The Stata Journal (2009)
9, Number 2, pp. 252–264

Multiple imputation of missing values: New features for mim

Patrick Royston
Hub for Trials Methodology Research
MRC Clinical Trials Unit and University College London
London, UK
pr@ctu.mrc.ac.uk

John B. Carlin
Clinical Epidemiology and Biostatistics Unit
Murdoch Children's Research Institute and University of Melbourne
Parkville, Australia

Ian R. White
MRC Biostatistics Unit
Institute of Public Health
Cambridge, UK

Abstract. We present an update of `mim`, a program for managing multiply imputed datasets and performing inference (estimating parameters) using Rubin's rules for combining estimates from imputed datasets. The new features of particular importance are an option for estimating the Monte Carlo error (due to the sampling variability of the imputation process) in parameter estimates and in related quantities, and a general routine for combining any scalar estimate across imputations.

Keywords: st0139_1, mim, multiple imputation, missing data, missing at random, ice, MICE

1 Introduction

The presence of missing data and the consequent loss of observations from a multivariable dataset raise two potential threats: bias due to selection mechanisms that may be related to the variables of interest, and loss of precision due to the reduced sample size. In recent years, researchers have realized the importance of working with techniques that permit cases containing missing data to be used in analysis. The technique known as multiple imputation (MI) of missing observations (Rubin 1987; Schafer 1997) has gained popularity and now appears to be the dominant method.

Briefly, MI comprises two stages. First, copies of the original dataset are created, in each of which the missing values are imputed using an appropriate modeling procedure. Second, standard analyses are performed on each of these imputed datasets by using complete-data statistical methods. The results (i.e., parameter estimates that are of

substantive interest—typically, regression coefficients) are then combined according to "Rubin's rules" (Rubin 1987) to obtain a set of final estimates and standard errors (SEs).

For MI, Carlin, Galati, and Royston (2008) provided `mim`, a toolkit for performing analyses of an ensemble of datasets that includes multiple copies of the original data with imputations of missing values. The tools are based on a simple data-management paradigm in which the imputed datasets are all stored along with the original data in one dataset with a vertically stacked format. `mim` can validly fit most of the regression models available in Stata to multiply imputed datasets, giving parameter estimates and confidence intervals computed according to Rubin's rules. Additionally, `mim` provides some postestimation facilities (`testparm`, `lincom`, `predict`) with multiply imputed data and data manipulation commands (`reshape`, `append`, `merge`) for multiply imputed data.

In this article, we report on functionality that has been added to `mim` since the original publication. The key additions are an estimate of the amount of Monte Carlo (MC) (simulation) error in an estimate, a `category(combine)` feature that allows Rubin's rules to be applied at the user's discretion to just about any scalar quantity, and extensions to `mim: predict`.

2 Jackknife estimates of MC error: The mcerror option

2.1 Background

If MI were performed with an infinite number of imputations (if $m = \infty$, in standard notation), there would be no purely random contribution to the parameter estimates of a model—the "MC error" would be zero. According to Rubin's theoretical work, for finite m, the SE of an estimated regression coefficient $\widehat{\beta}$ is given, to a good approximation, by

$$\text{SE}\left(\widehat{\beta}\right) = \sqrt{W + B + B/m} \tag{1}$$

where W is the average within-imputation variance of the $\widehat{\beta}_i$ and B is the variance of the $\widehat{\beta}_i$ across imputations. As m is increased, $\text{SE}\left(\widehat{\beta}\right)$ gets smaller (on average) as the MC error gets smaller (on average). The result is an increase in precision for $\widehat{\beta}$, with a consequent increase in the absolute t statistic and reduction in the p-value for testing $\beta = 0$. The quantity $\sqrt{B/m}$, the MC SE of $\widehat{\beta}$, is itself imprecise and should be regarded only as a guide.

For practical reasons, and because there is little gain in theoretical efficiency from using larger values (Rubin 1987; Schafer 1997), small values of m (between 5 and 20) are typically used. However, users have generally paid little attention to the resulting sampling error associated with MI-based parameter estimates. Although some researchers, including Horton and Lipsitz (2001) and Graham, Olchowski, and Gilreath (2007), advocate larger values (see further comments below; it is possible informally to assess the MC error in a parameter estimate by increasing m and observing changes in $\widehat{\beta}$ and other

relevant quantities), the strategy is inefficient and computationally expensive with a large dataset. Typing `mim, mcerror` after fitting a model provides an approximate SE for the random component of each quantity in the table of results, including the parameter estimates, their SEs, and the corresponding t statistics. The SEs are computed using the jackknife approach, of which more details are given below. When coefficients are presented in exponentiated form, jackknife SEs are approximated from the SEs for untransformed coefficients by using the delta method.

2.2 The jackknife

The jackknife is used to obtain estimates of SEs that are awkward or impossible by using standard methods. According to Efron and Gong (1983), "The advantage... is an easy generalizability to any estimator." We follow the Stata 10 manual entry [R] **jackknife**. Suppose we have n observations $y_1, \ldots, y_n$ and an estimator $\widehat{\theta} = \widehat{\theta}(y_1, \ldots, y_n)$. Let $\widehat{\theta}_{(i)} = \widehat{\theta}(y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n)$ be the estimate of θ omitting the ith observation. Let

$$\psi_i = n\widehat{\theta} - (n-1)\,\widehat{\theta}_{(i)}$$

The ψ_i are known as *pseudovalues*. Then the jackknife estimator of the SE of $\widehat{\theta}$ is

$$\sigma_J = \left\{ \frac{1}{n\,(n-1)} \sum_{i=1}^{n} \psi_i^2 \right\}^{1/2} \tag{2}$$

We apply (2) to all relevant quantities in the table of regression output from `mim`. Estimates for each imputed dataset are withheld in turn to compute $\widehat{\theta}_{(i)}$ and hence (2). Here $n = m$, i.e., the sum is over imputations.

For $\widehat{\theta} = \widehat{\beta}$, a regression coefficient, Rubin's rules imply that $\widehat{\theta} = (1/m)\sum_{i=1}^{m} \widehat{\beta}_i$. It follows that $\psi_i = \widehat{\beta}_i$, i.e., the coefficient in the ith imputed dataset, and so σ_J reduces to the standard formula $\sqrt{B/m}$. However, such a simple formula does not apply to the other quantities.

2.3 Example

We revisit the breast cancer example described in Carlin, Galati, and Royston (2008, 15–17). First, `ice` (Royston 2007) is used to impute missing data in the variables `mx1`, `mx4a`, `mx5e`, `mx6`, and `mhormon`, each of which has about 20% of observations artificially missing completely at random. Five imputations (i.e., $m = 5$) are generated and stored in memory by using the `clear` option:

```
. use brcaex
(German breast cancer data)
. ice mx1 mx4a mx5e mx6 mhormon lnt _d, clear match(mx6) m(5) seed(101)
```

#missing values	Freq.	Percent	Cum.
0	231	33.67	33.67
1	290	42.27	75.95
2	126	18.37	94.31
3	33	4.81	99.13
4	6	0.87	100.00
Total	686	100.00	

Variable	Command	Prediction equation
mx1	regress	mx4a mx5e mx6 mhormon lnt _d
mx4a	logit	mx1 mx5e mx6 mhormon lnt _d
mx5e	regress	mx1 mx4a mx6 mhormon lnt _d
mx6	regress	mx1 mx4a mx5e mhormon lnt _d
mhormon	logit	mx1 mx4a mx5e mx6 lnt _d
lnt		[No missing data in estimation sample]
_d		[No missing data in estimation sample]

```
Imputing 1..2..3..4..5..[note: imputed dataset now loaded in memory]
Warning: imputed dataset has not (yet) been saved to a file
```

Next a suitable fractional polynomial model for the time-to-event outcome (recurrence-free survival) is fit to the multiply imputed dataset by using `mim`. The `nohr` option is used to force regression coefficient estimates to be displayed rather than hazard ratios:

```
. fracgen mx1 -2 -0.5
-> gen double mx1_1 = X^-2
-> gen double mx1_2 = X^-0.5
   (where: X = mx1/10)
. fracgen mx6 0.5
-> gen double mx6_1 = X^0.5
   (where: X = (mx6+1)/1000)
. mim: stcox mx1_1 mx1_2 mx4a mx5e mx6_1 mhormon, nohr
Multiple-imputation estimates (stcox)                    Imputations =       5
                                                         Minimum obs =     686
                                                         Minimum dof =     8.2
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Int.]	FMI
mx1_1	38.3469	20.7274	1.85	0.100	-9.23066	85.9245	0.743
mx1_2	-16.4381	8.60385	-1.91	0.089	-35.9189	3.04267	0.716
mx4a	.663452	.336683	1.97	0.058	-.022936	1.34984	0.385
mx5e	-1.74223	.250471	-6.96	0.000	-2.23975	-1.2447	0.213
mx6_1	-2.12092	.463089	-4.58	0.000	-3.07056	-1.17128	0.414
mhormon	-.409152	.161789	-2.53	0.017	-.740712	-.077591	0.411

The table is similar to the one in Carlin, Galati, and Royston (2008), except that the column titled `MI.df` has been replaced with one titled `FMI`. We return to that issue later.

Next we use `mim, mcerror` to give the MC SE of all quantities in the table of coefficient estimates. We must again use the `nohr` option of `mim`; otherwise, we will get SEs of exponentiated quantities.

```
. mim, mcerror nohr
Multiple-imputation estimates (stcox)                    Imputations =       5
                                                         Minimum obs =     686
                                                         Minimum dof =     8.2
[Values displayed beneath estimates are Monte Carlo jackknife standard errors]
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Int.]	FMI
mx1_1	38.3469	20.7274	1.85	0.100	-9.23066	85.9245	0.743
	7.02075	5.32528	0.59	.0969	19.176	20.2413	0.173
mx1_2	-16.4381	8.60385	-1.91	0.089	-35.9189	3.04267	0.716
	2.85307	1.91661	0.49	.0817	7.59108	6.69266	0.164
mx4a	.663452	.336683	1.97	0.058	-.022936	1.34984	0.385
	.081162	.040689	0.35	.0441	.142677	.139772	0.191
mx5e	-1.74223	.250471	-6.96	0.000	-2.23975	-1.2447	0.213
	.045424	.0226	0.54	2.9e-08	.090021	.044201	0.169
mx6_1	-2.12092	.463089	-4.58	0.000	-3.07056	-1.17128	0.414
	.115629	.050929	0.61	5.5e-04	.168925	.20287	0.157
mhormon	-.409152	.161789	-2.53	0.017	-.740712	-.077591	0.411
	.040251	.022795	0.25	.0172	.095837	.045636	0.232

We now see that each quantity in the table is subject to considerable random uncertainty, due to using only 5 imputations. We can study the effect of increasing m by rerunning the sequence of analyses with $m = 100$ instead of $m = 5$ and repeating `mim, mcerror nohr`, with the following results:

```
. mim, mcerror nohr
Multiple-imputation estimates (stcox)                    Imputations =     100
                                                         Minimum obs =     686
                                                         Minimum dof =   188.6
[Values displayed beneath estimates are Monte Carlo jackknife standard errors]
```

_t	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Int.]	FMI
mx1_1	36.4644	17.7323	2.06	0.041	1.4852	71.4435	0.554
	1.31008	.746788	0.15	.014	2.56225	1.18428	0.050
mx1_2	-15.0414	7.29785	-2.06	0.040	-29.4257	-.657102	0.510
	.517423	.275975	0.13	.0127	.475857	.962491	0.048
mx4a	.608436	.302504	2.01	0.045	.014207	1.20266	0.233
	.014514	.005255	0.05	.0056	.0159	.01966	0.027
mx5e	-1.86289	.261403	-7.13	0.000	-2.37638	-1.34941	0.232
	.012495	.004499	0.12	2.7e-12	.016655	.013954	0.026
mx6_1	-1.89201	.433128	-4.37	0.000	-2.74348	-1.04054	0.318
	.024248	.009518	0.11	7.7e-06	.030477	.031102	0.031
mhormon	-.396602	.147469	-2.69	0.007	-.686306	-.106899	0.243
	.007215	.002778	0.05	.0012	.010669	.007161	0.028

The MC error in the $\widehat{\beta}$ for `mhormon`, for example, is reduced from 10% to a more reasonable 2%. As expected, because the SE in (1) reduces on average as m increases, (most of) the t statistics have increased in absolute value.

We may use the MC errors as an indication of how secure the β estimates are and how many imputations would be needed to achieve a given precision. For example, the $\widehat{\beta}$ for `mx5e` is -1.863 with an MC SE of 0.012. If we wanted this $\widehat{\beta}$ to be accurate to two decimal places, we would need to reduce its MC SE by a factor of 2.4 to 0.005, requiring $2.4^2 = 5.76$ times as many imputations, i.e., 576 instead of 100. The magnitude of the MC error shows the major effect that missing data may have on inference. Although it will not always be feasible to perform such a large number of imputations, and the practical importance of determining the estimate to the second decimal place in an example like this is likely to be minimal, it is important to appreciate that this is a way in which missing data undermine the principle that two researchers using the same data and methods should arrive at the same estimates (at least, to a given precision). With anything other than insignificant amounts of MC error, reproducibility cannot be achieved.

To further exemplify the uncertainties in MI and how MC error can throw light on them, we repeated the breast cancer analysis in 10 independent replications, 5 with $m = 5$ and 5 with $m = 100$. The resulting $\widehat{\beta}$ estimates for `mx4a` with $m = 5$ were $(0.559, 0.626, 0.451, 0.515, 0.654)$ compared with $(0.609, 0.579, 0.611, 0.583, 0.604)$ for $m = 100$. The corresponding jackknife SEs (σ_J) were $(0.019, 0.068, 0.030, 0.087, 0.076)$ and $(0.013, 0.015, 0.013, 0.014, 0.014)$. The values of σ_J with $m = 5$ are highly variable and much less informative than with $m = 100$, for which $\sigma_J = 0.014 \pm 0.001$. The

values of $\widehat{\beta}$ with $m = 100$ vary narrowly, between 0.58 and 0.61, whereas for $m = 5$ they range between 0.45 and 0.65. It seems clear that more than 5 imputations would be needed for a serious analysis of this dataset. Finally, the p-values for `mx4a` are also instructive; with $m = 5$ they are $(0.037, 0.052, 0.088, 0.141, 0.055)$ compared with $(0.041, 0.056, 0.039, 0.050, 0.044)$ for $m = 100$. As already discussed, because of larger m increasing the precision of $\widehat{\beta}$, the p-values are somewhat lower with $m = 100$ than with $m = 5$.

We do recommend that users replicate their MI analysis at least once—that is, having selected a "reasonable" value of m, to create two independent sets of m imputations—and compare the results of running `mim`, with respect to substantive values such as $\widehat{\beta}$ and their MC errors. Running analyses with larger numbers of imputations is much cheaper than obtaining additional subjects for the research!

3 mim, category(combine)

Principally, `mim` is an engine to fit Stata regression models in several imputed datasets and apply Rubin's rules to combine estimates across imputations, also calculating appropriate SEs for them. In principle, Rubin's rules may be applied in similar fashion to any scalar quantity. However, Rubin's rules should be applied only to quantities that are estimators of well-defined parameters; they must not, for example, be applied to p-values or to Wald χ^2 or likelihood-ratio χ^2 statistics.

`mim, category(combine)` provides options for computing a scalar quantity (and, optionally, its SE) in each of m imputations and combining the m values according to Rubin's rules. To do the calculations, `mim` harnesses the power of another Stata command: `statsby`. The output from `mim, category(combine)` displays the `statsby` command that `mim` has used to do the desired analysis and estimate the scalar statistic for each positive value of `_mj`, the imputation indicator.

We illustrate using the breast cancer data. First, a dataset with $m = 5$ imputations is created as in the previous section. Suppose that we wish to predict the probability of a woman being postmenopausal as a function of age and then use the concordance index (c-index) as a measure of the strength of the association. Rubin's rules may validly be applied to the c-index because it estimates a parameter (the probability that a randomly selected postmenopausal woman is older than a randomly selected premenopausal woman). The c-index and its SE may be calculated by using the Stata command `roctab`. The variable `x2` is coded as 1 for premenopausal and 2 for postmenopausal, and `mx1` is the woman's age. `x2` is complete, whereas `mx1` is about 20% missing observations. First, we create a binary variable for menopausal status:

```
. generate byte meno = (x2 == 2)
. mim, category(combine) est(r(area)) se(r(se)): roctab meno mx1
Applying Rubin´s rules, using statsby for analysis:
-> statsby est = (r(area)) se = (r(se)), by(_mj) nodots clear: roctab meno mx1
      command:  roctab meno mx1
          est:  r(area)
           se:  r(se)
           by:  _mj
```

Combined estimate	Mean	Std. Err.	[95% Conf.	Interval]	FMI
r(area)	.887999	.0180201	.8502271	.9257709	0.513

Here `mim` requires three options: `category(combine)` to signify the type of operation, `est()` to tell `statsby` how to compute the estimate, and `se()` to tell `statsby` how to compute the SE of the estimate. (The `se()` option may be omitted, in which case only the average of the estimate is reported.) After the usual colon comes the Stata command, here `roctab meno mx1`, that `statsby` uses to do the work. `mim` tells `statsby` to execute `roctab meno mx1` in each of the m imputations, and `mim` then collects the resulting estimates returned by `statsby` in `r(area)` and the SEs in `r(se)`. The locations `r(area)` and `r(se)` have to be determined either empirically, by running the target command once and issuing a `return list` to see what it has produced, or by consulting the *Saved results* section of the help file on the target command.

The result in the above example is that the c-index is estimated as 0.888 with an SE of 0.018. `mim` stores these quantities in `r(Q)` and `r(se)`, respectively, and the lower and upper confidence limits in `r(lb)` and `r(ub)`, respectively.

A rather faster alternative to `statsby` is the user-written program `byvar` (Royston 1996). The syntax is similar. The additional option `byvar` is supplied to `mim`:

```
. mim, category(combine) est(r(area)) se(r(se)) byvar: roctab meno mx1
Applying Rubin´s rules, using byvar for analysis:
-> byvar _mj, r(area se) unique generate: roctab meno mx1
```

Combined estimate	Mean	Std. Err.	[95% Conf.	Interval]	FMI
area	.887999	.0180201	.850227	.9257709	0.513

A copy of `byvar` is provided and should be installed alongside `mim`.

Further details on how to specify the `est()` and `se()` options may be obtained from relevant parts of the help files for `mim`, `statsby`, and `byvar`.

3.1 Advanced use

For `category(combine)` to work, one needs a single command that returns the required scalar quantity and its SE in either an `r()` or an `e()` saved result. If more than one command is necessary to achieve this, the user has to write an r-class program (see `help program`) that returns the requisite quantities for use by `statsby` or `byvar`.

For example, suppose we wanted a 95% confidence interval (CI) for a Pearson correlation coefficient, R, computed in an MI dataset. The standard normalizing transformation for R is Fisher's z function (see Cox [2008]),

$$z = \frac{1}{2} \ln \frac{1+R}{1-R} = \operatorname{atanh}(R)$$

whose asymptotic variance is $1/(n-3)$, with n being the sample size. We compute z and its confidence limits, and then back-transform using the inverse function:

$$R = \frac{\exp(2z) - 1}{\exp(2z) + 1} = \tanh(z)$$

The simple program (ado-file) listed below, `fisher`, computes z and SE(z), given two variables to be correlated:

```
. program define fisher, rclass
  1.     syntax varlist(min=2 max=2) [if] [in]
  2.     quietly correlate `varlist´ `if´ `in´
  3.     return scalar z = atanh( r(rho) )
  4.     return scalar sez = sqrt( 1 / (r(N) - 3) )
  5. end
```

The `[if]` part of the syntax is essential, because `fisher` will filter the imputation indicator variable, `_mj`. The values of z and its SE are returned by `fisher` in `r(z)` and `r(sez)`, respectively.

Here is an example. Suppose for the breast cancer data that we want the correlation and its 95% CI between ln (1 + `mx6`) and `mx5e`. The necessary work, using `fisher` and `mim, category(combine)`, is as follows:

```
. use brcaeximp, clear
(German breast cancer data)
. gen mx61 = ln(1 + mx6)
(127 missing values generated)
. mim, category(combine) est(r(z)) se(r(sez)): fisher mx5e mx61
Applying Rubin´s rules, using statsby for analysis:
-> statsby est = (r(z)) se = (r(sez)),  by(_mj) nodots clear: fisher  mx5e
> mx61
      command:  fisher mx5e mx61
          est:  r(z)
           se:  r(sez)
           by:  _mj

Combined estimate |       Mean     Std. Err.      [95% Conf. Interval]    FMI
------------------+------------------------------------------------------------
r(z)              |    .1324091     .0600462       .0007296    .2640887  0.651
. display "Combined R = " %7.4f tanh(r(Q)) " 95% CI = " %7.4f tanh(r(lb)) ","
> %7.4f tanh(r(ub))
Combined R =  0.1316 95% CI =  0.0007, 0.2581
```

`mim` presents the combined value of z and its 95% CI as 0.1324 $[0.0007, 0.2641]$. The final line of the example code picks up these three quantities, saved by `mim` in `r(Q)`,

`r(lb)`, and `r(ub)`, as described above. The required CI for R is computed "manually" using Stata's `tanh()` function and is presented as 0.1316 [0.0007, 0.2581].

Correct SEs are obtained only if the sampling distribution of the estimate in complete data is normal. However, it is not known how robust this assumption is to departures from normality. In the above example, we used the Fisher transformation of R to try to ensure that we fulfilled the normality assumption and to produce plausible CIs.

4 Fraction of missing information

The fraction of missing information (FMI) is now reported for every model that `mim` fits. This quantity is an estimate of the relative loss of efficiency, or increase in variance, of a parameter estimate because of missing data (Schafer 1997). For each predictor, FMI is a function of the ratio of the between- to within-imputation variance of the estimated coefficient and the associated approximate degrees of freedom (df):

$$\text{FMI} = \left(r + \frac{2}{\text{df} + 3} \right) / (r + 1)$$

where r is the "relative increase in variance due to nonresponse" (Rubin's terminology) and is estimated by $\left(1 + m^{-1}\right) B/W$. Because df is always positive, FMI lies between 0 and 1, and because df is usually considerably larger than 3, FMI is approximately $r/(r+1)$. The larger the value of FMI, the greater the loss of information (hence loss of precision) that has been induced in the estimated coefficient by the missing data.

It is important to remember that FMI, as reported by `mim`, is an estimate. For few imputations, FMI is likely to be imprecise. Just how imprecise may be gauged using the `mim, mcerror` replay command, described above.

5 Extension of mim: predict

`mim: predict` (by default, i.e., without further options) computes the linear predictor (`xb`) from the most recently fit model and averages the predictions across imputations, storing the resulting means in the appropriate `_mj==0` locations. `mim: predict` now accepts other `predict` options, according to the `category(fit)` command that has been used. With `stcox`, for example, the `hr` option of `predict` gives the mean relative hazard for each individual across imputations, that is, an estimate of the hazard for each individual relative to the baseline hazard function:

```
. mim: stcox mx1_1 mx1_2 mx4a mx5e mx6_1 mhormon, nohr
  (output omitted)
. mim: predict hr, hr
[predicting hr]
```

6 Minor extensions

6.1 Saved results from mim: testparm

The `testparm` command is a standard postestimation feature of most of Stata's regression commands. It applies a Wald test to (a subset of) the covariates fit in a model. If the null distribution of the test statistic is assumed to be chi-squared on d df, then `testparm` returns the chi-squared statistic in `r(chi2)` and the df in `r(df)`; if it is assumed to be F on d_1, d_2 df, then `testparm` returns the F statistic in `r(F)`, d_1 in `r(df)`, and d_2 in `r(df_r)`. In both cases, the p-value is returned in `r(p)`.

`mim: testparm` mimics `testparm` with MI data. Because theory suggests that the null distribution of the Wald statistic in MI is best approximated by the F distribution, `mim: testparm` returns the same quantities as `testparm` does in the F distribution case. For example,

```
mim: testparm mx4a mhormon
 ( 1)  mx4a = 0
 ( 2)  mhormon = 0

       F(  2,  25.9) =    5.00
            Prob > F =    0.0145
. return list
scalars:
               r(df_r) =  25.9190852241141
                  r(F) =  5.0029667920515
                 r(df) =  2
                  r(p) =  .0145422160391562
```

Because `mim: testparm` performs an F test with denominator df, `r(df_r)`, estimated from the data, even in cases (such as logistic regression) in which a chi-squared test statistic would be expected with complete data, an F test is reported.

6.2 from() and to() options

The `from()` and `to()` options simply limit the range of imputation numbers to which a given `mim` command is to be applied; for example,

```
. mim, from(2) to(4): stcox mx1_1 mx1_2 mx4a mx5e mx6_1 mhormon, nohr
```
(*output omitted*)

would estimate a Cox model using imputations 2, 3, and 4 only. These options may be pedagogically useful; e.g., one could create a dataset with 50 imputations and run `mim` in batches of 5 imputations to study the variability of the resulting estimates.

7 Conclusion

As with `ice` (Royston 2007), development of `mim` continues as new features are requested by users or considered by the authors to be worthwhile. In particular, the `category(combine)` and `mcerror` features should prove to be useful extensions to the practical use of MI.

8 References

Carlin, J. B., J. C. Galati, and P. Royston. 2008. A new framework for managing and analyzing multiply imputed data in Stata. *Stata Journal* 8: 49–67.

Cox, N. J. 2008. Speaking Stata: Correlation with confidence, or Fisher's z revisited. *Stata Journal* 8: 413–439.

Efron, B., and G. Gong. 1983. A leisurely look at the bootstrap, the jackknife, and cross-validation. *American Statistician* 37: 36–48.

Graham, J. W., A. E. Olchowski, and T. D. Gilreath. 2007. How many imputations are really needed? Some practical clarifications of multiple imputation theory. *Prevention Science* 8: 206–213.

Horton, N. J., and S. R. Lipsitz. 2001. Multiple imputation in practice: Comparison of software packages for regression models with missing variables. *American Statistician* 55: 244–254.

Royston, P. 1996. ip9: Repeat Stata command by variable(s). *Stata Technical Bulletin* 27: 3–5. Reprinted in *Stata Technical Bulletin Reprints*, vol. 5, pp. 67–69. College Station, TX: Stata Press.

———. 2007. Multiple imputation of missing values: Further update of ice, with an emphasis on interval censoring. *Stata Journal* 7: 445–464.

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.

About the authors

Patrick Royston is a medical statistician with 30 years of experience who has a strong interest in biostatistical methods and in statistical computing and algorithms. He now works in cancer clinical trials and related research issues. Currently, he is focusing on problems of model building and validation with survival data, including prognostic factor studies; on parametric modeling of survival data; on multiple imputation of missing values; and on novel clinical trial designs.

John Carlin is a biostatistician with experience across a wide range of collaborative research relating mainly to child and adolescent health; he has current methodological research interests in the handling of missing data in large longitudinal studies. He is director of the Clinical

Epidemiology and Biostatistics Unit, Murdoch Children's Research Institute, at the Royal Children's Hospital in Melbourne, Australia, and has professorial appointments in the Department of Paediatrics and School of Population Health at the University of Melbourne.

Ian White is a senior statistician at the MRC Biostatistics Unit in Cambridge, UK. His research interests include missing data, noncompliance and measurement error in clinical trials, observational studies, and meta-analysis.