7

The Imputation Phase of Multiple Imputation

7.1 CHAPTER OVERVIEW

Recall from previous chapters that maximum likelihood estimation uses a log-likelihood function to identify the population parameter values that are most likely to have produced the observed data. The estimation process essentially auditions different parameter values until it identifies the estimates that minimize the standardized distance to the observed data. This process does not involve imputation. Rather, maximum likelihood estimates the parameters directly from the available data, and it does so in a way that does not require individuals to have complete data records. Multiple imputation is an alternative to maximum likelihood estimation and is the other state-of-the-art missing data technique that methodologists currently recommend (Schafer & Graham, 2002). The imputation approach outlined in this chapter makes the same assumptions as maximum likelihood estimation—missing at random (MAR) data and multivariate normality—but takes the very different tack of filling in the missing values prior to analysis.

A multiple imputation analysis consists of three distinct steps: the imputation phase, the analysis phase, and the pooling phase. Figure 7.1 shows a graphical depiction of the process. The **imputation phase** creates multiple copies of the data set (e.g., $m = 20$), each of which contains different estimates of the missing values. Conceptually, this step is an iterative version of stochastic regression imputation, although its mathematical underpinnings rely heavily on Bayesian estimation principles. As its name implies, the goal of the **analysis phase** is to analyze the filled-in data sets. This step applies the same statistical procedures that you would have used had the data been complete. Procedurally, the only difference is that you perform each analysis m times, once for each imputed data set. The analysis phase yields m sets of parameter estimates and standard errors, so the purpose of the **pooling phase** is to combine everything into a single set of results. Rubin (1987) outlined relatively straightforward formulas for pooling parameter estimates and standard errors. For example, the pooled parameter estimate is simply the arithmetic average of the m estimates from the

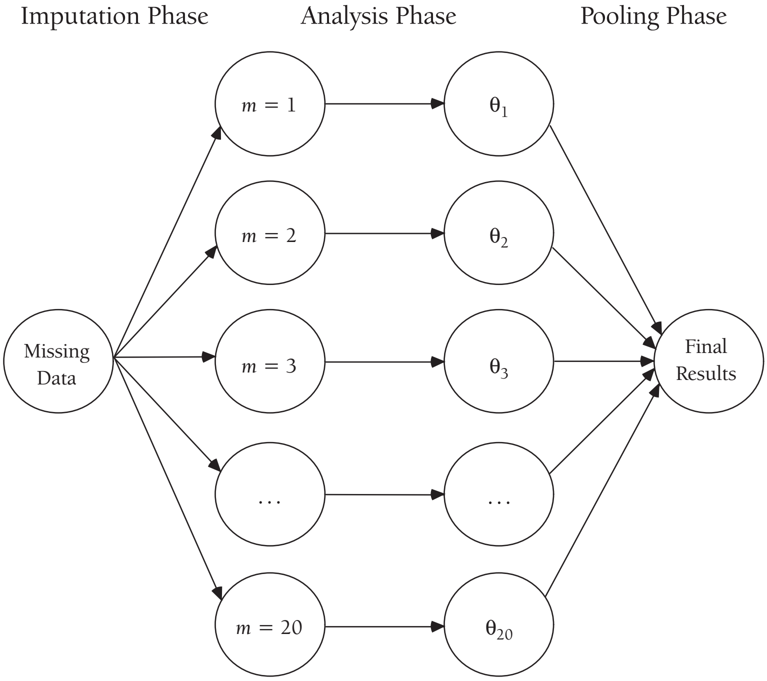

FIGURE 7.1. Graphical depiction of a multiple imputation analysis. The imputation phase creates multiple copies of the data set (i.e., $m = 20$) and imputes each with different missing values. The analysis phase estimates the model parameters using each of the complete data sets. The pooling phase combines the parameter estimates and standard errors into a single set of of results.

analysis phase. Combining the standard errors is slightly more complex but follows the same logic. The process of analyzing multiple data sets and pooling the results sounds very tedious, but multiple imputation software packages completely automate the procedure. The imputation phase is arguably the most difficult aspect of a multiple imputation analysis, so I devote Chapter 7 to this topic and outline the analysis and pooling phases in Chapter 8.

Multiple imputation is actually a broad term that encompasses a collection of techniques. The three-step process (i.e., imputation, analysis, pooling) is common to all multiple imputation procedures, but methodologists have proposed a variety of algorithms for the imputation phase (King, Honaker, Joseph, & Scheve, 2001; Lavori, Dawson, & Shera, 1995; Raghunathan, Lepkowski, Van Hoewyk, & Solenberger, 2001; Royston, 2005; Schafer, 1997, 2001; van Buuren, 2007). These algorithms address different types of problems (e.g., categorical versus continuous data, longitudinal versus cross-sectional data, monotone missing data patterns versus general patterns), so no single procedure works best in every situation. Because the normal distribution is arguably one of the most widely used data models in the social and behavioral sciences, I devote this chapter to an imputation approach that assumes multivariate normality. This so-called data augmentation algorithm (Schafer, 1997; Tanner & Wong, 1987) is perhaps the most widely used imputation approach and is readily available in a number of commercial and freeware software packages. I briefly outline a few alternative imputation algorithms in Chapter 9.

As an important aside, researchers often object to imputation on grounds that the procedure is somehow cheating by "making up data." This concern is ungrounded for at least three reasons. First, it is important to remember that the primary goal of a statistical analysis is to estimate the population parameters. In truth, multiple imputation is nothing more than a mathematical tool that facilitates that task, so imputation itself is ancillary to the end goal. Second, multiple imputation and maximum likelihood estimation are asymptotically (i.e., in very large samples) equivalent and tend to produce the same results. The fact that the two procedures—only one of which fills in the data—are effectively interchangeable underscores the point that imputation is not inherently problematic. Finally, unlike other imputation routines, multiple imputation explicitly accounts for the uncertainty associated with the missing data. By repeatedly filling in the data, multiple imputation yields parameter estimates that average over a number of plausible replacement values, so the process never places faith in a single set of imputations. This is in stark contrast to imputation techniques that treat a single set of filled-in values as real data (e.g., the single imputation methods from Chapter 2).

I use the small data set in Table 7.1 to illustrate ideas throughout this chapter. I designed these data to mimic an employee selection scenario where prospective employees complete an IQ test and a psychological well-being questionnaire during their interview. The company subsequently hires the applicants that score in the upper half of the IQ distribution, and a supervisor rates their job performance following a 6-month probationary period. Note that the job performance scores are missing at random (MAR) because they are systematically missing as a function of IQ scores (i.e., individuals in the lower half of the IQ distribution were never hired and thus have no performance rating). In addition, I randomly deleted three of the well-being scores in order to mimic a missing completely at random (MCAR) mechanism (e.g., the human resources department inadvertently loses an applicant's well-being questionnaire). This data set is far too small for a serious application of multiple imputation, but it is useful for illustrating the basic mechanics of the imputation phase.

TABLE 7.1. Employee Selection Data Set

IQ	Psychological well-being	Job performance
78	13	—
84	9	—
84	10	—
85	10	—
87	—	—
91	3	—
92	12	—
94	3	—
94	13	—
96	—	—
99	6	7
105	12	10
105	14	11
106	10	15
108	—	10
112	10	10
113	14	12
115	14	14
118	12	16
134	11	12

7.2 A CONCEPTUAL DESCRIPTION OF THE IMPUTATION PHASE

Rubin (1987) developed multiple imputation in the Bayesian framework, and data augmentation relies heavily on Bayesian methodology. The imputation phase has relatively intuitive logic (e.g., repeatedly impute the data and update the parameters), but its reliance on Bayesian principles can make it difficult to grasp. This section gives a conceptual description of data augmentation that does not rely on Bayesian statistics. The goal of this section is to lay the foundation for the more precise explanation that I give in the next section, but also to provide an overview of data augmentation for researchers who want to use multiple imputation without necessarily mastering its mathematical underpinnings. I use the IQ and job performance scores from Table 7.1 to illustrate the imputation phase. A bivariate analysis with a single incomplete variable is a very basic application of data augmentation, but the ideas in this section readily generalize to multivariate analyses.

The I-Step

The data augmentation algorithm is a two-step procedure that consists of an imputation step (I-step) and a posterior step (P-step). Procedurally, the **I-step** is identical to the stochastic regression procedure from Chapter 2. Specifically, the I-step uses an estimate of the mean vector and the covariance matrix to build a set of regression equations that predict the incomplete variables from the observed variables. The bivariate analysis example is straightforward because there is only one pattern with missing data (the subset of cases with missing job performance scores), and thus only one regression equation. The imputation equation is

$$JP_i^* = [\hat{\beta}_0 + \hat{\beta}_1(IQ_i)] + z_i \tag{7.1}$$

where JP_i^* is the imputed job performance rating for case i, the brackets contain the regression coefficients that generate the predicted job performance rating for that individual, and z_i is a random residual from a normal distribution. The normal curve that generates the residuals has a mean of zero and a variance equal to the residual variance from the regression of job performance on IQ (i.e., $\sigma^2_{JP|IQ}$). Consistent with stochastic regression imputation, substituting an IQ score into the bracketed terms yields a predicted job performance rating. The predicted scores fall directly on a regression line (or a regression surface, in the multivariate case), so adding a normally distributed residual term to each predicted value restores variability to the imputed data.

The P-Step

The ultimate goal of the imputation phase is to generate m complete data sets, each of which contains unique estimates of the missing values. Creating unique imputations requires different estimates of the regression coefficients at each I-step, and the purpose of the **P-step** is to generate alternate estimates of the mean vector and the covariance matrix (the building blocks of the I-step regression equations). Although this process relies heavily on Bayesian estimation principles, it is straightforward to understand at a conceptual level. Specifically, the P-step begins by using the filled-in data from the preceding I-step to estimate the mean vector and the covariance matrix. Next, the algorithm generates a new set of parameter values by adding a random residual term to each element in $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$. Randomly perturbing the parameter values is akin to drawing a new set of plausible estimates from a sampling distribution (or alternatively, a posterior distribution).

To illustrate the P-step in more detail, suppose that the mean and the variance of the filled-in job performance scores from a particular I-step are $\hat{\mu}_{JP} = 10$ and $\hat{\sigma}^2_{JP} = 9$, respectively. The sampling distribution (or in the Bayesian context, the posterior distribution) of the mean is a normal curve with a standard deviation of $\hat{\sigma}/\sqrt{N}$, so a new sample of 20 job performance scores should produce a mean that deviates from the current estimate by $3/\sqrt{20} = 0.67$ points, on average. To generate an alternate estimate of the mean, the P-step uses Monte Carlo simulation to draw a random residual term from a normal distribution with a mean of zero and a standard deviation of 0.67. Adding this residual to $\hat{\mu}_{JP} = 10$ gives a new estimate of the job performance mean that randomly differs from that of the filled-in data. The same process generates new covariance matrix elements, but these parameters require a different residual distribution (the inverse Wishart distribution from Chapter 6).

Adding residual terms to the elements in the mean vector and the covariance matrix produces parameter values that randomly differ from those that produced the regression coefficients at the preceding I-step. Carrying the updated estimates forward to the next I-step yields a new set of regression coefficients and a different set of imputations. The new imputations carry forward to the next P-step, where the algorithm generates another set of plausible parameter estimates. Repeating this two-step procedure a large number of times creates multiple copies of the data, each of which contains unique estimates of the missing values.

7.3 A BAYESIAN DESCRIPTION OF THE IMPUTATION PHASE

The previous description of the data augmentation algorithm is conceptual in nature and omits many of the mathematical details. This section expands the previous ideas and gives a more precise explanation of the I-step and the P-step. In particular, I illustrate how the Bayesian estimation principles from Chapter 6 apply to the imputation phase. In doing so, I continue to use the IQ and job performance scores from Table 7.1. Again, a bivariate analysis with a single incomplete variable is a very basic example, but surprisingly few changes occur when applying data augmentation to multivariate data.

The I-Step

As I explained in the previous section, the computational details of the I-step are identical to stochastic regression (i.e., use regression equations to predict the incomplete variables from the observed variables and add random residuals to the predicted scores). To illustrate the imputation process graphically, the top panel of Figure 7.2 shows a scatterplot of a set of imputed job performance ratings. The solid regression line corresponds to the predicted job performance scores (i.e., the values generated by the bracketed terms in Equation 7.1), and the dashed lines represent the random residuals (i.e., the z_i values).

From a Bayesian perspective, the imputed values are random draws from a **conditional distribution** that depends on the observed data and the estimates of the mean vector and the covariance matrix at a particular I-step. (Bayesian texts sometimes refer to this distribution as the **posterior predictive distribution**.) The bottom panel of Figure 7.2 imposes normal residual distributions over the regression line at IQ values of 80, 90, and 100. Each of the normal curves represents the conditional distribution of job performance ratings, given the particular IQ score on the horizontal axis (i.e., the distribution of performance ratings for a hypothetical subsample of cases that share the same IQ). The regression line intersects each distribution at its mean, so the predicted job performance ratings (i.e., the bracketed terms in Equation 7.1) are **conditional means** (i.e., the expected performance rating for a hypothetical subsample of cases that share the same IQ). The normal curves represent the distri-

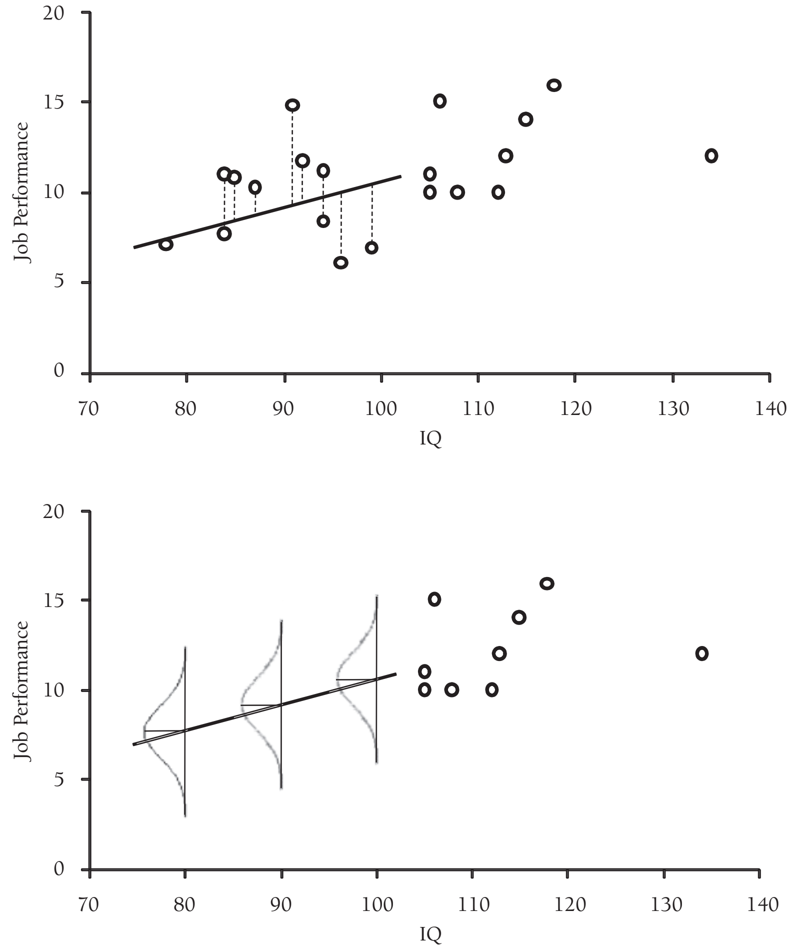

FIGURE 7.2. The top panel shows a hypothetical imputed data set. The solid regression line denotes the predicted job performance scores, and the dashed lines represent the random residuals. The bottom panel shows normal curves imposed over the regression line at IQ values of 80, 90, and 100. These curves represent the conditional distribution of job performance ratings at three different IQ scores (i.e., the distribution of performance ratings for a hypothetical subsample of cases that share the same IQ). The imputed values are random draws from the conditional distributions.

bution of the residuals, so adding a z_i value to each predicted score effectively simulates a random draw from a distribution of plausible replacement values that is contingent on the observed IQ data.

More formally, the following equation summarizes the I-step

$$Y_t^* \sim p(Y_{\text{mis}} \mid Y_{\text{obs}}, \theta_{t-1}^*) \tag{7.2}$$

where Y_t^* represents the imputed values at I-step t, Y_{mis} is the missing portion of the data (e.g., the missing job performance ratings), Y_{obs} is the observed portion of data (e.g., the observed IQ scores), and θ_{t-1}^* denotes the mean vector and the covariance matrix from the preceding P-step (i.e., the parameter values that generate the imputation regression equations). In words, Equation 7.2 says that the imputed values at a particular I-step are random draws from a distribution (the ~ symbol means "distributed as") of plausible replacement values that depends on the observed data and the current parameter estimates. Regardless of how you conceptualize the I-step, the computational details amount to stochastic regression imputation.

The P-Step

The P-step is essentially a standalone Bayesian analysis that describes the posterior distributions of the mean vector and the covariance matrix. Recall that a Bayesian analysis consists of three steps: specify a prior distribution, estimate a likelihood function, and define the posterior distribution. This section presents the relevant posterior distributions but provides no background on their derivations. Chapter 6 describes the Bayesian analytic steps in some detail, so it may be useful to review Sections 6.8 through 6.10 before proceeding.

Creating multiple sets of imputed values requires different estimates of the mean vector and the covariance matrix at each I-step, and the purpose of the P-step is to generate alternate parameter values. The Bayesian framework is ideally suited for this task because it views a parameter as a random variable that has a distribution of values. In the previous section, I stated that the P-step generates new parameter estimates by adding a random residual term to each element in $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$. This description is conceptually accurate, but mathematically imprecise. More accurately, the P-step randomly draws a new mean vector and a new covariance matrix from their respective posterior distributions. Throughout the chapter, I refer to these new estimates as **simulated parameters** because Monte Carlo computer simulation techniques generate their values.

To begin, the P-step uses the filled-in data from the preceding I-step to compute the sample means and the sample sum of squares and cross products matrix (i.e., $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Lambda}}$, respectively). Having obtained these quantities, note that the posterior distribution of the covariance matrix is

$$p(\boldsymbol{\Sigma} \mid \hat{\boldsymbol{\mu}}, \mathbf{Y}) \sim W^{-1}(N-1, \hat{\boldsymbol{\Lambda}}) \tag{7.3}$$

where $p(\boldsymbol{\Sigma} \mid \hat{\boldsymbol{\mu}}, Y)$ denotes the posterior, $\hat{\boldsymbol{\mu}}$ is the vector of sample means, $\mathbf{Y}$ is the filled-in data matrix from the preceding I-step, $\sim W^{-1}$ represents the inverse Wishart distribution, $N-1$ is the degrees of freedom (i.e., the distribution's expected value), and $\hat{\mathbf{\Lambda}}$ is the sample sum of squares and cross products matrix (i.e., the matrix that defines the spread of the distribution). Notice that this posterior distribution has the same form as the one from Chapter 6 (see Equation 6.20). Having defined the shape of the posterior distribution, the data augmentation algorithm uses Monte Carlo simulation techniques to "draw" a new covariance matrix from the posterior. Procedurally, this amounts to using a computer to generate a matrix of random numbers from the distribution in Equation 7.3. To avoid confusion with the sample estimates, I denote the simulated covariance matrix as $\mathbf{\Sigma}^*$.

The algorithm uses a similar procedure to create a new set of means. Specifically, the sample means and the simulated covariance matrix define the posterior distribution of the mean vector, as follows:

$$p(\hat{\boldsymbol{\mu}}|\mathbf{Y}, \mathbf{\Sigma}) \sim MN(\hat{\boldsymbol{\mu}}, N^{-1}\mathbf{\Sigma}^*) \tag{7.4}$$

where $p(\hat{\boldsymbol{\mu}}|\mathbf{Y}, \mathbf{\Sigma})$ is the posterior, $\sim MN$ denotes a multivariate normal distribution, $\hat{\boldsymbol{\mu}}$ is the vector of sample means, and $\mathbf{\Sigma}^*$ is the simulated covariance matrix. Again, this posterior distribution is the same as the one described in Chapter 6 (see Equation 6.9). Finally, Monte Carlo computer simulation techniques generate a new set of means from the distribution in Equation 7.4. I denote the resulting estimates as $\hat{\boldsymbol{\mu}}^*$.

After drawing new parameter values from the posterior distributions, the subsequent I-step uses the updated estimates to construct a new set of regression coefficients and a different set of imputations. The new imputations carry forward to the next P-step, where the algorithm draws another set of plausible parameter estimates. Repeating the two-step procedure a number of times generates multiple copies of the data, each of which contains unique estimates of the missing values.

More formally, the following equation summarizes the P-step

$$\boldsymbol{\theta}_t^* \sim p(\boldsymbol{\theta}|\mathbf{Y}_{\text{obs}}, Y_t^*) \tag{7.5}$$

where $\boldsymbol{\theta}_t^*$ denotes the simulated parameter values from P-step t (i.e., $\boldsymbol{\mu}^*$ and $\mathbf{\Sigma}^*$), Y_{obs} is the observed data (e.g., the observed IQ scores), and Y_t^* contains the imputed values from the preceding I-step. In words, Equation 7.5 says that the simulated parameter values from P-step t are random draws from a distribution that depends on the observed data and the filled-in values from the preceding I-step. A lack of familiarity with Bayesian estimation can make it difficult to grasp the nuances of the P-step, but the process described above is conceptually straightforward: use the filled-in data to estimate the mean vector and the covariance matrix and generate a new set of plausible parameter values by adding a random residual to each element in $\hat{\boldsymbol{\mu}}$ and $\hat{\mathbf{\Sigma}}$.

7.4 A BIVARIATE ANALYSIS EXAMPLE

Having outlined data augmentation in more detail, I use the IQ and job performance scores in Table 7.1 to illustrate a worked example. Multiple imputation software programs fully automate the data augmentation procedure, so there is no need to perform the computational steps manually. Nevertheless, examining what happens at each step of the process is instructive and gives some insight into the inner workings of the "black box."

Consistent with a maximum likelihood analysis, data augmentation requires an initial estimate of the mean vector and the covariance matrix to get started. For reasons discussed later, maximum likelihood parameter estimates make good starting values, so I use the estimates from Chapter 4 for this purpose.

$$\hat{\boldsymbol{\mu}}_0 = \begin{bmatrix} \hat{\mu}_{IQ} \\ \hat{\mu}_{JP} \end{bmatrix} = \begin{bmatrix} 100.000 \\ 10.281 \end{bmatrix}$$

$$\hat{\boldsymbol{\Sigma}}_0 = \begin{bmatrix} \hat{\sigma}^2_{IQ} & \hat{\sigma}_{IQ,JP} \\ \hat{\sigma}_{JP,IQ} & \hat{\sigma}^2_{JP} \end{bmatrix} = \begin{bmatrix} 189.600 & 23.392 \\ 23.392 & 8.206 \end{bmatrix}$$

Throughout this section, I use a numeric subscript to index each data augmentation cycle, and the value of zero indicates that these parameter estimates are starting values that precede the first I-step.

The initial I-step uses the elements in $\hat{\boldsymbol{\mu}}_0$ and $\hat{\boldsymbol{\Sigma}}_0$ to derive the regression equation that fills in the missing data. The necessary estimates are

$$\hat{\beta}_1 = \frac{\hat{\sigma}_{IQ,JP}}{\hat{\sigma}^2_{IQ}} \tag{7.6}$$

$$\hat{\beta}_0 = \hat{\mu}_{JP} - \hat{\beta}_1 \hat{\mu}_{IQ} \tag{7.7}$$

$$\hat{\sigma}^2_{JP|IQ} = \hat{\sigma}^2_{JP} - \hat{\beta}^2_1 \hat{\sigma}^2_{IQ} \tag{7.8}$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are the intercept and slope coefficients, respectively, and $\hat{\sigma}^2_{JP|IQ}$ is the residual variance from the regression of job performance on IQ. The means, variances, and covariances that appear on the right side of the equations are elements from the mean vector and the covariance matrix.

To begin, substituting the appropriate elements of $\hat{\boldsymbol{\mu}}_0$ and $\hat{\boldsymbol{\Sigma}}_0$ into Equations 7.6 through 7.8 produces the following regression estimates: $\hat{\beta}_0 = -.057$, $\hat{\beta}_1 = 0.123$, and $\hat{\sigma}^2_{JP|IQ} = 5.320$. Next, substituting the regression coefficients and the observed IQ scores into the bracketed terms in Equation 7.1 generates predicted job performance ratings for the 10 incomplete cases. The predicted scores fall directly on a regression line, so adding normally distributed residual terms restores variability to the imputed data. I used Monte Carlo simulation methods to generate these residuals from a normal distribution with a mean of zero and a variance equal to 5.320 (the previous residual variance estimate), and I subsequently added these terms to each predicted job performance rating. Table 7.2 summarizes the imputation steps and shows the predicted scores, residual terms, and the imputed values. Again, each imputed value is a random draw from a distribution of plausible job performance ratings that is conditional on a particular IQ score.

The P-step is a standalone Bayesian analysis, the goal of which is to describe the posterior distributions of the mean vector and the covariance matrix. To begin, the P-step uses the

TABLE 7.2. Imputed Values from the Initial I-Step of the Bivariate Example

IQ	Job performance	Predicted score	Random residual	Imputed value
78	—	7.567	1.247	8.814
84	—	8.307	1.023	9.330
84	—	8.307	–1.586	6.721
85	—	8.430	1.285	9.716
87	—	8.677	–0.228	8.449
91	—	9.171	0.469	9.640
92	—	9.294	–3.663	5.631
94	—	9.541	–2.389	7.152
94	—	9.541	–0.329	9.212
96	—	9.787	–0.189	9.598
99	7	—	—	—
105	10	—	—	—
105	11	—	—	—
106	15	—	—	—
108	10	—	—	—
112	10	—	—	—
113	12	—	—	—
115	14	—	—	—
118	16	—	—	—
134	12	—	—	—

complete data set from the preceding I-step to estimate the mean vector and the covariance matrix. The data in Table 7.2 yield the following estimates.

$$\hat{\boldsymbol{\mu}}_1 = \begin{bmatrix} 100.000 \\ 10.063 \end{bmatrix}$$

$$\hat{\boldsymbol{\Sigma}}_1 = \begin{bmatrix} 199.579 & 25.081 \\ 25.081 & 7.270 \end{bmatrix}$$

Again, the numeric subscript denotes the fact that $\hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\Sigma}}_1$ are estimates from the first data augmentation cycle.

The ultimate goal of the P-step is to sample new estimates of the mean vector and the covariance matrix from their respective posterior distributions, so that the next I-step can use these updated parameter values to construct a different set of regression coefficients. The posterior distribution of the covariance matrix depends on the sample size, the sample means, and the sum of squares and cross products matrix, $\hat{\boldsymbol{\Lambda}}_1 = (N - 1)\hat{\boldsymbol{\Sigma}}_1$. Substituting $\hat{\boldsymbol{\Lambda}}_1$ into Equation 7.3 gives the following posterior distribution.

$$p(\boldsymbol{\Sigma} \,|\, \hat{\boldsymbol{\mu}}_1, \mathbf{Y}) \sim W^{-1}(N - 1, \hat{\boldsymbol{\Lambda}}_1)$$

Next, I used Monte Carlo simulation to draw a new covariance matrix from this posterior. Procedurally, this amounts to programming a computer to generate a matrix of random numbers from an inverse Wishart distribution with 19 degrees of freedom and a sum of squares and cross products matrix equal to $\hat{\mathbf{\Lambda}}_1$. Interested readers can consult Schafer (1997, p. 184) for specific programming instructions. Monte Carlo simulation generated the following covariance matrix.

$$\mathbf{\Sigma}_1^* = \begin{bmatrix} 488.873 & 36.663 \\ 36.663 & 7.493 \end{bmatrix}$$

Consistent with the previous section, the asterisk denotes the fact that the covariance matrix is a simulated estimate.

The sample means and the simulated covariance matrix define the posterior distribution of the mean vector, as follows:

$$p(\boldsymbol{\mu}|\mathbf{Y}, \mathbf{\Sigma}) \sim MN(\hat{\boldsymbol{\mu}}_1, N^{-1}\mathbf{\Sigma}_1^*)$$

To draw a new estimate of the mean vector from its posterior, I used Monte Carlo simulation to generate two data points from a multivariate normal distribution with a mean vector of $\hat{\boldsymbol{\mu}}_1$ and a covariance matrix equal to $N^{-1}\mathbf{\Sigma}_1^*$. This gave the following estimates.

$$\boldsymbol{\mu}_1^* = \begin{bmatrix} 87.929 \\ 8.162 \end{bmatrix}$$

Conceptually, using computer simulation procedures to generate $\boldsymbol{\mu}_1^*$ and $\mathbf{\Sigma}_1^*$ is akin to adding a random residual term to each element in $\hat{\boldsymbol{\mu}}_1$ and $\hat{\mathbf{\Sigma}}_1$. Regardless of how you think about it, this process yields new parameter values that randomly differ from the estimates that generated the regression coefficients at the initial I-step.

Having completed the first cycle, data augmentation returns to the I-step and uses the simulated parameter values to generate a new set of imputations. To illustrate, I estimated the regression parameters for the second I-step by substituting the appropriate elements of $\boldsymbol{\mu}_1^*$ and $\mathbf{\Sigma}_1^*$ into Equations 7.6 through 7.8. Doing so produced the following estimates: $\hat{\beta}_0 = 2.564$, $\hat{\beta}_1 = 0.075$, and $\hat{\sigma}^2_{JP|IQ} = 4.743$. Table 7.3 shows the predicted scores, residual terms, and imputed values from the second I-step. As before, the bracketed terms in Equation 7.1 generate the predicted job performance ratings for the 10 incomplete cases, and I augmented each predicted score with a random residual term from a normal distribution with a mean of zero and a variance equal to 4.743. The regression coefficients from the second I-step are randomly different from those at the previous I-step, so it follows that the imputations in Table 7.3 are different from those in Table 7.2.

The second P-step is procedurally identical to the first. As before, the P-step uses the filled-in data to estimate the mean vector and the covariance matrix. The data in Table 7.3 yield the following estimates.

$$\hat{\boldsymbol{\mu}}_2 = \begin{bmatrix} 100.000 \\ 10.767 \end{bmatrix}$$

$$\hat{\mathbf{\Sigma}}_2 = \begin{bmatrix} 199.579 & 18.624 \\ 18.624 & 5.818 \end{bmatrix}$$

TABLE 7.3. Imputed Values from the Second I-Step of the Bivariate Example

IQ	Job performance	Predicted score	Random residual	Imputed value
78	—	8.413	0.261	8.675
84	—	8.863	1.358	10.221
84	—	8.863	–1.576	7.287
85	—	8.938	1.914	10.852
87	—	9.088	–0.297	8.791
91	—	9.388	2.725	12.113
92	—	9.463	–0.510	8.953
94	—	9.613	3.000	12.613
94	—	9.613	–1.399	8.214
96	—	9.763	0.865	10.628
99	7	—	—	—
105	10	—	—	—
105	11	—	—	—
106	15	—	—	—
108	10	—	—	—
112	10	—	—	—
113	12	—	—	—
115	14	—	—	—
118	16	—	—	—
134	12	—	—	—

The sample size, the sample means, and the sum of squares and cross products matrix define the posterior distribution of the covariance matrix

$$p(\mathbf{\Sigma} \,|\, \hat{\boldsymbol{\mu}}_2, \mathbf{Y}) \sim W^{-1}(N-1, \hat{\mathbf{\Lambda}}_2)$$

and using Monte Carlo simulation to generate a random draw from this distribution produced the following estimates:

$$\mathbf{\Sigma}_2^* = \begin{bmatrix} 258.754 & 26.418 \\ 26.418 & 7.929 \end{bmatrix}$$

The sample means and the simulated covariance matrix define the posterior distribution of the mean vector

$$p(\boldsymbol{\mu} \,|\, \mathbf{Y}, \mathbf{\Sigma}) \sim MN(\hat{\boldsymbol{\mu}}_2, N^{-1}\mathbf{\Sigma}_2^*)$$

and I again used Monte Carlo procedures to draw a new pair of means from this distribution.

$$\boldsymbol{\mu}_2^* = \begin{bmatrix} 101.277 \\ 10.339 \end{bmatrix}$$

TABLE 7.4. Simulated Parameters from the First 20 P-Steps of the Bivariate Example

P-Step	μ^*_{IQ}	μ^*_{JP}	σ^{2*}_{IQ}	$\sigma^*_{JP,IQ}$	σ^{2*}_{JP}
1	87.929	8.162	488.873	36.663	7.493
2	101.277	10.339	258.754	26.418	7.929
3	105.008	11.088	234.612	35.607	9.631
4	104.608	11.414	186.003	31.542	10.205
5	100.621	11.080	311.717	38.136	9.161
6	99.774	9.929	191.862	19.771	6.655
7	95.161	9.959	316.123	34.109	7.641
8	106.298	11.451	308.825	26.468	10.873
9	99.470	9.862	218.068	18.509	10.136
10	102.117	11.976	349.522	27.239	13.159
11	99.774	10.797	221.643	–0.813	7.077
12	97.273	11.903	261.294	0.813	4.329
13	92.820	10.882	234.744	19.840	12.870
14	99.974	10.424	256.293	4.937	4.881
15	98.452	10.573	327.198	3.915	4.365
16	103.664	11.705	216.647	10.612	5.964
17	103.860	11.306	202.434	21.347	12.383
18	97.445	11.595	384.950	3.103	3.795
19	99.501	11.560	218.074	11.698	5.258
20	93.604	11.099	127.753	10.401	7.271

As you might have guessed, the next I-step constructs a new regression equation from $\boldsymbol{\mu}^*_2$ and $\boldsymbol{\Sigma}^*_2$ and uses this equation to generate another set of imputations. The subsequent P-step uses the parameter estimates from filled-in data (i.e., $\hat{\boldsymbol{\mu}}_3$ and $\hat{\boldsymbol{\Sigma}}_3$) to define the posterior distributions, from which it draws yet another set of plausible parameter values.

Data augmentation repeatedly cycles between the I-step and the P-step, often for several thousand iterations. Unlike maximum likelihood estimation, the algorithm generates parameter estimates that randomly vary across successive P-steps, so the elements in $\boldsymbol{\mu}^*$ and $\boldsymbol{\Sigma}^*$ never converge to a single value. For example, Table 7.4 shows the simulated parameters from the first 20 P-steps of the bivariate analysis. Notice that the estimates randomly bounce around from one cycle to the next and never land on a stationary value. This is true for every parameter, including those associated with the complete IQ variable (i.e., μ_{IQ} and σ^2_{IQ}). The random behavior of the parameter estimates across the P-steps leads to a very different definition of convergence and adds a layer of complexity that was not present with maximum likelihood estimation. I discuss the issue of convergence in considerable detail later in the chapter.

7.5 DATA AUGMENTATION WITH MULTIVARIATE DATA

The previous bivariate illustration is relatively straightforward because the missing values are isolated to a single variable. Applying data augmentation to multivariate data is typically more complex because each missing data pattern requires a unique regression equation (or set of equations). Despite this complication, the basic procedure is the same and only requires a slight modification to the I-step. To illustrate the changes to the I-step, I use the full data set in Table 7.1. Data augmentation with three variables is still relatively straightforward, but the logic of this example generalizes to data sets with any number of variables. Finally, note that the procedural details of the P-step are unaffected by the shift from bivariate to multivariate data, so there is no need for further discussion of this aspect of the procedure.

Not including the complete cases, there are three missing data patterns in Table 7.1: cases that are missing (1) job performance ratings only, (2) well-being scores only, and (3) both job performance and well-being scores. The presence of multiple missing data patterns complicates the imputation process somewhat because each missing data pattern requires a unique regression equation. To illustrate, Table 7.5 shows the regression equations for the three missing data patterns. Consistent with the bivariate example, the I-step uses the mean vector and the covariance matrix from the preceding P-step to estimate the regression coefficients and the corresponding residual variances. After constructing the regression equations, the algorithm generates predicted values by substituting the observed data into the relevant regression equation, and it augments each predicted score with a normally distributed residual term. Each regression equation now requires its own residual distribution, but the basic idea is the same as before. Finally, whenever two or more variables are missing, the residual distribution is multivariate normal with a mean vector of zero and a covariance matrix equal to the residual covariance matrix from the multivariate regression of the incomplete variables on the complete variables. For example, the third missing data pattern (i.e., the subset of cases with missing job performance and well-being scores) requires residuals from a multivariate normal distribution with a covariance matrix equal to the residual covariance matrix from the multivariate regression of job performance and well-being on IQ.

Estimating unique regression equations for each missing data pattern is the only procedural change associated with multivariate data. The number of missing data patterns can often be quite large, but a computational algorithm called the sweep operator simplifies the imputation process. The sweep operator repeatedly applies a series of transformations to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and yields new matrices that contain the desired regression coefficients and residual variances. A number of detailed descriptions of the sweep operator are available to readers who are interested in additional details (e.g., Dempster, 1969; Goodnight, 1979; Little & Rubin, 2002). The changes to the I-step have no bearing on the P-step, and the process of simulating new parameter values is identical to the earlier bivariate example.

TABLE 7.5. I-Step Regression Equations for a Multivariate Analysis

Missing variables	Regression equation	Residual distribution
Job performance	$JP_i^* = \hat{\beta}_0 + \hat{\beta}_1(IQ_i) + \hat{\beta}_2(WB_i) + z_i$	$z_i \sim N(0, \hat{\sigma}^2_{JP\|IQ,WB})$
Well-being	$WB_i^* = \hat{\beta}_0 + \hat{\beta}_1(IQ_i) + \hat{\beta}_2(JP_i) + z_i$	$z_i \sim N(0, \hat{\sigma}^2_{WB\|IQ,JP})$
Job performance and well-being	$JP_i^* = \hat{\beta}_0 + \hat{\beta}_1(IQ_i) + z_i$ $WB_i^* = \hat{\beta}_0 + \hat{\beta}_1(IQ_i) + z_i$	$\mathbf{Z}_i \sim MN(0, \hat{\boldsymbol{\Sigma}}_{JP,WB\|IQ})$

7.6 SELECTING VARIABLES FOR IMPUTATION

Deciding which variables to include in the imputation phase is an important aspect of a multiple imputation analysis. At a minimum, the imputation process should include any variable that you intend to use in a subsequent statistical analysis. Excluding an analysis model variable will attenuate its associations with other variables, even if the data are MCAR or MAR. This underscores the importance of variable selection because an inadequate imputation model can introduce biases that would not occur in a maximum likelihood analysis. Fortunately, including too many variables in the imputation process is unlikely to produce bias, so adopting a liberal approach to variable selection is usually a good strategy. The primary downside of including too many variables is the possibility of convergence problems (as an upper limit, the number of variables cannot exceed the number of cases).

In addition to including analysis model variables, the imputation phase should preserve any higher-order effects that are of interest in the analysis phase as well as any other special features of the data. In particular, researchers in the behavioral and the social sciences are often interested in estimating interaction (i.e., moderation) effects where the magnitude of the association between two variables depends on a third variable (e.g., a regression model where gender moderates the association between psychological well-being and job performance). In addition, many common statistical analyses address implicit interaction effects. For example, multiple group structural equation models and multilevel models do not necessarily contain interaction terms, but they do posit group differences in the mean structure, the covariance structure, or both. Regardless of whether the higher-order effect is an explicit part of the statistical analysis or a hidden feature of the data, it is necessary to specify an imputation model that preserves any complex associations among the variables. Again, failing to do so can bias the subsequent analysis results, regardless of the missing data mechanism. I address this topic in detail in Chapter 9, but for now, it is important to raise awareness of the issue.

Chapter 5 introduced the idea of an inclusive analysis strategy that incorporates a number of auxiliary variables into the missing data handling procedure (Collins, Schafer, & Kam, 2001). Recall that an auxiliary variable is one that is ancillary to the substantive research questions but is a potential correlate of missingness or a correlate of an incomplete analysis model variable. Methodologists have long recommended the use of auxiliary variables in a multiple-imputation analysis. For example, Rubin (1996, p. 479) stated that "the advice has always been to include as many variables as possible when doing multiple imputation." Using auxiliary variables in a multiple imputation analysis is particularly straightforward because the variables only play a role in the imputation phase. Including auxiliary variables in the imputation process infuses the filled-in values with the auxiliary information, so there is no need to include the extra variables in the subsequent analysis phase. This is in contrast to maximum likelihood estimation, which incorporates auxiliary variables via the slightly awkward saturated correlates approach. As an aside, multiple imputation can generally handle a larger set of auxiliary variables than a maximum likelihood analysis, so there is usually no reason to limit the number of auxiliary variables. Chapter 5 describes the process of identifying auxiliary variables, so that information need not be reiterated here.

Finally, although it is important to include all analysis variables in the imputation phase, it makes no difference whether a particular variable will ultimately serve as an explanatory variable or an outcome variable. For example, Chapter 8 illustrates a multiple regression analysis in which psychological well-being and job satisfaction predict job performance. Both predictor variables have missing data, but the imputation model uses the observed job performance scores to impute the missing values. At first glance, using an outcome variable to impute an incomplete independent variable may seem incorrect and somewhat circular. However, the addition of a random residual term to each imputed value eliminates any bias that might result from doing so (Little & Rubin, 2002). In fact, multiple imputation programs make no distinction between independent and dependent variables and only require you to specify a set of input variables.

7.7 THE MEANING OF CONVERGENCE

The data augmentation algorithm belongs to a family of **Markov Chain Monte Carlo** (i.e., MCMC) procedures (Jackman, 2000). The goal of a Markov chain Monte Carlo algorithm is to simulate random draws from a distribution (e.g., random draws from the posterior distribution or from the distribution of missing values). Repeatedly cycling between the I- and P-steps creates a so-called data augmentation chain, as follows:

$$Y_1^*, \theta_1^*, Y_2^*, \theta_2^*, Y_3^*, \theta_3^*, Y_4^*, \theta_4^*, \ldots, Y_t^*, \theta_t^*$$

where Y_t^* represents the imputed values at I-step t and θ_t^* contains the simulated parameter values at P-step t. Over the course of a long enough chain, the I-step generates imputations from a large array of plausible parameter values, so the Y_t^* values are effectively drawn from a distribution that averages over the entire range of the posterior distribution. Similarly, the P-step generates parameters from a large number of plausible Y_t^* values, so the simulated parameters form a posterior distribution that averages over all possible values of the missing data.

Simulating random draws from a distribution requires a new definition of convergence. Whereas maximum likelihood converges when the parameter estimates no longer change across successive iterations, data augmentation converges when the *distributions* become stable and no longer change in a systematic fashion (i.e., the distributions become **stationary**). The complicated aspect of this definition is that each step in the data augmentation chain is dependent on the previous step. That is, the simulated parameters at P-step t depend on the imputed values at the preceding I-step, the imputations at I-step $t + 1$ depend on the simulated parameters from P-step t, the simulated parameters at P-step $t + 1$ depend on the imputed values at I-step $t + 1$, and so on. Although the behavior of the data augmentation algorithm is seemingly random from one cycle to the next, the mutual dependence of the I- and P-steps induces a correlation between the simulated parameters from successive P-steps. By extension, analyzing data sets from successive I-steps is inappropriate because the resulting imputations are also dependent (i.e., imputations from adjacent I-steps do not originate from a stable distribution).

Researchers often assess convergence by determining the number of data augmentation cycles that need to lapse before the imputations at iteration $t + k$ are independent of those at iteration t. Monitoring the behavior of the simulated parameter values across a large number of P-steps is one way to do this. For example, suppose that 10 data augmentation cycles separate two sets of simulated parameter values, θ_t^* and θ_{t+10}^*. A correlation between θ_t^* and θ_{t+10}^* suggests that the posterior distribution is systematically changing after 10 cycles. Consequently, analyzing data sets that are separated by only 10 data augmentation cycles is inappropriate because the imputed values are also dependent. In contrast, suppose that θ_t^* is uncorrelated with the simulated parameters from 50 cycles later in the chain. The lack of correlation suggests that θ_t^* and θ_{t+50}^* originate from a stable posterior distribution, so the two sets of parameter values should produce independent imputations. From a practical perspective, this implies that at least 50 data augmentation cycles need to separate the data sets that you analyze in the subsequent analysis phase.

7.8 CONVERGENCE DIAGNOSTICS

Methodologists have proposed dozens of techniques for assessing the convergence of data augmentation, the majority of which are computationally complex and difficult to implement (e.g., Gelman & Rubin, 1992; Geweke, 1992; Geyer, 1992; Johnson, 1996; Mykland, Tierney, & Yu, 1995; Ritter & Tanner, 1992; Roberts, 1992; Zellner & Min, 1995). A comprehensive review of convergence diagnostics is beyond the scope of this chapter, but interested readers can consult Cowles and Carlin (1996) for an overview of some of these procedures. I focus primarily on the use of graphical displays (time-series plots and autocorrelation function plots) because these methods are readily available in multiple imputation software packages. Graphical techniques are certainly not foolproof, but they are straightforward to implement and are relatively easy to understand.

Assessing convergence requires an exploratory data augmentation chain. The purpose of the exploratory analysis is to gather the simulated parameter values from a large number of P-steps and use graphical displays to examine their behavior (the literature sometimes refers to this as an **output analysis**). Establishing guidelines for the length of the exploratory chain is difficult because a number of factors influence convergence speed (e.g., the missing data rate, the choice of starting values for the mean vector and the covariance matrix). Running the data augmentation algorithm for several thousand cycles is probably sufficient in most situations, but data sets with a large proportion of missing values may require longer chains. In this section, I use the small data set in Table 7.1 to illustrate graphical diagnostic techniques. I generated an exploratory data augmentation chain of 5,000 cycles and saved the simulated parameters from each P-step to a file for further analysis. As you will see, the information from this exploratory analysis is important for planning the final data augmentation chain that generates the imputed data sets.

What Does EM Tell You about Convergence?

Using the EM algorithm (an algorithm that generates maximum likelihood estimates of the mean vector and the covariance matrix; see Chapter 4) to estimate the mean vector and the covariance matrix is a useful precursor to a multiple imputation analysis. EM estimates make good starting values for data augmentation because they tend to be representative of the posterior distribution. Consequently, data augmentation will generally converge more rapidly from a set of EM starting values. In addition, the number of EM iterations is a useful diagnostic for assessing convergence. Schafer and colleagues (Schafer, 1997; Schafer & Olsen, 1998) suggest that EM converges more slowly than data augmentation, so researchers often estimate convergence speed by doubling the number of EM iterations. This approach is far from ideal, and blindly relying on the "two times the number of EM iterations" rule of thumb is not a good way to assess convergence. Nevertheless, the EM algorithm is a good starting point. Returning to the data in Table 7.1, note that the EM algorithm converged in 60 iterations, so data augmentation may require even fewer cycles to converge. However, doubling the number of EM iterations can provide a more conservative initial guess about convergence speed.

7.9 TIME-SERIES PLOTS

A **time-series plot** displays the simulated parameter values from the P-step on the vertical axis and the data augmentation cycles along the horizontal axis. To illustrate, consider the exploratory data augmentation chain that I generated from the small job performance data set. Figure 7.3 shows time-series plots for the job performance and the psychological well-being means. I arbitrarily chose to plot the parameter values from the first 200 data augmentation cycles, but did so after inspecting the plots over the entire chain. The top panel of Figure 7.3 suggests that the well-being means bounce around in a seemingly random fashion with no discernible long-term trends. The absence of trend is an ideal situation and suggests that this parameter quickly converges to a stable distribution. In contrast, the bottom panel of the figure shows a time-series plot that is somewhat less ideal (though not bad). Specifically, notice that the job performance means exhibit systematic upward and downward trends that last for 40 iterations or more. These systematic trends suggest that this parameter's posterior distribution requires at least 40 P-steps to converge (i.e., 40 data augmentation cycles need to lapse before the simulated parameters become independent). I examined the time-series plots for all of the means and covariance matrix elements, and they were largely consistent with those in Figure 7.3.

Figure 7.3 emphasizes that the simulated parameters can converge at different rates. For example, the job performance mean systematically wandered up and down while the well-being mean settled into a random pattern almost immediately. Perhaps not surprisingly, the missing data rate—or more accurately, the fraction of missing information—is responsible for these differences. The **fraction of missing information** quantifies the proportion of a parameter's sampling error that is due to missing data. I describe this concept in more detail in Chapter 8, but you can think of missing information as a measure that combines the missing data rate and the magnitude of the correlations among the variables. For example, the fraction of missing information and the proportion of missing data are roughly equal when variables are uncorrelated, but the missing information is typically less than the missing data rate when variables are correlated because the shared variability among the variables mitigates the loss of information.

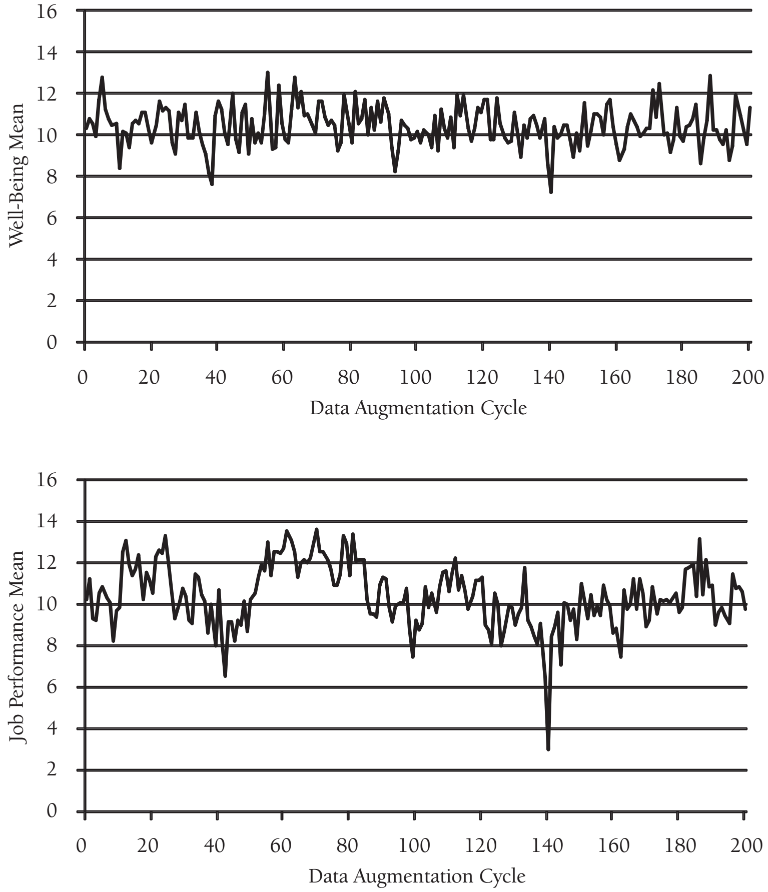

FIGURE 7.3. Time-series plots for the simulated well-being and job performance means. The top panel shows a time-series plot that exhibits no systematic trends. The bottom panel shows systematic trends that last for 40 iterations or more.

Because the fraction of missing information tends to vary across the elements in the mean vector and the covariance matrix, you should attempt to examine time-series plots for every parameter that is affected by missing data. Paying particularly close attention to parameters with high rates of missing information (i.e., high missing data rates) is a good idea because these parameters tend to converge most slowly. Multivariate data sets often have a prohibitively large number of covariance matrix elements, so the fraction of missing information can serve as a screening device for identifying the most important time-series plots (multiple imputation programs typically report these values). As shown in the next chapter, the fraction of missing information influences the magnitude of the multiple-imputation standard errors, so inspecting these values is often useful in and of itself.

Worst Linear Function

In addition to inspecting the behavior of individual parameters, it is useful to examine a summary measure that Schafer (1997) terms the worst linear function of the parameters. The

worst linear function combines the simulated parameters from each P-step into a single composite that weights each parameter according to its convergence speed. The idea behind the worst linear function is to create a summary measure that converges more slowly than the individual parameters, so the time series plot of the worst linear function should provide a conservative gauge of convergence speed. However, Schafer (1997) cautions that the worst linear function is not a definitive diagnostic tool, because other combinations of the simulated parameters may converge at an even slower rate.

The worst linear function is a weighted sum of the simulated parameters at P-step *t*

$$\text{WLF}_t = \boldsymbol{\upsilon}^T \boldsymbol{\theta}_t^* \tag{7.9}$$

where $\boldsymbol{\theta}_t^*$ is a column vector that contains the simulated parameter values and $\boldsymbol{\upsilon}$ is a weight vector that quantifies the change in the corresponding maximum likelihood estimates at the final EM iteration. Parameters that converge slowly exhibit the greatest change at the final iteration, so $\boldsymbol{\upsilon}$ assigns larger weights to parameters that converge slowly. The complete-data parameters do not change at the final EM iteration, so they do not contribute to the function (the weights are zero for these parameters). Finally, note that the worst linear function can take on positive or negative values because it centers the parameters in $\boldsymbol{\theta}_t^*$ at their maximum likelihood estimates.

With regard to the exploratory data chain from the small job performance data set, Figure 7.4 shows the time-series plot of the worst linear function. Notice that the function exhibits systematic upward and downward trends that last for approximately 50 iterations. Taken together, Figures 7.3 and 7.4 suggest that the joint posterior distribution is stable (i.e., the simulated parameter values are no longer dependent) after about 50 data augmentation cycles, although certain parameters (e.g., the well-being mean) converge far more rapidly. From a practical perspective, this implies that at least 50 data augmentation cycles need to separate the data sets that you analyze in the subsequent analysis phase. Doubling or tripling this number provides an extra margin of safety.

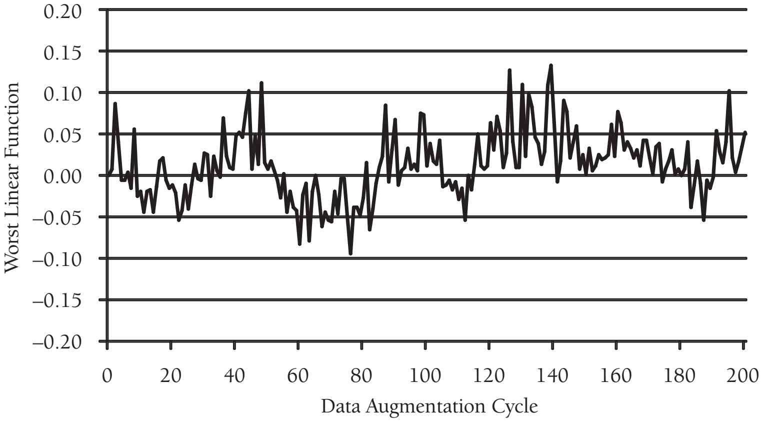

FIGURE 7.4. Time-series plot for the worst linear function of the parameters. The worst linear function shows systematic trends that last for approximately 60 iterations.

7.10 AUTOCORRELATION FUNCTION PLOTS

The systematic trends in the previous time-series plots suggest that certain parameters are serially dependent across successive data augmentation cycles. The **autocorrelation** quantifies the magnitude and duration of this dependency and is an important diagnostic tool for assessing convergence. The lag-*k* autocorrelation is the Pearson correlation between sets of parameter values separated by *k* iterations in the data augmentation chain. To illustrate, reconsider the exploratory chain of 5,000 data augmentation cycles that I generated from the data in Table 7.1. The Lag-1 columns of Table 7.6 show the simulated job performance means from P-steps 1 through 10 and 4,991 through 5,000. Notice that the one row (i.e., one data augmentation cycle) offsets the parameter values, such that the mean at P-step 2 is coupled to the mean at P-step 1, the mean at P-step 3 is linked to the mean at P-step 2, and so on. Computing the Pearson correlation between the 4,999 pairs of parameter values gives the lag-1 autocorrelation, $r_1 = 0.61$. This correlation indicates that the job performance mean at P-step *t* is highly dependent on the mean at the preceding iteration. Computing additional lag-*k* correlations can help determine the duration of this dependency. For example, Table 7.6 also shows data excerpts for the lag-2 and the lag-3 autocorrelations. The lag-2 autocorrelation quantifies the dependency between estimates separated by two iterations (e.g., the mean from P-step 3 is linked to the estimate from P-step 1, the mean at P-step 4 is coupled with the mean from P-step 2, and so on), and the lag-3 autocorrelation separates the simulated

TABLE 7.6. Data for the Lag–1, Lag–2, and Lag–3 Autocorrelations

Simulated values		Parameter values for autocorrelation computations					
P-step	μ^*_{JP}	Lag-1		Lag-2		Lag-3	
1	8.16	8.16	—	8.16	—	8.16	—
2	10.34	10.34	8.16	10.34	—	10.34	—
3	11.09	11.09	10.34	11.09	8.16	11.09	—
4	11.41	11.41	11.09	11.41	10.34	11.41	8.16
5	11.08	11.08	11.41	11.08	11.09	11.08	10.34
6	9.93	9.93	11.08	9.93	11.41	9.93	11.09
7	9.96	9.96	9.93	9.96	11.08	9.96	11.41
8	11.45	11.45	9.96	11.45	9.93	11.45	11.08
9	9.86	9.86	11.45	9.86	9.96	9.86	9.93
10	11.98	11.98	9.86	11.98	11.45	11.98	9.96
…	…	…	…	…	…	…	…
4991	10.66	10.66	11.29	10.66	10.88	10.66	9.53
4992	11.11	11.11	10.66	11.11	11.29	11.11	10.88
4993	12.13	12.13	11.11	12.13	10.66	12.13	11.29
4994	10.54	10.54	12.13	10.54	11.11	10.54	10.66
4995	11.22	11.22	10.54	11.22	12.13	11.22	11.11
4996	10.63	10.63	11.22	10.63	10.54	10.63	12.13
4997	9.94	9.94	10.63	9.94	11.22	9.94	10.54
4998	12.17	12.17	9.94	12.17	10.63	12.17	11.22
4999	11.79	11.79	12.17	11.79	9.94	11.79	10.63
5000	11.34	11.34	11.79	11.34	12.17	11.34	9.94

parameter values by three iterations. The estimates of the lag-2 and lag-3 correlations are $r_2 = 0.52$ and $r_3 = 0.46$, respectively.

An **autocorrelation function plot** (also known as a **correlogram**) is a graphical summary that displays the autocorrelation values on the vertical axis and the lag values on the horizontal axis. For example, Figure 7.5 shows the autocorrelation function plots for the job performance and the well-being means. The horizontal dashed lines represent the two-tailed critical values for an alpha level of 0.05 (Bartlett, 1946). The top panel of Figure 7.5 shows that autocorrelation in the well-being means drops to within sampling error of zero almost immediately. This suggests that the parameter's distribution becomes stable after a very small number of data augmentation cycles. In contrast, the bottom panel of the figure shows autocorrelations that exceed chance levels (i.e., fall outside the critical values) for nearly 60 data augmentation cycles. This suggests that the posterior distribution of the job performance

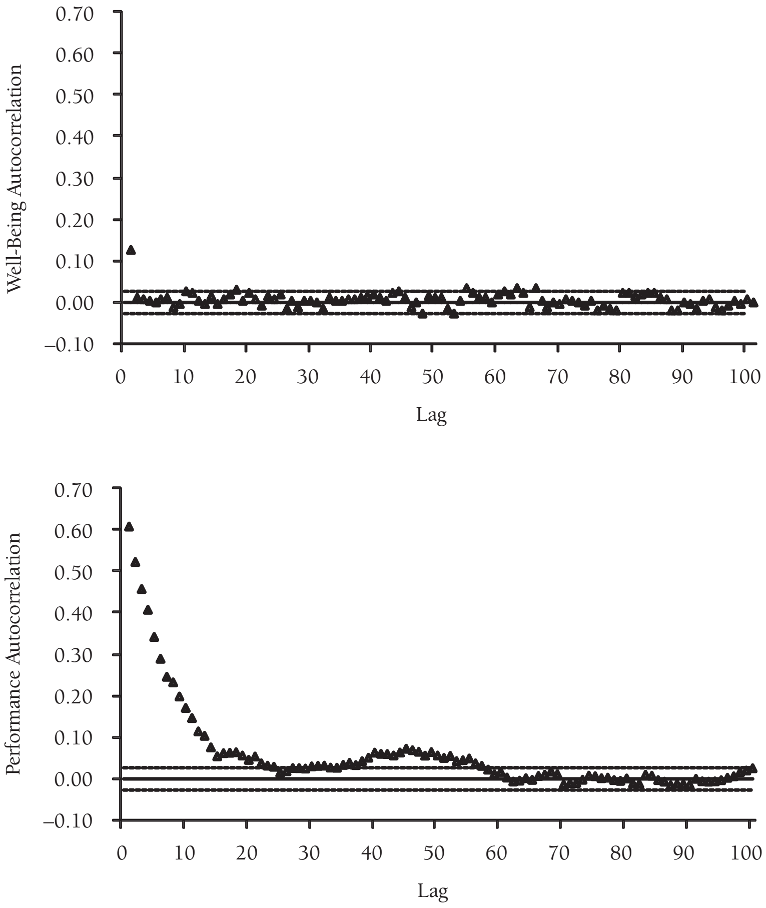

FIGURE 7.5. Autocorrelation function plots (correlograms) for the simulated well-being and job performance means. The top plot shows autocorrelations (denoted by a triangle symbol) that drop to within sampling error of zero almost immediately. The bottom plot shows nonzero autocorrelations that persist for nearly 60 iterations.

mean requires approximately 60 data augmentation cycles to become stationary. As an aside, autocorrelations are subject to considerable sampling fluctuation, so data augmentation chains that are several thousand cycles in length will provide the best assessment of serial dependencies.

Figures 7.3 and 7.5 are largely consistent with one another. For example, the time-series plot indicates that the job performance mean has systematic trends lasting for at least 40 data augmentation cycles, and the corresponding autocorrelation plot indicates serial dependencies that last for approximately 60 iterations. In contrast, both plots suggest that distribution of the well-being mean stabilizes almost immediately. Taken together, the diagnostic information suggests that the slowest parameters are stationary (i.e., become independent) after about 60 iterations, although some distributions are stable well before that. Again, multiplying this value by a factor of two or three is a conservative strategy for planning the final data augmentation run.

7.11 ASSESSING CONVERGENCE FROM ALTERNATE STARTING VALUES

Thus far, I have only considered using EM estimates as starting values for data augmentation. EM estimates are ideal in the sense that they are often located near the center of the posterior distribution. However, methodologists disagree on whether a single set of starting values is sufficient for assessing convergence (Gelman & Rubin, 1992; Geyer, 1992; Raftery & Lewis, 1992). For example, Raftery and Lewis (1992) argue that a single exploratory data augmentation chain is usually sufficient, whereas Gelman and Rubin (1992) recommend using multiple exploratory data augmentation chains, each of which uses starting values for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that are far from the center of their respective posterior distributions.

Multiple exploratory chains are useful for assessing whether idiosyncratic features of the data influence convergence and can yield a more conservative gauge of convergence speed. However, generating alternate starting values can be computationally complex and difficult to implement (Gelman & Rubin, 1992). One straightforward approach is to use the bootstrap to generate starting values for each exploratory data augmentation chain (Schafer, 1997). The bootstrap treats the data as a miniature population from which it draws samples of size *N* with replacement (see Chapter 5 for additional information on the bootstrap). The bootstrap procedure can generate a small number of alternate estimates of the mean vector and the covariance matrix. Because the ultimate goal is to start data augmentation with parameter values that are far from the center of their posterior distributions, the bootstrap estimates should be somewhat noisy and unrepresentative of their true values. To accomplish this, Schafer (1997) recommends drawing bootstrap samples with half as many cases as the original data set because the additional sampling error is likely to yield estimates from the tails of the posterior distribution. After generating a small number of alternative starting values, you can run multiple exploratory data augmentation chains and use graphical diagnostic techniques to examine the convergence of each chain. Some multiple imputation programs generate bootstrap starting values, so implementing this approach is relatively straightforward.

7.12 CONVERGENCE PROBLEMS

You may occasionally encounter situations in which data augmentation fails to converge. For example, Figure 7.6 shows what the time-series and autocorrelation function plots would look like when data augmentation fails to converge. The times-series plot indicates the presence of systematic trends lasting for several hundred iterations, and the autocorrelation function plot shows serial dependencies that persist for an extended period (e.g., the lag-200 correlation is approximately $r_{200} = 0.70$).

Convergence problems can occur because some of the parameters are inestimable or because the number of variables is close to the number of cases. Eliminating the problematic variables is one way to solve convergence problems, but this solution may not be ideal, particularly if it alters the substantive research goals. An alternate strategy is to use a so-called

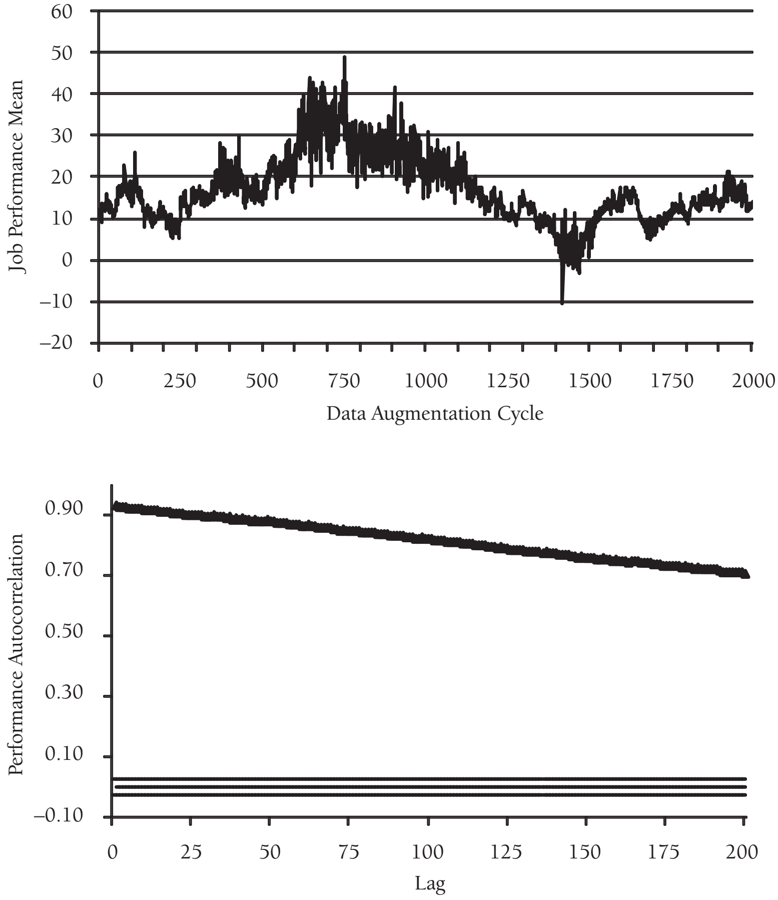

FIGURE 7.6. Time-series and autocorrelation function plot for parameters that do not converge. The top panel shows a time-series plot that exhibits systematic trends that last for hundreds of iterations and simulated parameter values that are outside of the plausible score range of 1 to 20. The bottom panel shows autocorrelations (denoted by a triangle symbol) that are close to $r = 0.70$ at lag-200.

ridge prior distribution for the covariance matrix. The basic idea behind the ridge prior is to add a small number of imaginary data records from a hypothetical population where the variables are uncorrelated. Adding these extra cases can stabilize estimation and eliminate convergence problems. Chapter 9 describes the ridge prior in more detail.

7.13 GENERATING THE FINAL SET OF IMPUTATIONS

After assessing convergence, you can begin planning the data augmentation run that will generate the imputed data sets for the subsequent analysis phase. As I explained previously, an important objective of the imputation phase is to generate data sets that mimic independent draws from the distribution of the missing values. There are two strategies for generating independent imputations: sample imputed data sets at regular intervals in the data augmentation chain (e.g., save and analyze the imputed data set from every 200$^{\text{th}}$ I-step), or generate several data augmentation chains and save the imputed data at the final I-step in each chain. The multiple imputation literature refers to these two approaches as sequential and parallel data augmentation chains, respectively.

Sequential Data Augmentation Chains

One way to generate independent imputations is to sample imputed data sets at regular intervals in a single data augmentation chain (e.g., save and analyze the data from every 200th I-step). The literature sometimes refers to this approach as **sequential data augmentation**. The difficulty with sequential data augmentation is determining the number of iterations that need to lapse between each saved file (i.e., the number of **between-imputation iterations**). Choosing too large an interval is not a problem, but specifying too few between-imputation iterations can result in correlated imputations and negatively biased standard errors. Fortunately, the time-series and autocorrelation function plots provide the necessary information to specify the number of between-imputation iterations. For example, if the longest serial dependency lasts for 20 data augmentation cycles, then the between-imputation interval should be at least 20 iterations. Again, the graphical diagnostics are far from perfect, so doubling or tripling that value is probably a safe strategy.

To illustrate sequential data augmentation, reconsider the data in Table 7.1. Suppose that the goal is to generate $m = 20$ complete data sets for the subsequent analysis phase. The graphical diagnostics from the earlier example suggest that the slowest parameters converged (i.e., became independent) after about 60 iterations, so between-imputation interval should be at least 60 cycles, if not longer. Specifying 200 between-imputations is probably sufficient because this interval is more than three times larger than the slowest convergence rate. Consequently, the final data augmentation chain consists of 4,000 cycles. Specifically, an initial **burn-in period** of 200 cycles precedes the first data set, and 200 between-imputation cycles separate each of the remaining data sets. The burn-in iterations give the parameter distributions time to stabilize, and the between-imputation iterations ensure that the resulting imputations are independent.

Parallel Data Augmentation Chains

Parallel data augmentation is a second method for generating independent imputations. Rather than saving data sets at specified intervals in the chain, this approach generates several chains and saves the imputed data at the final I-step in each chain. For example, generating 20 imputations from the data in Table 7.1 would require 20 separate data augmentation chains, each of which is comprised of 200 iterations. The *m* chains can originate from a common set of starting values or from different estimates of the mean vector and the covariance matrix. The primary consideration is to generate chains that are long enough to ensure that the distribution of missing values has stabilized and that the imputations are independent of the starting values. As with sequential approach, graphical diagnostics can determine the length of the data augmentation chains.

Methodologists have debated on whether to use sequential or parallel data augmentation chains. Much of this discussion centers on the detection of convergence problems (e.g., Gelman & Rubin, 1992; Geyer, 1992; Raftery & Lewis, 1992), but computational efficiency is also a consideration (e.g., Schafer, 1997, pp. 137–138; Smith & Roberts, 1993). If the parameter distributions converge properly, it probably makes little difference whether a single chain or multiple chains generate the final imputations. Because sequential chains are somewhat easier to implement in existing software packages, the final decision may be one of convenience. My advice is to explore convergence using a relatively small number of parallel chains that originate from a diverse set of starting values. If you are comfortable that the algorithm is converging properly, choose a conservative number of burn-in and between-imputation iterations and generate the final set of imputations from a single data augmentation chain.

7.14 HOW MANY DATA SETS ARE NEEDED?

Choosing the number of imputed data sets to save and analyze is one of the most basic decisions in a multiple imputation analysis. Conventional wisdom suggests that multiple imputation analyses require relatively few imputations, and the literature historically recommends between three and five imputed data sets (e.g., Rubin, 1987, 1996; Schafer, 1997; Schafer & Olsen, 1998). However, there are good reasons to use many more imputations. In the next chapter, I show that multiple imputation standard errors decrease as the number of imputations increases, and analyzing an infinite number of imputed data sets yields the lowest possible standard error. Obviously, it is not feasible to analyze an infinite number of data sets, but this property suggests that using a large number of imputations can improve power. Power issues aside, some of the multiparameter significance tests outlined in Chapter 8 become more accurate as *m* increases, so analyzing a large number of data sets can improve the validity of these tests.

Relative Efficiency

The recommendation to use between three and five data sets follows from the fact that the resulting standard errors are not appreciably larger than their hypothetical minimum values. **Relative efficiency** quantifies the magnitude of a multiple imputation standard error (or more precisely the sampling variance, or squared standard error) relative to its theoretical minimum

$$\mathrm{RE} = \left(1 + \frac{\mathrm{FMI}}{m}\right)^{-1} \tag{7.10}$$

where m is the number of imputed data sets and FMI is the fraction of missing information (Rubin, 1987). I describe the fraction of missing information in Chapter 8, but for now, you can think of it as being roughly equal to the proportion of missing data. To illustrate, suppose that $m = 5$ and the fraction of missing information for a particular parameter is 0.20 (e.g., there is a 20% missing data rate). Equation 7.10 suggests that the sampling variance (i.e., squared standard error) based on an infinite number of imputations is 96% as large as the sampling variance based on only $m = 5$ imputations. From a practical standpoint, this means that analyzing five imputed data sets should produce a standard error that is only $\sqrt{1 + (0.20/5)} = 1.02$ times larger than its hypothetical minimum value.

Table 7.7 shows the relative efficiency and proportional increase in the standard error for different fractions of missing information and different numbers of imputations. The table shows two noticeable trends. First, the largest gains in efficiency (or alternatively, largest reductions in the standard error) occur between 3 and 10 imputations, and using more than 10 data sets has little additional benefit. Second, using a large number of imputations is most beneficial when the fraction of missing information is large. Researchers have traditionally relied on relative efficiency estimates such as those in Table 7.6 when choosing the number of imputations. Doing so has led to the common recommendation to analyze between three and five imputed data sets. Interestingly, this common rule of thumb does not necessarily maximize power.

The Number of Imputations and Power

Graham, Olchowski, and Gilreath (2007) used computer simulation studies to show that the number of imputations has a more dramatic impact on power than it does on relative

TABLE 7.7. Relative Efficiency and Proportional Increase in Standard Error for Different Fractions of Missing Information and Numbers of Imputations

	$m = 3$		$m = 5$		$m = 10$		$m = 20$	
FMI	R.E.	P.S.E.	R.E.	P.S.E.	R.E.	P.S.E.	R.E.	P.S.E.
0.10	0.97	1.02	0.98	1.01	0.99	1.00	1.00	1.00
0.20	0.94	1.03	0.96	1.02	0.98	1.01	0.99	1.00
0.30	0.91	1.05	0.94	1.03	0.97	1.01	0.99	1.01
0.40	0.88	1.06	0.93	1.04	0.96	1.02	0.98	1.01
0.50	0.86	1.08	0.91	1.05	0.95	1.02	0.98	1.01
0.60	0.83	1.10	0.89	1.06	0.94	1.03	0.97	1.01
0.70	0.81	1.11	0.88	1.07	0.93	1.03	0.97	1.02

Note. R.E. = relative efficiency; P.S.E. = proportional increase in standard error; m = number of imputations; FMI = fraction of missing information.

efficiency. For example, returning to Table 7.7, the combination of $m = 5$ and FMI = .50 yields a relative efficiency value of .91. In contrast, Graham et al. show that the power for this set of conditions is 13% below its ideal value. Decreasing the number of imputations to $m = 3$ reduces relative efficiency to .86, but reduces power to 75% of its optimal level.

Contrary to conventional wisdom, the Graham et al. study indicates that using more than 10 imputations has a beneficial impact on statistical power. Considered as a whole, their simulations suggest that 20 imputations are sufficient for many realistic situations, and increasing the number of imputations beyond 20 will only affect power if the fraction of missing information is very high (e.g., FMI > 0.50). The Graham et al. study also shows that an analysis based on 20 imputations yields comparable power to a maximum likelihood analysis, so generating a *minimum* of 20 imputed data sets seems to be a good rule of thumb for many situations.

Other Considerations

Power issues aside, there are other good reasons to use a large number of imputations. As I mentioned previously, analyzing a large number of data sets can improve the validity of the multiparameter significance tests in the next chapter. In addition, the estimates of missing information that most imputation programs report can be very noisy when the number of imputations is small (Graham et al., 2007; Harel, 2007; Schafer, 1997), and stable estimates require between 50 and 100 imputations (Harel, 2007). Obtaining accurate estimates of the missing information is usually not an important analytic goal, but these estimates are useful for assessing the impact of missing data on standard errors. Taken as a whole, there are many issues to consider when deciding on the number of imputed data sets to save and analyze. Although $m = 20$ appears to be a good rule of thumb, increasing the number of imputations beyond this point is a good idea and often adds very little to the total processing time.

7.15 SUMMARY

Multiple imputation is an alternative to maximum likelihood estimation and is the other state-of-the-art missing data technique that methodologists currently recommend. The imputation approach outlined in this chapter makes the same assumptions as maximum likelihood estimation—MAR data and multivariate normality—but takes the very different tack of filling in the missing values prior to the analysis. A multiple imputation analysis consists of three distinct steps: the imputation phase, the analysis phase, and the pooling phase. The imputation phase creates multiple copies of the data set (e.g., $m = 20$), each of which contains different estimates of the missing values. The purpose of the analysis phase is to analyze the filled-in data sets. This step applies the same statistical procedures that you would have used had the data been complete. Procedurally, the only difference is that you perform each analysis m times, once for each imputed data set. Finally, the pooling phase uses Rubin's (1987) rules to combine the m sets of parameter estimates and standard errors into a single set of results. Because of its complexity, the imputation phase was the primary focus of this chapter.

The imputation phase uses an iterative data augmentation algorithm that consists of an I-step and a P-step. The I-step uses an estimate of the mean vector and the covariance matrix to build a set of regression equations where the complete variables for a given missing data pattern predict the incomplete variables for that pattern. Substituting the observed data into these equations generates predicted scores for the missing variables. The predicted scores fall directly on a regression surface, so the imputation procedure restores variability to the data by adding a normally distributed residual term to each predicted value. From a Bayesian perspective, each imputed value is a random draw from the conditional distribution of the missing values, given the observed data (i.e., draws from the posterior predictive distribution). However, from a procedural standpoint, the I-step amounts to stochastic regression imputation.

The ultimate goal of the imputation phase is to generate *m* complete data sets, each of which contains different estimates of the missing values. Creating unique sets of imputations requires different estimates of the mean vector and the covariance matrix at each I-step, and the purpose of the P-step is to generate these estimates. The P-step begins by using the filled-in data from the preceding I-step to estimate the mean vector and the covariance matrix, after which it generates alternative parameter estimates by randomly drawing new values from their respective posterior distributions. Conceptually, the algorithm generates new parameter values by adding a random residual term to each element in the complete-data mean vector and covariance matrix. The subsequent I-step uses these simulated parameter values to construct a new set of regression coefficients, and the process begins anew. Repeating the two-step procedure a number of times generates multiple copies of the data, each of which contains unique estimates of the missing values.

Unlike maximum likelihood estimation, data augmentation generates parameter values that constantly vary across successive P-steps. Although the behavior of the data augmentation algorithm is seemingly random from one cycle to the next, the parameter values and the imputations from successive iterations are correlated. Because the ultimate goal is to simulate independent draws from a distribution of plausible values, it is inappropriate to save and analyze the filled-in data sets from successive I-steps. One way to simulate independent draws from the distribution of missing data is to sample imputed data sets at regular intervals in the data augmentation chain (e.g., save and analyze the data from every 200th I-step). Time-series and autocorrelation function plots can help determine if the number of between-imputation iterations is large enough to produce independent sets of imputed values.

The convergence diagnostics play an important role in planning the final data augmentation run that generates the complete data sets. Choosing the number of imputed data sets to save and analyze is one of the most basic decisions in a multiple imputation analysis. Conventional wisdom suggests that multiple imputation analyses require relatively few imputations, and the literature historically recommends between three and five imputed data sets. However, contemporary research suggests that analyzing 20 data sets will maximize power in most situations. Although $m = 20$ appears to be a good rule of thumb, there is no downside (other than computer processing time) to using far more imputations (e.g., $m = 50$ or $m = 100$).

The next chapter describes the analysis and pooling phases. The purpose of the analysis phase is to analyze the filled-in data sets from the preceding imputation phase. This step consists of *m* statistical analyses, one for each imputed data set. The analysis phase yields several sets of parameter estimates and standard errors, so the goal of the pooling phase is to combine everything into a single set of results. Rubin (1987) outlined relatively straightforward formulas for pooling parameter estimates and standard errors. Because the analysis phase is relatively straightforward, most of Chapter 8 is devoted to the pooling phase and related inferential procedures. At the end of Chapter 8, I revisit some of the data analysis examples from Chapter 4 and illustrate how to analyze the data using multiple imputation.

7.16 RECOMMENDED READINGS

Allison, P. D. (2002). *Missing data*. Newbury Park, CA: Sage.

Graham, J. W. (2009). Missing data analysis: Making it work in the real world. *Annual Review of Psychology*, *60*, 549–576.

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

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. New York: Wiley.

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association*, *91*, 473–489.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. New York: Chapman.

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, *7*, 147–177.

Schafer, J. L., & Olsen, M. K. (1998). Multiple imputation for multivariate missing-data problems: A data analyst's perspective. *Multivariate Behavioral Research*, *33*, 545–571.

Sinharay, S., Stern, H. S., & Russell, D. (2001). The use of multiple imputation for the analysis of missing data. *Psychological Methods*, *6*, 317–329.