

REALCOM-IMPUTE: multiple imputation using MLwiN. Modified September 2014

by

Harvey Goldstein,

Centre for Multilevel Modelling, University of Bristol

This description is divided into two sections. In the first we show how MLwiN can be used in conjunction with the REALCOM macros (www.cmm.bristol.ac.uk/REALCOM) to carry out multiple imputation for continuous or discrete data at up to 2 levels. In the second section we describe how MLwiN can be used in conjunction with the REALCOM macros to carry out a wide range of general multilevel multivariate response models that can be specified straightforwardly from within MLwiN. In both cases MLwiN commands and a menu system have been developed to make the analyses as straightforward as possible.

1. Imputation

Multiple imputation (MI) is an efficient and very general procedure for handling missing data in either response or predictor variables in generalised linear models and in the present case extended to multilevel models. A detailed description of MI can be found at www.missingdata.org.uk which also has a set of MLwiN macros for carrying out multiple imputation. These macros, however, have certain limitations.

- In general they only apply where all the variables containing missing values have normal distributions
- They cannot be used where a higher level variable has missing data

As part of the Bristol REALCOM project it is now possible to overcome these limitations. Specially written REALCOM macros for handling multivariate models with responses at two levels are described in Chapter 3 of the training materials for REALCOM available at www.cmm.bristol.ac.uk/REALCOM. The theory behind the use of these for multiple imputation is described and is also discussed in Goldstein et al. (2009). The advantage of the procedure over existing procedures is that it deals properly with categorical as well as normal data and also with multilevel structures. The following describes how such a full imputation procedure can be implemented straightforwardly using MLwiN in conjunction with REALCOM-IMPUTE.

Currently, only 2-level hierarchical data can be handled, although in some cases it will be possible to substitute fixed for random effects. Thus, in a 3-level structure a fixed (dummy) variable for each level 3 unit could be used and a similar procedure for a cross classified model.

We should emphasise that the MLwiN team would welcome any feedback and will do their best to respond to queries. The MLwiN team cannot be held responsible for any consequences arising from the use of these procedures as they have been implemented.

The procedure has three stages. The first stage is to set up a model in MLwiN where some of the variables have missing data. This can be done using commands or a menu. The second stage is to run REALCOM-IMPUTE which is a special version of REALCOM. The third stage

is to run MLwiN using the results output from REALCOM-IMPUTE to produce the model estimates.

Stage 1

The model of interest (MOI) should be set up in MLwiN. If you are using MLwiN 2.11 or earlier you will not see the option for specifying weights (see later).

Note that prior to MLwiN version 2.19 you will obtain an error if you attempt to use a multicategory variable containing missing values as a response or predictor. To avoid this do the following:

Set up the imputation specification *before* setting up the model of interest in MLwiN. Then set up the model of interest in MLwiN, but changing all missing values in such a variable to a legitimate code before declaring the variable in the model. Since the MOI is only run with the imputed datasets this will not cause problems.

Versions 2.19 onwards have fixed this problem.

Using the menu

Click on **imputation** on the **model** menu and the **save imputation specification** and you will see the following screen:

Number of response variables	1	Level 1 weights column	[none]
Number of auxiliary variables	1	Level 2 weights column	[none]
Level 2 identifier			

Variables to be used as responses in the imputation		Auxiliary variables (include constant term)
Column	Type	Column

Done

Enter number of responses and under variables to be used as responses enter all the variables that have any missing data. You can also enter variables with no missing values but these can also be entered as auxiliary variables and that will often be preferable. By each response enter the variable type.

Enter number of auxiliary variables. Under auxiliary variables enter all the remaining variables (these must have no missing data) that you wish to use in the imputation model in REALCOM to assist in the imputation (prediction) of the missing data. You can enter variables that are not in the MOI.

Enter the column numbers for the level 1 and level 2 weights – these do not need to be standardised. If these are left at the default value **[none]** equal weights will be used. There should be no missing values in these columns. For a description of weighting see the MLwiN help topic.

Note that you can stretch the boxes to incorporate long names.

When you click **done** you will be asked for the (full path) name of the file that is read in by REALCOM

NOTE that the constant term **must** be included as an auxiliary variable (no missing values) and called 'CONS'

Note that if you wish to carry out a single level imputation you will still need to enter a column for the level 2 ID, but will not be used if you remove it on the REALCOM-IMPUTE menu (see below).

Using commands

The following three commands can also be used.

IMPU C1 C2 Cn Type1 Type2....Typen Where C1...Cn are variables with missing data to be imputed, Type1...Typen are data types (as in REALCOM)

AUXI C1 C2....Cn

C1 is the level 2 ID, C2 is 'CONS' C3...Cn are any auxiliary (fully observed) variables.

NOTE that the constant term **must** be included as an auxiliary variable (no missing values) and called 'CONS'. If using the AUXI command rather than the menu note also that the first auxiliary variable must be the level 2 ID.

See later for a description of the command for weights.

NOTE: level 2 variables must follow all level 1 variables. The types are: 1=normal, 2=unordered categorical, 3=ordered categorical. *Note that binary variables must be coded 2.* Note also that any set of integer codes for categorical variables can be used.

ISAVE " path where file for input to MATLAB is to be saved ". End file name with .txt

REALCOM-IMPUTE

You will need to install this program by running the file Realcom-impute.msi which can be downloaded from the REALCOM web site www.cmm.bristol.ac.uk/research/realcom/index.shtml . You will need to have installed the matlab runtime installer as described.

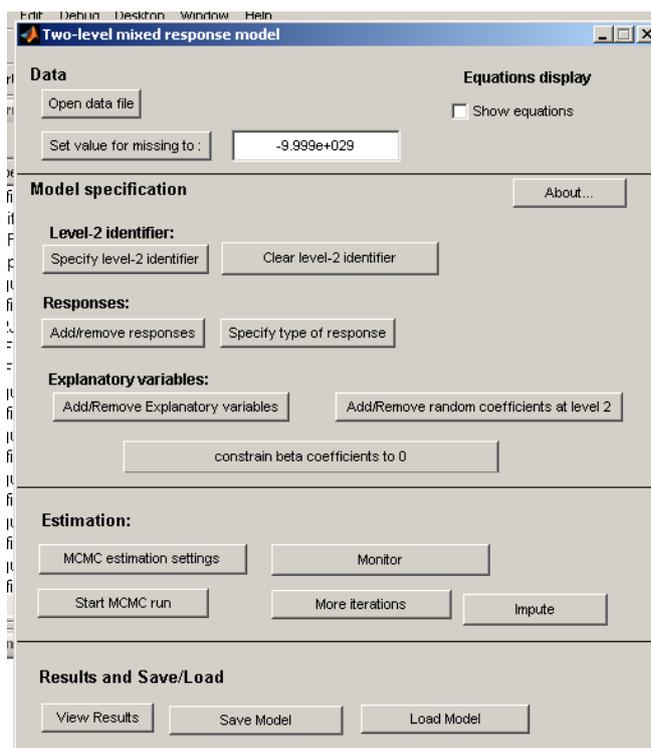
Start the program and you will see the GUI screen (see below)

Alternatively, if MATLAB is available and source macros are available, you can do the following:

In directory where MATLAB files are located start MATLAB by double clicking on "gui_2levmixresp.m"

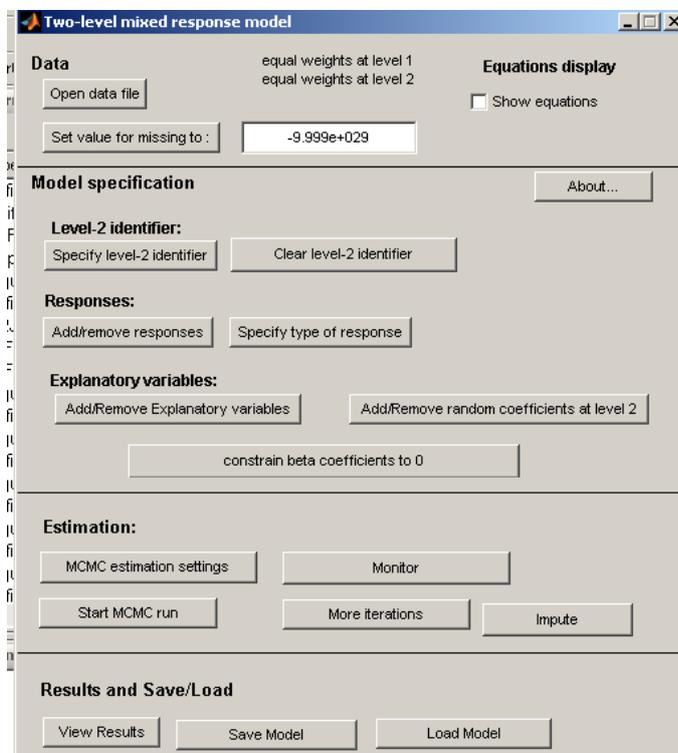
This will open MATLAB and you will be presented with the matlab file "gui_2levmixresp.m" – press F5 to run the GUI.

The GUI screen will appear as follows (Note that there is now an extra button allowing you to remove the level 2 identifier if you wish to run a single level imputation).



Now, in the GUI click on ‘open data file’ and enter (browse for) the input file specified above.

The following screen will now appear with a description of the weights – in this case equal weights. Note that the possibility of differential weights is not currently available – see section below.



In the GUI you can view the equations. Then you should specify the MCMC estimation settings and then click on ‘impute’ to specify the iteration cycles where you want imputed datasets to be produced. It is suggested that you should allow 500 iterations between imputations to ensure independence, although sometimes fewer may be adequate. You will

also typically need 10+ imputed sets. Thus, if you specify 10 sets and a gap of 500, you will need 4501 total iterations assuming you designate the first iteration after the burn in for imputing. You should browse to specify a directory for these – best to make it a subdirectory of the one where the data are. Click ‘start MCMC run’ and wait till it has finished (you can monitor one or more parameters).

By default, the first time the ‘Impute’ window is opened, a set of ten values will be displayed, being equally spaced among the iterations you have specified. You can change these and the changes will be remembered for the current run of the software. You can also revert to the default set by clicking on ‘Defaults’.

It is suggested that you do monitor one or two parameters so that you can see when iterations have finished. If you have specified a categorical variable it is only necessary that the category values have integer codes; REALCOM-IMPUTE will detect these codes in the data and carry out the appropriate imputations.

Now go back to MLwiN and make sure you have loaded up the worksheet that contains the data (including missing data) and the MOI. You may now use the menu or commands.

Menu

Select the menu as before but now select **retrieve imputation** and you will be asked for the folder which contains the file ‘impvals.txt’ containing all the imputed datasets as in the IRETR command below (which is more flexible, allowing you to select the data sets to be used).

Select the menu as before but now select **Start analysis** to run the model with the imputed datasets with results displayed in the equations window.

Commands

Type:

IRETR “path+filename” where impuses are stored. The actual filename you will need (it contains details of number of imputations etc) is ‘impvals.txt’

Type ISTAR n_1, n_2, \dots, n_k

This will use completed data sets n_1, n_2, \dots, n_k listed in ‘impvals.txt’ where numbering starts at 0. Thus e.g. “ISTAR 0 2 4” will use data sets 1,3 and 5. If no numbers are specified all the completed sets will be used.

The results will be computed and displayed in the equations window. Note that you may need to refresh the equations window by clicking **estimates** a few times.

General

If you are fitting a multivariate model then you need to have set this up using the multivariate screen. When specifying the variables to be imputed and the auxiliary variables, use the original variables from which the multivariate variables have been created.

If there are interactions in the model, use the interactions menu to specify these and when specifying the variables to be imputed and the auxiliary variables include only the basic variables and not the created interaction variables. If there are polynomial terms these should be created using the interactions menu.

For both interactions and with multivariate data, mlwin will create the necessary variables when retrieving the completed data sets and fitting the model of interest.

For each pair of unordered categorical variables you should ensure that there are no empty cells in their cross-tabulation. If this happens the level 2 covariance matrix may become ill conditioned since the corresponding covariance is unidentified. The software will detect this, output a message, and will stop with current values after this has occurred at 10 iterations. In such cases you will need to group categories. In addition, if there are cells with very small numbers of level 1 units the level 2 covariance matrix can become ill-conditioned and a similar warning will be issued. Again, you will need to consider grouping categories. Automatic procedures for dealing with such situations will be incorporated in future versions.

File structures

The file produced by MLwiN for REALCOM has the following format:

Number of response variables (NR)

Number of auxiliary variables; first auxiliary variable is level 2 ID, second is CONS

Type of response; NR values

Variable names, responses followed by auxiliary variables

Data records with variables in the above order.

For example, for 3 responses with 4 auxiliary (including level 2 ID and CONS) we might have the following:

```

3
4
1      1      2
EXAMVR  standlrt  girl  school  cons  VR2  VR3
0.26132 -9.999e+029      1      1      1      0      0
0.13407 -9.999e+029      1      1      1      1      0
-1.7239 -9.999e+029      0      1      1      0      1
0.96759 0.2058      1      1      1      1      0
0.54434 0.3711      1      1      1      1      0
1.7349 -9.999e+029      0      1      1      0      0
1.0396 -9.999e+029      0      1      1      0      1
-9.999e+029 -1.034      0      1      1      1      0
-9.999e+029 -9.999e+029      1      1      1      1      0
-1.2195 -1.4472      0      1      1      0      1
.....

```

The file impvals.txt produced by REALCOM has the following format:

Variable names for responses

Number of level 1 records for each data set (N1)

Number of imputed data sets (NI)

NI names one for each imputed data set. The names have the format imputeiterx, where x is the iteration number at which the imputation is made.

Thus, for the above example, if we have 10 imputations at 1000, ...10000 and the number of level 1 units is 4059, we would have:

```
EXAMVVR      standlrt  girl
4059
10
imputeiter1000  imputeiter2000  imputeiter3000  imputeiter4000  imputeiter5000  imputeiter6000
                imputeiter7000  imputeiter8000  imputeiter9000  imputeiter10000
```

Other software packages can be used to produce these files if required. A series of commands for STATA has been written – see website for details.

NOTE that in the REALCOM input and output files all file name lists should have tab separators between file names.

An example

We illustrate the procedure using a subset of the tutorial data set.

Open MLwiN and load this data set. We shall use just the first 15 schools, so type the command

```
CHOOSE 1 15 c1 C2-C10 C1 C2-C10
```

Now set up a model with NORMEXAM as response, with CONS, STANDLRT, GIRL and VRBAND (a 3-category variable) as predictors with SCHOOL as level 2 ID and STUDENT as level 1 ID. Run this model and you will obtain the following results:

$$\text{normexam}_{ij} \sim N(\mathcal{X}B, \Omega)$$

$$\text{normexam}_{ij} = \beta_{0ij}\text{cons} + 0.485(0.040)\text{standlrt}_{ij} + 0.222(0.057)\text{girl}_{ij} + -0.236(0.070)\text{vb2}_{ij} + -0.338(0.136)\text{vb3}_{ij}$$

$$\beta_{0ij} = 0.168(0.097) + u_{0ij} + e_{0ij}$$

$$\begin{bmatrix} u_{0ij} \end{bmatrix} \sim N(0, \Omega_u) : \Omega_u = \begin{bmatrix} 0.074(0.030) \end{bmatrix}$$

$$\begin{bmatrix} e_{0ij} \end{bmatrix} \sim N(0, \Omega_e) : \Omega_e = \begin{bmatrix} 0.593(0.025) \end{bmatrix}$$

-2*loglikelihood(IGLS Deviance) = 2604.176(1110 of 1110 cases in use)

We are now going to introduce missing values randomly chosen. Type the following commands:

```
URAN 1110 C20
Calc C21=(C20>0.25)
Chan 0 C21 missing C21
Calc C3=C21*C3
Note this makes random 25% NORMEXAM values missing
CALC C21=(C20<=0.2 | C20>=0.3)
CHAN 0 C21 missing C21
```

```
CALC C5=C5*C21
```

Note this makes 10% LRT missing with 5% records missing both

```
CALC C21=(C20<=0.15 | C20>=0.25)
```

```
CHAN 0 C21 missing C21
```

```
CALC C6=C6*C21
```

Note this makes 10% Girls with missing values and 5% of records missing all three.

If you now run the model again you will obtain a result similar to

$$\text{normexam}_{ij} \sim N(XB, \Omega)$$

$$\text{normexam}_{ij} = \beta_{0ij}\text{cons} + 0.458(0.048)\text{standlrt}_{ij} + 0.198(0.068)\text{girl}_{ij} + -0.360(0.084)\text{vb2}_{ij} + \\ -0.435(0.167)\text{vb3}_{ij}$$

$$\beta_{0ij} = 0.273(0.103) + u_{0ij} + e_{0ij}$$

$$\begin{bmatrix} u_{0ij} \end{bmatrix} \sim N(0, \Omega_u) : \Omega_u = \begin{bmatrix} 0.061(0.028) \end{bmatrix}$$

$$\begin{bmatrix} e_{0ij} \end{bmatrix} \sim N(0, \Omega_e) : \Omega_e = \begin{bmatrix} 0.608(0.031) \end{bmatrix}$$

$-2*\text{loglikelihood(IGLS Deviance)} = 1840.140(775 \text{ of } 1110 \text{ cases in use})$

Notice the reduced number of records used and the increase in the fixed part standard errors. Where any variable is missing MLwiN will by default omit this record from the analysis.

We now set up C3, C5, C6 as responses and the intercept and VRBAND as auxiliary variables. The following commands can be used or the menu

```
IMPUte C3 C5 C6 1 1 2
```

```
AUXiliary C1 C4 C10
```

```
ISAVE "D:\IMPUTE\test.txt"
```

You can choose any convenient location for the ISAVE command. Note that we are treating the binary variable 'girl' as an unordered category.

Now open REALCOM-IMPUTE and open the "test.txt" file (or whatever you specified) as data input. Select the settings as described above and start the MCMC iterations. These may be very slow so that you might like, for testing purposes, to just have a burn in of 100 and impute 5 datasets with a spacing of 100 iterations.

Once this has finished go back to MLwiN, make sure you have the original MOI set up and read in the "impvals.txt" file using the menu or the following commands

```
IRETRieve "D:\IMPUTE\impvals.txt"
```

Note we assume that we have specified the output to go into "D:\IMPUTE"

```
ISTAR
```

The following output (or very similar values) will be displayed once iterations have completed. The above REALCOM-IMPUTE settings have been used.

$\text{normexam}_{ij} \sim N(\lambda B, \Omega)$

$\text{normexam}_{ij} = \beta_{0ij}\text{cons} + 0.443(0.043)\text{standlrt}_{ij} + 0.189(0.069)\text{girl}_{ij} + -0.352(0.082)\text{vb2}_{ij} + -0.525(0.147)\text{vb3}_{ij}$

$\beta_{0ij} = 0.286(0.104) + u_{0ij} + e_{0ij}$

$\begin{bmatrix} u_{0ij} \end{bmatrix} \sim N(0, \Omega_u) : \Omega_u = \begin{bmatrix} 0.063(0.028) \end{bmatrix}$

$\begin{bmatrix} e_{0ij} \end{bmatrix} \sim N(0, \Omega_e) : \Omega_e = \begin{bmatrix} 0.619(0.028) \end{bmatrix}$

Notice that the standard errors for the fixed part parameters are mostly intermediate between the full data set and the listwise deleted data set. Note also that you will not have a loglikelihood value since this is not defined for the combined data. For inference you can use the standard errors given. For inferences on linear functions of parameters, for example if you have a multicategory variable such as verbal reasoning band, you can use the ‘intervals and tests’ menu as with standard models.

Fitting multivariate 2-level models with mixed response types at one or both levels.

The REALCOM macros allow the fitting of such models and a full description is given on the REALCOM web site.

The commands and menus described above can also be used to fit these models straightforwardly from within MLwiN in conjunction with REALCOM-IMPUTE.

Suppose we have three responses at level 1 or level 2, y_1, y_2, y_3 and two predictors including the intercept, x_0, x_1 . We can use the above commands or menus to define the responses and their data types and then define the level 2 ID and x_0, x_1 as auxiliary variables. The responses may or may not have missing values. The REALCOM file will then be set up automatically with the required model and the user will just need to enter the MCMC estimation settings and decide what to monitor. In addition, if required, constraints can be imposed and further level 2 random effects defined for any level 1 responses. There will be no need to enter any IMPUTE specifications if the purpose is simply to fit the REALCOM model. Once iterations are finished the results can be viewed from the REALCOM screen.

Using sampling weights

In MLwiN there is currently a facility to specify weights attached to units at any level. See the help item in MLwiN for details. Research is currently underway into the possible use of such weights when making imputations. Preliminary results suggest that weights are not needed at the imputation stage and REALCOM-IMPUTE does not at the moment allow the use of differential weights.

Hypothesis testing

When MLwiN combines the imputed datasets it produces a combined estimate of the full covariance matrices of the fixed and random parameters (in C1099 and C1097). You can therefore use the interval and tests menu in MLwiN to estimate confidence intervals for linear functions of parameters and to test hypotheses involving several parameters.

MCMC

Currently, the fitting and combination of completed datasets within MLwiN cannot be done when using MCMC estimation. To do this the user will need to combine datasets manually.

STATJR

This new software will now carry out the same set of analyses as REALCOM-IMPUTE and is considerably faster. Details are available at <http://www.bris.ac.uk/cmm/software/statjr>

References

Goldstein, H., Carpenter, J., Kenward, M. and Levin, K. (2009). "Multilevel Models with multivariate mixed response types." *Statistical Modelling*. 9(3): 173–197.

Carpenter, J., Goldstein, H. and Kenward, M. G.. (2011). REALCOM-IMPUTE software for multilevel multiple imputation with mixed response types. *J. Statistical Software*, 45,5, 1-14.

Troubleshooting

You may find that the results you get look strange, or there may be warning messages displayed on the MATLAB command interface screen or the program may crash. The following are some things to check.

1. Make sure that there are no missing values in any of the auxiliary (explanatory) variables or in the weights.
2. Make sure that there are no linear dependencies among any of the variables
3. Check that the data are ordered within level 2 units
4. Check that the level 2 responses follow the level 1 responses
5. You may not have allowed enough IGLS iterations, especially for generalised linear models. Change this using the command: `MAXIt n`. We suggest $n=50$ is safe.
6. If all level responses are missing for a record then this should ideally be removed from the imputation since it will add noise, unless you also have an auxiliary that is reasonably highly correlated with one or more responses.
7. If you are adding a level 2 identifier to fit a 2-level model make sure you specify one (e.g. CONS) or more random coefficients at level 2.
8. Make sure that binary variables are coded as 'unordered categorical'.
9. Ensure that there are enough level 2 units for the number of responses in your model. For example, if you have 6 responses each with a level 2 random intercept, this implies 21 variances and covariances at level 2. If you only have 20 level 2 units, REALCOM will not be able to fit the model. You should, as a rule of thumb, aim for at least three times as many level 2 units as there are level 2 parameters.