



The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

THE STATA JOURNAL

Editors

H. JOSEPH NEWTON
Department of Statistics
Texas A&M University
College Station, Texas
editors@stata-journal.com

NICHOLAS J. COX
Department of Geography
Durham University
Durham, UK
editors@stata-journal.com

Associate Editors

CHRISTOPHER F. BAUM, Boston College
NATHANIEL BECK, New York University
RINO BELLOCCO, Karolinska Institutet, Sweden, and
University of Milano-Bicocca, Italy
MAARTEN L. BUIS, WZB, Germany
A. COLIN CAMERON, University of California–Davis
MARIO A. CLEVES, University of Arkansas for
Medical Sciences
WILLIAM D. DUPONT, Vanderbilt University
PHILIP ENDER, University of California–Los Angeles
DAVID EPSTEIN, Columbia University
ALLAN GREGORY, Queen’s University
JAMES HARDIN, University of South Carolina
BEN JANN, University of Bern, Switzerland
STEPHEN JENKINS, London School of Economics and
Political Science
ULRICH KOHLER, WZB, Germany

FRAUKE KREUTER, Univ. of Maryland–College Park
PETER A. LACHENBRUCH, Oregon State University
JENS LAURITSEN, Odense University Hospital
STANLEY LEMESHOW, Ohio State University
J. SCOTT LONG, Indiana University
ROGER NEWSON, Imperial College, London
AUSTIN NICHOLS, Urban Institute, Washington DC
MARCELLO PAGANO, Harvard School of Public Health
SOPHIA RABE-HESKETH, Univ. of California–Berkeley
J. PATRICK ROYSTON, MRC Clinical Trials Unit,
London
PHILIP RYAN, University of Adelaide
MARK E. SCHAFER, Heriot-Watt Univ., Edinburgh
JEROEN WEESIE, Utrecht University
NICHOLAS J. G. WINTER, University of Virginia
JEFFREY WOOLDRIDGE, Michigan State University

Stata Press Editorial Manager

LISA GILMORE

Stata Press Copy Editors

DAVID CULWELL and DEIRDRE SKAGGS

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-STAT-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	
1-year subscription	\$ 79	1-year subscription	\$115
2-year subscription	\$155	2-year subscription	\$225
3-year subscription	\$225	3-year subscription	\$329
3-year subscription (electronic only)	\$210	3-year subscription (electronic only)	\$210
1-year student subscription	\$ 48	1-year student subscription	\$ 79
1-year university library subscription	\$ 99	1-year university library subscription	\$135
2-year university library subscription	\$195	2-year university library subscription	\$265
3-year university library subscription	\$289	3-year university library subscription	\$395
1-year institutional subscription	\$225	1-year institutional subscription	\$259
2-year institutional subscription	\$445	2-year institutional subscription	\$510
3-year institutional subscription	\$650	3-year institutional subscription	\$750

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.



Copyright © 2012 by StataCorp LP

Copyright Statement: The *Stata Journal* and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LP. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

The articles appearing in the *Stata Journal* may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the *Stata Journal*, in whole or in part, on publicly accessible websites, file servers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the *Stata Journal* or the supporting files understand that such use is made without warranty of any kind, by either the *Stata Journal*, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the *Stata Journal* is to promote free communication among Stata users.

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata, **STATA**, Stata Press, Mata, **MATA**, and NetCourse are registered trademarks of StataCorp LP.

Diagnostics for multiple imputation in Stata

Wesley Eddings
StataCorp
College Station, TX
weddings@stata.com

Yulia Marchenko
StataCorp
College Station, TX
ymarchenko@stata.com

Abstract. Our new command `midiaplots` makes diagnostic plots for multiple imputations created by `mi impute`. The plots compare the distribution of the imputed values with that of the observed values so that problems with the imputation model can be corrected before the imputed data are analyzed. We include an example and suggest extensions to other diagnostics.

Keywords: `st0263`, `midiaplots`, multiple imputation, diagnostics, model checking, imputed values, missing data, missing at random

1 Introduction

Multiple imputation ([Rubin 1987](#)) is a principled method for handling missing data, but it relies on a model for imputing the missing values. An inappropriate imputation model can lead to biased estimates, so it is important to check the model. A few simple checks are now available in our command `midiaplots`. Most of the methods are graphical, but there are also Kolmogorov–Smirnov tests for comparing the distribution of the observed values with the distribution of the imputed values. The foremost reference for the diagnostics is [Abayomi, Gelman, and Levy \(2008\)](#).

1.1 Methods for handling missing data

The theory of missing data assumes that missingness follows a probability model so that we may speak of the probability that data are missing. The probability model assumed to create the missing values is called the missing-data mechanism. To analyze incomplete data, we must make assumptions about the missing-data mechanism.

The default missing-data analysis in Stata is complete-case analysis, which makes a strong assumption about the missing-data mechanism. A complete-case analysis omits every observation that has a missing value for any of the model variables. So if we type `regress y x1 x2`, Stata will omit any observation that has a missing value for `y`, `x1`, or `x2`. Such omission is typically justified only if the data are missing completely at random (MCAR), the most stringent missing-data mechanism. The data are MCAR only if the missing values are like a simple random sample of all values so that missingness is not correlated with any variable, observed or unobserved.

MCAR is a severe restriction, and complete-case analysis may be biased if the data are not MCAR. A less restrictive method is multiple imputation, which may be performed under a weaker assumption, missing at random (MAR). An MAR mechanism

allows missingness to be correlated with observed variables so long as it remains conditionally independent of the unobserved values. So the observed variables must suffice for predicting missingness. If the MAR assumption does not hold, resulting in missingness being correlated with the unobserved values even after conditioning on the observed values, the data are said to be missing not at random (MNAR) or nonignorably missing. For rigorous definitions of the missing-data mechanisms, see [Little and Rubin \(2002, sec. 1.3\)](#).

1.2 Multiple imputation in Stata

Multiple imputation imputes each missing value multiple times. A regression model is created to predict the missing values from the observed values, and multiple predicted values are generated for each missing value to create the multiple imputations. Each imputation is a separate, filled-in dataset that can be analyzed on its own with standard methods. The separate results are then combined to produce a single multiple-imputation result. The method accounts for the uncertainty in the imputed values provided that the imputation and analysis models are appropriate. For more information, please see the documentation entry [MI] **intro substantive** and its references.

Multiple imputation was first added to Stata in the user-written packages `mitools`, `ice`, and `mim` ([Carlin et al. 2003](#); [Royston 2004](#); [Carlin, Galati, and Royston 2008](#)). The official `mi` commands were introduced in Stata 11 and expanded in Stata 12. The key commands are `mi impute`, for creating multiple imputations; `mi estimate`, for analyzing the multiple imputations; and special commands for managing the multiply imputed datasets. For more information on multiple imputation in Stata, type `help mi`.

`mi impute` requires the data to be MAR, so the missing values can be imputed using only the observed values and an imputation model.¹ The MAR assumption is not testable, because it is not possible to check the distribution of the unobserved values. But if we tentatively assume MAR, it is possible to check the imputation model.

Our new command `midiaplots` helps check the fit of an imputation model. The command compares the imputed values with the observed ones, so implausible imputed values may be detected before the primary analysis. For continuous variables, there are three graphical methods (cumulative distribution functions, kernel density estimates, and histograms) and Kolmogorov–Smirnov tests; for categorical variables, there are both graphs and tables (of proportions or frequencies).

1. Multiple imputation itself does not require the MAR assumption. If the data are MNAR though, the probability model for the missing-data mechanism must be incorporated in the imputation model.

2 The `middiagplots` command

2.1 Syntax

```
middiagplots [ impvars ] [ if ] [ , m(numlist) plotype(plotspec)
      sample(plotsample) ncategories(#) separate combine by(varlist)
      sort(varlist) tabfreq sleep(#) more ksmirnov nograph notable
      plot1opts(plotopts) plot2opts(plotopts) plot3opts(plotopts) graph_options ]
```

2.2 Description

`middiagplots` performs diagnostics for multiply imputed data. By default, the command plots the distributions of continuous variables and tabulates categorical variables. A variable is considered categorical if it has no more than five distinct observed values; use the `ncategories()` option to specify a different number of values.

The diagnostics compare the distributions of the observed, imputed, and completed values. (The completed data combine the observed and imputed data.) If the distributions differ greatly (possibly after conditioning on predictors of missingness), we may suspect a problem with the imputation model.

By default, there is one plot or table per imputed variable per imputation. A typical command is

```
. middiagplots age income, m(1/5) sample(all) plotype(cumul)
```

The option `m(1/5)` requests diagnostics for the first five imputations; the other options specify `all` samples (observed, imputed, and completed) and plots of cumulative distribution functions. (The choice for `sample()` is the default; we are assuming that `age` and `income` are continuous.)

Each plot would show three overlaid cumulative distribution functions, one for each of the observed, imputed, and completed samples. There would be 10 plots, 5 for `age` and 5 for `income`. By default, `middiagplots` shows one plot at a time and waits for three seconds before going on to the next plot; you can change the waiting time with the `sleep()` option. Or you can specify `more`, which pauses after each plot until you press a key. You can `combine` the plots across imputations into one figure or `separate` the samples for each imputation into separate plots. The `separate` option is especially useful for `plotype(histogram)`.

For categorical variables, `middiagplots` displays tables. Each table shows the distributions of the observed, imputed, and completed samples. Proportions are shown by default; to see frequencies instead, use the `tabfreq` option. Use `plotype(histogram)` to supplement the tables with histograms. (For categorical variables, the other plot types are not available.)

The most useful multiple-imputation diagnostics are graphical, but `middiagplots` also includes significance tests. The `ksmirnov` option uses the Kolmogorov–Smirnov

test to compare the observed and imputed distributions; a significant result means that the distribution of the imputed data differs significantly from that of the observed data. The results of `ksmirnov` should not be taken too seriously though, because the imputed data are not independent of the observed data and because the distributions will differ for MAR data even if the imputation model is correct (Abayomi, Gelman, and Levy 2008, 280).

If no *impvars* are specified, the command defaults to all variables registered as imputed.

`midiaplots` works only with `mi` data that have been `mi set`. Data from `ice` may be converted to `mi` data with the official command `mi import ice`.

2.3 Options

`m(numlist)` specifies which imputations to use. The default is `m(1)`.

`plottype(plotspec)` specifies the type of plot. *plotspec* is one of

`kdensity` [*, kden_opts*] | `histogram` [*, hist_opts*] | `cumul` [*, cumul_opts*]

`kdensity` requests kernel density estimates, the default. *kden_opts* are any of the options allowed by `twoway kdensity`; see [G-2] **graph twoway kdensity**.

`histogram` requests histograms. The `discrete` option is automatically applied to categorical variables. *hist_opts* are any of the options allowed by `twoway histogram`; see [G-2] **graph twoway histogram**.

`cumul` requests plots of cumulative distribution functions. Plots of cumulative distribution functions may be desirable because they do not require tuning, unlike histograms and kernel density estimates (which are affected by the number of bins or the bandwidth). *cumul_opts* are `freq`, `equal`, and *connect_options*; see [R] **cumul** and [G-3] *connect_options*.

Options specified in *plotspec* are applied to each plot.

`sample(plotsample)` specifies which samples to plot. *plotsample* may be `all` or any combination of `observed`, `imputed`, and `completed`. The default is `sample(all)`. The option does not affect tables for categorical variables, which always show all three samples.

`ncategories(#)` specifies that variables with no more than `#` distinct values should be considered categorical. The default is `ncategories(5)`.

`separate` requests a separate plot for each *plotsample*; the separate plots are presented in one figure. By default, the distributions are instead overlaid onto one plot. `separate` may not be specified with `combine` or with `twoway's legend()` option.

`combine` combines all of a variable's imputation plots into one figure. `combine` implies all imputations, unless `m()` is specified. `combine` may not be specified with `separate`.

`by(varlist)` requests separate diagnostics for the subgroups defined by *varlist*; also see the `by()` option of `twoway`.

`sort(varlist)` sorts the data on the variables in *varlist*. Without sorting, plots may depend slightly on the active `mi` style if there are tied observations. If there are no ties, sorting has no effect.

`tabfreq` requests that tables display frequencies instead of proportions. Plots are not affected.

`sleep(#)` specifies a length of *#* milliseconds between the plots. The default is `sleep(3000)`.

`more` causes Stata to pause after each plot until you press a key.

`ksmirnov` requests Kolmogorov–Smirnov statistics comparing the observed and imputed distributions of each continuous variable in *impvars*. Tests are not reported for categorical variables. `ksmirnov` may not be combined with `by()`.

`nograph` suppresses all graphs and is intended for use with `ksmirnov`.

`notable` suppresses the tables produced by default for categorical variables.

`plot1opts(plotopts)` modifies the plot of the observed values.

`plot2opts(plotopts)` modifies the plot of the imputed values.

`plot3opts(plotopts)` modifies the plot of the completed values.

plotopts are any of the options documented in [G-3] **connect_options**, [G-2] **graph twoway histogram**, or [G-2] **graph twoway kdensity** applicable to the specified `plottype()`.

graph_options specify the overall look of the graph. If the `separate` option is used, then *graph_options* are any of the options documented in [G-2] **graph combine**. Otherwise, *graph_options* are *twoway_options*—any of the options documented in [G-3] **twoway_options**.

3 Example

We will use a study of breast cancer that has illustrated multiple imputation in several other *Stata Journal* articles (Royston 2004; Carlin, Galati, and Royston 2008). There are 686 patients, and the outcome is recurrence-free survival. The data were modified by Royston (2004, 234) to have 20% of the values MCAR. We want to impute the missing predictors, check the imputations, and fit a model to predict recurrence-free survival. Because we are imputing survival data, the imputation model should include as predictors the censoring indicator and the Nelson–Aalen estimate of the cumulative hazard (White, Royston, and Wood 2011, 384). The Nelson–Aalen estimate is available in `sts generate`.


```

. use brcaex
(German breast cancer data)
. stset rectime, failure(censrec)
      failure event:  censrec != 0 & censrec < .
obs. time interval:  (0, rectime]
exit on or before:   failure

```

```

      686 total obs.
      0   exclusions

```

```

      686 obs. remaining, representing
      299 failures in single record/single failure data
      771400 total analysis time at risk, at risk from t =      0
               earliest observed entry t =      0
               last observed exit t =      2659

```

```

. sts generate cumhaz = na

```

Carlin, Galati, and Royston (2008, 62–64) used `ice` to impute the five missing predictors `mx1`, `mx4a`, `mx5e`, `mx6`, and `mhormon`. We will use `mi impute chained`.

```

. mi set wide
. mi register imputed mx1 mx4a mx5e mx6 mhormon
. set seed 912346
. mi impute chained (regress) mx1 mx5e (pmm) mx6
> (logit) mx4a mhormon = _d cumhaz, add(5)
Conditional models:
      mx6: pmm mx6 i.mx4a i.mhormon mx1 mx5e _d cumhaz
      mx4a: logit mx4a mx6 i.mhormon mx1 mx5e _d cumhaz
      mhormon: logit mhormon mx6 i.mx4a mx1 mx5e _d cumhaz
      mx1: regress mx1 mx6 i.mx4a i.mhormon mx5e _d cumhaz
      mx5e: regress mx5e mx6 i.mx4a i.mhormon mx1 _d cumhaz
Performing chained iterations ...
Multivariate imputation          Imputations =      5
Chained equations                added =      5
Imputed: m=1 through m=5        updated =      0
Initialization: monotone        Iterations =     50
                                burn-in =     10

      mx1: linear regression
      mx5e: linear regression
      mx6: predictive mean matching
      mx4a: logistic regression
      mhormon: logistic regression

```

Variable	Observations per <i>m</i>			
	Complete	Incomplete	Imputed	Total
mx1	554	132	132	686
mx5e	538	148	148	686
mx6	559	127	127	686
mx4a	557	129	129	686
mhormon	557	129	129	686

(complete + incomplete = total; imputed is the minimum across *m* of the number of filled-in observations.)

Figure 1 gives a diagnostic plot for `mx1`, patients' ages in years:

```
. midiagplots mx1
(M = 5 imputations)
(imputed: mx1 mx4a mx5e mx6 mhormon)
```

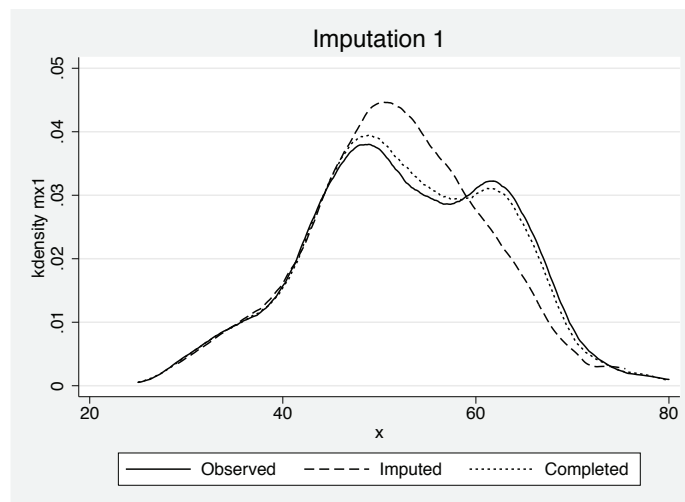


Figure 1. Diagnostic plot for patients' ages in years

In figure 1, the observed ages are bimodal, but the imputed values are unimodal (as we would expect from a linear imputation model with normal errors). To correct the discrepancy, we may reimpute the variable by predictive mean matching (`mi impute pmm`), which does not assume normality. For more information, please see the documentation entry [MI] **`mi impute pmm`**.

```
. mi impute chained (regress) mx5e (pmm) mx1 mx6 (logit) mx4a
> mhormon = _d cumhaz, replace
Conditional models:
      mx6: pmm mx6 i.mx4a i.mhormon mx1 mx5e _d cumhaz
      mx4a: logit mx4a mx6 i.mhormon mx1 mx5e _d cumhaz
      mhormon: logit mhormon mx6 i.mx4a mx1 mx5e _d cumhaz
      mx1: pmm mx1 mx6 i.mx4a i.mhormon mx5e _d cumhaz
      mx5e: regress mx5e mx6 i.mx4a i.mhormon mx1 _d cumhaz
Performing chained iterations ...
Multivariate imputation          Imputations =      5
Chained equations                added =      0
Imputed: m=1 through m=5        updated =      5
Initialization: monotone        Iterations =     50
                                burn-in =     10

      mx5e: linear regression
      mx1: predictive mean matching
      mx6: predictive mean matching
      mx4a: logistic regression
      mhormon: logistic regression
```

Variable	Observations per m			Total
	Complete	Incomplete	Imputed	
mx5e	538	148	148	686
mx1	554	132	132	686
mx6	559	127	127	686
mx4a	557	129	129	686
mhormon	557	129	129	686

(complete + incomplete = total; imputed is the minimum across m of the number of filled-in observations.)

```
. midiagplots mx1
(M = 5 imputations)
(imputed: mx1 mx4a mx5e mx6 mhormon)
```

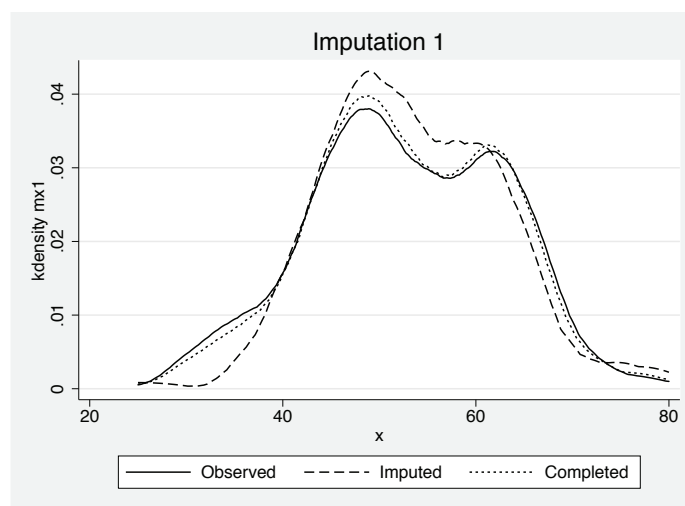


Figure 2. Patients' ages imputed by predictive mean matching

We see in figure 2 that ages (mx1) imputed by method `pmm` no longer have a unimodal distribution, resulting in a distribution that more closely matches the observed distribution. We did not specify the `knn()` option, so `mi impute pmm` used the default setting of one “nearest neighbor”. Using more than one nearest neighbor would decrease variance but increase bias.

The imputations for variable `mx5e` could also be improved because they include values that are impossible in the observed data (figure 3):

```
. midiagplots mx5e, plotype(histogram) separate
(M = 5 imputations)
(imputed: mx1 mx4a mx5e mx6 mhormon)
```

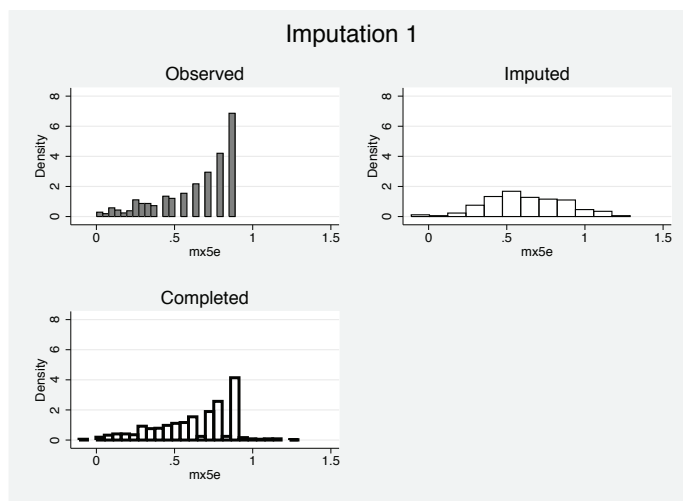


Figure 3. The imputed values of `mx5e` lie outside $(0, 1)$

The variable `mx5e` in figure 3 is an exponential transformation $f(x) = \exp(-0.12x)$ of the patient's number of positive lymph nodes, and the transformation always produces observed data between 0 and 1. (Every patient in the study had at least one positive node.) The imputation model, however, does not respect the bounds, and some of the imputed values lie outside $(0, 1)$.

To handle the outlying imputed values, we will reimpute `mx5e` by using predictive mean matching. Method `pmm` guarantees that the imputed values lie within the extremes of the observed data.

```
. mi impute chained (pmm) mx1 mx6 mx5e (logit) mx4a
> mhormon = _d cumhaz, replace
Conditional models:
      mx6: pmm mx6 i.mx4a i.mhormon mx1 mx5e _d cumhaz
      mx4a: logit mx4a mx6 i.mhormon mx1 mx5e _d cumhaz
      mhormon: logit mhormon mx6 i.mx4a mx1 mx5e _d cumhaz
      mx1: pmm mx1 mx6 i.mx4a i.mhormon mx5e _d cumhaz
      mx5e: pmm mx5e mx6 i.mx4a i.mhormon mx1 _d cumhaz
Performing chained iterations ...
Multivariate imputation          Imputations =      5
Chained equations                added =      0
Imputed: m=1 through m=5        updated =      5
Initialization: monotone        Iterations =     50
                                burn-in =     10

      mx1: predictive mean matching
      mx6: predictive mean matching
      mx5e: predictive mean matching
      mx4a: logistic regression
      mhormon: logistic regression
```

Variable	Observations per <i>m</i>			
	Complete	Incomplete	Imputed	Total
mx1	554	132	132	686
mx6	559	127	127	686
mx5e	538	148	148	686
mx4a	557	129	129	686
mhormon	557	129	129	686

(complete + incomplete = total; imputed is the minimum across *m* of the number of filled-in observations.)

```
. midiagplots mx5e, plotype(histogram) separate xscale(range(0 1))
(M = 5 imputations)
(imputed: mx1 mx4a mx5e mx6 mhormon)
```

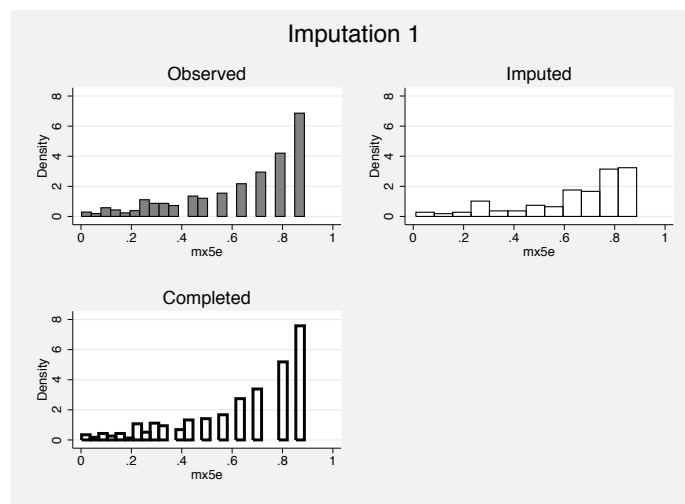


Figure 4. The revised imputations for `mx5e` lie in $(0, 1)$

The imputations for `mx5e` in figure 4 now lie in $(0, 1)$.

There is another way to handle the transformation: instead of imputing the transformed variable `mx5e`, we could impute `mx5`, the untransformed number of positive nodes, and then transform the imputations by using $f(x) = \exp(-0.12x)$ to produce values lying in $(0, 1)$.

We can use the `combine` option to check all imputations for variable `mx6` (concentration of progesterone receptors). The `combine` option combines all imputations into one graph shown in figure 5:

```
. midiagplots mx6, combine
(M = 5 imputations)
(imputed: mx1 mx4a mx5e mx6 mhormon)
(all imputations assumed with combine)
```

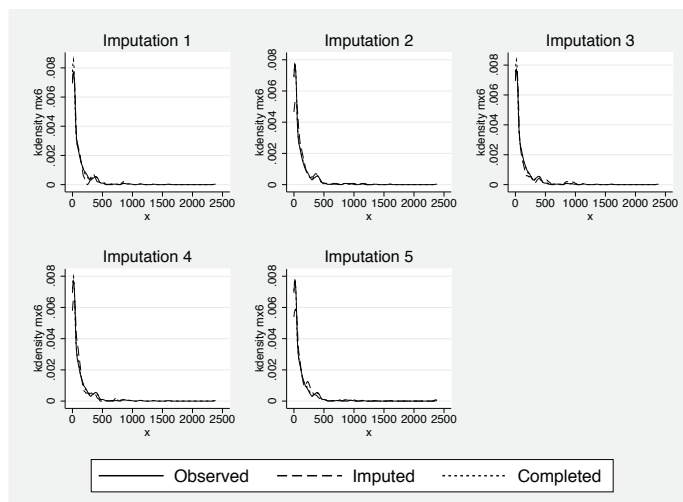


Figure 5. Kernel density estimates for progesterone receptor concentration (`mx6`) for all imputations

So far we have examined only the continuous variables, but `midiagplots` supports categorical variables too. By default, there is no graph; instead, the command tabulates the observed, imputed, and completed distributions. Let us tabulate the binary variables `mx4a` (tumor grade) and `mhormon` (hormonal therapy) for the fourth imputation:

```
. midiagplots mx4a mhormon, m(4)
(M = 5 imputations)
(imputed: mx1 mx4a mx5e mx6 mhormon)
Proportions of mx4a for m=4
Number of observed =      557
Number of imputed  =      129
Number of completed =      686
```

mx4a	Observed	Imputed	Completed
0	0.102	0.093	0.101
1	0.898	0.907	0.899

```
Proportions of mhormon for m=4
```

```
Number of observed =      557
Number of imputed  =      129
Number of completed =      686
```

mhormon	Observed	Imputed	Completed
0	0.643	0.628	0.640
1	0.357	0.372	0.360

For each variable, the three distributions are similar. To tabulate frequencies instead of proportions, use the `tabfreq` option.

Once we are satisfied with our imputation model, we can fit an analysis model with `mi estimate: stcox`.

```
. mi estimate: stcox mx1 mx6 mx5e i.mx4a i.mhormon
Multiple-imputation estimates      Imputations      =      5
Cox regression: Breslow method for ties      Number of obs      =     686
                                      Average RVI      =     0.3986
                                      Largest FMI      =     0.4888
DF adjustment: Large sample      DF:      min      =     20.53
                                      avg      =     251.75
                                      max      =     700.07
Model F test:      Equal FMI      F( 5, 173.8) =     15.51
Within VCE type:      OIM      Prob > F      =     0.0000
```

_t	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
mx1	.0039181	.0075758	0.52	0.609	-.0115861	.0194222
mx6	-.0021616	.0005791	-3.73	0.000	-.0032986	-.0010245
mx5e	-1.903025	.2836718	-6.71	0.000	-2.481607	-1.324442
1.mx4a	.6920571	.2801742	2.47	0.014	.1415335	1.242581
1.mhormon	-.3603974	.1698982	-2.12	0.046	-.7142081	-.0065867

Carlin, Galati, and Royston (2008, 64) used fractional polynomials to model the variables `mx1` and `mx6`. For the other variables, our estimates are similar to theirs.

4 Conclusion

`midiaplots` adds to Stata several multiple-imputation diagnostics, and the command may be extended as new diagnostics are published. Extensions may include plots of fitted values and residuals (Abayomi, Gelman, and Levy 2008; Marchenko and Eddings 2011), propensity score diagnostics (Raghunathan and Bondarenko 2007), and cross-validation (Gelman, King, and Liu 1998, 853–855).

`midiaplots` and other diagnostics can help check an imputation model provided that the data are MAR. But the diagnostics cannot check the MAR assumption itself. If the assumption is in doubt, an MNAR model may be used.

MNAR models are not identifiable though, for the same reason that the MAR assumption is not testable. So it is important to perform a sensitivity analysis—to make assumptions to identify the MNAR model and then vary the assumptions to see how the conclusions change. For an introduction to MNAR selection models and pattern-mixture models, see chapter 10 of Enders (2010).

5 References

- Abayomi, K., A. Gelman, and M. Levy. 2008. Diagnostics for multivariate imputations. *Journal of the Royal Statistical Society, Series C* 57: 273–291.
- 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.
- Carlin, J. B., N. Li, P. Greenwood, and C. Coffey. 2003. Tools for analyzing multiple imputed datasets. *Stata Journal* 3: 226–244.
- Enders, C. K. 2010. *Applied Missing Data Analysis*. New York: Guilford Press.
- Gelman, A., G. King, and C. Liu. 1998. Not asked and not answered: Multiple imputation for multiple surveys. *Journal of the American Statistical Association* 93: 846–857.
- Little, R. J. A., and D. B. Rubin. 2002. *Statistical Analysis with Missing Data*. 2nd ed. Hoboken, NJ: Wiley.
- Marchenko, Y. V., and W. Eddings. 2011. A note on how to perform multiple-imputation diagnostics in Stata. <http://www.stata.com/users/ymarchenko/midiagnote.pdf>.
- Raghunathan, T., and I. Bondarenko. 2007. Diagnostics for multiple imputations. Working Paper. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1031750.
- Royston, P. 2004. Multiple imputation of missing values. *Stata Journal* 4: 227–241.
- Rubin, D. B. 1987. *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley.
- White, I. R., P. Royston, and A. M. Wood. 2011. Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine* 30: 377–399.

About the authors

Wes Eddings is a senior statistician at StataCorp.

Yulia Marchenko is the director of biostatistics at StataCorp.