The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$ 98	1-year subscription	$138
2-year subscription	$165	2-year subscription	$245
3-year subscription	$225	3-year subscription	$345
1-year student subscription	$ 75	1-year student subscription	$ 99
1-year institutional subscription	$245	1-year institutional subscription	$285
2-year institutional subscription	$445	2-year institutional subscription	$525
3-year institutional subscription	$645	3-year institutional subscription	$765
Electronic only		**Electronic only**	
1-year subscription	$ 75	1-year subscription	$ 75
2-year subscription	$125	2-year subscription	$125
3-year subscription	$165	3-year subscription	$165
1-year student subscription	$ 45	1-year student subscription	$ 45

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata, Stata, Stata Press, Mata, Mata, and NetCourse are registered trademarks of StataCorp LP.

The Stata Journal (2014)
14, Number 2, pp. 418–431

Application of multiple imputation using the two-fold fully conditional specification algorithm in longitudinal clinical data

Catherine Welch
University College London
London, UK
catherine.welch@ucl.ac.uk

Jonathan Bartlett
London School of Hygiene & Tropical Medicine
London, UK
jonathan.bartlett@lshtm.ac.uk

Irene Petersen
University College London
London, UK
i.petersen@ucl.ac.uk

Abstract. Electronic health records of longitudinal clinical data are a valuable resource for health care research. One obstacle of using databases of health records in epidemiological analyses is that general practitioners mainly record data if they are clinically relevant. We can use existing methods to handle missing data, such as multiple imputation (MI), if we treat the unavailability of measurements as a missing-data problem. Most software implementations of MI do not take account of the longitudinal and dynamic structure of the data and are difficult to implement in large databases with millions of individuals and long follow-up. Nevalainen, Kenward, and Virtanen (2009, *Statistics in Medicine* 28: 3657–3669) proposed the two-fold fully conditional specification algorithm to impute missing data in longitudinal data. It imputes missing values at a given time point, conditional on information at the same time point and immediately adjacent time points. In this article, we describe a new command, `twofold`, that implements the two-fold fully conditional specification algorithm. It is extended to accommodate MI of longitudinal clinical records in large databases.

Keywords: st0345, twofold, multiple imputation, longitudinal data

1 Introduction

Electronic health records of routinely collected clinical information are a potentially valuable resource for epidemiological investigations and health care research. One example are primary care databases such as The Health Improvement Network (THIN) (CSD 2011), which provide longitudinal records of routinely collected clinical data. One obstacle when using databases of health records in epidemiological analyses is that the general practitioner (GP) mainly collects the data if they are relevant to the clinical care of the individual. If health indicators are required to analyze the data, such as weight and systolic blood pressure measured at a particular time (for example, relative to registration with a GP), values are unavailable for many individuals because the

health indicators were not measured at that time as part of their clinical care. A GP (family physician) will regularly record weight or blood pressure for individuals in poor health, but it is unnecessary for healthy men and women. For example, measurements of health indicators are more regularly recorded for individuals with previous cardiovascular events, or at high risk, compared with individuals without previous cardiovascular events. This difference increased after the introduction of the Quality Outcome Framework (QOF 2012) in the UK in 2004. As a consequence, calculating statistics of interest becomes problematic because of the unavailability of health indicator measurements.

If we treat measurements as if they were intended to be recorded on a regular basis, the unavailability of measurements is now a missing-data problem, and we can use existing methods to handle missing data. From a missing-data perspective, these datasets present formidable challenges: the "missing data" (or data we want to analyze but were not measured) generally have an intermittent pattern of missingness over time (nonmonotone) and are not missing completely at random, so approaches such as complete-case analysis are inefficient and potentially biased (Little and Rubin 2002).

Rubin (1987) developed multiple imputation (MI), a popular approach to handle missing data. MI replaces each missing value with multiple imputed values, usually random draws from the distribution of an imputation regression model that conditions on the observed data. The end result is multiple complete datasets. The user fits a substantive model to each imputed dataset, and the parameter estimates and standard errors from the substantive model are combined using Rubin's rules (Rubin 1987). This takes into account the uncertainty of the estimates due to the missing data. Inferences from imputed data are valid provided the imputation model is correctly specified and data are missing at random (MAR) (Rubin 1987). Schafer (1997) recommends richly specifying the imputation model, including all available explanatory variables as covariates. This increases the plausibility of the MAR assumption, reduces uncertainty in the imputed values, and gives more efficient inferences.

Provided the MAR assumption is valid, and imputations are drawn from correctly specified models, the resulting estimates are unbiased and are efficient in the sense that optimal use of the observed information is used. Depending on the patterns and level of missingness and on the substantive model, MI can result in substantial gains in efficiency compared with complete-case analysis, even when the latter is unbiased. MI is therefore an attractive option to consider for tackling the missing-data problem in electronic health records of routinely collected clinical information.

A popular MI approach is fully conditional specification (FCS), which specifies separate univariate imputation models for each variable with missing data conditional on all other variables (van Buuren, Boshuizen, and Knook 1999). Therefore, we can choose a model appropriate to the variable type (that is, continuous, count, ordered categorical, unordered categorical). This method is easier than directly specifying a multivariate distribution for a mixture of continuous and categorical variables with missing data, as required in parametric MI's original form.

In longitudinal studies where individuals' characteristics are measured at fixed times, we can treat measurements of health indicators at each "time" as distinct variables

and impute using FCS. An imputation model for a health indicator at a particular time point includes the variables corresponding to measurements of other health indicators at the same time point and the measurements at all other time points, across all health indicators, as explanatory variables. However, with a moderate number of health indicators and time points, the imputation model has many explanatory variables, potentially causing numerical problems because of overfitting. To overcome this, Nevalainen, Kenward, and Virtanen (2009) recently proposed a modification of the FCS approach to MI, the two-fold FCS algorithm. Missing values at a given time point are imputed from a model that only uses information from that time point and immediately adjacent time points. The rationale is that measurements of health indicators at time points before or after the time point with imputed measurements are more unlikely to provide substantial additional information than measurements at immediately adjacent time points. This simplifies the imputation models and reduces problems because of overfitting. However, this simplification may induce bias in parameter estimates if the measurements excluded from imputation models have independent effects.

In this article, we describe a new command, `twofold`, that implements an extension of the two-fold FCS algorithm. In the next section, we describe our implementation of the two-fold FCS algorithm. In section 3, we explain the syntax of the `twofold` command. In section 4, we illustrate the command with data derived from a primary-care longitudinal clinical database. In section 5, we conclude with some final remarks.

2 Two-fold FCS MI

In this section, we implement the two-fold FCS algorithm as the `twofold` command. We assume time is discretized into q time points. Let $X_t = (X_{t1}, \ldots, X_{tp})$ denote the vector of values of the p variables at time point t. Let $X = (X_1, \ldots, X_q)$ denote the vector of values of the p variables at all q time points. Let Y denote the outcome variables in the substantive model, which we assume is fully observed. Let $Z = (Z_1, \ldots, Z_r)$ denote a vector of time-independent variables, some (or all) of which may have missing values. Our aim is to impute missing values in X and Z.

For each time point t ($t = 1, \ldots, q$) and variable j ($j = 1, \ldots, p$) such that X_{tj} contains missing values, we specify an imputation model for

$$f(X_{tj}|Y, Z, X_{t,-j}, X_{t-1}, X_{t+1}) \tag{1}$$

where $X_{t,-j}$ denotes the vector of variables at time point t excluding the jth variable. At time point $t = 1$, measurements at $t - 1$ are missing for all individuals; X_{t-1} is excluded from (1); and similarly, at the last time point when $t + 1$ does not exist, X_{t+1} is excluded from (1).

If a component of Z, say, Z_k, contains missing values, we also specify an imputation model for Z_k, which similarly includes a subset of the other variables as covariates. We implemented the `twofold` command, so missing values in Z_k are imputed from an imputation model that conditions on the outcome Y, the other time-independent variables Z_{-k}, and the values of the time-dependent variables at an individual-specific "baseline" time b:

$$f(Z_k|Y, Z_{-k}, X_b)$$

We chose to include values of time-dependent variables at "baseline" when imputing time-independent variables to simplify the imputation model and avoid collinearity issues that would arise if measurements of time-dependent variables at all time points were included as covariates. An alternative solution is to treat the time-independent variables as time dependent. Hence, values of time-dependent variables at later times also inform the imputation of time-independent variables.

As with standard FCS MI, `twofold` initially replaces each missing value with a randomly selected observed value from the same variable (measured at the same time).

Next any components of Z with missing values are imputed in order of ascending missingness. Once the time-independent variables are imputed, `twofold` moves on to impute the time-dependent variables X. The time-dependent variables at time point t are imputed by cycling around the specified imputation models, performing a certain number of cycles (option `bw(`*bw*`)`), or "within-time iterations". The variables at time point t+1 are imputed next, again using *bw* iterations. Each time point is chronologically updated. Once *bw* iterations are performed at the last time point, the first "among-time iteration" (option `ba(`*ba*`)`) is complete (figure 1). Further *ba* iterations are performed, each one starting at the first time point. At each step, the most recent imputations of missing values are carried forward to the next step. When the *ba* iterations are complete, the current imputations of missing values, together with the originally observed values, form the first imputed dataset. The whole process is repeated to create as many imputed datasets as desired, using the original dataset as starting values to ensure imputations are independent.

Each univariate imputation step in the two-fold FCS algorithm is identical to the corresponding step in standard FCS: the postulated imputation model is fit to individuals with the variable observed, conditioning on the observed and current imputations of the imputation model's explanatory variables; a draw of the imputation model's parameters is taken from their posterior distribution (assuming standard noninformative priors); and lastly, the missing values are imputed using these newly drawn parameter values.

2.1 Time window width

As proposed by Nevalainen, Kenward, and Virtanen (2009), and as described thus far, the two-fold FCS algorithm conditions on measurements at time $t-1$ and $t+1$ when imputing missing values at time t. This is reasonable if measurements of variables at the other (excluded) time points are independent of the variable's values at time point

t, conditional on values at $t-1$ and $t+1$ and the outcome Y and Z. Sometimes, it is desirable to increase the "time window" width and condition on variable measurements at other times. For example, with a time window of 2, we condition on values at times $t-2$, $t-1$, $t+1$, and $t+2$. The width of the time window can be specified in the `twofold` command.

2.2 Late entry and loss to follow-up

`twofold` automatically imputes all missing values at the q time points for all individuals. This is undesirable in some contexts. For example, in many longitudinal clinical databases, individuals are registered with the corresponding health body only for a portion of the time points $1, \ldots, q$, as represented in figure 1. The missing measurements before entry and after exit times are also imputed. In this situation, we usually want to impute missing values only within the period the individual was registered.

The `twofold` command can specify an entry and exit time for each individual. When the two-fold FCS algorithm completes each imputation, those imputed values falling outside an individual's registration period are changed back to missing. For example, if we impute missing values for the data represented in figure 1 using `twofold`, individual 2 would have values imputed at time t, but at the end of each imputation, these values would be changed back to missing. This is also true for the time points $t+1$, $t+2$, and $t+3$.

An option for `twofold` retains the imputed values before and after follow-up. However, it is important to consider the appropriateness of this. For example, it may not be appropriate to keep imputed values after individuals die because `twofold` treats these individuals as if they survived. The individuals' imputations are based on those individuals who did survive to this time, and the imputed values are estimates of the measurements they would have had if they had survived.

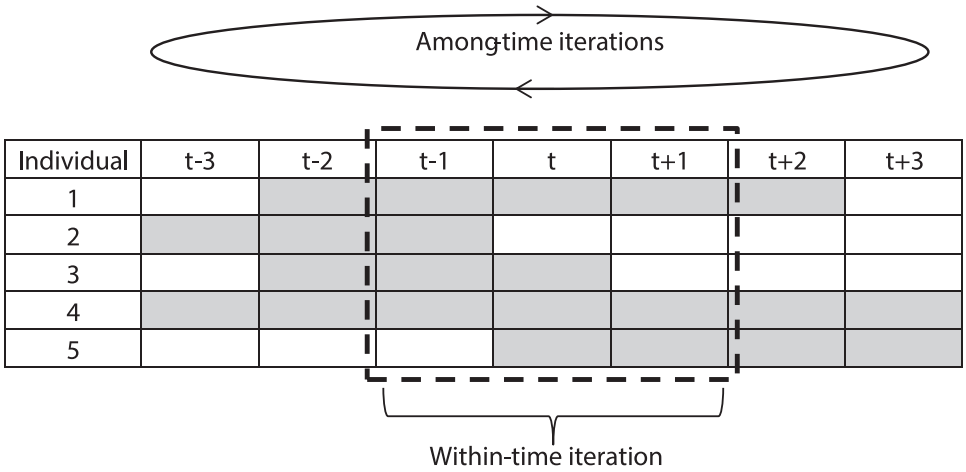

Figure 1. Lengths of follow-up for individuals in longitudinal clinical data (gray indicates the time point is included in the individuals' follow-up)

2.3 Incorporation of the outcome

When imputing missing values in covariates with missing data, we must include the outcome variable Y from the substantive model as an explanatory variable. Omitting the outcome variable from the imputation model results in distorted associations between covariates and outcome, leading to bias (Sterne et al. 2009). The appropriate way to incorporate the outcome Y in covariate imputation models depends on the type and specification of the substantive model fit to the imputed datasets (Bartlett et al. 2013). When the outcome is a censored time to event, Y consists of two variables: the time to event (or censoring) and an event indicator. White and Royston (2009) showed that imputation models for binary and continuous (to an approximation) covariates should include the event indicator and the cumulative baseline hazard function as covariates. Their simulation results suggest that including the event indicator and the time-to-event variable as covariates typically gives estimates with small biases. An option of `twofold` specifies the outcome variables included in the imputation models. We caution the user to ensure variables are imputed from models compatible with the substantive models subsequently fit to the imputed data: incompatibility between these may cause bias (Bartlett et al. 2013). Lastly, it is possible to condition on multiple outcomes in the imputation models, which may be used when substantive models for different outcomes are to be fit to the imputed datasets.

3 The twofold command

3.1 Syntax

`twofold, timein(`*varname*`) timeout(`*varname*`)`

{`clear`|`saving(`*filename*[`, replace`])} [`table base(`*varname*`)`

`indmis(`*varlist*`) depmis(`*varlist*`) indobs(`*varlist*`) depobs(`*varlist*`)`

`outcome(`*varlist*`) cat(`*varlist*`) m(`#`) ba(`*ba*`) bw(`*bw*`) width(`#`)`

`conditionon(`*varlist*`) condvar(`*varlist*`) condval(`*string*`) im keepoutside`

`trace(`*filename*[`, `*string*`])`]

3.2 Options

`timein(`*varname*`)` specifies a variable *varname* indicating the time point each individual entered the study. Missing values are imputed only for time points between and including an individual's `timein()` and `timeout()` (see `timeout()`). `timein()` is required.

`timeout(`*varname*`)` specifies a variable *varname* indicating the time point each individual exited the study. Missing values are imputed only for time points between and including an individual's `timein()` and `timeout()` (see `timein()`). `timeout()` is required.

`clear` specifies that the original memory be cleared and the combined datasets be loaded into the memory. The dataset must be saved manually. `saving()` or `clear` is required.

`saving(`*filename*[`, replace`]`)` specifies that the original dataset as well as the imputed datasets will be saved to *filename*. `replace` allows *filename* to be overwritten with the new data. `saving()` or `clear` is required.

`table` produces a table showing the percentage of missing values for the time-independent variables with missing values and the time-dependent variables with missing values at each time point for all individuals, regardless of when they enter and exit the study.

`base(`*varname*`)` specifies the variable containing the baseline time point for each individual. The time-independent variables with missing data are imputed conditional on other time-independent variables and time-dependent variables recorded at baseline. The baseline time point must be within the individual's follow-up time, specified by `timein()` and `timeout()`.

`indmis(`*varlist*`)` specifies time-independent variables with missing values, imputed at the beginning of each among-time iteration.

`depmis(`*varlist*`)` specifies the variable name stems of the time-dependent variables with missing values. The variable names for the same measurements will have a stem and a number to represent the time point. For example, with weight measurements at each time point, with time beginning at 1, the dataset contains the variables `weight1`, `weight2`, etc., so the stem `weight` is passed using the `depmis()` option. If one variable is passed to `depmis()`, `twofold` performs only one within-time iteration.

`indobs(`*varlist*`)` specifies fully observed time-independent variables, included as explanatory variables in imputation models.

`depobs(`*varlist*`)` specifies the stem of any time-dependent variables that are fully observed at all time points within the follow-up time specified by `timein()` and `timeout()`. When these variables are imputed, the values of the `depobs()` variables at the time point specified using `base()` are included as explanatory variables in the imputation model. Similarly, when time-dependent variables are imputed at time t, the values of the variables in `depobs()` at time point t are included as explanatory variables in the imputation model. Only the stem is specified using *varlist*.

`outcome(`*varlist*`)` specifies the fully observed outcome variables, included as explanatory variables in imputation models. For survival models, both outcome indicator and survival-time variables are specified using *varlist*.

`cat(`*varlist*`)` specifies the categorical variables with two or more categories. These variables with missing values are imputed assuming a multinomial logistic model. If they are complete, they will be categorical auxiliary variables. If a binary variable is coded as `0/1` and specified as a categorical variable, `twofold` will recognize the variable is binary and assume a logistic distribution. If it is a time-dependent categorical variable, only the stem is specified using *varlist*.

`m(#)` specifies the number of imputations to be created. The default is `m(5)`.

`ba(`*ba*`)` specifies the number of among-time iterations. The default is `ba(10)`.

`bw(`*bw*`)` specifies the number of within-time iterations. The default is `bw(5)`.

`width(#)` specifies the width of the time window. When you impute time-dependent variables at time t, the values of other time-dependent variables within `width()` time units are included as explanatory variables. The default is `width(1)`, so measurements recorded at time $t-1$ and $t+1$ are included in the imputation model to inform imputation of missing values at time t. If the window width is 2, measurements recorded at time $t-2$, $t-1$, $t+1$, and $t+2$ are included in the imputation model.

`conditionon(`*varlist*`)` is the variable `condvar()` conditions on.

`condvar(`*varlist*`)` specifies that the variables passed to `condvar()` are only imputed for individuals if the variable specified by `conditionon()` is equal to the value `condval()`. `conditionon()` can be specified as the stem for time-dependent variables, one of the time-dependent variables at a specific time or a time-independent

variable, that is, `weight`, `weight2001`, or `gender`. If the stem is specified, measurements at all time points are imputed if the variable specified by `conditionon()` is equal to `condval()`. If measurements at a single time point are specified, only the measurements at this point are imputed if the variable specified by `conditionon()` is equal to `condval()`.

For example, a variable `smoker` = 1 if an individual is a smoker, and `smoker` = 0 otherwise. Another variable, `nocigs`, indicates the reported number of cigarettes that smoking individuals smoke. Ordinarily, we do not want to impute the number of cigarettes for nonsmokers. This is achieved by specifying `conditionon(smoker) condval(1) condvar(nocigs)`.

`condval(`*string*`)` is the value `condvar()` conditions on.

`im` displays the `mi impute` commands. To avoid duplication, the `mi impute` commands are only shown for each among-time iteration of the first imputation. For the first among-time iteration, each command imputes missing values at more than one time point because there are missing values at each time point. For subsequent among-time iterations, only missing values at each time point in turn are imputed because the missing values at other time points are replaced with previously imputed values.

`keepoutside` retains imputed values in the imputed datasets before the individual enters the study and after the individual exits the study. `twofold` replaces values imputed with missing values if this option is not specified.

`trace(`*filename*[`,` *string*]`)` saves the imputation number, the among-time iteration number, and the mean and standard deviation of imputed variables within `timein()` and `timeout()` after each among-time iteration to *filename*. The results monitor the convergence of `twofold`. To assess convergence, initially investigate the means and standard deviations for one imputation (that is, `m(1)`) and many among-time iterations (that is, `ba(30)`). Convergence occurs when the pattern of the imputed means is random.

If just the *filename* is specified, the means and standard deviations are found for all time-dependent and time-independent variables with missing data.

It is possible to specify a single variable or group of variables as follows:

1. Time-independent variable: enter variable name in the `trace()` option, that is, for variable `height` `trace(`*filename*`, height)`.
2. Time-dependent variable at each time point: enter the stem of the variables in the `trace()` option, that is, for variable `weight` `trace(`*filename*`, weight)`.
3. Time-dependent variable at one time point: enter the stem of the variables and the time point in the `trace()` option, that is, for variable `weight` at time point 5 `trace(`*filename*`, weight 5)`.

3.3 Implementation details

The `twofold` command imputes using Stata's `mi impute chained`. The imputed data are `mi set` ready to analyze using `mi estimate`.

3.4 Using the twofold command

Implementation of the two-fold FCS MI algorithm in the `twofold` command assumes the data are in wide form, so each individual has one observation in the dataset and separate variables for measurements at each time point. For example, if weight was measured at each time point beginning at time point 1, the dataset contains variables `weight1`, `weight2`, etc. All time points must be positive integer values with one-unit increments. `twofold` does not support `if` or `in` and imputes all individuals in the dataset.

4 Example

We illustrate the `twofold` command using a sample dataset, simulated from the distribution of health indicators in THIN. THIN is a large, longitudinal, clinical primary care database widely used in epidemiological research (CSD 2011). The data are broadly representative of the entire UK population (Blak et al. 2011). THIN contains data from over 10 million individuals registered to approximately 500 practices since 1988. The recording of both consultations and prescriptions are similar to national consultation and prescription statistics (Bourke, Dattani, and Robinson 2004; McClure, Lee, and Wilson 2003). Available data include individual characteristics, medical (symptoms and diagnoses), and prescription information.

Annual measurements for the years 2000–2009 on specific health indicators (weight, height, and systolic blood pressure) were simulated for males registered with participating general practices (family physician) before 2000 and aged between 40 and 100 years in 2000. Age was split into 10 categories. Systolic blood pressure and weight were continuous time-dependent variables, and height was considered time independent. Systolic blood pressure measurements were selected completely at random and changed to missing, so the observed values were representative of the missing data.

For our substantive model, we assumed an exponential time-to-event model relating the hazard of a nonfatal coronary heart disease (CHD) event (between 2000 and 2009) to measurements of the health indicators recorded in the year 2000. Each individual's outcome time was calculated as the time between 1 January 2001 and the date of the first CHD event. The individuals without an observed CHD event were censored at the earliest of date of death, date of transfer out of the practice, or 31 December 2009. See table 1 for a description of the variables from our simulated data.

Table 1. Description of variables in simulated data

Variable name	Description
`firstyear`	Calendar year the individual entered the study; this is year 2000 for all individuals
`lastyear`	Calendar year the individual exited the study; the last year data is recorded
`age`	Age in 2000
`height`	Height
`chd`	Binary variable indicating whether the individual had CHD event
`chdtime`	Time from 2000 to CHD event/end of follow-up
`weight2000`	Weight measurement in the year 2000
⋮	⋮
`weight2009`	Weight measurement in the year 2009
`sys2000`	Systolic blood pressure measurement in the year 2000
⋮	⋮
`sys2009`	Systolic blood pressure measurement in the year 2009

To impute the missing values (here only occurring in the systolic blood pressure variable), we used the `twofold` command:

```
. use simulated
. twofold, timein(firstyear) timeout(lastyear) clear depmis(sys)
> indobs(age height) outcome(chd chdtime) depobs(weight) cat(age chd) m(2)
> ba(3) bw(5) width(1) table
There is only 1 variable with missing values, so only one within-time iteration
>  required
Time-dependent       | Percentage of missing values
variables            | at each time point
with missing values  | 2000  2001  2002  2003  2004  2005  2006  2007  2008  2009
---------------------+-----------------------------------------------------------
                 sys | 29.7  29.6  28.8  45.5  44.1  42.1  40.6  38.5  38.6  36.5

Imputation number 1
-------------------

Among-time iteration 1
Imputing time-dependent variables . . . . . .
Among-time iteration 2
Imputing time-dependent variables . . . . . .
Among-time iteration 3
Imputing time-dependent variables . . . . . .
```

```
Imputation number 2
-------------------

Among-time iteration 1
Imputing time-dependent variables . . . . . .
Among-time iteration 2
Imputing time-dependent variables . . . . . .
Among-time iteration 3
Imputing time-dependent variables . . . . . .
[note: imputed dataset now loaded in memory]
```

To instruct `twofold` to impute only missing values during individuals' follow-up time, we passed the `firstyear` and `lastyear` variables to the `timein()` and `timeout()` options. If instead we want `twofold` to impute at all time points for all individuals, the `timein()` and `timeout()` variables should equal 2000 and 2009 for all individuals. We specified the stem of the time-dependent systolic blood pressure variable with missing values by specifying `depmis(sys)`. The fully observed time-independent variables were specified using the `indobs()` option. The fully observed time-dependent weight variable was included as an explanatory variable in imputation models by using the `depobs()` option. The fully observed outcome variables were specified using the `outcome()` option. Finally, the categorical variable `age` was specified using the `cat()` option.

Finally, we fit the model of interest to the imputed data, properly combining the estimates using `mi estimate`.

5 Comments

The `twofold` command implements an extension of the two-fold FCS algorithm to accommodate MI of longitudinal, clinical records in large datasets. A previous implementation in SAS by Nevalainen, Kenward, and Virtanen (2009) was designed to impute only time-dependent variables, and all individuals entered and exited the study at the same time points. Our more flexible implementation can impute time-independent variables and allow users to specify the width of the time window.

The distinguishing characteristic of the two-fold FCS algorithm is the use of simplified imputation models: values at a given time are imputed using only measurements at nearby times (plus outcome and time-independent variables). This reduces the complexity of the imputation models relative to conventional application of FCS MI and is less prone to issues of collinearity and overfitting. In all settings, we must carefully consider whether the simplification is reasonable for the data. For example, if an exploratory analysis finds that measurements further away in time provide independent information given the adjacent time points, we can increase the time window width. An important issue to consider when using MI generally is to ensure that imputations are generated from models that are compatible with the substantive model or analysis that will be performed on the imputed datasets (Bartlett et al. 2013). The two-fold command can condition on one or more outcome variables when imputing other variables.

In principle, this allows the user to fit multiple substantive models to the imputed data with different outcome variables. However, further research is needed to clarify how to ensure unbiased estimates of parameters in multiple substantive models. A possibly preferable alternative approach is to generate separate imputed datasets for each outcome of interest, with imputation models specified to ensure they are compatible with a given outcome and substantive model. Furthermore, we caution users that it is difficult to impute compatibly using standard imputation models when substantive models contain interactions or nonlinear covariate effects (Bartlett et al. 2013). Further research is needed to explore how longitudinal data should be imputed in such settings.

As with standard FCS MI, the two-fold FCS algorithm is iterative, and a sufficient number of iterations must be performed to ensure that the algorithm has converged to its stationary distribution. Unlike standard FCS MI, with two-fold FCS algorithm, the user must separately specify the number of among-time and within-time iterations. In our experience from simulation studies, we found a relatively small number of within-time iterations (that is, 5) and a large number of among-time iterations (10–20) give good convergence of the algorithm, leading to the command's default choices of these options. However, as with standard FCS MI, diagnostics (such as plotting means and standard deviations by iteration number) can and should be used to empirically assess convergence and, if necessary, use more iterations.

6 Acknowledgments

The `twofold` command was developed as part of the project titled "Missing data imputation in clinical databases: Development of a longitudinal model for cardiovascular risk factors", funded by the UK Medical Research Council. We would like to thank Ian White, Louise Marston, and Sarah Hardoon for testing the command and their suggested improvements to the code and article.

Jonathan Bartlett was supported by a grant from the ESRC Follow-On Funding scheme (RES-189-25-0103) and MRC grant G0900724.

7 References

Bartlett, J. W., S. R. Seaman, I. R. White, and J. R. Carpenter. 2013. Multiple imputation of covariates by fully conditional specification: Accommodating the substantive model. http://arxiv.org/abs/1210.6799.

Blak, B. T., M. Thompson, H. Dattani, and A. Bourke. 2011. Generalisability of The Health Improvement Network (THIN) database: Demographics, chronic disease prevalence and mortality rates. *Informatics in Primary Care* 19: 251–255.

Bourke, A., H. Dattani, and M. Robinson. 2004. Feasibility study and methodology to create a quality-evaluated database of primary care data. *Informatics in Primary Care* 12: 171–177.

CSD. 2011. Cegedim Strategic Data. http://csdmruk.cegedim.com/our-data/our-data.shtml.

Little, R. J. A., and D. B. Rubin. 2002. *Statistical Analysis with Missing Data*. 2nd ed. Hoboken, NJ: Wiley.

McClure, F. D., J. K. Lee, and D. B. Wilson. 2003. Validity of the percent reduction in standard deviation outlier test for screening laboratory means from a collaborative study. *Journal of AOAC International* 86: 1045–1055.

Nevalainen, J., M. G. Kenward, and S. M. Virtanen. 2009. Missing values in longitudinal dietary data: A multiple imputation approach based on a fully conditional specification. *Statistics in Medicine* 28: 3657–3669.

QOF. 2012. Quality and outcomes framework, The Information Centre. http://www.qof.ic.nhs.uk/.

Rubin, D. B. 1987. *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley.

Schafer, J. L. 1997. *Analysis of Incomplete Multivariate Data*. Boca Raton, FL: Chapman & Hall/CRC.

Sterne, J. A. C., I. R. White, J. B. Carlin, M. Spratt, P. Royston, M. G. Kenward, A. M. Wood, and J. R. Carpenter. 2009. Multiple imputation for missing data in epidemiological and clinical research: Potential and pitfalls. *British Medical Journal* 338: b2393.

van Buuren, S., H. C. Boshuizen, and D. L. Knook. 1999. Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine* 18: 681–694.

White, I. R., and P. Royston. 2009. Imputing missing covariate values for the Cox model. *Statistics in Medicine* 28: 1982–1998.

About the authors

Catherine Welch is a research associate at University College London and is also a part-time PhD student. Her thesis topic is to implement and evaluate multiple imputation to impute missing values in longitudinal data.

Jonathan Bartlett is a lecturer at the London School of Hygiene & Tropical Medicine, with research interests in methods for handling missing data and covariate measurement error. Recently, he has been working specifically on methods for imputing covariates in the presence of interactions and nonlinearities and for imputing in large, routinely collected longitudinal datasets.

Irene Petersen is a senior lecturer in epidemiology and statistics at University College London. She works on a range of studies using electronic health records and is principal investigator for this project to develop new methods for multiple imputation of longitudinal data.