A note on how to perform multiple-imputation diagnostics in Stata

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

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

Abstract. Multiple-imputation (MI) diagnostics are an important step of multiple-imputation analysis. We present a short tutorial on performing multiple-imputation diagnostics in Stata.

1 Introduction

Multiple imputation is a principled statistical method for handling missing data, but like any other method, it relies on certain assumptions. One assumption, that the data are missing at random, is not testable. But if we tentatively assume the data are missing at random, other assumptions are testable. We can check, for instance, that the imputation models fit the observed data well, and that the imputed values themselves are reasonable (e.g., Abayomi, Gelman, and Levy [2008], Raghunathan and Bondarenko [2007]).

We are working on a new unofficial command `midiagplots` to implement some of the MI diagnostics in Stata. In the meantime you can perform them manually, as we explain in this note.

Here are the steps for performing MI diagnostics manually:

- Convert the `mi` data to the `wide` style, so that imputations will be stored in extra variables and not in extra observations. The `wide` style makes it easier to compare imputed values to observed values.

- Create indicator variables to flag the observations with missing values.

- Fit the imputation models to the observed data with a non-`mi` estimation command like `regress`, and use regression diagnostics to check the fit. (The models should not be used for imputation if they fit the observed data poorly.)

- Use `mi impute` to impute the missing values.

- Check that the imputed values are reasonable. The key command is `mi xeq`, which executes a given command on particular imputations.

2 Example

Consider the heart-attack example from *First use* under *Remarks* in **[MI] mi impute monotone**:

```
. webuse mheart5s0
(Fictional heart attack data; bmi and age missing)
. mi describe
  Style:  mlong
          last mi update 30mar2011 12:46:48, 22 days ago
  Obs.:   complete          126
          incomplete         28  (M = 0 imputations)
          ------------------------
          total             154
  Vars.:  imputed:  2; bmi(28) age(12)
          passive:  0
          regular:  4; attack smokes female hsgrad
          system:   3; _mi_m _mi_id _mi_miss
         (there are no unregistered variables)
```

Our data are already `mi set`. There are two variables registered as imputed: `age` and `bmi`. The remaining variables are complete and are registered as regular. Our primary analysis is a logistic regression of heart attacks on smoking, adjusted for other factors such as age, body mass index, and gender.

We want to include all available data on heart attacks and smoking in our analysis, so we'll use multiple imputation to fill in the missing values of `age` and `bmi`. According to `mi misstable nested`, the variables `age` and `bmi` follow a monotone missing-data pattern:

```
. mi misstable nested
     1.  age(12) -> bmi(28)
```

The pattern is monotone because `age` is missing only when `bmi` is missing, so we'll use `mi impute monotone` to fill in missing values of `age` and `bmi`.

Our first step is to build an imputation model. We start with a simple model. `age` and `bmi` are continuous variables so we choose to impute them using linear regression, and we use all of our complete variables as predictors. Here are the imputation models we'll fit:

```
. mi impute monotone (regress) age bmi = attack smokes hsgrad female, dryrun
Conditional models:
               age: regress age attack smokes hsgrad female
               bmi: regress bmi age attack smokes hsgrad female
```

The `dryrun` option displays the imputation models but does not fit them. There is one equation for each imputed variable; `age` will be imputed first because it has the fewest

missing values. Then the imputed `age` variable will be used as a predictor to impute `bmi`.

Now that we have our imputation models, we are ready to check if they fit the observed data well. For the purpose of illustration, we only check the imputation model for `age` below. We'll demonstrate another MI diagnostic using the imputation model for `bmi` later.

Following the steps from the introduction, we convert to the `wide` style:

```
. mi convert wide
```

and create indicator variables identifying the missing values in `age` and `bmi` for later use:

```
. qui mi xeq: generate byte Mis_age = missing(age)
. qui mi xeq: generate byte Mis_bmi = missing(bmi)
. mi register regular Mis_age Mis_bmi
```

The variables could also be created in the observed data with `misstable summarize, generate()`.

Now we fit the model for `age` to the observed data and plot the residuals against the fitted values with `rvfplot`. (A list of diagnostic commands is in [**R**] **regress postestimation**.)

```
. regress age attack smokes hsgrad female
```

Source	SS	df	MS	Number of obs =	142
				F(4, 137) =	1.20
Model	643.790243	4	160.947561	Prob > F =	0.3116
Residual	18300.7516	137	133.582129	R-squared =	0.0340
				Adj R-squared =	0.0058
Total	18944.5419	141	134.358453	Root MSE =	11.558

age	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
attack	3.51913	2.061631	1.71	0.090	-.5576042	7.595864
smokes	.5643832	2.093235	0.27	0.788	-3.574845	4.703612
hsgrad	-1.419099	2.236254	-0.63	0.527	-5.841138	3.00294
female	1.759922	2.254749	0.78	0.436	-2.69869	6.218534
_cons	55.18863	2.279613	24.21	0.000	50.68086	59.69641

```
. rvfplot
```

Figure 1 does not indicate that the model fit is poor.

If a model does not fit the observed data well, we should try to improve its fit before we impute. We may, for example, include additional predictors, or use [**R**] **mfp** to determine a more appropriate functional form for the existing predictors.

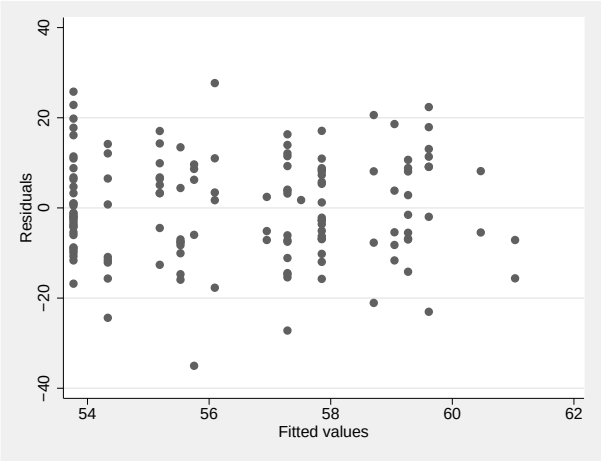

Figure 1: Residual versus fitted plot for `age`

Once we are comfortable with the observed-data model fit we can proceed with imputation.

```
. mi impute monotone (regress) age bmi = attack smokes hsgrad female, add(5) rseed(288403)
Conditional models:
               age: regress age attack smokes hsgrad female
               bmi: regress bmi age attack smokes hsgrad female

Multivariate imputation                     Imputations =        5
Monotone method                                   added =        5
Imputed: m=1 through m=5                        updated =        0
               age: linear regression
               bmi: linear regression
```

	Observations per m			
Variable	Complete	Incomplete	Imputed	Total
age	142	12	12	154
bmi	126	28	28	154

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

We created 5 imputations and specified the `rseed()` option for reproducibility. Let's `save` our imputations:

```
. save mheart5_imputed
file mheart5_imputed.dta saved
```

Now we should check that the imputed values are reasonable—that the imputed and observed values have similar distributions.

We can use `mi xeq` to obtain summaries of the observed, imputed, and completed data. The summary statistics for `age` in the first two imputations are:

```
. mi xeq 1/2: summarize age if Mis_age==0;   ///
>             summarize age if Mis_age==1;   ///
>             summarize age

m=1 data:
-> summarize age if Mis_age==0
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         age |        142    56.43324    11.59131   20.73613   83.78423
-> summarize age if Mis_age==1
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         age |         12     54.0485    10.25337   37.96242   68.38442
-> summarize age
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         age |        154    56.24742       11.48   20.73613   83.78423

m=2 data:
-> summarize age if Mis_age==0
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         age |        142    56.43324    11.59131   20.73613   83.78423
-> summarize age if Mis_age==1
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         age |         12    57.19998    7.643524   44.46599   72.91214
-> summarize age
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         age |        154    56.49299    11.31651   20.73613   83.78423
```

The summaries look reasonable. For example, the means of the observed values are similar to the means of the imputed values.

We can move beyond summaries and compare for the first imputation the distributions of `bmi` in the observed, imputed, and completed data:

```
. qui mi xeq 1: twoway (kdensity bmi if Mis_bmi==0) ||   ///
>                      (kdensity bmi if Mis_bmi==1) ||   ///
>                      (kdensity bmi),                 ///
>                        legend(label(1 "Observed") label(2 "Imputed") label(3 "Completed"))
```

(*Continued on next page*)

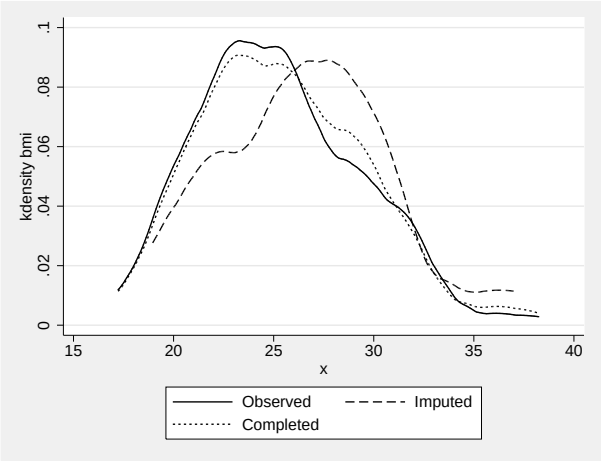

Figure 2: Distributions of `bmi` in the observed, imputed, and completed samples

Figure 2 shows that the shape of the distribution of the imputed values of `bmi` differs from that of the observed values. But that is not necessarily a problem—the distributions should be similar only if the data are missing completely at random (MCAR). The two distributions may differ if the data are missing at random (MAR) or missing not at random (MNAR). If you can reasonably assume that data are missing at random, you should verify that the observed differences make sense scientifically. If you suspect that data are missing not at random, you should use an imputation method that also models the missing-data mechanism. Most commonly-used imputation methods assume MAR.

You can test the assumption of MCAR data against MAR data. For example, you can perform logistic regression of the missing-data indicator for each imputed variable on other explanatory variables to test for associations. If data are MCAR, there should be no strong associations. It is impossible to test the assumption of MAR data against MNAR data formally without additional information about the process that generated the missing data. Sensitivity analysis (e.g. Kenward and Carpenter 2007 and references therein) is often used to test the plausibility of the MAR assumption against MNAR.

We created our hypothetical data to be missing completely at random, so we expect to see similar distributions of `bmi` in the observed and imputed data. The differences in figure 2 suggest the our imputation model for `bmi` may be inappropriate. The distribution of `bmi` is not normal, so predictive mean matching (`pmm`) would perhaps be a better choice for imputing `bmi`.

It is also important to remember that multiple imputation is a stochastic procedure, so "bad imputations" may be simply the result of randomness. As such, it is important to look at all imputations and verify that the majority of them are reasonable. You can do this by executing the preceding command on all imputations:

```
qui mi xeq 1/5: twoway (kdensity bmi if Mis_bmi==0) ||   ///
>                      (kdensity bmi if Mis_bmi==1) ||   ///
>                      (kdensity bmi),                   ///
>                        legend(label(1 "Observed") label(2 "Imputed") label(3 "Completed")); ///
>                set more on; more; set more off
  (output omitted)
```

Note that you can execute multiple commands with `mi xeq` by separating them with semicolons. Above, we use this feature to pause Stata using `more` after each graph is generated so that we can study the graph for as long as we like. When we are ready for the next graph, we simply click on `more` in the Results window.

If you have enough data and wish to compare the distributions formally, you can perform Kolmogorov-Smirnov tests (Abayomi et al. 2008, 280–281) with `mi xeq`:

```
. mi xeq 1/5: ksmirnov bmi, by(Mis_bmi)
  (output omitted)
```

Another way to check an imputation model is to compare its predictions to the observed data. With `mi` data we can plot the observed and imputed values against predictions from the imputation model (Su et al. 2011, 18–19). We'll demonstrate how to produce such a graph for the imputation model of `bmi`.

To make the graph we'll need to re-fit the imputation model for `bmi` to the observed data and compute linear predictions from a specific imputation using the observed-data estimates of the coefficients.

We could use `mi xeq`, as before, to produce the graph. Instead, we'll show how to do this by using `mi extract`.

(*Continued on next page*)

We use `mi extract 0` to retrieve the observed data and then re-fit the regression model for `bmi`:

```
. use mheart5_imputed, clear
(Fictional heart attack data; bmi and age missing)
. mi extract 0
. regress bmi age attack smokes hsgrad female
```

Source	SS	df	MS		
Model	73.1449691	5	14.6289938	Number of obs	= 126
Residual	1956.28756	120	16.3023963	F(5, 120)	= 0.90
				Prob > F	= 0.4853
Total	2029.43253	125	16.2354602	R-squared	= 0.0360
				Adj R-squared	= -0.0041
				Root MSE	= 4.0376

bmi	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
age	-.0240674	.0318764	-0.76	0.452	-.0871805	.0390457
attack	1.545	.7775581	1.99	0.049	.0054888	3.084511
smokes	-.2470361	.7820658	-0.32	0.753	-1.795472	1.3014
hsgrad	-.4092541	.8225038	-0.50	0.620	-2.037754	1.219246
female	-.109108	.8372703	-0.13	0.897	-1.766845	1.548629
_cons	26.31799	1.961478	13.42	0.000	22.4344	30.20158

We now compute linear predictions in, for example, the second imputation using the above estimates. We reload our imputed data and use `mi extract 2` to extract the second imputation. The estimation results from `regress` are current, so we use `predict` to compute linear predictions. We then plot the observed and imputed values against the model's predictions. We also overlay separate `lowess` curves for the observed and imputed values:

```
. use mheart5_imputed, clear
(Fictional heart attack data; bmi and age missing)
. mi extract 2
. predict xb, xb
. twoway (scatter bmi xb if Mis_bmi==0, msymbol(oh)) ///
>        (scatter bmi xb if Mis_bmi==1) ///
>        (lowess bmi xb if Mis_bmi==0)  ///
>        (lowess bmi xb if Mis_bmi==1),  ///
>           ytitle(bmi)                  ///
>           title(Bivariate scatter with overlaid lowess fit for m=2) ///
>           legend(label(1 "Observed") label(2 "Imputed") ///
>                  label(3 "Lowess (observed)") label(4 "Lowess (imputed)"))
```

(*Continued on next page*)

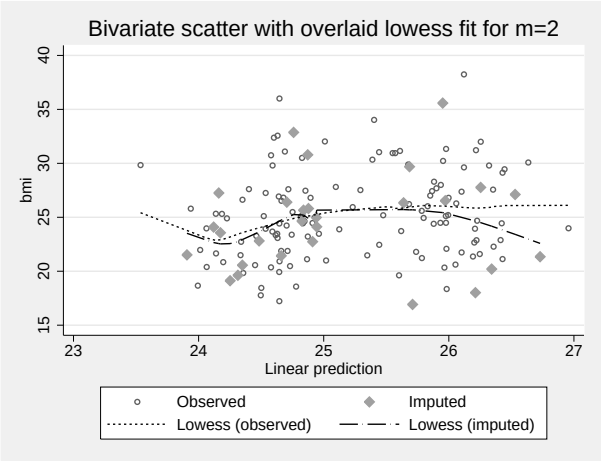

Figure 3: Bivariate scatter plot for `bmi`

From figure 3, the `lowess` curves are similar and suggest that the imputations in the second dataset are reasonable.

3 References

Abayomi, K., A. Gelman, and M. Levy. 2008. Diagnostics for multivariate imputations. *Applied Statistics* 57(3): 273–291.

Kenward, M. G., and J. R. Carpenter. 2007. Multiple imputation: Current perspectives. *Statistical Methods in Medical Research* 16: 199–218.

Raghunathan, T., and I. Bondarenko. 2007. Diagnostics for Multiple Imputations. http://ssrn.com/abstract=1031750.

Su, Y.-S., A. Gelman, J. Hill, and M. Yajima. 2011. Multiple imputation with diagnostics (mi) in R: Opening windows into the black box. Forthcoming in Journal of Statistical Software. Available at http://www.stat.columbia.edu/∼gelman/research/published/mipaper.rev04.pdf.