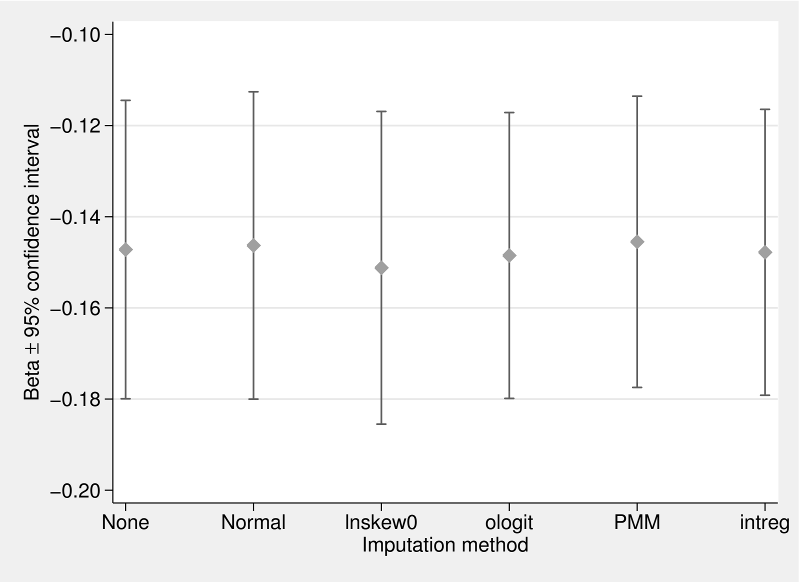

Figure 3: Regression coefficients (with 95% confidence intervals) for albumin on age after six different methods for imputing missing albumin. Key to horizontal axis labels: None, no imputation (complete cases); Normal, conditionally normal; lnskew0, shifted log transformation; ologit, ordinal logistic regression; PMM, predictive mean matching; intreg, interval censored age.

`figo`, `histol`, `ctype`, `surg`, `ca125` and `alp`.

The two continuous variables, `ca125` and `alp`, are both extremely positively skew. We reduce the skewness by transforming them to their logarithms, `lca125` and `lalp`, respectively. (Had these two variables been of greater importance in the analysis model, we would probably have transformed them more completely toward normality, for example, by applying a shifted log transformation.) The missing values of `lca125` and `lalp` are imputed assuming conditional normality.

Of the two binary variables, `ascites` and `surg`, the first has 5.5% missing values and the second is complete. No special treatment is required for these variables; `ascites` is imputed using logistic regression.

One complete ordinal variable (`ctype`) and three incomplete ones (`grade`, `resdis` and `ps`) are present. The latter are imputed using ordinal logistic regression.

One complete (`histol`) and one incomplete nominal variable (`figo`) are present. The latter is imputed using multinomial logistic regression.

We first present the complete `ice` command for the problem, then comment on specific aspects.

7.2. The `ice` run

We create $M = 100$ imputations according to the following command:

```
. ice alb age o.grade o.resdis o.ps ascites m.figo i.histol i.ctype surg
> lca125 lalp, m(100) seed(11) saving(multivar)
```

```
=> xi: ice alb age grade i.grade resdis i.resdis ps i.ps ascites figo i.figo
> i.histol i.ctype surg lca125 lalp, cmd(figo:mlogit, grade resdis ps:ologit)
> substitute(grade:i.grade, resdis:i.resdis, ps:i.ps, figo:i.figo) m(100)
> seed(11) saving(multivar, replace)
```

The second group of lines is output by `ice`'s preprocessor. It is a 'translation' of the user-entered command into the form that `ice` actually executes. The preprocessor recognizes three special prefixes to variable names, namely `i.`, `m.` and `o.`, which ease the task of specifying the imputation model for categorical variables.

A term of the form `i.`*variable_name* says (a) that *variable_name* has no missing values and (b) that *variable_name* is a categorical variable to be converted to its dummy variables when *variable_name* is used as a predictor in another variable's imputation model. The option `substitute()` associates *variable_name* with its dummy variables. For example, `substitute(grade:i.grade)` says 'whenever `grade` is a predictor, substitute it with the dummy variables created by `xi:i.grade`'. The 'prefix command' `xi:` is Stata's dummy variable generator. `grade` is a three-level ordinal variable taking values 1, 2 and 3. `xi:i.grade` creates two dummy variables, which (according to its default naming conventions) happen to be called `_Igrade_2` and `_Igrade_3`. By default, `xi:` drops the dummy variable associated with the lowest value (1) of `grade`, creates `_Igrade_2` equal to 1 when `grade` equals 2 and 0 otherwise, and creates `_Igrade_3` equal to 1 when `grade` equals 3 and 0 otherwise. The default behaviour of `xi:` may be modified in different ways, as described in the Stata documentation of the command.

A term of the form `m.`*variable_name* says (a) that *variable_name* has at least one missing value, (b) that *variable_name* is an unordered categorical variable to be imputed with multinomial logistic regression using Stata's `mlogit` command, and (c) that when *variable_name* is a predictor in another variable's imputation model, it is to be converted to its dummy variables, as just described for the `i.` prefix.

The form `o.`*variable_name* is identical to `m.`*variable_name* except that *variable_name* is assumed to be an ordinal variable to be imputed with ordinal logistic regression using Stata's `ologit` command.

As is clear from the above example, when several categorical variables are present, `ice`'s preprocessor feature greatly reduces both the amount of typing and the likelihood of making a syntax error. The tasks of mentioning all relevant variables 'before the comma' and of correctly specifying the `substitute()` and `cmd()` options can be onerous.

Finally, the `clear` option has been replaced with the `saving(`*filename* [`,replace`]`)` option which permanently stores the original and imputed data, here to a Stata-format file called `multivar.dta`. We include the `replace` sub-option to overwrite `multivar.dta` in case it happens to exist.

Note that the `o.` prefix recognized by `ice` should not be confused with Stata 11's internal (rarely-used) `o.` operator. The latter specifies the term that should be dropped from the model in the presence of collinearity. Also, the `i.`*varname* notation of `ice` should be distinguished from the `i.` factor-variable notation introduced in Stata 11. `ice` uses the 'old' way of including dummy variables via the `xi:` prefix.

7.3. Fitting the analysis model

An article primarily about imputation would be almost meaningless without an analysis model or models in mind. Having created the 100 imputations as just described, we use mim to fit the MI regression of alb on age, adjusting for all available confounders:

```
. use multivar, replace
. xi: mim: regress alb age i.grade i.resdis i.ps ascites i.figo i.histol
> i.ctype surg lca125 lalp
i.grade           _Igrade_1-3         (naturally coded; _Igrade_1 omitted)
i.resdis          _Iresdis_1-3        (naturally coded; _Iresdis_1 omitted)
i.ps              _Ips_0-3            (naturally coded; _Ips_0 omitted)
i.figo            _Ifigo_1-4          (naturally coded; _Ifigo_1 omitted)
i.histol          _Ihistol_1-7        (naturally coded; _Ihistol_1 omitted)
i.ctype           _Ictype_0-2         (naturally coded; _Ictype_0 omitted)

Multiple-imputation estimates (regress)                Imputations =     100
Linear regression                                      Minimum obs =    1189
                                                       Minimum dof =   158.8

------------------------------------------------------------------------------
       alb |     Coef.  Std. Err.     t    P>|t|    [95% Conf. Int.]     FMI
-----------+------------------------------------------------------------------
       age | -.081365   .016196   -5.02   0.000   -.113286 -.049444   0.528
 _Igrade_2 | -.273271   .678623   -0.40   0.688   -1.61174   1.0652   0.567
 _Igrade_3 | -.575581   .630512   -0.91   0.362   -1.81773  .666572   0.502
_Iresdis_2 | -.147188   .480532   -0.31   0.760   -1.09165  .797269   0.322
_Iresdis_3 | -.701506   .460626   -1.52   0.129   -1.60686  .203844   0.324
     _Ips_1 | -1.86618   .394812   -4.73   0.000   -2.64297 -1.08938   0.413
     _Ips_2 | -3.56228   .581297   -6.13   0.000   -4.70632 -2.41824   0.435
     _Ips_3 | -5.45046   .896262   -6.08   0.000   -7.21877 -3.68216   0.583
   ascites |  .694876   .379975    1.83   0.068   -.052936  1.44269   0.434
   _Ifigo_2 |  -.31739   .635984   -0.50   0.618   -1.56886  .934076   0.423
   _Ifigo_3 | -1.46409    .54969   -2.66   0.008   -2.54538 -.382797   0.397
   _Ifigo_4 | -1.42044   .687881   -2.06   0.040    -2.7729 -.067978   0.355
_Ihistol_2 | -.832649   .657984   -1.27   0.207   -2.12871  .463411   0.492
_Ihistol_3 |  -.13152   .412515   -0.32   0.750   -.942362  .679323   0.330
_Ihistol_4 |  -.61942   .624708   -0.99   0.322   -1.84894  .610097   0.437
_Ihistol_5 | -1.61377   .910045   -1.77   0.077   -3.40507  .177534   0.445
_Ihistol_6 |  .074369    1.1317    0.07   0.948   -2.15752  2.30625   0.562
_Ihistol_7 | -.895017   1.11046   -0.81   0.421   -3.07676  1.28673   0.284
 _Ictype_1 |  .180672   .551454    0.33   0.744   -.908458   1.2698   0.631
 _Ictype_2 |  1.08158   .635551    1.70   0.090   -.171166  2.33433   0.533
      surg |  .977838   1.00073    0.98   0.330   -.994651  2.95033   0.531
    lca125 | -.451467   .126678   -3.56   0.000   -.700903  -.20203   0.469
      lalp | -1.58529   .368222   -4.31   0.000   -2.30975 -.860833   0.411
     _cons |  54.2405   2.40758   22.53   0.000    49.4963  58.9848   0.517
------------------------------------------------------------------------------
```

It is clear from the table of regression output that several of the confounders are strongly associated with `alb` in a multivariable context. The main message is that after adjustment, the combined value of $\widehat{\beta}$ for `age` has been approximately halved, from -0.146 (SE 0.017) to -0.081 (SE 0.016).

If we fit the above regression model to the 358 complete cases available for the same variables in the original data, we obtain $\widehat{\beta} = -0.066$ (SE 0.022). We have been able to use only 30% of the observations. The complete-cases analysis is both potentially biased and inefficient.

To give an idea of timing, the total time taken by `ice` and `mim` to create 100 imputed datasets and fit the analysis model on a 3GHz dual-core PC was 13.8 min.

8. Other features of `ice`

It would be inappropriate here to give a complete description of all the features and options of `ice`. The help file of course provides most of the information, and the various articles on `ice` already referenced expand and illustrate details. We restrict ourselves to mentioning briefly a few distinctive features that are useful in practical applications.

8.1. Imputing with time-to-event data

Although the topic is not specific to `ice`, there is an issue as to how one should include survival-time variables t and d as predictors in the imputation model when the analysis model is some kind of survival analysis. Here, t denotes a possibly censored time to event and d the censoring indicator. van Buuren *et al.* (1999) suggest heuristically using t, $\log t$ and d. White and Royston (2009) showed that if the analysis model is a proportional hazards regression and there is just one binary predictor, an appropriate functional form is the cumulative hazard function for each individual and d.

8.2. Stratified imputation

It is sometimes desirable to impute separate strata of a dataset independently. The `ice` option `by(`*variable_name*`)` imputes in independent subsets according to the levels of *variable_name*. Stratification may be appropriate in a randomized trial, when *variable_name* denotes treatment arm and one wishes to allow for different relationships among the variables in the different treatment arms. It amounts to allowing for interactions between treatment and the other variables in the imputation model. Similar remarks apply when *variable_name* denotes centers, countries or some other large grouping of individuals whose multivariate structures are plausibly expected to differ. Clearly it should be avoided when the sample size may be small in some groups, since unstable imputations may be obtained.

8.3. Conditional imputation

Consider a dataset comprising the variables `age`, `female`, and `pregnant`, where female is coded 1 for females, 0 for males, and `pregnant` is coded 1 for pregnant, 0 for not pregnant. Males are coded `pregnant` = 0. All three variables may have missing values. Since males cannot be pregnant, we wish to impute missing values of `pregnant` using data only from females. Code with `ice` is as follows:

```
. ice age pregnant female, conditional(pregnant: female==1) clear
```

The prediction equation for `age` is `pregnant female`, whereas the equations for `pregnant` and for `female` are just `age`.

8.4. Monotone imputation

A monotone missing data pattern may be imputed in `ice` by specifying the `monotone` option. By default, `ice` orders variables to be imputed in order of increasing missingness. For variables $x_1, \ldots, x_k$ thus ordered, the imputation equations are x_1 on [nothing], x_2 on x_1, x_3 on x_1 x_2, $\ldots$, x_k on $x_1, \ldots, x_{k-1}$. When the missingness pattern really is monotone, only one cycle of MICE is required, so the default is `cycles(1)`. In addition, `ice` reports a 'non-monotonicity score'. The score is defined as 100× (sum of numerators) / (sum of denominators), where the sums are taken over all $k-1$ pairs of adjacent variables in $x_1, \ldots, x_k$. Consider two variables, x_1 and x_2. The numerator component for x_1 and x_2 is the number of observations in the estimation sample for which x_1 is missing and x_2 is observed. If the numerator is positive, x_1 and x_2 show a non-monotone pattern. The denominator for x_1 and x_2 is the the number of observations in the estimation sample for which x_2 is observed.

A relaxed view is taken by `ice` when the non-monotonicity score is positive. It issues a warning message, but goes ahead with the imputations anyway. Note, however, that non-monotone imputed values have the wrong association with later variables, e.g., imputed values of x_1 are independent of x_2, unlike with the standard `ice` approach.

Note that monotone imputation can also be performed by Stata 11's `mi impute monotone` command.

8.5. Perfect prediction

A potential problem with MICE arises when a conditional regression model exhibits perfect prediction. Perfect prediction affects logistic, ordered logistic and multinomial logistic regression models. In logistic regression, perfect prediction occurs when there is a category of any predictor variable in which the outcome is always 0 (or always 1), i.e., if the two-way table of a predictor variable by the outcome variable contains a zero cell. Perfect prediction results in infinite parameter estimates and difficulties in estimating the variance-covariance matrix of the parameter estimates. The result may be inappropriate imputations. The problem is discussed in White *et al.* (2010).

Various solutions exist (White *et al.* 2010). The one implemented in `ice` involves 'augmenting' the data by adding a few extra, temporary observations to the dataset (with small weight) so that no prediction is perfect. At the time of writing, perfect prediction still causes problems in other software, including `mi impute` in Stata 11.

9. Comments and conclusions

We have illustrated some of the features and use of `ice` in multiple imputation of multivariate missing data. Judging by the emails we receive, many people have used and are still using `ice` to good effect in their practical work. The main advantages of `ice` are its flexibility and wide range of options. We acknowledge, of course, that `ice` is work in progress and has limitations. No facility is available automatically to simplify imputation model(s), for example by stepwise deletion of variables. (Some would regard omission as a good thing!)

Apart from models for interval-censored data, `ice` does not implement range restrictions on imputed values. Imputing large numbers of variables in moderate-sized datasets is highly likely to cause estimation difficulties in some of the imputation models; the only sensible remedy is to prune the models, possibly quite severely. To facilitate the process, `ice` provides `eq()` and `eqdrop()` options.

It is straightforward to specify a set of incompatible conditional distributions for which no multivariate density exists (van Buuren 2007). In general, little is known theoretically or practically about the effects of incompatible conditional distributions on the quality of imputations. However, van Buuren *et al.* (2006) performed some simulations under severely incompatible models and observed that the adverse effects on the estimates following MI were 'minimal'.

It is important to realize that all imputation methods can fail (see section 10 of White *et al.* (2011) for further discussion). The user can and should carry out simple checks of the quality of imputations, e.g., as in Figure 1.

Imputations are based on drawing parameter values in an imputation model from a normal approximation to the distribution of the parameter estimates. An alternative, implemented in the `ice` option `boot`, is the approximate Bayesian bootstrap.

Acknowledgments

We thank Dr Taane G. Clark for kind permission to use the ovarian cancer data. We are grateful to Yulia Marchenko and an anonymous reviewer for helpful suggestions which enabled us to improve the manuscript. Patrick Royston is supported by MRC grant MC_US_A737_0002 to the MRC Clinical Trials Unit and Ian White by MRC grant MC_US_A030_0014 to the MRC Biostatistics Unit.

References

Barnard J, Meng XL (1999). "Applications of Multiple Imputation in Medical Studies: From AIDS to NHANES." *Statistical Methods in Medical Research*, **8**, 17–36.

Carlin JB, Galati JC, Royston P (2008). "A New Framework for Managing and Analysing Multiply Imputed Data in Stata." *The Stata Journal*, **8**, 49–67.

Clark TG, Stewart ME, Altman DG, Gabra H, Smyth JF (2001). "A Prognostic Model for Ovarian Cancer." *British Journal of Cancer*, **85**, 944–952.

Galati JC, Carlin JB (2009). "**inorm**: Stata Module to Perform Multiple Imputation Using Schafer's Method." Statistical Software Components, Boston College Department of Economics. URL `http://EconPapers.RePEc.org/RePEc:boc:bocode:s456966`.

Horton NJ, Kleinman KP (2007). "Much Ado about Nothing: A Comparison of Missing Data Methods and Software to Fit Incomplete Data Regression Models." *The American Statistician*, **61**, 79–90.

Kenward MG, Carpenter J (2007). "Multiple Imputation: Current Perspectives." *Statistical Methods in Medical Research*, **16**, 199–218.

Little RJA, Rubin DB (2002). *Statistical Analysis with Missing Data.* 2nd edition. John Wiley & Sons, New York.

Novo AA, Schafer JL (2010). ***norm**: Analysis of Multivariate Normal Datasets with Missing Values.* R package version 1.0-9.2, URL http://CRAN.R-project.org/package=norm.

Royston P (2004). "Multiple Imputation of Missing Values." *The Stata Journal*, **4**, 227–241.

Royston P (2005a). "Multiple Imputation of Missing Values: Update." *The Stata Journal*, **5**, 188–201.

Royston P (2005b). "Multiple Imputation of Missing Values: Update of **ice**." *The Stata Journal*, **5**, 527–536.

Royston P (2007). "Multiple Imputation of Missing Values: Further Update of **ice**, with an Emphasis on Interval Censoring." *The Stata Journal*, **7**, 445–464.

Royston P (2009). "Multiple Imputation of Missing Values: Further Update of **ice**, with an Emphasis on Categorical Variables." *The Stata Journal*, **9**, 466–477.

Royston P, Carlin JB, White IR (2009). "Multiple Imputation of Missing Values: New Features for **mim**." *The Stata Journal*, **9**, 252–264.

Schafer JL (1997). *Analysis of Incomplete Multivariate Data.* Chapman and Hall, London.

Schafer JL (1999). "Multiple Imputation: A Primer." *Statistical Methods in Medical Research*, **8**, 3–15.

StataCorp (2009). *Stata Statistical Software: Release 11.* StataCorp LP, College Station, TX. URL http://www.stata.com/.

StataCorp (2011). *Stata Data Analysis Statistical Software: Release 12.* StataCorp LP, College Station, TX. URL http://www.stata.com/.

Sterne JAC, White IR, Carlin JB, Spratt M, Royston P, Kenward MG, Wood AM, Carpenter JR (2009). "Multiple Imputation for Missing Data in Epidemiological and Clinical Research: Potential and Pitfalls." *British Medical Journal*, **338**, b2393.

van Buuren S (2007). "Multiple Imputation of Discrete and Continuous Data by Fully Conditional Specification." *Statistical Methods in Medical Research*, **16**, 219–242.

van Buuren S, Boshuizen HC, Knook DL (1999). "Multiple Imputation of Missing Blood Pressure Covariates in Survival Analysis." *Statistics in Medicine*, **18**, 681–694.

van Buuren S, Brand JPL, Groothuis-Oudshoorn K, Rubin DB (2006). "Fully Conditional Specification in Multivariate Imputation." *Journal of Statistical Computation and Simulation*, **76**, 1049–1064.

White IR, Daniel R, Royston P (2010). "Avoiding Bias due to Perfect Prediction in Multiple Imputation of Incomplete Categorical Variables." *Computational Statistics & Data Analysis*, **54**, 2267–2275.

White IR, Royston P (2009). "Imputing Missing Covariate Values for the Cox Model." *Statistics in Medicine*, **28**, 1982–1998.

White IR, Royston P, Wood AM (2011). "Multiple Imputation Using Chained Equations: Issues and Guidance for Practice." *Statistics in Medicine*, **30**, 377–399.

Affiliation:

Patrick Royston
Hub for Trials Methodology Research
Medical Research Council
Clinical Trials Unit
and
University College London
Aviation House
125 Kingsway
London WC2B 6NH, United Kingdom
E-mail: pr@ctu.mrc.ac.uk

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/

Volume 45, Issue 4 *Submitted:* 2009-09-12
December 2011 *Accepted:* 2011-05-30