The Stata Journal (2003)
3, Number 3, pp. 1–20

Tools for analyzing multiple imputed datasets

John B. Carlin, Ning Li, Philip Greenwood, and Carolyn Coffey
Clinical Epidemiology and Biostatistics Unit
Murdoch Children's Research Institute and
University of Melbourne Department of Paediatrics
Royal Children's Hospital, Parkville, Victoria 3052, Australia

Abstract. The method of multiple imputation (MI) is used increasingly for analyzing datasets with missing observations. Two sets of tasks are required in order to implement the method: (a) generating multiple complete datasets in which missing values have been imputed by simulating from an appropriate probability distribution and (b) analyzing the multiple imputed datasets and combining complete data inferences from them to form an overall inference for parameters of interest. An increasing number of software tools are available for task (a), although this is difficult to automate, because the method of imputation should depend on the context and available covariate data. When the quantity of missing data is not great, the sensitivity of results to the imputation model may be relatively low. In this context, software tools that enable task (b) to be performed with similar ease to the analysis of a single dataset should facilitate the wider use of multiple imputation. Such tools need not only to implement techniques for inference from multiple imputed datasets but also to allow standard manipulations such as transformation and recoding of variables. In this article, we describe a set of Stata commands that we have developed for manipulating and analyzing multiple datasets.

Keywords: st0000, missing data, multiple imputation, Rubin's rule of combination, overall estimates

1 Introduction

The presence of missing data is a frequent source of difficulties in statistical practice. In large surveys, subjects may decline to participate or fail to respond to particular items in a questionnaire. The situation may be worse in longitudinal studies where investigators seek participation from individuals over several occasions—subjects may decline to respond to certain questions at each occasion ("item missingness") or fail to participate entirely at some occasions ("occasion missingness").

A simple way to deal with incompletely observed data is to omit any case that has a missing value for any variables required in the analysis of interest. This so-called complete-case analysis may, however, produce biased inferences when the subgroup with complete data differs systematically from the target population. It may also lead to inefficient use of the collected information, since the number of individuals with complete data may be substantially less than the total sample size.

© 2003 Stata Corporation

A slightly more sophisticated approach is to use single imputation, where missing values in a variable are simply filled in by a plausible estimate, such as the mean or median for that variable on other participants. Better estimates may be obtained by using the predicted means from a regression model or a hotdeck procedure (Little and Rubin 1987). Relative to the complete-case approach, single imputation has the advantage of retaining the full sample size, and in some situations, it can produce unbiased estimates of target parameters. It cannot, however, provide valid standard errors and confidence intervals, as it ignores the uncertainty implicit in the fact that the imputed values are not the actual values.

Under certain assumptions, valid inferences can, however, be obtained with the more sophisticated technique of multiple imputation introduced by Rubin (1987). The heuristic basis of this approach is that if several different complete datasets (rather than just one) are obtained by imputing missing values, then appropriate account can be taken of the uncertainty involved in imputing the missing values by examining the variation between inferences obtained in each of the completed datasets. In practice, the statistician (not necessarily the same analyst who will make final substantive analyses of the data) produces several complete datasets using plausible modeling assumptions. The data analyst then analyzes each of these using standard complete-data methods, and then combines the results according to appropriate rules to produce overall estimates with confidence intervals and p-values. The overall estimates incorporate the missing-data uncertainty as well as sampling variation.

The key assumption with the method of multiple imputation, as with most approaches to missing data problems, is that the missing data are *missing at random* (MAR) in the sense that the probability of a value being missing may only depend on observed data for the individual in question and not on the unobserved missing values. The original formal definition of MAR was provided by Rubin (1976); Little and Rubin (1987) and Schafer (1997) provide more recent discussions. An essential requirement for the MI method to work in practice is that imputation should be performed under a model that is general enough to make the MAR assumption defensible, even if this model uses variables that are not of substantive interest for later data analysis.

Generating proper imputations for missing values in a given dataset is not straightforward. The general principle is that missing values should be imputed by simulation from their posterior predictive distribution (PPD) under a plausible model fitted to the observed data. For general model specifications, this in principle requires Markov Chain Monte Carlo (MCMC) methods, and Schafer (1997) has developed a set of reasonably flexible software that covers a range of models. Imputation under a multivariate normal model for all variables (both those to be imputed and others used because of their association with the variable requiring imputation) is also now available in the S-PLUS and SAS packages. Another widely available tool is the SOLAS package (Statistical Solutions Ltd.), which implements an imputation method that provides considerable flexibility. It gives proper imputations when the pattern of missingness is monotone (variables can be arranged in order from those with least missing values to those with most) and otherwise employs pragmatic methods to force the data into a monotone missingness pattern. Raghunathan et al. (2001) have recently proposed another more flexible ap-

proach using conditional distributions and a SAS program is available to implement this. Finally, for small quantities of missing data, the approximate Bayesian bootstrap method implemented as the `hotdeck` command by Mander and Clayton (1999) may be adequate.

When you implement MI, it is usually sufficient to obtain a relatively small number of imputed datasets, often as few as 3 or 5, because the relative gains in precision from using larger numbers are usually minor, unless the fraction of missing data is extremely large (in which case the MAR assumption becomes increasingly tenuous in any case). See Rubin (1987) for details.

2 Statistical inference from multiple imputed datasets

The Stata commands presented in this article have been written to provide a large number of useful data manipulations and transformations, enabling data analysts to have maximum flexibility in their analyses. Ultimately, the analyst will need to perform inferences using the multiple datasets, so we have implemented the simple method derived by Rubin (1987).

Suppose that initially our primary interest lies in a scalar estimand Q. In a typical case, this might be a regression coefficient, for example, the log odds ratio in a logistic regression. Suppose that we have imputed m complete datasets using an appropriate model. In each of these datasets, we use standard complete-data methods to obtain an estimate of Q with an associated estimated variance (or equivalently, standard error). Let $\widehat{Q}^{(k)}$ and $U^{(k)}$ denote the point estimate and variance respectively from the kth $(k = 1, 2, \ldots, m)$ dataset.

As might be expected, the multiple imputation point estimate of Q is the average of the m complete-data estimates:

$$\overline{Q} = \frac{1}{m} \sum_{t=1}^{m} \widehat{Q}^{(t)}$$

Obtaining a valid standard error for this estimate requires combining information on within-imputation and between-imputation variation. The latter is important in reflecting variability due to imputation uncertainty.

First, a within-imputation variance component is obtained as the average of the complete-data variance estimates:

$$W = \frac{1}{m} \sum_{t=1}^{m} U^{(t)}$$

Second, between-imputation variance is calculated by a simple empirical combination of the complete-data point estimates:

$$B = \frac{1}{m-1}\sum_{t=1}^{m}(\widehat{Q}^{(t)} - \overline{Q})^2$$

The total variance in the combined estimate of Q is then given by

$$T = W + \left(1 + \frac{1}{m}\right)B \tag{1}$$

and Rubin (1987) shows that, approximately,

$$T^{-1/2}(Q - \overline{Q}) \sim t_\gamma$$

where the degrees of freedom γ are given by

$$\gamma = (m-1)\left\{1 + \frac{W}{(1+1/m)B}\right\}^2 \tag{2}$$

In the usual way, a $100(1-\alpha)\%$ interval estimate for Q is

$$\overline{Q} \pm t_{\gamma,1-\alpha/2}\sqrt{T}$$

The $(1 + 1/m)$ term in these simple equations indicates why it is not necessary to a create large number of imputed datasets, especially if there is a low ratio of between-imputation to within-imputation variance. The latter is typically the case unless there is a large fraction of missing data, along with relatively precise estimation of parameters within each dataset.

On occasion, it is of interest to perform inference for a multidimensional (vector) quantity, for example, when assessing a batch of effects related to a multi-category factor or examining interaction terms in a regression model. The simple methods described above for scalar estimands do not generalize immediately, but we have implemented an approximate method given by Li, Raghunathan, and Rubin (1991) to give a p-value for the null hypothesis that all components of a vector Q are equal to zero. Schafer (1997) cautions that for the procedure to work well, you should have a large sample and make sure that the scale in which Q is expressed is such that the usual normal-theory inferences for complete data are valid.

3 Syntax

`miset` [`using` *filename-prefix*] [`, mimps(#) clear`]

`mireset`

`miappend using` *filename-prefix* [`, nolabel keep(`*varlist*`)`]

`mimerge` [`varlist`] `using` *filename-prefix* [`, keep(`*varlist*`) unique uniqmaster uniqusing nolabel nokeep _merge(`*varname*`)`]

`misave` *filename-prefix* [`, replace`]

`mido` *Stata-command*

`mici` [`, indiv`] : *varlist* [`in` *range*] [`if` *exp*] [`, level(`#`) binomial poisson exposure(`*varname*`)`]

`mifit` [`, indiv`] : [`xi :`] *estimation command*

`milincom` [`, indiv`] : *exp* [`in` *range*] [`if` *exp*] [`, level(`#`) or irr eform`]

`mitestparm` *varlist*

where *estimation command* may be `regress`, `logit`, `probit`, `clogit`, `glm`, `logistic`, `poisson`, `svyreg`, `svylogit`, `svyprobit`, `svypoisson`, `xtgee`, or `xtreg` followed by standard arguments and options for that command.

where *exp* is any linear combination of coefficients that is valid syntax for `test`. Note that *exp* must not contain an equal sign (see [R] **lincom**).

4 Description

`miset` creates temporary copies of imputed datasets so that subsequent `mi` (multiple imputation) commands can be executed on these data. The other `mi` commands can only be used after the multiple datasets have been declared by `miset`. The imputed datasets are assumed to be created from an original dataset by a "proper" imputation method. These datasets must have the same variables and the same number of observations.

The following naming convention must be used: filenames of the imputed datasets must consist of a common word followed by a consecutive number, followed by the normal extension `.dta`; for example, `foo1.dta`, `foo2.dta`, ..., `foo5.dta`. Only the common word or *filename-prefix* ("`foo`" in this case) is to be specified after the `using` command.

`miset` creates temporary files `_mitemp1.dta`, `_mitemp2.dta`, and so on by copying the using files. All subsequent commands will be executed on these temporary files, leaving the original using files unchanged. The temporary files remain in the working directory until `mireset` is issued. To save the updated temporary files after a series of `mi` commands, see `misave`.

`mireset` erases the temporary files created by `miset` and clears the global macros created by `miset`.

`misave` saves the multiple temporary datasets declared by the last `miset` with any subsequent changes made with the `mi` commands. File names are constructed from the *filename-prefix* followed by the numbers 1, 2, ..., m.

`mido` executes a Stata command on each of the datasets created by `miset`. Most commands can be used, except estimation and post-estimation commands. You cannot define value labels with `mido`.

For execution of estimation-class commands with a combination of results over imputed datasets, see `mifit`; for post-estimation inferences, see `milincom` and `mitestparm`; and for manipulation of multiple datasets, see `mimerge`, `miappend`, and `misave`.

`miappend` appends the multiple datasets of the using files to the currently loaded `miset` datasets. The first using file is appended to the first `miset` file, the second to the second, and so on. The number of using datasets must be the same as the number of datasets declared in `miset`.

`mimerge` merges each of the `miset` datasets with each of the using datasets. The number of using datasets must be the same as that of the master datasets.

`mici` calculates confidence intervals separately for each of the `miset` datasets and calculates overall confidence intervals by Rubin's rule of combination (see section 2).

`mifit` carries out the *estimation command* on each of the `miset` datasets and then computes multiple-imputation point estimates of the regression parameters and associated overall variance estimates using Rubin's rule. Confidence intervals for the estimated model parameters are given. The `xi` prefix command can be used in the usual way within `mifit` to create dummy variables and interaction terms.

`milincom` computes multiple-imputation point estimates, standard errors, p-values t statistics, and confidence intervals for linear combinations of regression coefficients after `mifit` using Rubin's combining rule (1). Results can be displayed optionally in exponentiated form (giving odds ratios or incidence rate ratios in appropriate circumstances).

`mitestparm` is the `mi` version of the post-estimation command `testparm`. It obtains an approximate F-test (Li, Raghunathan, and Rubin 1991) to test the hypothesis that the specified coefficients are all equal to zero, analogous to the standard Wald test.

5 Options

In general, commands inherit the standard Stata options for their parent command. Most options and prefixes are acceptable with `mido`; for example, `mido by w: tab x y, sum(z) freq` is a legal syntax. However, the parent options `update` and `replace` are not options for `mimerge` because these options take effect only when there are missing values in the master file, which should not occur with multiple imputed datasets. `mifit` accepts all options available in the *estimation command*.

5.1 Options for miset

`mimps(#)` specifies the number, m, of datasets to be used. The default is 5. A minimum of 2 and a maximum of 9 datasets can be specified. If there are more than m datasets, only the first m datasets are used. As mentioned earlier, usually as few as 3 or 5 multiple imputed datasets is sufficient.

`clear` permits the data to be loaded, even if there is a dataset already in memory and even if that dataset has changed since the data were last saved.

5.2 Options for miappend

`nolabel` prevents copying the value label definitions from the using dataset.

`keep(`*varlist*`)` specifies the variables to be kept from the using data. If `keep()` is not specified, all variables are kept.

These options are inherited from the parent command `append`.

5.3 Options for mimerge

`keep(`*varlist*`)` specifies the variables to be kept from the `using` data. If `keep()` is not specified, all variables are kept.

`unique`, `uniqmaster`, and `uniqusing` specify that the match variables in a match-merge uniquely identify the observations.

`unique` specifies that the match variables uniquely identify the observations in the master data and in the using data. For most match-merges, you should specify `unique`. `merge` does nothing differently when you specify the option, unless the assumption that you are making is false. In that case, an error message is issued and the data are not merged.

`uniqmaster` specifies that the match variables uniquely identify the observations in memory, the master data, but not necessarily the ones in the using data.

`uniqusing` specifies that the match variables uniquely identify the observations in the using data but not necessarily the ones in the master data.

`nolabel` prevents Stata from copying the value label definitions from the using dataset.

`nokeep` causes merge to ignore observations in the using data that have no corresponding observation in the master.

`_merge(`*varname*`)` specifies the name of the variable that will mark the source of the resulting observation. The default is `_merge(_merge)`.

These options are all inherited from the parent `merge` command.

5.4 Option for misave

`replace` permits `misave` to overwrite existing datasets.

5.5 Options for mici

`indiv` reports the confidence intervals from each of the individual `miset` datasets, as well as the overall confidence intervals. By default, only the overall confidence interval is displayed.

`level` specifies the desired level of significance in calculating the confidence intervals. The default is to use the current system value defined by `set level` (whose initial value is 95%).

`binomial` specifies that each of the variables in *varlist* has a binomial distribution and that `mici` will calculate the binomial standard error for each of the `miset` datasets and then calculate an overall confidence interval for the corresponding proportion using Rubin's combining rule. If any variable in *varlist* is not binary (value 0 or 1), an error message is given. Note that the "exact" binomial method is not used since it does not enable application of Rubin's rule.

`poisson` specifies that the variables are Poisson-distributed counts. Poisson standard errors are calculated from individual `miset` datasets, and overall confidence intervals are computed using Rubin's rule.

`exposure(`*varname*`)` is used only with `poisson`. *varname* contains the total exposure during which the number of events recorded in *varlist* was observed.

5.6 Options for mifit

`indiv` causes an estimation result to be output for each of the `miset` datasets. The default is that only overall estimation results are displayed.

In principle, any options of the original *estimation command* can be passed along to `mifit`.

5.7 Options for milincom

`indiv` causes `milincom` to report estimates of the linear combination obtained from each of the `miset` datasets, as well as an overall estimate obtained by the combining rule.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals; see [R] **level**.

`or`, `irr`, and `eform` all report coefficient estimates as $\exp(b)$ rather than b. Standard errors and confidence intervals are similarly transformed. `or` is the default after `logistic`. The only difference in these options is how the output is labeled; see [R] **lincom**.

5.8 Options for mitestparm

The `testparm` options `equal` and `equation` have not yet been implemented.

6 Examples

We illustrate the `mi` commands using two sets of datasets: `smiF1.dta`, ..., `smiF5.dta` and `smiM1.dta`, ..., `smiM5.dta`. The datasets were imputed separately for gender, using a multivariate linear mixed effects model from data that contained 25% to 47% missing observations across variables. The data are extracted from a cohort study of adolescent health and consist mainly of self-reported indicators of substance use and other behavioral outcomes over 6 timepoints (`wave`s), along with some fixed covariates.

First, we load the females' datasets `smiF*.dta`:

```
. miset using smiF
smiF1.dta to smiF5.dta were loaded to _mitemp1.dta to _mitemp5.dta respectively
```

The five datasets have now been set up for `mi` operation. Note that the first dataset is always the one currently in memory after any `mi` command.

(*Continued on next page*)

```
. describe
Contains data from _mitemp1.dta
  obs:           600                          Imputed females data No.1
 vars:            10                          11 Jun 2003 10:16
 size:        10,200 (99.9% of memory free)   (_dta has notes)
```

variable name	storage type	display format	value label	variable label
id	long	%9.0g		
wave	byte	%9.0g		wave 1 to wave 6
mmetro	byte	%9.0g		school in mmetro
parsmk	byte	%9.0g		either parent smokes
drkfre	byte	%6.0g	drkfre	drink frequency
alcdos	byte	%6.0g	alcdos	av units/drinking day
alcdhi	byte	%6.0g		alcday>=5 units at least once
smk	byte	%6.0g	smk	none, occasional, daily
cistot	byte	%6.0g		cis total score (0,56)
mdrkfre	byte	%9.0g	mis	missing drkfre

```
Sorted by:  id  wave
```

With a view to appending the males' data, we generate a variable `sex=1` for females

```
. mido gen byte sex=1
-> Applying gen to dataset1 (_mitemp1.dta).
-> Applying gen to dataset2 (_mitemp2.dta).
-> Applying gen to dataset3 (_mitemp3.dta).
-> Applying gen to dataset4 (_mitemp4.dta).
-> Applying gen to dataset5 (_mitemp5.dta).
```

and we save the five datasets by issuing

```
. misave temp, replace
file temp1.dta saved
file temp2.dta saved
file temp3.dta saved
file temp4.dta saved
file temp5.dta saved
```

Similarly, we set up the males' datasets and generate a variable `sex=0`.

```
. miset using smiM
smiM1.dta to smiM5.dta were loaded to _mitemp1.dta to _mitemp5.dta respectively
. mido gen byte sex=0
-> Applying gen to dataset1 (_mitemp1.dta).
-> Applying gen to dataset2 (_mitemp2.dta).
-> Applying gen to dataset3 (_mitemp3.dta).
-> Applying gen to dataset4 (_mitemp4.dta).
-> Applying gen to dataset5 (_mitemp5.dta).
```

We may then join the two datasets into a single one by appending `smiF1.dta` to `smiM1.dta`, `smiF2.dta` to `smiM2.dta`, and so on to form new datasets `both1.dta`, `both2.dat`, and so on.

```
. miappend using temp
. mido label data "male & female"
-> Applying label to dataset1 (_mitemp1.dta).
-> Applying label to dataset2 (_mitemp2.dta).
-> Applying label to dataset3 (_mitemp3.dta).
-> Applying label to dataset4 (_mitemp4.dta).
-> Applying label to dataset5 (_mitemp5.dta).
. qui mido labsex
. mido sort id wave
-> Applying sort to dataset1 (_mitemp1.dta).
-> Applying sort to dataset2 (_mitemp2.dta).
-> Applying sort to dataset3 (_mitemp3.dta).
-> Applying sort to dataset4 (_mitemp4.dta).
-> Applying sort to dataset5 (_mitemp5.dta).
. misave both, replace
file both1.dta saved
file both2.dta saved
file both3.dta saved
file both4.dta saved
file both5.dta saved
```

Note that in the above, the values of `sex` were labeled by calling a user-written program `labsex` using `mido`. `labsex` was defined by an ado-file as follows:

```
program define labsex
        version 7
        label def sexlb 0 "male" 1 "female"
        label val sex sexlb
end
```

As Stata does not differentiate between built-in and ado-defined commands, `mido` also accepts commands defined by ado-files.

Now, we base our analysis on the datasets `both`. The variable `mdrkfre` is an index equal to 1 if `drkfre` was missing in the original data (before imputation), and zero otherwise. So, `mdrkfre=0` corresponds to actual observations and `mdrkfre=1` corresponds to imputed observations. We see that there are a total of 163 observations missing in `drkfre`. These missing values are filled in differently in the five datasets, as shown in column two of the following tables.

(*Continued on next page*)

```
. miset using both
both1.dta to both5.dta were loaded to _mitemp1.dta to _mitemp5.dta respectively
. mido tab drkfre mdrkfre
-> Applying tab to dataset1 (_mitemp1.dta).
```

drink frequency	missing drkfre 0	1	Total
Non drinker	438	51	489
not in last wk	326	69	395
<3 days last wk	203	36	239
>=3 days last wk	40	7	47
Total	1,007	163	1,170

```
-> Applying tab to dataset2 (_mitemp2.dta).
```

drink frequency	missing drkfre 0	1	Total
Non drinker	438	44	482
not in last wk	326	69	395
<3 days last wk	203	38	241
>=3 days last wk	40	12	52
Total	1,007	163	1,170

```
-> Applying tab to dataset3 (_mitemp3.dta).
```

drink frequency	missing drkfre 0	1	Total
Non drinker	438	49	487
not in last wk	326	70	396
<3 days last wk	203	37	240
>=3 days last wk	40	7	47
Total	1,007	163	1,170

```
-> Applying tab to dataset4 (_mitemp4.dta).
```

drink frequency	missing drkfre 0	1	Total
Non drinker	438	49	487
not in last wk	326	68	394
<3 days last wk	203	39	242
>=3 days last wk	40	7	47
Total	1,007	163	1,170

```
-> Applying tab to dataset5 (_mitemp5.dta).
```

drink frequency	missing drkfre 0	1	Total
Non drinker	438	56	494
not in last wk	326	57	383
<3 days last wk	203	42	245
>=3 days last wk	40	8	48
Total	1,007	163	1,170

Suppose that we are interested in the risk of drinking on a regular (at least weekly) basis. We may recode the variable `drkfre` in each dataset to a binary indicator and obtain the frequency of this level of drinking at each `wave` as follows.

```
. mido gen drkreg = drkfre>=2
-> Applying gen to dataset1 (_mitemp1.dta).
-> Applying gen to dataset2 (_mitemp2.dta).
-> Applying gen to dataset3 (_mitemp3.dta).
-> Applying gen to dataset4 (_mitemp4.dta).
-> Applying gen to dataset5 (_mitemp5.dta).
. for num 1/6: mici: drkreg if wave==X, bin
->  mici: drkreg if wave==1, bin
(male & female)
Overall estimates                                       -- Binomial --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+--------------------------------------------------------------
      drkreg |        195    .1230769    .0238581        .0763051    .1698487
->  mici: drkreg if wave==2, bin
(male & female)
Overall estimates                                       -- Binomial --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+--------------------------------------------------------------
      drkreg |        195    .2061539    .0309131        .1452904    .2670173
->  mici: drkreg if wave==3, bin
(male & female)
Overall estimates                                       -- Binomial --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+--------------------------------------------------------------
      drkreg |        195    .2441026    .0328858        .1793391     .308866
->  mici: drkreg if wave==4, bin
(male & female)
Overall estimates                                       -- Binomial --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+--------------------------------------------------------------
      drkreg |        195    .2912821     .035184        .2218792    .3606849
->  mici: drkreg if wave==5, bin
(male & female)
Overall estimates                                       -- Binomial --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+--------------------------------------------------------------
      drkreg |        195    .2646154    .0338951        .1978309    .3313999
->  mici: drkreg if wave==6, bin
(male & female)
Overall estimates                                       -- Binomial --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+--------------------------------------------------------------
      drkreg |        195    .3558975    .0407032        .2740175    .4377774
```

Examining the `wave` 6 outcome in more detail, the `indiv` results reveal a moderate amount of between-imputation variability, resulting in an overall SE about 20% greater than the apparent SE within each imputed dataset.

```
. mici, indiv : drkreg if wave==6, bin
(male & female)
```

Variable / Data	Obs	Mean	Std. Err.	-- Binomial -- [95% Conf.	Interval]
drkreg					
_mitemp1	195	.348718	.0341275	.2818293	.4156066
_mitemp2	195	.3435898	.0340087	.2769338	.4102456
_mitemp3	195	.3333333	.033758	.2671689	.3994977
_mitemp4	195	.3794872	.0347501	.3113782	.4475962
_mitemp5	195	.374359	.0346569	.3064328	.4422852
Overall	195	.3558975	.0407032	.2740175	.4377774

There is a clear trend for the prevalence of regular drinking to increase with `wave`, and we may use logistic regression to examine whether this trend is similar for males and females. Note the use of the option `cluster(id)` to obtain robust standard errors allowing for within-subject correlation.

```
. mifit : xi: logistic drkreg i.sex*wave, cl(id)
```

Overall estimates

Number of obs = 1170

drkreg	Odds Ratio	Std. Err.	t	P>|t|	[95% Conf.	Interval]	MI.df
_Isex_1	.52254	.20336	-1.67	0.095	.24369	1.1205	66938.36
wave	1.2254	.07173	3.47	0.001	1.0921	1.375	308.41
_IsexXwave_1	1.038	.08448	0.46	0.647	.88467	1.2178	681.17

Next, we assess potential nonlinearity (in the logistic scale) by including in the model a (centered) quadratic term in `wave`.

```
. mido gen wave2 = (wave-2.5)^2
-> Applying gen to dataset1 (_mitemp1.dta).
-> Applying gen to dataset2 (_mitemp2.dta).
-> Applying gen to dataset3 (_mitemp3.dta).
-> Applying gen to dataset4 (_mitemp4.dta).
-> Applying gen to dataset5 (_mitemp5.dta).
. mifit: logistic drkreg sex wave wave2, cl(id)
```

Overall estimates

Number of obs = 1170

drkreg	Odds Ratio	Std. Err.	t	P>|t|	[95% Conf.	Interval]	MI.df
sex	.60251	.14386	-2.12	0.035	.37667	.96374	323.55
wave	1.3483	.09245	4.36	0.000	1.1786	1.5425	1036.19
wave2	.9656	.02415	-1.40	0.167	.91844	1.0152	57.74

Slightly more precise estimates can be obtained using the GEE method (with the default assumption of exchangeable working correlations).

```
. mifit: xtgee drkreg sex wave, fam(binom) i(id) eform

 Overall estimates
                                                Number of obs   =       1170
```

drkreg	Odds Ratio	Std. Err.	t	P>|t|	[95% Conf.	Interval]	MI.df
sex	.6102	.14544	-2.07	0.039	.38139	.97629	202.30
wave	1.2457	.04823	5.67	0.000	1.1535	1.3454	87.39

It may also be of interest to examine the association with other covariates; for example, a measure of psychological morbidity grouped into 3 categories:

```
. mido gen cisgp = cistot
-> Applying gen to dataset1 (_mitemp1.dta).
-> Applying gen to dataset2 (_mitemp2.dta).
-> Applying gen to dataset3 (_mitemp3.dta).
-> Applying gen to dataset4 (_mitemp4.dta).
-> Applying gen to dataset5 (_mitemp5.dta).
. mido recode cisgp 0/5=1 6/11=2 12/100=3
-> Applying recode to dataset1 (_mitemp1.dta).
(cisgp: 1077 changes made)
-> Applying recode to dataset2 (_mitemp2.dta).
(cisgp: 1071 changes made)
-> Applying recode to dataset3 (_mitemp3.dta).
(cisgp: 1077 changes made)
-> Applying recode to dataset4 (_mitemp4.dta).
(cisgp: 1069 changes made)
-> Applying recode to dataset5 (_mitemp5.dta).
(cisgp: 1076 changes made)
. mifit: xi: xtgee drkreg sex wave i.cisgp, fam(binom) i(id) eform

 Overall estimates
                                                Number of obs   =       1170
```

drkreg	Odds Ratio	Std. Err.	t	P>|t|	[95% Conf.	Interval]	MI.df
sex	.55838	.13396	-2.43	0.016	.34826	.89529	303.65
wave	1.2807	.05258	6.03	0.000	1.1802	1.3897	81.89
_Icisgp_2	1.0057	.19269	0.03	0.976	.68955	1.4668	241.80
_Icisgp_3	1.7755	.35374	2.88	0.004	1.2	2.6272	358.64

A test of the overall null hypothesis of no differences between the three "CIS groups" can be obtained using the `mitestparm` command:

```
. mitestparm _Icis*
 ( 1)  _Icisgp_2 = 0
 ( 2)  _Icisgp_3 = 0
       F(  2,  3996) =    5.45
            Prob > F =    0.0043
```

There is clearly strong evidence against the null hypothesis—not surprising, given the individual effect estimates in the regression table, which reveal that this is entirely due to an elevated odds in the third group.

Although there was no evidence of `sex` by `wave` interaction, we refit this model to illustrate the use of `milincom`.

```
. mifit: xi: logistic drkreg i.sex*wave
 Overall estimates
                                                  Number of obs   =       1170
```

drkreg	Odds Ratio	Std. Err.	t	P>|t|	[95% Conf.	Interval]	MI.df
_Isex_1	.52254	.18247	-1.86	0.063	.26356	1.036	43384.52
wave	1.2254	.07201	3.46	0.001	1.0916	1.3756	313.20
_IsexXwave_1	1.038	.08936	0.43	0.665	.87658	1.229	852.87

```
. milincom: wave + _IsexXwave_1
 Overall estimates
```

drkreg	Odds Ratio	Std. Err.	t	P>|t|	[95% Conf.	Interval]	MI.df
(1)	1.271952	.0837888	3.65	0.000	1.117311	1.447996	303.59

This provides an estimate of the OR per `wave` among females; as the interaction term is very small, this is almost identical to that among males (1.2254).

Now, we reshape the data to examine longitudinal changes in more detail; for example, below examining the uptake of drinking between `wave`s 1 and 2.

```
. mido gen drkany = drkfre>=1
-> Applying gen to dataset1 (_mitemp1.dta).
-> Applying gen to dataset2 (_mitemp2.dta).
-> Applying gen to dataset3 (_mitemp3.dta).
-> Applying gen to dataset4 (_mitemp4.dta).
-> Applying gen to dataset5 (_mitemp5.dta).
. mido keep id wave drkany cisgp sex
-> Applying keep to dataset1 (_mitemp1.dta).
-> Applying keep to dataset2 (_mitemp2.dta).
-> Applying keep to dataset3 (_mitemp3.dta).
-> Applying keep to dataset4 (_mitemp4.dta).
-> Applying keep to dataset5 (_mitemp5.dta).
```

(*Continued on next page*)

```
. mido reshape wide drkany cisgp, i(id) j(wave)
-> Applying reshape to dataset1 (_mitemp1.dta).
(note: j = 1 2 3 4 5 6)
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                     1170   ->     195
Number of variables                   5   ->      14
j variable (6 values)              wave   ->   (dropped)
xij variables:
                                drkany   ->   drkany1 drkany2 ... drkany6
                                 cisgp   ->   cisgp1 cisgp2 ... cisgp6
-----------------------------------------------------------------------------
-> Applying reshape to dataset2 (_mitemp2.dta).
(note: j = 1 2 3 4 5 6)
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                     1170   ->     195
Number of variables                   5   ->      14
j variable (6 values)              wave   ->   (dropped)
xij variables:
                                drkany   ->   drkany1 drkany2 ... drkany6
                                 cisgp   ->   cisgp1 cisgp2 ... cisgp6
-----------------------------------------------------------------------------
-> Applying reshape to dataset3 (_mitemp3.dta).
(note: j = 1 2 3 4 5 6)
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                     1170   ->     195
Number of variables                   5   ->      14
j variable (6 values)              wave   ->   (dropped)
xij variables:
                                drkany   ->   drkany1 drkany2 ... drkany6
                                 cisgp   ->   cisgp1 cisgp2 ... cisgp6
-----------------------------------------------------------------------------
-> Applying reshape to dataset4 (_mitemp4.dta).
(note: j = 1 2 3 4 5 6)
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                     1170   ->     195
Number of variables                   5   ->      14
j variable (6 values)              wave   ->   (dropped)
xij variables:
                                drkany   ->   drkany1 drkany2 ... drkany6
                                 cisgp   ->   cisgp1 cisgp2 ... cisgp6
-----------------------------------------------------------------------------
-> Applying reshape to dataset5 (_mitemp5.dta).
(note: j = 1 2 3 4 5 6)
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                     1170   ->     195
Number of variables                   5   ->      14
j variable (6 values)              wave   ->   (dropped)
xij variables:
                                drkany   ->   drkany1 drkany2 ... drkany6
                                 cisgp   ->   cisgp1 cisgp2 ... cisgp6
-----------------------------------------------------------------------------
```

```
. mici: drkany2 if drkany1==0, bin
(male & female)
Overall estimates                                          -- Binomial --
    Variable | Obs(min) Obs(max)     Mean    Std. Err.   [95% Conf. Interval]
-------------+-------------------------------------------------------------
     drkany2 |     136      137  .3034993    .0401668    .2247364    .3822623*
```

This provides an estimate of the (cumulative) incidence of alcohol use between `wave`s 1 and 2. The asterisk at the end of the output line is a hyperlink to a warning message indicating that the complete data analyses are not based on the same number of observations (because the `if` condition defines a slightly different subset in each imputation). Logistic regression may be used (below) to examine association between incidence and covariates of interest.

```
. mifit: xi: logistic drkany2 i.cisgp1 if drkany1==0
 Overall estimates
                                          Number of obs (min)   =        136
                                          Number of obs (max)   =        137
------------------------------------------------------------------------------
     drkany2 | Odds Ratio   Std. Err.      t    P>|t|     [95% Conf. Interval]  MI.df
-------------+----------------------------------------------------------------
  _Icisgp1_2 |     .8641    .40652    -0.31   0.756     .34345     2.174    1968.65*
  _Icisgp1_3 |  1.0686     .48456     0.15   0.884     .43906    2.6007    1603.00*
------------------------------------------------------------------------------
```

Finally, you might choose to do a little cleaning up before ending the session.

```
. for num 1/5: erase tempX.dta
->  erase temp1.dta
->  erase temp2.dta
->  erase temp3.dta
->  erase temp4.dta
->  erase temp5.dta
. for num 1/5, nohead : erase bothX.dta
. mireset
```

7 Saved Results

`mici` saves results in `r()`:

Macros

`r(mimps)`	number of multiple imputed datasets
`r(level)`	confidence level

Matrices

`r(overall)`	multiple-imputation results and related quantities of interest

`mifit` saves results in `e()`:

Scalars	
`e(dr_r)`	maximum multiple-imputation degrees of freedom for estimates of regression coefficients
`e(df_m)`	model degrees of freedom
`e(obs_mi`i`)`	number of observations in the ith imputed dataset
Macros	
`e(cmd)`	original estimation command issued in the call of mifit
`e(depv)`	name of the dependent variable
`e(mi_level)`	confidence level
Matrices	
`e(MI_b)`	regression coefficient vector
`e(MI_V)`	diagonal elements, which are estimates of multiple-imputation variance for the coefficients. Note that it is *not* a genuine variance–covariance matrix
`e(MI_df)`	vector of multiple-imputation degrees of freedom, see expression (2) in section 2
`e(b_`i`)`	coefficient vector for the ith imputed dataset
`e(V_`i`)`	variance–covariance matrix for the ith imputed dataset

8 References

Li, K., T. Raghunathan, and D. Rubin. 1991. Large sample significance levels from multiply-imputed data using moment-based statistics and an F reference distribution. *Journal of the American Statistical Association* 86: 1065–1073.

Little, R. and D. Rubin. 1987. *Statistical Analysis with Missing Data*. New York: John Wiley & Sons.

Mander, A. and D. Clayton. 1999. sg116: Hotdeck imputation. *Stata Technical Bulletin* 51: 32–34. In *Stata Technical Bulletin Reprints*, vol. 9, 196–199. College Station, TX: Stata Press.

Raghunathan, T., J. Lepkowski, J. V. Hoewyk, and P. Solenberger. 2001. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology* 27(1): 85–95.

Rubin, D. 1976. Inference and missing data (with discussion). *Biometrika* 63: 581–592.

—. 1987. *Multiple Imputation for Nonresponse in Surveys*. New York: John Wiley & Sons.

Schafer, J. 1997. *Analysis of Incomplete Multivariate Data*. London: Chapman & Hall.

About the Authors

John Carlin is a Professor in the Departments of Paediatrics and Public Health at the University of Melbourne and Director of the Clinical Epidemiology and Biostatistics Unit, Murdoch Children's Research Institute, at the Royal Children's Hospital in Melbourne.

Ning Li is a Ph.D. student at La Trobe University in Melbourne and was a Research Assistant in the Clinical Epidemiology and Biostatistics Unit in 2000–2001.

Philip Greenwood is a Research Assistant in the Clinical Epidemiology and Biostatistics Unit, Murdoch Children's Research Institute.

Carolyn Coffey is a Research Fellow with the Centre for Adolescent Health at the Murdoch Children's Research Institute and a senior investigator on the Victorian Adolescent Health Cohort Study.