Journal of Statistical Software

December 2011, Volume 45, Issue 5. *http://www.jstatsoft.org/*

REALCOM-IMPUTE Software for Multilevel Multiple Imputation with Mixed Response Types

James R. Carpenter
London School of Hygiene & Tropical Medicine

Harvey Goldstein
University of Bristol

Michael G. Kenward
London School of Hygiene & Tropical Medicine

Abstract

Multiple imputation is becoming increasingly established as the leading practical approach to modelling partially observed data, under the assumption that the data are missing at random. However, many medical and social datasets are multilevel, and this structure should be reflected not only in the model of interest, but also in the imputation model. In particular, the imputation model should reflect the differences between level 1 variables and level 2 variables (which are constant across level 1 units). This led us to develop the **REALCOM-IMPUTE** software, which we describe in this article. This software performs multilevel multiple imputation, and handles ordinal and unordered categorical data appropriately. It is freely available on-line, and may be used either as a standalone package, or in conjunction with the multilevel software **MLwiN** or Stata.

Keywords: multilevel multiple imputation, missing data, mixed response types.

1. Introduction

Multiple imputation is becoming increasingly established as the leading practical approach to analysing partially observed datasets (Sterne, White, Carlin, Spratt, Royston, Kenward, Wood, and Carpenter 2009; Klebanoff and Cole 2008). Although there are now an increasing number of software packages around, they vary in their accessibility to data analysts. More fundamentally, some software uses the full conditional specification approach. For an early example see van Buuren, Boshuizen, and Knook (1999), which does not explicitly model the joint distribution but forms univariate models for each incomplete variable in turn conditional on all the others. There is no guarantee in general that these correspond to a proper joint model.

Other software is based on an explicit joint model, as described for example in Schafer (1997).

Moreover, some software treats discrete data as continuous in the imputation model, and most packages do not allow for multilevel structure (Kenward and Carpenter 2007).

In this paper, we describe the **REALCOM-IMPUTE** software we have developed. This is an extension of the **REALCOM** software that we have developed to fit multivariate multilevel mixed response models (Goldstein, Carpenter, Kenward, and Levin 2009). Using a multivariate latent normal model allows us to properly handle discrete data, and also naturally allow for two-level structure. In particular, our software allows us to properly handle 'level 2' variables, which are constant over the observations at level 1. This work builds on that described in Carpenter and Goldstein (2005), where we described macros for multilevel multiple imputation in **MLwiN** (Centre for Multilevel Modelling. 2011) which use a multilevel normal imputation model. In that paper, we also demonstrated the need for multiple imputation to respect the multilevel structure of the data, and the multilevel structure in the model of interest, in order to avoid biasing the parameter estimates in the multilevel model and producing potentially invalid estimates of precision.

As before, a further aim in our development has been to build on the unique `Equations` window in **MLwiN** to make the process of multiple imputation, together with both the imputation model and the model of interest, as accessible as possible. We believe this is the key for data analysts, who may not have sufficient statistical training to use multiple imputation appropriately.

The article is structured as follows. In Section 2 we describe our example multilevel data set, and in Section 3 we give more details about our **REALCOM-IMPUTE** software. Section 4 describes the use of the software, and we conclude with a discussion in Section 5.

2. Example data set

To illustrate our approach we will use data from the class size study, made available to us by Peter Blatchford at the Institute of Education, London. This study sought to understand the effect of class size on development of literacy and numeracy skills in the first two years of English childrens' full time education. The analysis below is illustrative; for a fuller analysis and more details of the study see Blatchford, Goldstein, Martin, and Browne (2002).

The version of the dataset we analyse below relates to children in their first year of full time education in the UK, known as the reception year. Table 1 shows the five variables in the dataset, which is available with this article. Four measure literacy and numeracy skills when children start their reception class and at the end of their reception year, and the fifth is class size.

The respective literacy and numeracy test scores have been normalized to create the variables

Variable name	Description
`nlitpost`	Standardized literacy score at the end of 1st school year
`nmatpost`	Standardized numeracy score at the end of 1st school year
`nlitpre`	Standardized literacy score school entry
`nmatpre`	Standardized numeracy score at school entry
`csize`	Categorical class size variable: 1 is ≤ 19; 2 is 20–24; 3 is 25–29; 4 is ≥ 30

Table 1: Description of variables in class size data used in this analysis.

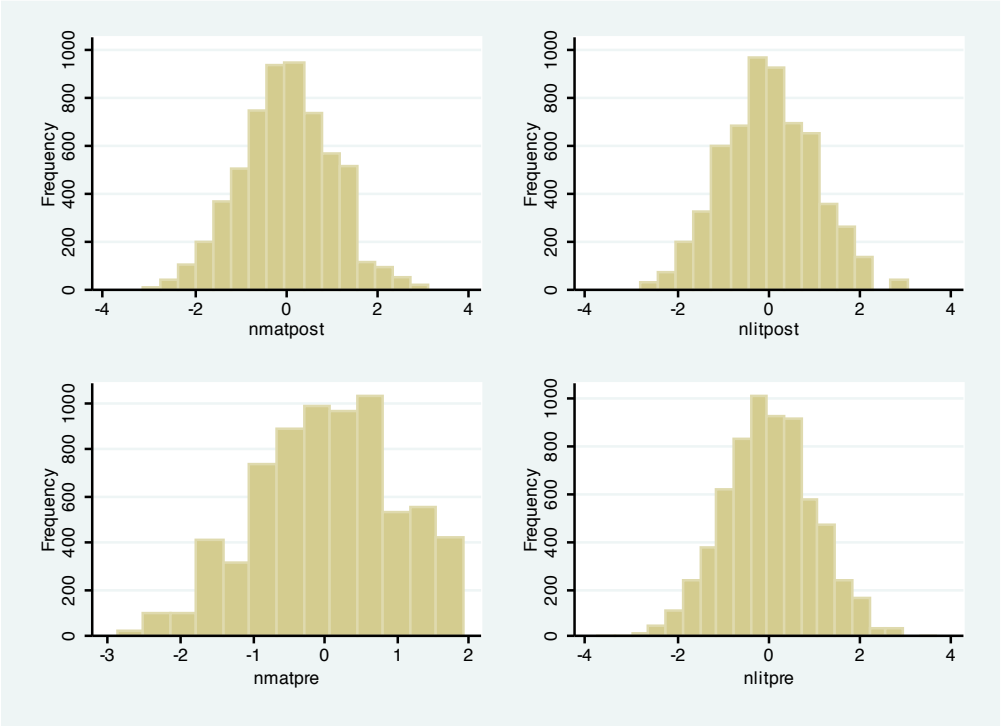

Figure 1: Histograms of literacy and numeracy variables.

`nlitpre`, `nmatpre`, `nlitpost` and `nmatpost`. This was done as follows. For each test, the pupils' results were ranked. Then for observation in rank order i, where n pupils sat the test, the normalized result was calculated as the inverse normal of $i/(n+1)$. Since many pupils got the same marks there are ties in the data. After this transformation, Figure 1 shows the test score data data are approximately normal.

The class size variable originally could change within classes, since in England children enter their first year class either in September or after Christmas, depending on their birthday. For the analysis here we avoid this additional complication by, for each class, calculating the class size as the average average class size over all the children in the class at the end of the first year. Where this could not be calculated, because class size was missing for one or more children in the class, we set class size to missing. We note that class size is not just the count of the number of children with the same class identifier in the dataset. Among other reasons this arises because not all children in a class were necessarily included in the study. In the original paper reporting the results, the effect of class size was modelled using a spline (Blatchford *et al.* 2002). To simplify the analysis, and illustrate how our software can handle categorical data, we here use a four-category version of class size.

The dataset is thus multilevel, with children at level 1 belonging to classes at level 2. Class size is a level 2 variable. Table 2 shows the missing value patterns. Class size is missing for 32 classes out of 329. Of 7406 records, 12 had no data on any of the variables in this data set. This left 7394 records, of which 5033 had no missing data for the variables in the model of interest below.

Let j denote class and i denote pupil and $1[\cdots]$ an indicator for the event in brackets. Our

Pattern	Variable				Number	Percent
	`nlitpost`	`nmatpost`	`nlitpre`	`nmatpre`		
1	+	+	+	+	5233	71.0%
2	–	–	+	+	1013	13.7%
3	+	+	–	+	292	3.9%
4	+	+	–	–	207	2.8%
5	–	+	+	+	191	2.6%
6	+	–	+	+	179	2.4%
7	–	–	–	+	150	2.0%
8	All other patterns				129	1.7%

Table 2: Principal missing value patterns in the class size data.

model of interest is is

$$\begin{aligned} \texttt{nmatpost}_{ij} &= \beta_{0ij} + \beta_1 \texttt{nmatpre}_{ij} + \beta_2 \texttt{nlitpre}_{ij} + \beta_3 1[20 \le \texttt{csize}_j \le 24] + \\ &\quad \beta_4 1[25 \le \texttt{csize}_j \le 29] + \beta_5 1[30 \le \texttt{csize}_j] \\ \beta_{0ij} &= \beta_0 + u_j + e_{ij} \\ u_j &\sim N(0, \sigma_u^2); \quad e_{ij} \sim N(0, \sigma_e^2). \end{aligned} \tag{1}$$

Parameter estimates from fitting this, our model of interest, to the 5033 complete cases are shown in Table 3 and the top panel of Figure 5. As expected, we see that literacy and numeracy when children start school strongly predicts their score at the end of the year. Increasing class size seems to reduce achievement at the end of the year, and this just passes significance at the 5% level if the class size is 25 or over.

3. REALCOM-IMPUTE software

The **REALCOM-IMPUTE** software is a free standing package, designed to have a smooth interface with **MLwiN**, although it can be used with any package. It fits multivariate response models to 2-level data, allowing for both level 1 and level 2 variables, and through this allows proper imputation of missing data. Continuous data are modelled using the multivariate normal distribution. The default is to have all the variables as responses, although fully observed variables can be included as covariates; in this way interactions with fully observed variables may be handled. For each level 1 response a mean and level 2 random intercept is fitted, together with a level 1 residual. For level 2 variables, only a mean and level 2 residual is fitted. Level 1 and level 2 residuals are assumed independent, with mean zero, with separate covariance matrices. If all variables are normal these are unstructured; otherwise these have appropriate structures for the latent normal model for discrete data (Goldstein *et al.* 2009).

As an illustration, consider multiple imputation for model (1), fitted to the class size data. In this example, there are three level-1 variables which (as they have been transformed) we treat as normal, and a level 2 variable, class size, which we treat as unordered categorical. In addition, we include literacy score at the end of the first year, `nlitpost`, as an auxiliary variable. That is, it is included in the imputation model both to increase the plausibility of the underlying assumption that data are missing at random and because it is a strong predictor of the other variables.

Covariate	Parameter	Estimates (std. errors) from Complete cases ($n = 5033$)	Multiple imputation ($n = 7394$)
`constant`	β_0	0.269 (0.141)	0.253 (0.155)
`nmatpre`	β_1	0.367 (0.015)	0.353 (0.014)
`nlitpre`	β_2	0.372 (0.015)	0.383 (0.015)
`class size 20-24`	β_3	−0.114 (0.157)	−0.093 (0.169)
`class size 25-29`	β_4	−0.318 (0.149)	−0.309 (0.160)
`class size ≥30`	β_5	−0.340 (0.160)	−0.289 (0.174)
Between class variance	σ_u^2	0.252 (0.025)	0.282 (0.027)
Between pupil, within class, variance	σ_e^2	0.375 (0.008)	0.386 (0.007)

Table 3: Parameter estimates from fitting model (1) to complete cases, and after multiple imputation. The model of interest is fitted using restricted maximum likelihood.

The details of how to use the software are described in the next subsection. We first describe the model that **REALCOM-IMPUTE** will propose in the absence of the level 2 variable `csize`. We then describe how this is extended when we introduce class size as an unordered categorical variable at level 2. As above, let i index children and j index class. **REALCOM-IMPUTE** proposes the following model (subscripts of random effects and coefficients are chosen to match the software output):

$$\begin{aligned}
\texttt{nmatpost}_{ij} &= \beta_1 + u_{1j} + e_{1ij} \\
\texttt{nlitpost}_{ij} &= \beta_2 + u_{2j} + e_{2ij} \\
\texttt{nmatpre}_{ij} &= \beta_3 + u_{3j} + e_{3i} \\
\texttt{nlitpre}_{ij} &= \beta_4 + u_{4j} + e_{4ij}
\end{aligned}$$

$$\begin{aligned}
\begin{pmatrix} u_{1j} \\ u_{2j} \\ u_{3j} \\ u_{4j} \end{pmatrix} &\sim N(0, \Omega_u^{cts}), \quad \text{where } \Omega_u^{cts} \text{ is an unstructured } 4 \times 4 \text{ covariance matrix;} \\
\begin{pmatrix} e_{1ij} \\ e_{2ij} \\ e_{3ij} \\ e_{4ij} \end{pmatrix} &\sim N(0, \Omega_e), \quad \text{where } \Omega_e \text{ is an unstructured } 4 \times 4 \text{ covariance matrix.}
\end{aligned} \tag{2}$$

We note that this imputation model is equivalent to a conditional model for each variable in which it is linearly regressed on all the other variables.

Recall from (1) we treat class size as a 4-level unordered categorical variable at level 2. We therefore extend (2), to include 3 latent normal variables, $Y_{1,j}, Y_{2,j}, Y_{3,j}$ which are related to the 4-level class size variable through the maximum indicant model (Aitchison and Bennett 1970). Under the implementation of this model in **REALCOM-IMPUTE**,

$$\begin{aligned}
\Pr\{(\text{class size category } 1)_j\} &= \Pr\{Y_{1,j} > 0 \text{ and } Y_{1,j} > Y_{2,j} \text{ and } Y_{1,j} > Y_{3,j}\} \\
\Pr\{(\text{class size category } 2)_j\} &= \Pr\{Y_{2,j} > 0 \text{ and } Y_{2,j} > Y_{1,j} \text{ and } Y_{2,j} > Y_{3,j}\} \\
\Pr\{(\text{class size category } 3)_j\} &= \Pr\{Y_{3,j} > 0 \text{ and } Y_{3,j} > Y_{1,j} \text{ and } Y_{3,j} > Y_{2,j}\} \\
\Pr\{(\text{class size category } 4)_j\} &= \Pr\{Y_{1,j} < 0 \text{ and } Y_{2,j} < 0 \text{ and } Y_{3,j} < 0\}.
\end{aligned}$$

These latent variables are included in (2) at level 2 by adding

$$\begin{aligned}
Y_{1j} &= \beta_{1,5} + u_{1,5j} \\
Y_{2j} &= \beta_{2,5} + u_{2,5j} \\
Y_{3j} &= \beta_{3,5} + u_{3,5j}
\end{aligned}$$

$$\text{where} \quad \begin{pmatrix} u_{1,5j} \\ u_{2,5j} \\ u_{3,5j} \end{pmatrix} \sim N\left(0, \Omega_u^{cat} = \begin{pmatrix} \sigma^2_{u_{1,5}} & 0 & 0 \\ 0 & \sigma^2_{u_{2,5}} & 0 \\ 0 & 0 & \sigma^2_{u_{3,5}} \end{pmatrix}\right)$$

The overall covariance matrix of the level 2 random effects is then

$$\Omega_u = \begin{pmatrix} \Omega_u^{cont} & \Omega_u^{cont,cat} \\ \Omega_u^{cat,cont} & \Omega_u^{cat} \end{pmatrix},$$

with the 4-by-3 submatrix $\Omega_u^{cont,cat}$ unstructured. For full details of this model see Goldstein *et al.* (2009). This extended model remains equivalent to a conditional model for each (latent) variable in which it is linearly regressed on all the other variables.

We can add covariates to this model, provided they have no missing data, and these covariates can have separate coefficients for each response, which may have random effects at level 2. Binary and discrete data can either be treated as normal—possibly after transformation (Bernaards, Belin, and Schafer 2007; Lee and Carlin 2010)—or, preferably in our view, properly modelled. Again, we use a latent normal structure to do this, via a probit link function, with the probit analogue of the proportional odds model for ordinal data. We reiterate that all these discrete variables can be included at either level 1 or two; the appropriate constraints on the covariance matrices are implemented automatically.

Once specified, the **REALCOM-IMPUTE** software fits the model using Markov Chain Monte Carlo. We use a Gibbs sampling approach, updating each set of parameters in turn, conditional on the others. Where possible, we sample direct from the appropriate conditional distribution. Otherwise we use Metropolis steps or rejection sampling. Full details are given in Goldstein *et al.* (2009). The user has the option of requesting plots of parameter chains, and these are displayed and updated as the MCMC sampler runs. After the user-specified burn in, the software displays the parameter estimates for the joint model. If requested, it will then continue updating the sampler, imputing the missing data, and creating a file of imputed datasets stacked vertically.

The **REALCOM-IMPUTE** software has recently been extended to allow for weights (e.g., survey weights) both at level 1 and at level 2 and this is the subject of a forthcoming paper.

3.1. Obtaining and using the REALCOM-IMPUTE software

The program **MLwiN** can be downloaded from `http://www.cmm.bris.ac.uk/`, and is freely distributed to the UK academic community. The **REALCOM-IMPUTE** software is free for

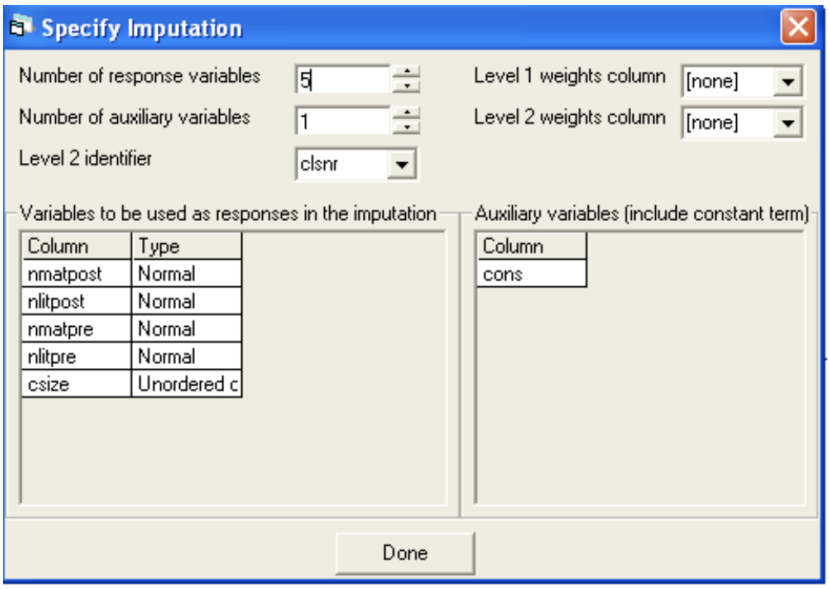

Figure 2: **MLwiN** dialogue box for exporting data to the **REALCOM-IMPUTE** software.

all users and can be downloaded from `http://www.cmm.bristol.ac.uk/research/Realcom/index.shtml`. Code to export data from Stata (StataCorp. 2011) to **REALCOM-IMPUTE** and vice-versa is available from `http://www.missingdata.org.uk/`. We now describe the steps required to obtain the results presented in Section 4 below; more detailed instructions are available in the on-line manual that comes with the software.

Multiple imputation in the **MLwiN** ↔ **REALCOM-IMPUTE** framework proceeds as follows:

1. Fit the model of interest in **MLwiN**. By default, this will only use complete cases for the estimation. Doing this for the class size data gives the results shown in the top panel of Figure 5 (parameter estimates and standard errors shown in green).

2. In **MLwiN**, from the toolbar, select `Model -> Imputation -> Save Imputation Specification`. A dialogue will open, asking for the number of response variables, number of auxiliary variables (these are included as covariates in the imputation model, and should not have any missing data) and the level 2 identifier. Then the user must specify the names of the response and auxiliary variables (including the constant) from a drop-down menu. In addition, the user must specify the 'type' for each response (continuous, ordinal, or unordered categorical). Binary variables may be specified as either ordinal or unordered categorical.

   The resulting dialogue for the class size analysis is shown in Figure 2. We first specify the number of response variables (all the variables in the model of interest which have missing observations, together with any partially observed auxiliary variables we wish to bring into the imputation), the number of fully observed variables (including any fully observed auxiliary variables) and the column identifying the level 2 groups. Here, these are classes, and the class identifier is `clsnr`. The four variables in the model of interest, `nmatpost`, `nmatpre`, `nlitpre` and `csize` are all responses, together with the (partially observed) auxiliary variable `nlitpost`. The only fully observed auxiliary variable in this setting is the intercept, `CONS`, i.e., a vector of 1's.

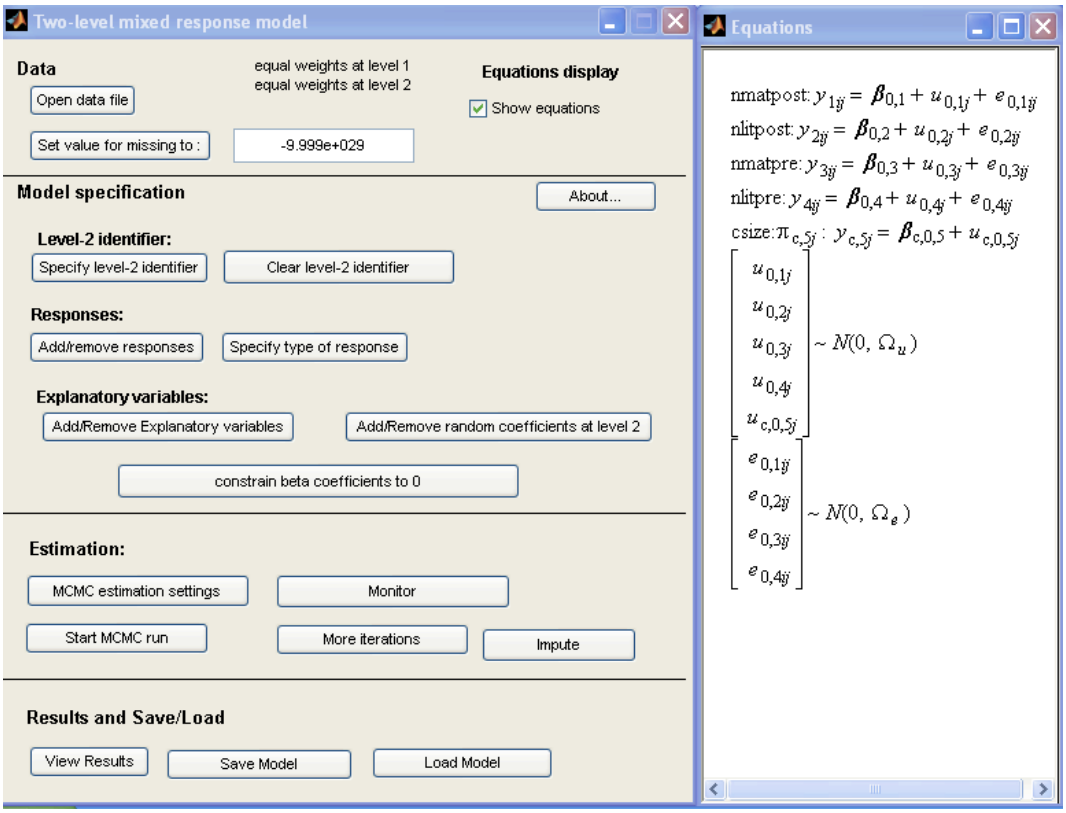

Figure 3: **REALCOM-IMPUTE** software: Dialogue box (left) and imputation equation window (right).

When the user clicks on `Done` at the bottom of Figure 2, **MLwiN** will ask for a file name for the data which is being exported to **REALCOM-IMPUTE**.

3. Next, start **REALCOM-IMPUTE** (from the `START` menu select `Realcom -> Mixed responses modelling`. After a few moments, a window similar to the left panel of Figure 3 appears. Click on `Open data file` to load the data exported from **MLwiN**.

   A window appears. Clicking on `open data file` prompts for a file name for loading the data file created by **MLwiN**. Then, click on the `show equations` box, to display the proposed imputation model (right panel of Figure 3). In most cases, this imputation model will be the appropriate one (given the data exported from **MLwiN**). If not, the `Model Specification` part of the **REALCOM-IMPUTE** dialogue, allows the user to change the variable which identifies clusters, add/remove response variables, explanatory variables and random coefficients. In addition, we can constrain certain coefficients.

   For our example, the right hand panel of Figure 3 shows the imputation equation. We see that the top four responses are modelled as normal, with level 2 and level 1 residuals. Class size is modelled as an unordered categorical. We do not model it as an ordinal variable, in order to allow for the fact it may not satisfy the proportionality assumption. Note the index c on $\pi_{c,5j}$, which indexes category ($c = 2, 3, 4$ as category 1 is a reference) and enables a more concise presentation of the model.

   Once the imputation model is specified as desired, details of the MCMC estimation can

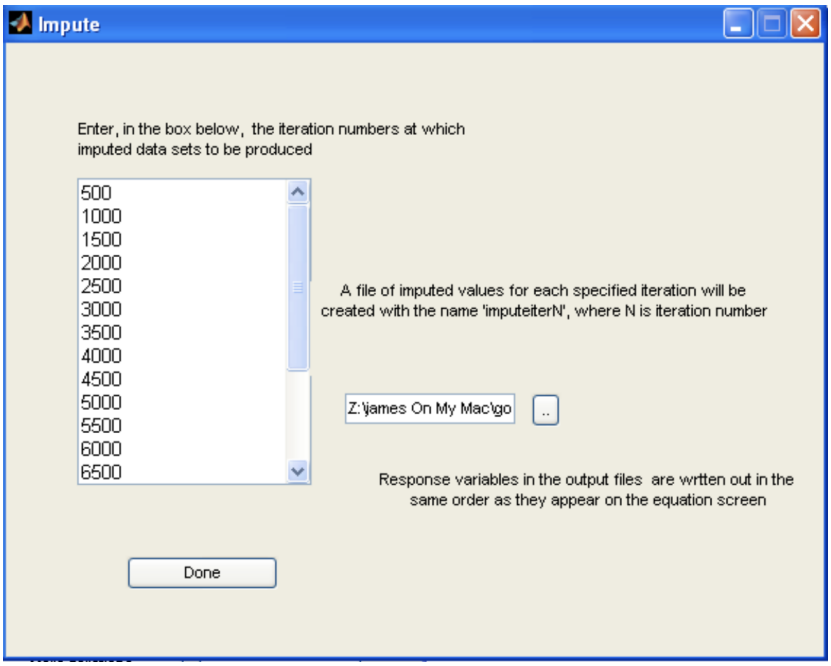

Figure 4: **REALCOM-IMPUTE** software: Specifying the iterations at which imputations are generated.

be specified using the `Estimation` part of the dialogue. In particular, clicking on `MCMC estimation settings` allows specification of the 'burn in' for the MCMC sampler, total number of iterations (post burn in) and the refresh rate. We now discuss these in more detail.

The 'burn in' should be long enough for the sampler to leave behind the initial values and converge to the posterior distribution (see Gilks, Richardson, and Spiegelhalter 1996, for example). This occurs with fewer updates if variables are mean-centred. While more complex models require a longer burn in, we believe 2000 is plenty and 500 may be sufficient for simpler models.

We wish successive imputed datasets to be (approximately) independent draws from the distribution of the missing data given the observed. In practice we believe that 500 updates in between imputations is sufficient to achieve this. If we adopted this, for 20 imputations we would need to specify 10,000 updates.

Lastly, the refresh rate is the frequency with which the graphical monitor of the MCMC process updates. This entails a (non-statistical) computational overhead. Too low a value slows the software considerably. We therefore recommend a value between 50 and 100.

To exit the `MCMC estimation settings` dialogue, click on `Done`. Next click on `Monitor` to specify which of the parameter chains are to be monitored. Again, monitoring all chains slows the software; we often focus on variance parameters and parameters relating to discrete variables.

Then, click on `Impute` to specify the iterations of the MCMC sample at which an imputed dataset will be generated; for example with 10,000 post-burn in iterations of

$$\text{nmatpost}_{ij} \sim \text{N}(XB,\ \Omega)$$
$$\text{nmatpost}_{ij} = \beta_{0ij}\text{cons} + 0.367(0.015)\text{nmatpre}_{ij} + 0.372(0.015)\text{nlitpre}_{ij} + \text{-}0.114(0.157)\text{csize20-24}_j + \text{-}0.318(0.149)\text{csize25-29}_j + \text{-}0.340(0.160)\text{csize>=30}_j$$
$$\beta_{0ij} = 0.269(0.141) + u_{0j} + e_{0ij}$$
$$\begin{bmatrix} u_{0j} \end{bmatrix} \sim \text{N}(0,\ \Omega_u) \ :\ \Omega_u = \begin{bmatrix} 0.252(0.025) \end{bmatrix}$$
$$\begin{bmatrix} e_{0ij} \end{bmatrix} \sim \text{N}(0,\ \Omega_e) \ :\ \Omega_e = \begin{bmatrix} 0.375(0.008) \end{bmatrix}$$

*-2*loglikelihood(IGLS Deviance)* = 9989.453(5033 of 7394 cases in use)

$$\text{nmatpost}_{ij} \sim \text{N}(XB,\ \Omega)$$
$$\text{nmatpost}_{ij} = \beta_{0ij}\text{cons} + 0.353(0.014)\text{nmatpre}_{ij} + 0.383(0.015)\text{nlitpre}_{ij} + \text{-}0.093(0.169)\text{csize20-24}_j + \text{-}0.309(0.160)\text{csize25-29}_j + \text{-}0.289(0.174)\text{csize>=30}_j$$
$$\beta_{0ij} = 0.253(0.155) + u_{0j} + e_{0ij}$$
$$\begin{bmatrix} u_{0j} \end{bmatrix} \sim \text{N}(0,\ \Omega_u) \ :\ \Omega_u = \begin{bmatrix} 0.282(0.027) \end{bmatrix}$$
$$\begin{bmatrix} e_{0ij} \end{bmatrix} \sim \text{N}(0,\ \Omega_e) \ :\ \Omega_e = \begin{bmatrix} 0.386(0.007) \end{bmatrix}$$

*-2*loglikelihood(IGLS Deviance) : not available*(5033 of 7394 cases in use)

Figure 5: Screen shots of **MLwiN** *Equations* window. Top: Fitting the model of interest (1) to the complete cases (using restricted maximum likelihood); bottom results of multiple imputation. See also columns 2, 3 of Table 3.

the MCMC sampler, 20 imputations could be created at iterations 500, 1000, 1500, ..., 10,000 (see the start of Section 4 for a discussion on the number of imputations). Within the `Impute` dialogue, click on the icon to specify the file name for the imputed data. Figure 4 shows this dialogue for our imputation for the class size data.

To start **REALCOM-IMPUTE** , return to the left hand dialogue and click on `Start MCMC Run`

4. When **REALCOM-IMPUTE** has finished, return to **MLwiN** and, from the toolbar, select `Model -> Imputation -> Retrieve Imputation`. This prompts for the name of the file of imputed data created by **REALCOM-IMPUTE**.

5. The final step is to fit the model of interest to each imputed dataset. To do this, from the **MLwiN** toolbar select `Model -> Imputation-> Start Analysis`. **MLwiN** then fits the model of interest to each imputed data set, combines the results using Rubin's rules, and displays them in the `Equations` window (in blue).

The bottom panel of Figure 5 shows the results of multiple imputation, and is discussed in more detail in Section 4.

3.2. Additional REALCOM-IMPUTE commands

The above commands are sufficient to use **REALCOM-IMPUTE** with **MLwiN** for multiple

imputation. However, there are a number of additional commands which can be called through the buttons shown in the left panel of Figure 3. We now briefly describe these; more detail and examples are given in the online manual available with the software.

The `Level-2 identifier` buttons allow us to change the variable which identifies level 2 units. The `Responses` buttons allow us to add/remove responses and specify whether they are continuous, ordinal or unordered categorical. The `Explanatory variables` buttons allow adding/removing explanatory variables and adding/removing random coefficients at level 2; we can also fix a β coefficient to be zero if desired. Lastly, the `Results Save/Load` display the parameter estimates for the imputation model, and allow the specified model to be saved and subsequently retrieved.

4. Application to class size data

Here we describe the results of multiple imputation using the **REALCOM-IMPUTE** software. We performed multiple imputation using the 7394 partially observed cases.

We have already noted that the results of fitting the model of interest (1) to the 5033 complete cases are shown in the Table 3. We followed the steps in Subsection 3.1, and took literacy score at the end of the first year as an auxiliary variable. Using a burn in of 2000 and 500 further updates between each of 20 imputations, multiple imputation gave the parameter estimates in the righthand column of Table 3. For a practical discussion of the number of imputations, as well as issues to consider in building imputation models, see Spratt, Sterne, Tilling, Carpenter, and Carlin (2010).

Relative to the complete case analysis, after multiple imputation we see similar estimated coefficients for pre-school numeracy and literacy (β_1, β_2). However, the effect of class size is weaker after multiple imputation. Classes over 25 have a lower post reception maths score on average, but this does not quite reach significance at the 5% level (β_3, β_4).

5. Discussion and conclusion

We have described standalone **REALCOM-IMPUTE** software for performing multiple imputation on 2-level data, and illustrated its use on data from a study of the effect of class size on achievement among children in their first year at school.

Assuming the data are missing at random given the variables in the model, we find the effect of class size on post-reception maths scores, adjusted for pre-reception literacy and numeracy, is slightly weaker than the complete cases analysis suggests: after multiple imputation classes over 25 have a lower post reception maths score on average, but this does not quite reach significance at the 5% level.

As described above, our software properly models partially observed data at level 1 and level 2, and can handle binary, discrete and ordinal data at both levels. Further details of the underlying model and estimation, together with a confirmatory simulation study, are given in Goldstein *et al.* (2009).

Our approach relies on joint modelling rather than full conditional specification. We have adopted this approach for several reasons. First, we believe that explicitly modelling the multilevel structure is a natural way to handle missing observations at level 2, congenial with

the model of interest we wish to fit to the data. Indeed, the practicality of a full conditional approach in a non-trivial multilevel setting is unclear. Within this framework, the latent normal model provides a natural extension to binary, ordinal and categorical data; this is also implemented in **REALCOM-IMPUTE**. The two-level joint model we have implemented could thus be naturally extended—for example to allow for weighting at each level, to allow for more levels in the hierarchy, and to allow for cross classified data. Second, we believe the joint model is more robust to potentially redundant variables in the imputation model. This problem is most acute with binary and categorical variables, and the chance of it occurring increases with the size of the data set. In other words, we believe this approach is more likely to scale up to routine use by data analysts on large datasets.

Multiple imputation involves the specification of a joint model for the data, either implicitly or explicitly. Our software uses the idea behind the unique `Equations` window in **MLwiN** to accessibly display this information. We believe our imputation model is extremely flexible, and one of few approaches that allows for complex multilevel variance structure and missing data in level 2 variables. However, a flexible imputation model alone is not sufficient; for multiple imputation point and interval estimates to be (first-order) unbiased the joint distribution of the full set of study variables needs to be congenial with the conditional model of interest (Meng 1994). In particular, as discussed in Section 3 conditional relationships among responses in our imputation model are linear and thus will be uncongenial with models of interest which include non-linear relationships. However, if the dependent variable in a non-linear relationship is (almost) fully observed, it can be included as a covariate in the imputation model. Another possible source of uncongeniality is that the model of interest is a logistic regression, whereas the **REALCOM-IMPUTE** model uses probit regression. However, as the probit and logit links are close when probabilities are away from 0 and 1, we believe this is unlikely to be an issue in practice. Multiple imputation puts the onus on the analyst to devise an appropriate imputation model—it is therefore thought-intensive as well as computer-intensive.

The software handles generalized linear models (including negative binomial) as the model of interest, as well as multivariate response models of interest. Currently, the speed of the matlab code is a limitation, but we are working to address this. We are also working on an extension to handle simple non-linear relationships and interactions.

In conclusion, we have described software for multiple multilevel imputation. Previously, we have shown that respecting the multilevel structure in the imputation is important to avoid biasing parameter estimates (Carpenter and Goldstein 2005). This is particularly the case if data are unbalanced, a typical feature of educational data. Our software is standalone, but designed to interface easily with **MLwiN**. Code for interfacing with R (R Development Core Team 2011) is in preparation; an interface from Stata is available from `http://www.missingdata.org.uk/`. We have demonstrated the use of our software on data from a study on the effect of class size on children in their first year at school, showing that it can naturally handle missing data at level 2. We encourage readers to download the software and explore it themselves.

Acknowledgments

James Carpenter is funded by ESRC research fellowship RES-063-27-0257.

REALCOM-IMPUTE was initially developed under ESRC research grant RES-000-23-0140.

We are indebted to Jon Rasbash and Christopher Charlton who between them designed the **MLwiN↔REALCOM-IMPUTE** menu driven interface. Thanks to Jonathan Bartlett who wrote the code for exporting data from Stata to **REALCOM-IMPUTE** and back, supported by ESRC grant RES-189-25-0103.

We thank the referees and editor, whose comments have substantially improved the manuscript.

References

Aitchison J, Bennett JA (1970). "Polychotomous Quantal Response by Maximum Indicant." *Biometrika*, **57**, 253–262.

Bernaards CA, Belin TR, Schafer JL (2007). "Robustness of a Multivariate Normal Approximation for Imputation of Incomplete Binary Data." *Statistics in Medicine*, **26**(6), 1368–1382.

Blatchford P, Goldstein H, Martin C, Browne W (2002). "A Study of Class Size Effects in English School Reception Year Classes." *British Educational Research Journal*, **28**, 169–185.

Carpenter J, Goldstein H (2005). "Multiple Imputation in **MLwiN**." *Multilevel Modelling Newsletter*, **16**, 9–18.

Centre for Multilevel Modelling (2011). ***MLwiN**: Resease 2.23*. University of Bristol, Bristol, UK. URL http://www.bristol.ac.uk/cmm/.

Gilks WR, Richardson S, Spiegelhalter DJ (1996). *Markov Chain Monte-Carlo in Practice*. Chapman and Hall, London.

Goldstein H, Carpenter JR, Kenward MG, Levin K (2009). "Multilevel Models with Multivariate Mixed Response Types." *Statistical Modelling*, **9**, 173–197.

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

Klebanoff MA, Cole SR (2008). "Use of Multiple Imputation in the Epidemiologic Literature." *American Journal of Epidemiology*, **168**, 355–357.

Lee KJ, Carlin JB (2010). "Multiple Imputation for Missing Data: Fully Conditional Specification versus Multivariate Normal Imputation." *American Journal of Epidemiology*, **171**, 624–632.

Meng XL (1994). "Multiple Imputation Inferences with Uncongenial Sources of Input." *Statistical Science*, **10**, 538–573.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

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

Spratt M, Sterne JAC, Tilling K, Carpenter JR, Carlin JB (2010). "Strategies for Multiple Imputation in Longitudinal Studies." *American Journal of Epidemiology*, **172**, 478–487.

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*, **339**, 157–160.

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

Affiliation:

James Carpenter
Department of Medical Statistics
London School of Hygiene & Tropical Medicine
Keppel Street
London, WC1E 7HT, United Kingdom
E-mail: James.Carpenter@lshtm.ac.uk
URL: http://www.missingdata.org.uk/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 45, Issue 5 *Submitted:* 2009-09-23
December 2011 *Accepted:* 2011-06-30