

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

---

Faculty Publications, Department of Psychology

Psychology, Department of

---

October 2005

# An SAS Macro for Implementing the Modified Bollen–Stine Bootstrap for Missing Data: Implementing the Bootstrap Using Existing Structural Equation Modeling Software

Craig K. Enders

*University of Nebraska - Lincoln*

Follow this and additional works at: <http://digitalcommons.unl.edu/psychfacpub>



Part of the [Psychiatry and Psychology Commons](#)

---

Enders, Craig K., "An SAS Macro for Implementing the Modified Bollen–Stine Bootstrap for Missing Data: Implementing the Bootstrap Using Existing Structural Equation Modeling Software" (2005). *Faculty Publications, Department of Psychology*. 307. <http://digitalcommons.unl.edu/psychfacpub/307>

This Article is brought to you for free and open access by the Psychology, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Faculty Publications, Department of Psychology by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

# An SAS Macro for Implementing the Modified Bollen–Stine Bootstrap for Missing Data: Implementing the Bootstrap Using Existing Structural Equation Modeling Software

Craig K. Enders  
*University of Nebraska*

The Bollen–Stine bootstrap can be used to correct for standard error and fit statistic bias that occurs in structural equation modeling (SEM) applications due to non-normal data. The purpose of this article is to demonstrate the use of a custom SAS macro program that can be used to implement the Bollen–Stine bootstrap with existing SEM software. Although this article focuses on missing data, the macro can be used with complete data sets as well. A series of heuristic analyses are presented, along with detailed programming instructions for each of the commercial SEM software packages.

Bootstrapping is a method that can be used to correct for the standard error and fit statistic bias that occurs in structural equation modeling (SEM) applications due to nonnormal data. Two forms of the bootstrap have been discussed in the SEM literature, the naive bootstrap and the Bollen–Stine bootstrap; the former is appropriate for generating standard error estimates, whereas the latter can be used to estimate standard errors (Yung & Bentler, 1996) and to correct for bias in the model fit statistic (Bollen & Stine, 1992). Although the Bollen–Stine bootstrap has performed well in a small number of empirical studies (e.g., Enders, 2001; Fouladi, 1998; Nevitt & Hancock, 2001), it appears that the method is used infrequently in applied studies.

One likely reason why the bootstrap is not widely used in applied studies is the availability of straightforward corrective procedures such as the rescaled test statistic and robust standard errors (Satorra & Bentler, 1994). With complete data, it is well known that rescaled statistics provide reasonably accurate inferences, even when normality assumptions are violated (e.g., see Finney & DiStefano, in press, for an overview). However, much less is known about the performance of these corrective procedures in the missing data context. Rescaling methods for nonnormal missing data were proposed by Yuan and Bentler (2000), and are currently available in the *Mplus* (version 2.1 or higher) and EQS (version 6 or higher) software packages.

Despite this widespread availability, few studies have examined the performance of rescaled statistics for missing data. Enders (2001) compared the performance of rescaled statistics and the bootstrap under missing at random (MAR) missing data (i.e., missingness is dependent on other observed variables), and found that the Bollen–Stine bootstrap produced slightly more accurate Type I error rates than the rescaled test statistic. Although robust and bootstrap standard errors both worked well, the two approaches exhibited different tendencies: The bootstrap yielded slightly liberal significance tests (i.e., confidence interval coverage values were, in some cases, lower than 95%), and robust standard errors produced somewhat conservative tests (i.e., coverage values generally exceeded 95%). Although maximum likelihood (ML) estimation with missing data assumes MAR missingness, it is important to note that robust standard errors require the more stringent missing completely at random (MCAR) assumption (i.e., missingness is unrelated to the data). Although the bootstrap does not explicitly rely on specific assumptions regarding the missingness, the performance of the procedure in this context merits further investigation.

In addition to the availability of rescaled statistics, another possible explanation for the lack of bootstrap applications is the limited availability of bootstrapping routines in commercial software packages. As such, the purpose of this article is to outline an SAS macro program capable of performing the Bollen–Stine bootstrap, and to demonstrate how this macro can be used in conjunction with existing SEM software packages to obtain corrected probability values for the model fit statistic and empirical standard error estimates. Although the macro is capable of handling complete data, the focus of this article is on missing data, as only one commercial software package, EQS 6.1, performs the modified Bollen–Stine bootstrap for missing data proposed by Enders (2002).

There are several reasons why this SAS macro might be useful to substantive researchers and methodologists. As already noted, little is known about the performance of rescaled test statistics in the missing data context, and this is further complicated by the fact that robust standard errors require restrictive assumptions about the missing data mechanism (i.e., MCAR). The limited simulations that have been conducted to date suggest that the Bollen–Stine bootstrap performs as well, if not better than, rescaled statistics in the missing data context (Enders,

2001). Although implementing the Bollen–Stine bootstrap with complete data is relatively straightforward, the computational complexities increase dramatically with missing data. To date, only EQS 6.1 implements the Bollen–Stine bootstrap for missing data, so the availability of this macro might facilitate the use of the bootstrap in substantive applications, but could also provide a research tool for methodologists who are interested in studying the behavior of this technique. Although not one of the explicit goals of this article, it should be noted that the process of fitting multiple bootstrap samples using SEM software is identical to the process one would use to fit computer simulation data generated by an external program such as SAS or SPSS. As such, the instructions provided in this article may also be informative for researchers interested in conducting Monte Carlo studies.

To illustrate the bootstrapping procedure, a confirmatory factor analysis (CFA) of the Eating Attitudes Test (EAT; Garner, Olmsted, Bohr, & Garfinkel, 1982) is used, a popular self-report instrument for assessing eating disorder risk. For simplicity, a subset of 10 manifest indicators was extracted from a model proposed by Doninger, Enders, and Burnett (2005). These indicators measured two latent factors (drive for thinness and food preoccupation), and a graphical depiction of the

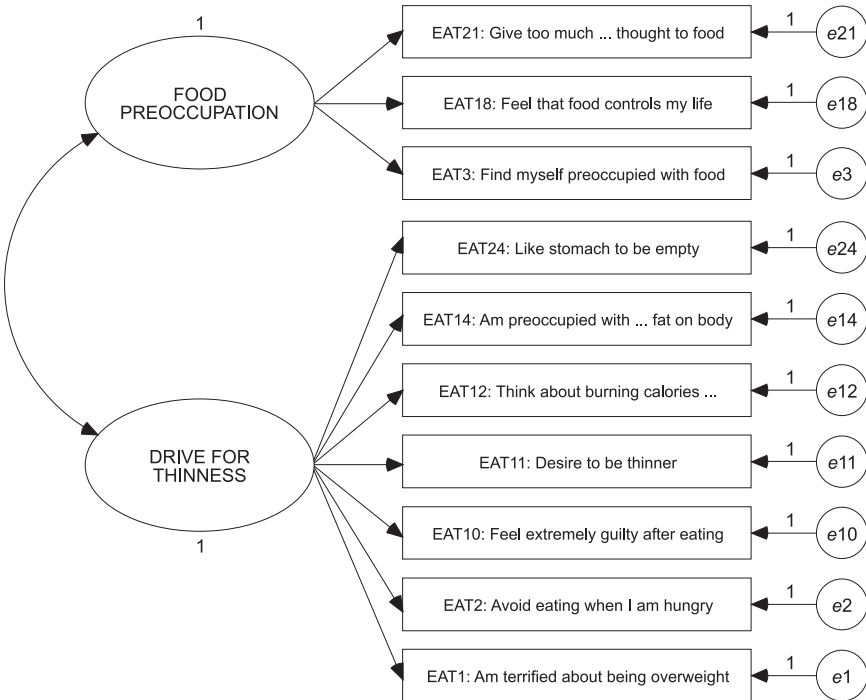


FIGURE 1 Two-factor CFA model for Eating Attitudes Test.

two-factor CFA model is shown in Figure 1. The original data had no missing values, so a 5% missing data rate was imposed on five items (EAT1, EAT10, EAT12, EAT18, and EAT24) in an MCAR fashion (the missing data mechanism is irrelevant to the current illustration, so MCAR was chosen for simplicity). The raw data ( $N = 200$ ) used in the heuristic analyses can be downloaded at <http://www.asu.edu/clas/psych/people/faculty/enders.htm> or obtained from the author.

## THE BOOTSTRAP

In the interest of space, only a brief description of the bootstrap is provided, but a number of detailed overviews are available to interested readers (e.g., Efron & Tibshirani, 1993; Stine, 1989). One of the central goals of most statistical analyses is to estimate the sampling variance of a parameter,  $\theta$ . In the SEM context, it is typical to rely on an estimator such as ML, although doing so often requires the adoption of distributional assumptions. The bootstrap provides an alternative method for estimating the sampling variance of a statistic, but does so without the usual normality assumptions (which is not to imply that the bootstrap is without its own tenuous assumptions).

In the bootstrapping procedure, the sample data are assumed to be an accurate representation of the population distribution function from which the sample was drawn. Essentially, the sample data, sometimes referred to as the *parent sample*, serve as a miniature population from which repeated samples will be drawn with replacement. For example, in the subsequent analyses, the EAT data set ( $N = 200$ ) serves as a surrogate for the population, and samples of  $N = 200$  are repeatedly drawn with replacement. Sampling with replacement implies that some cases may appear two or more times in any given bootstrap sample, and some cases may not appear at all. Usually, a large number of samples are drawn from the parent data (say  $B = 1,000$ ), and an estimate of the parameter of interest,  $\hat{\theta}$ , is obtained from each sample. The  $B$  estimates form an empirical sampling distribution, the standard deviation of which is the bootstrap standard error estimate.

## THE BOLLEN–STINE BOOTSTRAP

The so-called naive bootstrap previously described is appropriate for obtaining empirical estimates of parameter standard errors, and appears to provide accurate estimates in the SEM context. However, this form of the bootstrap is not appropriate for assessing model fit, because the covariance structure of the parent data matrix is not consistent with the null hypothesis. As such, model fit statistics estimated from the bootstrap samples will reflect both model misfit and sampling fluctuation.

The application of the bootstrap to testing covariance structures was first introduced by Beran and Srivastava (1985), and later popularized in the SEM context by Bollen and Stine (1992) in an unrelated work. When bootstrapping the likelihood ratio fit statistic, Bollen and Stine showed that it is first necessary to transform the parent data matrix so that its covariance structure is consistent with the hypothesized model (i.e., the null is true in the parent data). This is accomplished using the following transformation:

$$Z = (Y - \bar{Y})S^{-1/2}\hat{\Sigma}^{1/2} \quad (1)$$

where  $Y$  is the parent data matrix,  $\bar{Y}$  is the implied mean vector from the saturated model,  $S$  is the implied covariance matrix of the saturated model, and  $\hat{\Sigma}$  is the implied covariance matrix of the hypothesized model. Next,  $B$  bootstrap samples are drawn (with replacement) from the transformed data matrix,  $Z$ , and the hypothesized model is fit to each sample. The  $B$  model fit statistics are subsequently used as an empirical sampling distribution, and an adjusted probability value is obtained by computing the proportion of test statistics from the empirical sampling distribution that exceed the likelihood ratio test statistic obtained from the parent sample,  $Y$ .

The Bollen–Stine transformation shown in Equation 1 requires a complete data matrix, but Enders (2002) proposed a modification to the Bollen–Stine bootstrap for missing data. This modified Bollen–Stine transformation is applied separately to each case's data vector using only those parameter values for which case  $i$  has complete data. The modified Bollen–Stine transformation is

$$z_i = (y_i - \hat{\mu}_{S(i)}) \hat{\Sigma}_{S(i)}^{-1/2} \hat{\Sigma}_{M(i)}^{1/2} + \hat{\mu}_{M(i)} \quad (2)$$

where  $\hat{\mu}_{S(i)}$  is the implied mean vector of the saturated model,  $\hat{\Sigma}_{S(i)}$  is the implied covariance matrix of the saturated model, and  $\hat{\Sigma}_{M(i)}$  and  $\hat{\mu}_{M(i)}$  are the implied covariance matrix and mean vector associated with the hypothesized model, respectively. In the method proposed by Enders (2002), the parameter matrices are obtained using ML estimation for MAR missing data (often referred to as full information maximum likelihood [FIML]; Arbuckle, 1996).

The case subscript  $i$  in Equation 2 suggests that the parameter matrices may differ in size and content, but are constant within a group of cases with the same missing data pattern. It is important to note that the same parameter values are used for all cases, but the specific parameter elements that are used in the transformation may vary from case to case. To illustrate, consider a simple situation with three variables,  $y_1$ ,  $y_2$ , and  $y_3$ . For cases with no missing data, the FIML parameter matrices would not be altered, and the transformation would utilize the entire data vector—this is equivalent to the Bollen–Stine transformation given in Equation 1. In contrast, consider a subset of cases missing  $y_1$ . For these cases, the rows and col-

umns of the parameter matrices that correspond to  $y_1$  would be removed (as would the missing element in the raw data vector), and the transformation would utilize only those elements that correspond to  $y_2$  and  $y_3$ .

After applying the modified Bollen–Stine transformation to each case’s data vector, the bootstrap proceeds in the same fashion as it would with complete data; bootstrap samples are drawn with replacement from the transformed data matrix, the model is fit to each sample using the FIML estimator, and an empirical sampling distribution of the model fit statistics is used to obtain an adjusted probability value for the FIML likelihood ratio test.

To date, only the forthcoming version of EQS (version 6.1) performs the modified Bollen–Stine bootstrap for missing data, although AMOS (versions 4 and higher) can be used to implement the Bollen–Stine bootstrap with complete data. As stated previously, the purpose of this article is to illustrate how one would perform the bootstrap using existing SEM software. To accomplish this, a custom SAS macro program was written that generates bootstrap data sets from a transformed data matrix based on Equation 2. The hypothesized model can subsequently be fit to each of the bootstrap samples using existing SEM software. However, because SEM software packages require different input data structures, it was necessary to write a separate macro program for AMOS, LISREL, and *Mplus*, all of which are available for download at <http://www.asu.edu/clas/psych/people/faculty/enders.htm>. Again, these SAS programs can be used to perform the Bollen–Stine with complete data as well (with no missing data, Equations 1 and 2 are equivalent, although Equation 2 allows for a mean structure), and the instructions outlined herein are identical in both cases.

## USING THE SAS MACRO

Fitting the hypothesized model to each bootstrap sample is software-specific and is discussed in detail later. At this stage, the implementation of the SAS macro is described, as the process of transforming the parent data and drawing bootstrap samples is identical regardless of which package is ultimately used to analyze the data (the different versions of the macro program differ only with respect to the format of the output data sets).

As a first step, it is necessary to fit the hypothesized model to the parent data (not the transformed data) using the FIML estimator. Returning to the EAT example, the CFA model was estimated using *Mplus* 3.11, and the likelihood ratio fit statistic from this analysis was  $\chi^2(34) = 65.53, p = .001, N = 200$ . However, the accuracy of this test statistic is questionable given that the data were not multivariate normal, by definition; the univariate skewness and kurtosis values for the manifest indicators were not necessarily excessive (skewness ranged between .12 – 1.48, and kurtosis ranged between –1.04 – 2.18), but the questionnaire items were mea-

```

%let misscode = -9;                * MISSING VALUE CODE;
%let numvars = 10;                * NUMBER OF VARIABLES;
%let numboots = 1100;            * NUMBER OF BOOTSTRAP SAMPLES;
%let seed = 12345;               * SEED FOR BOOTSTRAPPING;
%let datafile = "c:\eatboot.txt"; * RAW DATA MATRIX;
%let sigmafile = "c:\impliedcovs.txt"; * MODEL-IMPLIED COVARIANCES;
%let imeansfile = "c:\impliedmeans.txt"; * MODEL-IMPLIED MEANS;
%let samplefile = "c:\samplecovs.txt"; * SATURATED MODEL COVARIANCES;
%let meansfile = "c:\samplemeans.txt"; * SATURATED MODEL MEANS;

```

FIGURE 2 SAS macro user-specified variables.

sured on a 6-point Likert scale ranging from (6) *always* to (1) *never*. As such, the Bollen–Stine bootstrap was used to construct empirical sampling distributions for the model fit statistic and parameter estimates. Before proceeding, it is worth noting that it may not be advisable to implement the bootstrap using a sample size of 200, but I do so for the purpose of illustrating the procedure.

To use the macro program, it is necessary to save the implied covariance matrices and mean vectors from both the saturated and hypothesized models (these matrices are standard output in commercial SEM programs). Each of the four parameter matrices should be saved as separate ASCII files, with the covariance matrices in lower diagonal format. Next, the parameter matrices and raw data matrix (also in ASCII format) are used as input data for the SAS macro program. Note that the raw data matrix must contain only those variables that appeared in the substantive model. The macro program is quite straightforward, and the user need only specify values for a small number of variables (the missing value code, number of manifest indicators, number of bootstrap samples, a seed value, and the file paths for the ASCII input files). To illustrate, a program excerpt showing the user-specified variables is given in Figure 2. Note that the AMOS macro requires an additional line of code that specifies the names of the manifest variables.

To illustrate the use of the SAS macro, 1,100 bootstrap samples were generated from the EAT data set described earlier,<sup>1</sup> and the two-factor CFA model was fit to each sample using the FIML estimator in *Mplus* (specific instructions for *Mplus* are given later). *Mplus*, like the other commercial SEM packages, saves the results from each sample replication to an ASCII file that can subsequently be analyzed using SPSS or SAS. In this case, the 1,100 model fit statistics were used to construct a reference sampling distribution for the likelihood ratio fit statistic obtained from the parent data. Although there were no convergence failures in this example,

<sup>1</sup>The initial goal was to generate 1,000 bootstrap samples. Anticipating convergence failures, I arbitrarily chose to generate 1,100 samples.



bootstrap samples that produce inadmissible estimates or fail to converge should be discarded from further analysis (Yung & Bentler, 1996).

Before proceeding, it is instructive to examine the empirical sampling distribution of the bootstrap fit statistics, and contrast it to the central chi-square distribution with 34  $df$  (the theoretical sampling distribution that was used to obtain the normal-theory probability value reported earlier). Kernel density graphs of empirical and central chi-square distributions are shown in Figure 3. In Figure 3, it is shown that the empirical sampling distribution is shifted to the right somewhat, and has a thicker tail (and thus a larger variance) than that of the central chi-square distribution. Recall that the expected value and variance of the central chi-square distribution is equal to  $df$  and  $2df$ , respectively, so the expected value of the central chi-square distribution is 34 with a variance of 68. In contrast, the mean of the empirical sampling distribution is 41.29, and its variance is 150.06. The important point to remember is that the shape of the empirical sampling distribution is not a result of model misfit, because the bootstrap samples were drawn from a transformed data set in which the null hypothesis was true. Rather, the discrepancy in the expected value and variance reflects sampling variation that occurs due to features of the data (e.g., nonnormality).

The empirical sampling distribution in Figure 3 can be used to obtain an adjusted probability value for the likelihood ratio fit statistic obtained from the parent sample. This is accomplished by computing the proportion of the 1,100 bootstrap chi-square values that exceeded that of the parent sample, which was  $\chi^2(34, N = 200) = 65.53$ ; this adjusted probability value can readily be determined from a fre-

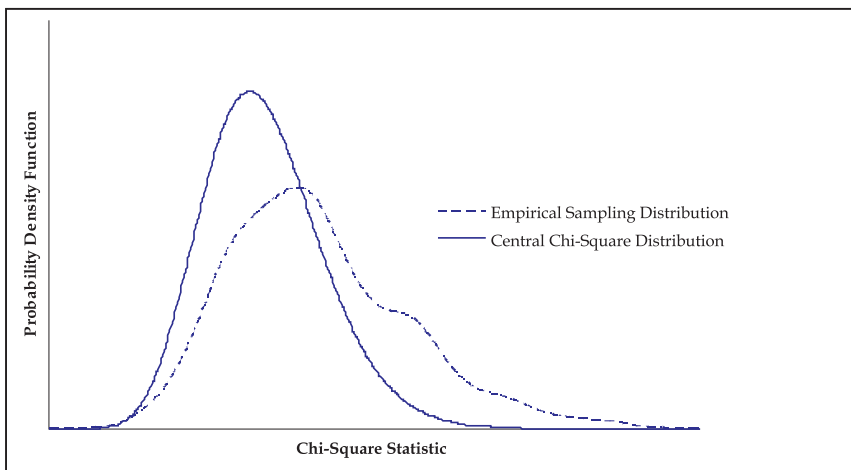


FIGURE 3 Kernel density graphs of empirical and central chi-square distributions.

quency distribution of the bootstrap fit statistics. In the EAT example, 51 of the 1,100 bootstrap chi-square statistics exceeded 65.53, so the Bollen–Stine adjusted probability value is  $51/1,100 = .046$ .

Finally, it should be noted that the parameter estimates from the 1,100 bootstrap samples can be used to obtain model-based empirical standard error estimates (Yung & Bentler, 1996). This is accomplished by simply computing the standard deviation of each parameter estimate over the 1,100 samples.

Although the use of the SAS macro is virtually identical across SEM software packages, the process of repeatedly fitting the model to each bootstrap sample differs considerably. As such, detailed programming instructions for each software package are outlined next.

### USING THE SAS MACRO WITH AMOS

Although AMOS readily performs the Bollen–Stine bootstrap with complete data (Arbuckle & Wothke, 1999), it is somewhat more labor intensive to implement the bootstrap with missing data, because a Visual Basic (VB) program must be used to fit the hypothesized model to each of the bootstrap samples. However, the AMOS software includes a VB programming environment (AMOS Basic), and the VB program given in Appendix A can be easily altered for a specific analytic purpose.

When using AMOS, it is necessary to save each bootstrap data set as a separate file within the same directory. Although the SAS macro performs this task, it is necessary to first create a folder named “bootsamples” in the root directory (i.e., C:\bootsamples). Executing the appropriate SAS program will generate  $B$  bootstrap data files within this directory, each of which is differentiated by a unique index number. To illustrate, the SAS macro was used to generate 1,100 bootstrap samples from the EAT data described previously. The macro generated 1,100 files within the C:\bootsamples directory with file names boot1.txt, boot2.txt, ..., boot1100.txt. As a reminder, the covariance structure of the bootstrap samples has been transformed to be consistent with the hypothesized model (i.e., the null hypothesis is true in the transformed parent data set).

Next, it is necessary to use the VB programming environment (AMOS Basic) to repeatedly fit the hypothesized CFA model to each of the bootstrap samples. The VB program used to fit the 1,100 EAT samples is given in Appendix A, and is also available for download at <http://www.asu.edu/clas/psych/people/faculty/enders.htm>. Briefly, the VB program (a) cycles through each of the bootstrap samples using a loop structure, (b) repeatedly calls the AMOS Engine to perform an analysis on the selected data set, and (c) writes the parameter estimates and fit statistics to an ASCII file. An overview of the VB programming language is beyond the scope of this article, but the program given in Appendix A has numerous comment lines that begin with an apostrophe. Fortunately, very few lines of code need to be

altered to perform the analysis, and these lines are clearly marked in the program. Specifically, the user need only specify the desired number of bootstrap samples and the number of model parameters, including the means or intercepts (these parameters must be estimated when you have missing data). In addition, the user must specify the hypothesized model—in this example, a two-factor CFA—using the VB language. For those who are familiar with the EQS or Simplis programming languages, the AMOS model specification should be fairly straightforward. A detailed tutorial for the AMOS Basic language can be found in the *AMOS 4.0 User's Guide* (Arbuckle & Wothke, 1999, p. 35).

On executing the VB program, one will notice that a new folder named tempboot was created in the root directory (i.e., C:\tempboot)—this folder can be deleted after the VB program is finished executing. This folder is created automatically so that the VB program can select the desired bootstrap sample (e.g., if the loop index equals 100, boot100.txt will be selected) and copy that file to the C:\tempboot folder, overwriting the previous replicate. Although this step seems unnecessary (VB could simply point AMOS to the appropriate file located in the C:\bootamples directory), the time required to fit the bootstrap samples is reduced dramatically if there is only a single data file residing in the target directory. On a related topic, it appears that processing time can also be reduced by using AMOS 4 rather than AMOS 5, or by using Microsoft Visual Basic rather than AMOS Basic.

After executing the VB program, two new files are generated in the root directory, chisquares.txt and parameters.txt, that contain the chi-square statistic and model parameters, respectively, from each of the bootstrap samples. These output statistics can be read into SPSS or SAS using free format (the exact order of the parameter values is listed on the first line of the parameter file). To illustrate, the SPSS syntax needed to import the AMOS results is shown in Appendix B. As described previously, an adjusted probability value for the parent sample is obtained by computing the proportion of bootstrap chi-square values that exceeded the chi-square statistic obtained from the parent sample, which can readily be accomplished by inspecting a frequency distribution. As before, bootstrap standard errors are obtained by computing the standard deviation of the parameter estimates across the 1,100 samples.

## USING THE SAS MACRO WITH LISREL

Unlike AMOS and *Mplus*, LISREL requires the bootstrap samples to be stacked vertically in a single file. Thus, executing the appropriate SAS macro program generates a single ASCII data file in the root directory, bootstacked.txt, that contains the  $B$  bootstrap samples. In the case of the EAT example, the 1,100 bootstrap samples (each of which is  $200 \times 10$  in dimension) were stacked in a single file that is  $220,000 \times 10$  in dimension, such that the first 200 records are associated with the

first bootstrap sample, the next 200 records with the second bootstrap sample, and so on. Again, each of the bootstrap samples is drawn from a transformed data set that has a covariance structure consistent with the null hypothesis.

Next, the hypothesized model is fit to each of the 1,100 bootstrap data sets. The Simplis syntax file for this analysis is given in Appendix C, and is available for download at <http://www.asu.edu/clas/psych/people/faculty/enders.htm>. Only minor programming modifications to the Simplis syntax are required to read and analyze a stacked data file. Specifically, the RP (replications) keyword is placed on the OPTIONS command line to specify the number of samples in the stacked file (it is assumed that each sample in the stacked file has the same  $N$ , which is true in this case). Also, the GF and LX keywords are used to specify the output files that contain the goodness-of-fit statistics and loadings from each bootstrap sample (the LX keyword is only necessary if bootstrap standard error estimates are desired).<sup>2</sup>

Like AMOS, the ASCII output files produced in LISREL can be read into SPSS or SAS using free format, the SPSS syntax for which is given in Appendix B. Consistent with the previous discussion, an adjusted probability value for the original likelihood ratio statistic can be determined by inspecting a frequency distribution of the bootstrap chi-square statistics, and bootstrap standard errors are obtained by computing the standard deviation of the parameter estimates contained in the file specified by the LX keyword.

## USING THE SAS MACRO WITH *MPLUS*

The Monte Carlo simulation facilities in *Mplus* can be used to estimate a structural equation model for each of the  $B$  bootstrap samples generated by the SAS macro. Because the most recent release of *Mplus* (version 3) handles Monte Carlo data differently than its predecessor, the bootstrapping procedure for *Mplus* version 2 and version 3 are described.

Like AMOS, it is necessary to save each bootstrap data set as a separate file within the same directory (the format of the ASCII data differs, however, which is why a separate macro program was written for *Mplus*). Again, the SAS macro program for this analysis is available for download at <http://www.asu.edu/clas/psych/people/faculty/enders.htm>. Before running the SAS macro, it is necessary to first create a folder named `bootsamples` in the root directory (i.e., `C:\bootsamples`). Executing the appropriate SAS program will generate  $B$  bootstrap data files within this directory, each of which is differentiated by a unique index number (e.g., `boot1.txt`, `boot2.txt`, ..., `boot1100.txt`).

---

<sup>2</sup>It may be necessary to download and install a patch for LISREL 8.5 ([www.ssicentral.com](http://www.ssicentral.com)) for the GF keyword to work properly with missing data.

Next, it is necessary to fit the hypothesized model to each of the bootstrap data sets. The *Mplus* 3 syntax for the EAT analysis is given in Appendix D. In *Mplus* version 3, externally generated Monte Carlo data (e.g., the *B* bootstrap data sets) are handled using the TYPE = MONTECARLO option of the DATA command. When specifying TYPE = MONTECARLO, the FILE option is used to specify a file that contains a list of the names of the bootstrap data sets that are to be analyzed. In the current example, the file replist.txt contains a list of the 1,100 file names found in the bootsamples directory. Note that the SAS macro automatically creates the replist.txt file in the root directory, so there is no need for the user to alter the DATA portion of the *Mplus* program shown in Appendix D. The VARIABLE, ANALYSIS, and MODEL commands are identical to those used to analyze the parent sample, so no discussion of these commands is given here; interested readers can consult the *Mplus User's Guide* (Muthén & Muthén, 2004) for further details.

Executing the *Mplus* program in Appendix D fits the two-factor CFA model to each of the bootstrap data sets. Conveniently, the TYPE = MONTECARLO option produces summary tables of the 1,100 replications, and these tables contain the critical values for the empirical chi-square distribution as well as the standard deviation of the parameter estimates across the 1,100 replications (i.e., the bootstrap standard errors). To illustrate, Figure 4 shows the *Mplus* summary

Chi-Square Test of Model Fit			
Degrees of freedom		34	
Mean		41.292	
Std Dev		12.250	
Number of successful computations		1100	
Proportions		Percentiles	
Expected	Observed	Expected	Observed
0.990	0.995	17.789	19.220
0.980	0.990	19.275	20.875
0.950	0.975	21.664	24.403
0.900	0.955	23.952	27.074
0.800	0.905	26.938	30.601
0.700	0.847	29.242	34.031
0.500	0.723	33.336	39.782
0.300	0.569	37.795	45.929
0.200	0.452	40.676	51.712
0.100	0.328	44.903	57.614
0.050	0.249	48.602	64.206
0.020	0.179	52.995	71.401
0.010	0.121	56.061	76.373

FIGURE 4 *Mplus* 3 Monte Carlo output.

table for the chi-square test statistic. The columns labeled Proportions Expected (column 1) and Percentiles Expected (column 3) give alpha values and the corresponding critical values from a central chi-square distribution with 34 *df*, respectively. From the output shown in Figure 4, it is seen that the chi-square critical value ( $df = 34$ ) associated with an alpha level of .05 is 48.60. The column labeled Proportions Observed (column 2) gives the proportion of bootstrap samples that exceeded a given critical value from the central chi-square distribution. In this example, 24.9% of the bootstrap samples would be deemed statistically significant using the critical value from the central chi-square distribution, which results from the fact that the empirical sampling distribution has a much thicker tail than that of the central chi-square distribution (see Figure 3). For the purpose of this demonstration, the column labeled Percentiles Observed (column 4) is important, as it contains the critical values for the empirical sampling distribution. As seen in Figure 4, the critical value associated with an alpha level of .05 is 64.206 (i.e., 5% of the bootstrap chi-square values exceeded 64.206). Because the chi-square statistic obtained from the parent sample ( $\chi^2(34, N = 200) = 65.53$ ) exceeds the critical value from the empirical reference distribution, the model would be rejected at  $p < .05$ .

For those who prefer an exact probability value over a dichotomous retain–reject decision, it is also possible to obtain an adjusted probability value from the empirical reference distribution. Returning to the *Mplus* program given in Appendix D, the RESULTS ARE option of the SAVEDATA command is used to save the parameter estimates, standard errors (which are not needed in this particular application), and fit indexes for each bootstrap sample to an ASCII file. In this example, the bootstrap results are output to a file called *mplusparams.txt*, which is located in the root directory.

As before, the output statistics can be read into SPSS or SAS using free format (the exact order of the statistics in the file is listed near the bottom of the *Mplus* output file). To illustrate, the SPSS syntax needed to import the *Mplus* results is shown in Appendix B. Consistent with the previous examples, an adjusted probability value is obtained by computing the proportion of the bootstrap chi-square values that exceeded the value of the likelihood ratio test statistic for the parent data.

Before concluding, a brief discussion of bootstrap standard errors is warranted. As described previously, the standard deviation of the parameter estimates across the 1,100 samples serves as an empirical estimate of the standard error for each parameter. Conveniently, the *Mplus* 3 output includes a summary table for the 1,100 sets of parameter estimates that includes the mean and standard deviation, among other things. As before, the standard deviation of the bootstrap parameter estimates can be used in lieu of the normal-theory standard errors. A more detailed descrip-

tion of the Monte Carlo summaries can be found in the *Mplus User's Guide* (Muthén & Muthén, 2004, p. 279).

A brief discussion of the *Mplus* 2 procedure is warranted, as this version of the software is likely enjoying widespread use at the time of this writing. Consistent with *Mplus* version 3, it is necessary to save each bootstrap data set as a separate file within the same directory. As such, the SAS macro can be used in conjunction with either version of *Mplus*. However, the procedure for fitting the model to each bootstrap sample differs considerably in *Mplus* version 2. In this case, a series of Monte Carlo utilities (e.g., `runall.bat`) must be downloaded from the *Mplus* home page ([www.statmodel.com](http://www.statmodel.com)). Specific instructions for editing these files is given on the *Mplus* Web site, but the procedure is quite straightforward: The `runstart.bat` file is edited, and the user must specify a number of variables contained within this file (e.g., the directory that contains the files to be analyzed, the name of the output file to which the parameter estimates will be saved, etc.). After specifying the appropriate variables in the `runstart.bat` file, the user executes a file called `runall.bat`. This file repeatedly fits the hypothesized model to each of the bootstrap data sets, and creates an ASCII file containing the parameter estimates and model fit statistics from each replicate. When the program is finished executing, the *Mplus* results can be analyzed using SPSS or SAS in the exact manner described earlier (e.g., see Appendix B).

## PERFORMING THE BOOTSTRAP USING EQS 6.1

As mentioned previously, the forthcoming release of EQS (version 6.1) can be used to perform the modified Bollen–Stine bootstrap for missing data. Although it is not necessary to use the SAS macro program in this context, the EQS syntax for performing this analysis which is given in Appendix E is briefly discussed. For brevity, the user is referred to the *EQS User's Guide* (Bentler & Wu, 2002) for instructions on implementing ML estimation in the missing data context. As seen in Appendix E, the `SIMULATION` command is added to the program, and the `MBB` and `REPLICATIONS` keywords are used to invoke the model-based (i.e., Bollen–Stine) bootstrap and to specify the desired number of bootstrap samples, respectively. Like *Mplus*, EQS provides summary statistics for a number of fit indexes and the parameter estimates. Returning to the EAT example, the critical value associated with an alpha level of .05 is 64.206 (i.e., 5% of the bootstrap chi-square values exceeded 64.206). Because the chi-square statistic obtained from the parent sample ( $\chi^2(34, N = 200) = 65.53$ ) exceeds the critical value from the empirical reference distribution, the model would be rejected at  $p < .05$ .

## DISCUSSION

The number of published bootstrapping applications has, to date, been limited, perhaps due to a lack of bootstrapping routines in commercial SEM software packages. As such, the purpose of this article was to demonstrate the use of a custom SAS macro program that can be used to implement the Bollen–Stine bootstrap with existing SEM software. The focus of the article was on missing data, although the macro could also be used with complete data sets. A series of heuristic analyses was performed using a CFA model of the EAT. Using the provided macro program, it is quite straightforward to implement the Bollen–Stine bootstrap. Although the bootstrap is more computationally demanding than rescaled statistics, this should not preclude its use. For example, generating and analyzing the 1,100 bootstrap samples from the EAT data set took less than 3 min. Detailed programming instructions were presented for each of the commercial SEM software packages, and it is hoped that this article facilitates the use of the bootstrap for those who are interested in implementing this technique in their own research studies, and enables researchers to further study the behavior of this corrective procedure under a variety of missing data conditions.

## ACKNOWLEDGMENTS

Craig K. Enders is now in the Department of Psychology, Arizona State University.

I would like to thank Yan Cheng and Gerhard Mels at Scientific Software for their valuable assistance with LISREL 8.5.

## REFERENCES

- Arbuckle, J. L. (1996). Full information estimation in the presence of incomplete data. In G. A. Marcoulides & R. E. Schumacker (Eds.), *Advanced structural equation modeling* (pp. 243–247). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Arbuckle, J. L., & Wothke, W. (1999). *AMOS 4.0 user's guide* [Computer software and manual]. Chicago: SmallWaters.
- Bentler, P. M., & Wu, E. J. C. (2002). *EQS 6 for Windows user's guide* [Computer software and manual]. Encino, CA: Multivariate Software.
- Beran, R., & Srivastava, M. S. (1985). Bootstrap tests and confidence regions for functions of a covariance matrix. *Annals of Statistics*, *13*, 95–115.
- Bollen, K. A., & Stine, R. A. (1992). Bootstrapping goodness-of-fit measures in structural equation models. *Sociological Methods and Research*, *21*, 205–229.
- Doninger, G. L., Enders, C. K., & Burnett, K. F. (2005). Validity evidence for Eating Attitudes Test scores in a sample of female college athletes. *Measurement in Physical Education and Exercise Science*, *9*, 35–49.
- Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York: Chapman & Hall.



- Enders, C. K. (2001). The impact of nonnormality on full information maximum likelihood estimation for structural equation models with missing data. *Psychological Methods*, 6, 352–370.
- Enders, C. K. (2002). Applying the Bollen–Stine bootstrap for goodness-of-fit measures to structural equation models with missing data. *Multivariate Behavioral Research*, 37, 359–377.
- Finney, S., & DiStefano, C. (in press). Dealing with nonnormality and categorical data in structural equation modeling. In G. R. Hancock & R. O. Mueller (Eds.), *A second course in structural equation modeling*. Greenwich, CT: Information Age.
- Fouladi, R. T. (1998, April). *Covariance structure analysis techniques under conditions of multivariate normality and nonnormality—Modified and bootstrap based test statistics*. Paper presented at the annual meeting of the American Educational Research Association, San Diego, CA.
- Garner, D. M., Olmsted, M. P., Bohr, Y., & Garfinkel, P. E. (1982). The Eating Attitudes Test: Psychometric features and clinical correlates. *Psychological Medicine*, 12, 871–878.
- Muthén, L. K., & Muthén, B. O. (2004). *Mplus user's guide* [Computer software and manual]. Los Angeles: Muthén & Muthén.
- Nevitt, J., & Hancock, G. R. (2001). Performance of bootstrapping approaches to model test statistics and parameter standard error estimation in structural equation modeling. *Structural Equation Modeling*, 8, 353–377.
- Satorra, A., & Bentler, P. M. (1994). Corrections to test statistics and standard errors in covariance structure analysis. In A. von Eye & C. C. Clogg (Eds.), *Latent variables analysis: Applications for developmental research* (pp. 399–419). Thousand Oaks, CA: Sage.
- Stine, R. (1989). An introduction to bootstrap methods: Examples and ideas. *Sociological Methods and Research*, 18, 243–291.
- Yuan, K.-H., & Bentler, P. M. (2000). Three likelihood-based methods for mean and covariance structure analysis with nonnormal missing data. In M. Becker & M. Sobel (Eds.), *Sociological methodology 2000* (pp. 165–200). Malden, MA: Blackwell.
- Yung, Y.-F., & Bentler, P. M. (1996). Bootstrapping techniques in analysis of mean and covariance structures. In G. A. Marcoulides & R. E. Schumacker (Eds.), *Advanced structural equation modeling: Issues and techniques* (pp. 195–226). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

## APPENDIX A VISUAL BASIC PROGRAM FOR FITTING BOOTSTRAP SAMPLES

```

` SPECIFY THE FOLLOWING INFORMATION IN THE SUBMAIN SECTION BELOW:
` (A) THE NUMBER OF DESIRED BOOTSTRAP SAMPLES (E.G., 1000)
` (B) NUMBER OF ESTIMATED PARAMETERS
` (C) THE OVERIDENTIFIED MODEL
` (D) THE SATURATED MODEL
Option Explicit
Sub Main()
` DO NOT ALTER THE FOLLOWING THREE LINES
    Dim chisq As Double, cminMod As Double, cminSat As Double
    Dim outfile As String, source As String, infile As String
    Dim projPath As String, outfilep As String
    Dim bsCount As Integer, boot As Integer, modStatus As Integer
    Dim satStatus As Integer, numFit As Integer, j As Integer
    Dim npars As Integer
` SPECIFY (A) AND (B) BELOW

```

```

` (A) SPECIFY THE NUMBER OF DESIRED BOOTSTRAP FIT STATISTICS
    numFit = 1000
` (B) SPECIFY THE NUMBER OF ESTIMATED MODEL PARAMETERS, INCLUDING MEANS
    Dim params(31)
    Dim names(31)
` DO NOT ALTER THE FOLLOWING CODE
    Mkdir("c:\tempboot")
` SPECIFY THE DIRECTORY THAT CONTAINS THE BOOTSTRAP SAMPLES
    projPath = "c:\bootsamples\"
` SPECIFY THE OUTPUT PATH FOR THE CHI-SQUARES AND PARAMETER ESTIMATES
    outfile = "c:\chisquares.txt"
    outfilep = "c:\parameters.txt"
` INITIALIZE COUNTERS
    bsCount = 0
    boot = 0
    Open outfile For Output As #1
    Open outfilep For Output As #2
` START BOOTSTRAP LOOP
    Do
        boot = boot + 1
` READ FILES
        source = projPath & "boot" & boot & ".txt"
        infile = "c:\tempboot\temp.txt"
        FileCopy source, infile
` CALL MODEL ESTIMATION SUBROUTINES
        Call AmosSemModel(infile, cminMod, modStatus, params, names, npars)
        Call AmosSat(infile, cminSat, satStatus)
` DO IF CONVERGED
        If modStatus = 1 And satStatus = 1 Then
            bsCount = bsCount + 1
            Debug.Print "Number of Bootstrap Samples Converged = " & bsCount
` COMPUTE FIT STATISTIC AND WRITE TO FILE
            chisq = cminMod - cminSat
            Write #1, chisq
` WRITE PARAMETER NAMES TO FILE ONLY FOR 1ST CONVERGED SAMPLE
            If bsCount = 1 Then
                For j = 1 To (npars)
                    Write #2, names(j)
                Next j
            Else
                End If
` WRITE PARAMETER VALUES TO A FILE AFTER NAMES
            For j = 1 To npars
                Write #2, params(j)
            Next j
` CLOSE DO IF CONVERGED
        Else
            End If
` END BOOTSTRAP SAMPLE LOOP

```

```

    Loop Until bsCount = numFit
  ` CLOSE OUTPUT FILES
    Close #1
    Close #2
End Sub
Sub AmosSemModel(infile, cminMod, modStatus, params, names, npars)
  Dim sem As New AmosEngine
  Dim Converge As Boolean, admiss As Boolean
  Dim p As Integer
  Converge = False
  admiss = False
  modStatus = 0
  sem.ModelMeansAndIntercepts
  On Error Resume Next
  ` (C) SPECIFY HYPOTHESIZED MODEL BELOW
  ` SPECIFY OVERIDENTIFIED MODEL
  sem.BeginGroup (infile)
    sem.Structure "driveft (1)"
    sem.Structure "foodpre (1)"
    sem.Structure "driveftfoodpre (flf2r)"
    sem.Structure "eat1 = (xbar1) + (load1) driveft + (1) e1"
    sem.Structure "eat2 = (xbar2) + (load2) driveft + (1) e2"
    sem.Structure "eat10 = (xbar10) + (load10) driveft + (1) e10"
    sem.Structure "eat11 = (xbar11) + (load11) driveft + (1) e11"
    sem.Structure "eat12 = (xbar12) + (load12) driveft + (1) e12"
    sem.Structure "eat14 = (xbar14) + (load14) driveft + (1) e14"
    sem.Structure "eat24 = (xbar24) + (load24) driveft + (1) e24"
    sem.Structure "eat3 = (xbar3) + (load3) foodpre + (1) e3"
    sem.Structure "eat18 = (xbar18) + (load18) foodpre + (1) e18"
    sem.Structure "eat21 = (xbar21) + (load21) foodpre + (1) e21"
    sem.Structure "e1 (e1)"
    sem.Structure "e2 (e2)"
    sem.Structure "e10 (e10)"
    sem.Structure "e11 (e11)"
    sem.Structure "e12 (e12)"
    sem.Structure "e14 (e14)"
    sem.Structure "e24 (e24)"
    sem.Structure "e3 (e3)"
    sem.Structure "e18 (e18)"
    sem.Structure "e21 (e21)"
  ` GET PARAMETER VALUES AND PARAMETER NAMES FOR OUTPUT
  For p = 1 To sem.NumberOfParameters
    params(p) = sem.ParameterValue(p)
    names(p) = sem.ParameterName(p)
  Next
  ` ASSESS CONVERGENCE AND ADMISSIBILITY STATUS
  npars = sem.NumberOfParameters
  Converge = (sem.FitModel() = 0)
  admiss = sem.Admissible

```

```

    cminMod = sem.Cmin
    If Converge = True And admis = True Then modStatus = 1
` REINITIALIZE AMOS ENGINE
    Set sem = Nothing
    DoEvents
End Sub
Sub AmosSat(infile, cminSat, satStatus)
    Dim sem As New AmosEngine
    Dim Converge As Boolean, admis As Boolean
    Converge = False
    admis = False
    satStatus = 0
    sem.ModelMeansAndIntercepts
    On Error Resume Next
` (D) SPECIFY SATURATED MODEL BELOW
` SPECIFY SATURATED MODEL
    sem.BeginGroup (infile)
        sem.Structure "eat1 = (xbar1)"
        sem.Structure "eat2 = (xbar2)"
        sem.Structure "eat10 = (xbar10)"
        sem.Structure "eat11 = (xbar11)"
        sem.Structure "eat12 = (xbar12)"
        sem.Structure "eat14 = (xbar14)"
        sem.Structure "eat24 = (xbar24)"
        sem.Structure "eat3 = (xbar3)"
        sem.Structure "eat18 = (xbar18)"
        sem.Structure "eat21 = (xbar21)"
` ASSESS CONVERGENCE AND ADMISSIBILITY STATUS
    Converge = (sem.FitModel() = 0)
    admis = sem.Admissible
    cminSat = sem.Cmin
    If Converge = True And admis = True Then satStatus = 1
` REINITIALIZE AMOS ENGINE
    Set sem = Nothing
    DoEvents
End Sub

```

## APPENDIX B

### SPSS SYNTAX FOR READING AMOS, MPLUS, AND LISREL OUTPUT FILES

```

* READ BOOTSTRAP CHI-SQUARES FROM AMOS OUTPUT.
data list free file = "c:\chisquares.txt"
  /chisq.
execute.
frequencies chisq.
* READ BOOTSTRAP PARAMETERS FROM AMOS OUTPUT.

```

```

data list free file = "c:\parameters.txt"
/f1f2cov icept1 load1 icept2 load2 icept10 load10 icept11 load11
  icept12 load12 icept14 load14 icept24 load24 icept3 load3
  icept18 load18 icept21 load21 res1 res2 res10 res11
  res12 res14 res24 res3 res18 res21.
execute.
descriptives f1f2cov to res21.
* READ BOOTSTRAP RESULTS FROM LISREL OUTPUT.
data list free file = 'c:\fit.txt'
/sample
  v2 v3
  df chisq pval
  v7 to v19
  rmsea lo hi rmseap
  v24 to v47.
frequencies chisq.
* READ BOOTSTRAP RESULTS FROM MPLUS OUTPUT.
data list free file = 'c:\mplusparams.txt'
/icept1 icept2 icept3 icept10 icept11
  icept12 icept14 icept18 icept21 icept24
  load1 load2 load3 load10 load11
  load12 load14 load18 load21 load24
  res1 res2 res3 res10 res11
  res12 res14 res18 res21 res24 f1f2cov
  se1 to se31 chisq fit1 to fit15.
execute.
frequencies chisq.
descriptives icept1 to f1f2cov.

```

## APPENDIX C

### SIMPLIS SYNTAX FOR FITTING BOOTSTRAP SAMPLES

```

Observed Variables: eat1 eat2 eat3 eat10 eat11 eat12 eat14 eat18
eat21 eat24
Missing Value Code -9
Sample Size: 200
Raw Data from File c:\bootstacked.txt
Latent Variables: driveft foodpre
Relationships:
  eat1 = const + driveft
  eat2 = const + driveft
  eat10 = const + driveft
  eat11 = const + driveft
  eat12 = const + driveft
  eat14 = const + driveft
  eat24 = const + driveft
  eat3 = const + foodpre

```

```

eat18 = const + foodpre
eat21 = const + foodpre
Set the error variances of eat1 - eat24 free
Set the variance of driveft to 1
Set the variance of foodpre to 1
Options: sc rp = 1100 lx = loadings.txt gf = fit.txt
End of Problem

```

## APPENDIX D MPLUS 3 SYNTAX FOR FITTING BOOTSTRAP SAMPLES

```

TITLE:
  Mplus external Monte Carlo program;
DATA:
  file is 'c:\replist.txt';
  type is montecarlo;
VARIABLE:
  names are eat1 eat2 eat3 eat10 eat11 eat12 eat14 eat18 eat21
  eat24;
  usevariables are eat1 - eat24;
  missing are all (-9);
ANALYSIS:
  type = h1 missing;
  estimator = ml;
MODEL:
  driveft by eat1* eat2 eat10 eat11 eat12 eat14 eat24;
  foodpre by eat3* eat18 eat21;
  driveft@1 foodpre@1;
  driveft with foodpre;
SAVEDATA:
  results are "c:\mplusparams.txt";

```

## APPENDIX E EQS 6.1 SYNTAX FOR PERFORMING THE BOLLEN-STINE BOOTSTRAP

```

/TITLE
  Modified Bollen-Stine Bootstrap for Missing Data
/SPECIFICATIONS
  DATA='C:\eatboot.ess';
  VARIABLES=10; CASES=200;
  METHOD=ML; ANALYSIS=MOMENT; MATRIX=RAW;
  MISSING=ML; SE=OBSERVED;

```

```

/LABELS
V1=eat1; V2=eat2; V3=eat3; V4=eat10; V5=eat11;
V6=eat12; V7=eat14; V8=eat18; V9=eat21; V10=eat24;
/EQUATIONS
V1 = *V999 + *F1 + E1;
V2 = *V999 + *F1 + E2;
V3 = *V999 + *F2 + E3;
V4 = *V999 + *F1 + E4;
V5 = *V999 + *F1 + E5;
V6 = *V999 + *F1 + E6;
V7 = *V999 + *F1 + E7;
V8 = *V999 + *F2 + E8;
V9 = *V999 + *F2 + E9;
V10 = *V999 + *F1 + E10;
F1 = D1;
F2 = D2;
/VARIANCES
V999 = 1;
E1 = *;
E2 = *;
E3 = *;
E4 = *;
E5 = *;
E6 = *;
E7 = *;
E8 = *;
E9 = *;
E10 = *;
D1 = 1;
D2 = 1;
/COVARIANCES
D2,D1 = *;
/PRINT
FIT=ALL;
TABLE=EQUATION;
/OUTPUT
DATA='EQSOUT.ETS';
/SIMULATION
MBB=200;
REPLICATIONS=1100;
SEED=12345.0;
SAVE=NO;
/END

```