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	$115	1-year subscription	$145
2-year subscription	$210	2-year subscription	$270
3-year subscription	$285	3-year subscription	$375
1-year student subscription	$ 85	1-year student subscription	$115
1-year institutional subscription	$345	1-year institutional subscription	$375
2-year institutional subscription	$625	2-year institutional subscription	$685
3-year institutional subscription	$875	3-year institutional subscription	$965
Electronic only		**Electronic only**	
1-year subscription	$ 85	1-year subscription	$ 85
2-year subscription	$155	2-year subscription	$155
3-year subscription	$215	3-year subscription	$215
1-year student subscription	$ 55	1-year student subscription	$ 55

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.

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* (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 (2015)
15, Number 3, pp. 833–844

Model specification and bootstrapping for multiply imputed data: An application to count models for the frequency of alcohol use

W. Scott Comulada
Department of Psychiatry and Biobehavioral Sciences
University of California, Los Angeles
Los Angeles, CA
wcomulada@mednet.ucla.edu

Abstract. Stata's `mi` commands provide powerful tools to conduct multiple imputation in the presence of ignorable missing data. In this article, I present Stata code to extend the capabilities of the `mi` commands to address two areas of statistical inference where results are not easily aggregated across imputed datasets. First, `mi` commands are restricted to covariate selection. I show how to address model fit to correctly specify a model. Second, the `mi` commands readily aggregate model-based standard errors. I show how standard errors can be bootstrapped for situations where model assumptions may not be met. I illustrate model specification and bootstrapping on frequency counts for the number of times that alcohol was consumed in data with missing observations from a behavioral intervention.

Keywords: st0407, multiple imputation, missing data, model specification, bootstrap

1 Introduction

Missing data are a common issue across most fields of study involving proper statistical analysis. When missing data are assumed to be dependent on observed variables (that is, missing at random [Rubin 1976]), multiple imputation (MI) (see Bartlett et al. [Forthcoming]; Belin et al. [2000]; Little and Rubin [2002]; Rubin [1987]; Schafer [2003]; Siddique and Belin [2008]) can potentially reduce estimation bias and increase precision. In Stata, `mi` commands are used to create multiple datasets where missing values are imputed based on observed variables in the data. The `mi estimate` command can then be used to combine estimated regression coefficients and standard errors (SEs) across imputed datasets for proper statistical inference based on Rubin's method (Rubin 1987). Over the past decade, many studies have conducted MI through Stata (Aloisio et al. 2014; Royston 2004; Royston 2005; Royston 2007; Royston 2009; Royston, Carlin, and White 2009).

Despite their utility, Stata's `mi` commands and the MI procedures of other standard software packages do not implement two statistical inferentially related key tasks. Before combining regression coefficients with `mi estimate` and selecting model covariates with `mi test`, one must specify an appropriate model. For example, model-fit statistics and diagnostic plots should be examined to determine the adequacy of outcome distributional assumptions, such as normality for a linear regression model. Model fit cannot be tested through the `mi` commands. This is understandable because discussion of methods to test model fit in the presence of MI is really just starting to emerge in the literature, let alone in statistical software packages. For example, see Johansson, Strålfors, and Cedersund (2014) and Schomaker and Heumann (2014).

Current MI procedures in Stata and other statistical software packages also still lack the ability to bootstrap SEs. This is especially important for nonlinear combinations of regression coefficients where assumptions for common methods for implementing variance estimation may not be tenable (for example, with the delta method implemented through the `margins` command). Furthermore, the formula for the SEs may be difficult or even mathematically intractable to specify (Guan 2003).

In this article, I show how to enhance the `mi` commands with additional programming to conduct model specification and bootstrapping. In both instances, I use a `forvalue` loop to iterate through imputed datasets. For testing model fit, I examine fit statistics across imputed datasets and favor the fit-statistic results that are in agreement across most imputed datasets. For bootstrapping, the SE is bootstrapped on each imputed dataset. The overall estimate of the SE is constructed using Rubin's rules for combining imputed data (Rubin 1987).

I illustrate model specification and bootstrapping in conjunction with MI using data from a behavioral intervention trial on the frequency of alcohol consumption over three months in HIV-positive adults. In section 2, I give details on the dataset. In section 3, I discuss candidate models for the frequency-count data. In section 4, I discuss the MI data structure in Stata to clarify how `forvalue` loops are used in the code. In section 5, I explain model specification for multiply imputed datasets. I estimate the SE for a quantity of interest in this dataset, giving the marginal effect of having an AIDS diagnosis on the expected count for the number of times that alcohol was consumed. In section 6.1, I give an example of SE estimation based on the delta method, and I contrast this with estimation of the SE through bootstrapping in section 6.2. In section 7, I provide further discussion. Analyses are conducted in Stata 14.

2 Healthy Living Project: A motivating example

Here we analyze baseline data ($n = 936$) from the Healthy Living Project (HLP). This study was designed to reduce HIV-transmission behaviors and improve the quality of life for adults living with HIV. Adults, ages 19 to 67, were recruited from social service agencies in four metropolitan areas in the United States from 2000 to 2002; the cities were Los Angeles, Milwaukee, New York City, and San Francisco. Study participants were queried on their sexual behavior and use of alcohol, tobacco, and other drugs. See additional study details in Wong et al. (2008) and Comulada et al. (2010).

Here we analyze the frequency of times that alcohol was consumed over the past 90 days (`alcoholx`) as an outcome, and we treat the self-reported CD4 counts as a primary regressor. We dichotomize the CD4 counts into a clinically relevant variable (`aids`) with categories for CD4 counts less than 200 cells per μL to indicate an AIDS diagnosis (1) or counts of 200 or greater (0). Approximately 1 in 5 study participants who reported their CD4 counts had counts less than 200 (17%; $n = 146$ of 867). We conduct analyses on study participants with complete data ($n = 928$), except for the `aids` variable that MI methods are illustrated on. Seven percent of the `aids` data are missing ($n = 61$ of 928), mostly because study participants reported that they did not know their CD4 counts. Other HLP measures include age (`age`); education level, dichotomized as high school or less (1) or more education (0) (`hsorless`); current employment (`work`; `1` = yes and `0` = no); study site, with `sitemi`, `siteny`, and `sitesf` coded as 1/0 binary indicators for Milwaukee, New York City, and San Francisco, respectively; and race and ethnicity, with 1/0 binary indicators for non-Latino African American ethnicity (`black`), Latino ethnicity (`latino`), and non-Latino White ethnicity (`white`). All other race and ethnic categories constitute the reference group. We also include an HIV-transmission risk group measure (CDC 2001), where male participants are coded into one of three 1/0 binary indicators based on the following hierarchy: 1) injecting drugs (`idu`); 2) not injecting drugs but having sex with other men (`msm`); and 3) not injecting drugs and not having sex with other men (`htm`). Female participants are the reference group. The following Stata code reads in the data:

```
. use hlp.dta, clear
. set seed 2014
. global nobsdata = _N // # observations in dataset
. // Specify values for 1st approach to estimate SE (improper approach)
. global nimpute = 25  // # imputations
. global nboot = 50    // # bootstrap samples on each imputed dataset
```

We set a seed value of 2014 for reproducibility on imputation and bootstrapping algorithms that will be carried out in subsequent sections. We use global macro variables to set numbers of observations in the dataset, imputations, and bootstrap samples to add flexibility to the Stata code for use in other MI data scenarios.

3 Candidate count models

We consider four commonly used count-data models for the frequency of alcohol consumption. We begin with Poisson regression as a base model. Count y_i for person i has mean $\mu_i = \exp x_i'\beta$, where β is a regression coefficient vector for covariate vector x_i. A key assumption of the Poisson model is that the mean and variance of the counts are equal, $\mathrm{Var}(y_i|x_i) = \mu_i$, which is typically not met in alcohol-use count data. Most individuals in our study were occasional alcohol consumers mixed with a few heavy consumers, resulting in data that appeared to be overdispersed beyond what a Poisson distribution can reasonably model. This is reflected in the sample variance of 660.5, which is much larger than the sample mean of 16.8 times.

An alternative model that handles overdispersed count data is the negative binomial regression model. An additional parameter α is introduced to allow the variance to be overdispersed, $\text{Var}(y_i|x_i) = \mu_i(1 + \alpha\mu_i)$. As part of the `nbreg` command to conduct negative binomial regression, a likelihood-ratio (LR) test is conducted to test against a null hypothesis that α is 0, H_0: $\alpha = 0$, and a Poisson distribution can be assumed.

Zero inflation is a feature related to overdispersion that also occurs in alcohol-use count data. This means there is a higher proportion of zeros than what can reasonably be explained through either a Poisson or a negative binomial distribution. In the HLP study, 33% of the participants did not report any alcohol use ($n = 303$ of 928), suggesting zero inflation. Two count models to handle zero inflation are the zero-inflated Poisson (Lambert 1992) and zero-inflated negative-binomial (ZINB) models (Greene 1994). The basic modeling approach assumes that some zeros occur from individuals in a "zero state" (for example, teetotalers in our data) and that other zeros are assumed to occur from individuals who are not in a zero state (for example, individuals who did not drink during the prior three months). Thus the zero-inflated model contains two parts: an inflation part for the probability of being in the zero state and a count part to model counts conditional on observations not being in the zero state.

We can test the fit of the four models by fitting the ZINB model through the `zinb` command. Two model-fit tests are offered. First, an LR test for α is conducted similar to the `nbreg` command to see whether overdispersion needs to be accounted for. Second, the Vuong test (Vuong 1989) is conducted to test against a null hypothesis of no zero inflation.

4 Multiple imputation: First steps

We use the `mi impute` command to impute missing values for the `aids` variable through a logistic model. We include all other measures listed in section 2 as covariates. We impute 25 imputed datasets as a lower boundary for the number of commonly accepted imputations. Stata code for this is as follows:

```
. mi set mlong

. mi register imputed aids // 61 marked as incomplete
(61 m=0 obs. now marked as incomplete)

. mi impute logit aids sitemi siteny sitesf black latino white msm idu htm
> hsorless age work, add($nimpute)

Univariate imputation                       Imputations =       25
Logistic regression                               added =       25
Imputed: m=1 through m=25                       updated =        0

-------------------------------------------------------------------
                   |            Observations per m
                   |----------------------------------------------
          Variable |   Complete   Incomplete   Imputed |     Total
-------------------+-----------------------------------+----------
              aids |        867           61        61 |       928
-------------------------------------------------------------------
(complete + incomplete = total; imputed is the minimum across m
 of the number of filled-in observations.)
```

We verify the adequacy of 25 imputations using the method proposed by White, Royston, and Wood (2011). We fit the candidate count models from section 3 across imputed datasets to alcohol-consumption counts. Models include covariates for `aids` and the HIV-transmission risk group measure that we hypothesize to be associated with the frequency of alcohol consumption. Using the `margins` command, we estimate the effect of an AIDS diagnosis on expected alcohol-consumption counts, that is, the marginal effect of AIDS. Note that the `aids` covariate should be preceded by an `i.` prefix to indicate that a categorical, and not a continuous, covariate is being modeled; the `margins` command will produce different estimates depending on the covariate type. We then examine the Monte Carlo (MC) error associated with estimating the marginal effect of AIDS across imputed datasets for each of the candidate models from section 3. The MC error is produced through the `mi estimate` command and `mcerror` option. The resulting MC errors from fitting the ZINB model follow and are representative of results from the other count models in terms of the adequate number of imputations.

```
. capture program drop mimargins
. program mimargins, eclass properties(mi)
  1.     version 14
  2.         args pvar
  3.         zinb alcoholx i.aids msm idu htm, inflate(i.aids msm idu htm)
  4.         margins, dydx(`pvar´) post
  5. end
. mi estimate, mcerror: mimargins aids
Multiple-imputation estimates                    Imputations       =         25
Average marginal effects                         Number of obs     =        928
                                                 Average RVI       =     0.0608
                                                 Largest FMI       =     0.0576
DF adjustment:   Large sample                    DF:     min       =   7,296.84
                                                         avg       =   7,296.84
Within VCE type: Delta-method                            max       =   7,296.84

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      1.aids |   1.922415   2.594921     0.74   0.459     -3.16438     7.00921
             |   .1218727   .0256568     0.04   0.026     .0987859    .1583907
------------------------------------------------------------------------------
Note: Values displayed beneath estimates are Monte Carlo error estimates.
```

Note that the MC error of the marginal effect for `aids` is less than 10% of the SE, the MC error of the t statistic is less than 0.1, and the MC error of the p-value is close to 0.03 and acceptable for larger p-values. These properties suggest that additional imputations are not needed.

Next, I discuss Stata's MI data structure to clarify how imputed datasets will be used in `forvalue` loops for model specification and bootstrapping in subsequent sections. The original HLP dataset contained 928 rows: one for each observation in the dataset. After conducting MI, Stata retains the original dataset and appends rows to the end of the original dataset for each imputed value. Because the `mlong` format was specified, there were 61 missing observations and 25 imputations per missing observation that were requested. Therefore, the `mi impute` command appended $61 \times 25 = 1525$ rows to the end of the original dataset. The MI dataset contains $928 + 1525 = 2453$ observations.

mi impute also adds two variables to the dataset. The _mi_m variable indicates the iteration number of the imputed observations, ranging from 1 to 25 in our case. The _mi_miss variable takes on values of 0 or 1 for observations in the original data to indicate whether an observation is missing or observed, respectively; _mi_miss is set to missing for imputed rows of observations. Therefore, if we wish to conduct an analysis on the first imputed dataset, for example, we can subset the analysis data by typing if _mi_miss == 0 | _mi_m == 1.

5 Multiple imputation and model specification

Next, model-fit tests are applied to each imputed dataset, and the results are displayed. In the Stata code, we use a forvalue loop to iterate through each imputed dataset. Within the ith imputed dataset, $i = 1 \ldots 25$, we fit a ZINB model to the alcoholx outcome with covariates for aids and three dummy-coded indicators for HIV-transmission risk: idu, msm, and htm. We include covariates in both the inflation and the count parts of the model. Statistics for the Vuong test for zero inflation and LR test for overdispersion from the e() returned results are saved into matrices mvuong and mch2, respectively. Stata does not save the p-values in the e() returned results. As in Buis (2007), the p-values are recalculated for the Vuong test using the normal() function and for the LR test using the chi2tail() function. Recalculated p-values are then saved as scalars, along with the test statistics. Test statistics and associated p-values for each imputed dataset are saved into matrix mattests and displayed as follows:

```
. // initialize matrix to hold test statistics
. // Use impossible value of -9 as check that each initial value is replaced
. matrix mattests = J($nimpute,4,-9)

. // Initialize local macro to store row names
. local names

. // Loop to run ZINB model and store test statistics
. forvalues i=1/$nimpute {
  2. quietly zinb alcoholx i.aids msm idu htm if _mi_miss == 0 | _mi_m == `i´,
> inflate(i.aids msm idu htm) nolog vuong zip
  3. matrix mvuong = e(vuong)
  4. matrix mch2 = e(chi2_cp)
  5. matrix mattests[`i´,1] =mvuong[1,1]            // Vuong test statistic
  6. matrix mattests[`i´,2] =normal(-1*mvuong[1,1]) // Vuong test p-value
  7. matrix mattests[`i´,3] =mch2[1,1]              // LR test stat, H0: alpha = 0
  8. matrix mattests[`i´,4] =chi2tail(1,mch2[1,1])  // LR p-value
  9. local names `names´ `i´
 10. }

. matrix rownames mattests = `names´ // Assign imputation numbers as row names

. matrix colnames mattests = Vuong Vuong_p LR LR_p  // Assign column names
. matrix list mattests // Output test statistic values
mattests[25,4]
         Vuong    Vuong_p          LR       LR_p
 1  10.793552  1.846e-27  14542.427          0
 2  10.643859  9.313e-27  14536.519          0
 3  10.553355  2.451e-26  14526.471          0
 4  10.910555  5.131e-28  14542.217          0
 5  10.751589  2.912e-27  14542.491          0
 6  10.843383  1.072e-27  14542.801          0
 7  11.220511  1.617e-29  14539.252          0
 8  10.846547  1.036e-27  14542.848          0
 9  11.063159  9.465e-29  14526.966          0
10  11.101006  6.202e-29  14538.338          0
11  10.877243  7.399e-28  14529.867          0
12  10.854434  9.499e-28  14540.268          0
13  10.956243  3.101e-28  14538.263          0
14  11.022121  1.495e-28  14539.027          0
15  10.879367  7.228e-28  14536.711          0
16  10.674114  6.727e-27  14543.062          0
17  10.887376  6.620e-28  14536.941          0
18  10.891531  6.325e-28  14543.108          0
19  10.768049  2.436e-27  14543.146          0
20  11.114011  5.362e-29  14541.313          0
21  10.526446  3.263e-26  14543.763          0
22  10.656714  8.111e-27   14543.36          0
23  10.712125  4.464e-27  14543.365          0
24  11.143318  3.859e-29   14540.06          0
25  10.675284  6.643e-27   14542.59          0
```

Across the 25 imputed datasets, p-values for the Vuong test are all less than 0.05, indicating that the null hypothesis of no zero inflation should be rejected. Similarly, p-values for the LR test across imputed datasets are all less than 0.05, indicating the null hypothesis of equidispersion should be rejected. When comparing the four candidate count models in section 3, we conclude that a ZINB model provides the best fit to the data.

6 Estimating the SE on multiply imputed data

6.1 Delta method

Proceeding from model specification in section 5, we focus on results from a ZINB regression, the best-fitting model. We specifically focus on the marginal effect of AIDS on expected alcohol-consumption counts. In section 4, we used the delta method to estimate the SE of the marginal effect across imputed datasets. Next, we combine the results for the estimated marginal effects and SE using Rubin's method (Rubin 1987) as follows. Let $\widehat{M_i}$ be the estimated marginal effect, and let $\widehat{W_i}$ be the variance within imputed dataset i, $i = 1, \ldots, N$. Here we set $N = 25$. We express the average estimated marginal effect $\overline{M}$ and average within-imputation variance $\overline{W}$ as

$$\overline{M} = \frac{1}{N} \sum_{i=1}^{N} \widehat{M_i} \tag{1}$$

and

$$\overline{W} = \frac{1}{N}\sum_{i=1}^{N} \widehat{W}_i, \tag{2}$$

respectively. The variance between imputed datasets is

$$B = \frac{1}{N-1}\sum_{i=1}^{n} \left(\widehat{M}_i - \overline{M}\right)^2 \tag{3}$$

and the total variance for the estimated marginal effect is

$$T = \overline{W} + \left(1 + \frac{1}{N}\right) B \tag{4}$$

The SE is then calculated as the square root of T. Next, I demonstrate how to bootstrap the SE.

6.2 Bootstrapping

As in the delta method, the variance of the estimated marginal effect is bootstrapped for each imputed dataset i, $i = 1 \dots N$. We set the number of bootstrap samples for each dataset to 50 using the `nboot` macro variable in section 2. Here we use (1) to (4) to aggregate variance terms across imputed datasets. The overall variance T and the SE follows from (4). To calculate T through one `forvalue` loop, we reformulate the calculation of (4) using an iterative method for mean and variance estimation (Knuth 1998, 232; Welford 1962). In addition to programming efficiency, rounding errors on estimates will be reduced. Based on the iterative algorithm, the average marginal effect $\overline{M}_n$, the sums of squares SS_n for the between-imputation variance B_n, and the average within-imputation variance $\overline{W}_n$ to iteration n can be expressed as follows:

$$\begin{aligned}
\delta_1 &= M_i - \overline{M}_{n-1} \\
\overline{M}_n &= \overline{M}_{n-1} + \frac{\delta_1}{n} \\
SS_n &= SS_{n-1} + \delta_1 \left(M_i - \overline{M}_n\right) \\
\delta_2 &= W_i - \overline{W_{n-1}} \\
\overline{W_n} &= \overline{W_{n-1}} + \delta_2 \left(W_i - \overline{W}_n\right)
\end{aligned}$$

At the last iteration, $n = N = 25$, and $T = \overline{W} + (1 + 1/n)SS_n/(n-1)$. Stata code follows.

```
. // Program to run ZINB model and estimate margins
. capture program drop mimargins2

. program mimargins2, rclass
  1. zinb alcoholx i.aids msm idu htm if _mi_miss == 0 | _mi_m == `1´,
> inflate(i.aids msm idu htm)
  2.         margins, dydx(aids) post
  3.         matrix m = r(b)
  4. end

. // Initialize values to 0 for iterative routine
. scalar n = 0            // Number of current imputed dataset in iteration

. scalar bmean = 0        // Estimated margin, current iteration

. scalar Emean = 0        // Estimated margin, cumulative

. scalar SSbet = 0        // Between variance sums of squares, cumulative

. scalar Vwit = 0         // Within variance on margin, cumulative

. forvalues i=1/$nimpute {
  2. // Bootstrap for each imputed dataset
. quietly bootstrap md=m[1,2], reps($nboot): mimargins2 `i´
  3. matrix bm = e(b)
  4. scalar bmean = bm[1,1]  // Save est margin from current iteration to bmean
  5. matrix bm2= e(V)
  6. scalar bvar   =bm2[1,1] // Save variance from current iteration to bvar
  7. scalar n = n + 1
  8. scalar delta = bmean - Emean
  9. scalar Emean = Emean + delta / n
 10. scalar SSbet = SSbet + delta*(bmean - Emean)
 11. scalar delta2 = bvar - Vwit
 12. scalar Vwit = Vwit + delta2 / n
 13. }

. display sqrt(Vwit + (1 + 1/n)*(SSbet / (n-1))) // Display SE
2.6299977
```

The SE is estimated to be 2.63 and only slightly larger than the SE that was estimated to be 2.59 through the delta method in section 4. Note that while SE estimates between delta and bootstrap methods were similar in this instance, they may differ to a larger degree in other datasets.

7 Discussion

In this article, I presented Stata code to extend the capabilities of the mi commands to test model fit. The proposed ad hoc approach to model specification is straightforward in following the "majority rule". Using our example dataset, the majority rule worked quite well. Model specification tests were in agreement across all imputed datasets. Of course, this will not always be the case, which may make it more difficult to select the best model. In some instances, it may be more desirable to average parameter estimates across various tenable models, as discussed in Schomaker and Heumann (2014). Unfortunately, model averaging or other more sophisticated model-specification methods are not yet available in Stata and other commercial software packages. Using our ad hoc approach, we found that additional programming to supplement the mi commands was minimal. Moreover, the Stata code can easily be modified to produce other model-specification tests and diagnostic plots that can be outputted across imputed datasets through a `forvalue` loop. Both the dataset and the Stata code that were used for analyses in this article have been provided as a supplementary resource.

I also presented Stata code to bootstrap SEs of a parameter or a combination of parameters as an alternative to the delta method. The delta method and bootstrap procedures were both applied to each imputed dataset, and results were combined using Rubin's method (Rubin 1987). Note that a possible alternative implementation to conducting MI and then bootstrapping each imputed dataset is conducting a multiply-impute-then-boot procedure. Shao and Sitter (1996) examined the scenario where one imputation is conducted instead of MI and advocated a boot-then-impute procedure. First, B_1 bootstrap samples are generated from the original dataset containing observed and missing data. Next, one dataset is imputed for each bootstrap sample. Variance terms are estimated across the B_1 bootstrapped-imputed datasets and averaged. The SE is the square root of the averaged variance. Many more bootstrap samples should be generated relative to the multiply-impute-then-boot procedure that bootstraps B_2 samples on each of N imputed datasets. Based on the number of imputations and bootstrap samples we used in our analyses, we could set $B_1 = N \times B_2 = 25 \times 50 = 1250$. For singly imputed data, the boot-then-impute procedure is necessary to capture variance inflation due to imputation and bootstrapping. To our knowledge, this issue has not fully been explored for MI. The added value of the boot-then-impute procedure is debatable in light of the additional variation that MI captures relative to single-imputation methods. We applied the boot-then-impute procedure to our data, setting $B_1 = 500$, and compared results with those in section 6.2 based on the multiply-impute-then-boot procedure; the SE was estimated to be 2.59 and less than the estimate of 2.63 from section 6.2.

We must also consider computing time and sample size. The boot-then-impute procedure with 500 bootstrap samples took several additional hours to run relative to the multiply-impute-then-boot procedure that was run on the HLP data, a moderately sized dataset at a little less than 1,000 observations. We attempted 1,250 bootstrap samples with the boot-then-impute procedure but stopped the code from running after several days without results. Analyses were conducted using Stata/IC and would have benefited from the ability of Stata/MP to harness parallel computing facilities. While run times increase dramatically as sample sizes increase, He (2006) aptly noted that combined MI and bootstrapping methods become less practical with increasing model complexity and decreasing sample size. In our case, ZINB models converged across imputed and bootstrapped datasets. For smaller datasets, the ZINB model may exhibit convergence problems in some MI-bootstrapped datasets but not in others. Iterating through datasets and combining results may be challenging and may require analysts to adopt other approaches.

8 Acknowledgments

The HLP study was conducted with the help of the National Institute of Mental Health (U10MH057615; U10MH057636; U10MH057631; U10MH057616; P30MH058107). My time was also supported by the National Institute of Mental Health (K01MH089270).

9 References

Aloisio, K. M., N. Micali, S. A. Swanson, A. Field, and N. J. Horton. 2014. Analysis of partially observed clustered data using generalized estimating equations and multiple imputation. *Stata Journal* 14: 863–883.

Bartlett, J. W., S. R. Seaman, I. R. White, and J. R. Carpenter. Forthcoming. Multiple imputation of covariates by fully conditional specification: Accommodating the substantive model. *Statistical Methods in Medical Research*.

Belin, T. R., M.-Y. Hu, A. S. Young, and O. Grusky. 2000. Using multiple imputation to incorporate cases with missing items in a mental health services study. *Health Services and Outcomes Research Methodology* 1: 7–22.

Buis, M. L. 2007. Stata tip 53: Where did my p-values go? *Stata Journal* 7: 584–586.

CDC. 2001. HIV/AIDS Surveillance Report, Vol. 13. 1–41.

Comulada, W. S., M. J. Rotheram-Borus, W. Pequegnat, R. E. Weiss, K. A. Desmond, E. M. Arnold, R. H. Remien, S. F. Morin, L. S. Weinhardt, M. O. Johnson, and M. A. Chesney. 2010. Relationships over time between mental health symptoms and transmission risk among persons living with HIV. *Psychology of Addictive Behaviors* 24: 109–118.

Greene, W. H. 1994. Accounting for excess zeros and sample selection in Poisson and negative binomial regression models. Working Paper Series EC-94-10, Department of Economics, Stern School of Business, New York University.

Guan, W. 2003. From the help desk: Bootstrapped standard errors. *Stata Journal* 3: 71–80.

He, Y. 2006. Missing data imputation for tree-based models. Doctoral dissertation, University of California, Los Angeles.

Johansson, R., P. Strålfors, and G. Cedersund. 2014. Combining test statistics and models in bootstrapped model rejection: It is a balancing act. *BMC Systems Biology* 8: 46.

Knuth, D. E. 1998. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. 3rd ed. Reading, MA: Addison–Wesley.

Lambert, D. 1992. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* 34: 1–14.

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

Royston, P. 2004. Multiple imputation of missing values. *Stata Journal* 4: 227–241.

———. 2005. Multiple imputation of missing values: Update of ice. *Stata Journal* 5: 527–536.

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

———. 2009. Multiple imputation of missing values: Further update of ice, with an emphasis on categorical variables. *Stata Journal* 9: 466–477.

Royston, P., J. B. Carlin, and I. R. White. 2009. Multiple imputation of missing values: New features for mim. *Stata Journal* 9: 252–264.

Rubin, D. B. 1976. Inference and missing data. *Biometrika* 63: 581–592.

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

Schafer, J. L. 2003. Multiple imputation in multivariate problems when the imputation and analysis models differ. *Statistica Neerlandica* 57: 19–35.

Schomaker, M., and C. Heumann. 2014. Model selection and model averaging after multiple imputation. *Computational Statistics and Data Analysis* 71: 758–770.

Shao, J., and R. R. Sitter. 1996. Bootstrap for imputed survey data. *Journal of the American Statistical Association* 91: 1278–1288.

Siddique, J., and T. R. Belin. 2008. Multiple imputation using an iterative hot-deck with distance-based donor selection. *Statistics in Medicine* 27: 83–102.

Vuong, Q. H. 1989. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica* 57: 307–333.

Welford, B. P. 1962. Note on a method for calculating corrected sums of squares and products. *Technometrics* 4: 419–420.

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.

Wong, F. L., M. J. Rotheram-Borus, M. Lightfoot, W. Pequegnat, W. S. Comulada, W. Cumberland, L. S. Weinhardt, R. H. Remien, M. Chesney, M. Johnson, and Healthy Living Trial Group. 2008. Effects of behavioral intervention on substance use among people living with HIV: The Healthy Living Project randomized controlled study. *Addiction* 103: 1206–1214.

About the author

W. Scott Comulada is an associate professor-in-residence in the Department of Psychiatry and Biobehavioral Sciences at the University of California, Los Angeles. He is also a scientist for the Methods Core through the Center for HIV Identification, Prevention and Treatment Services (CHIPTS; P30MH058107). He is currently part of a cross-disciplinary team of scientists who are developing research methods to assess and evaluate behavioral data from mobile phone-based health applications.