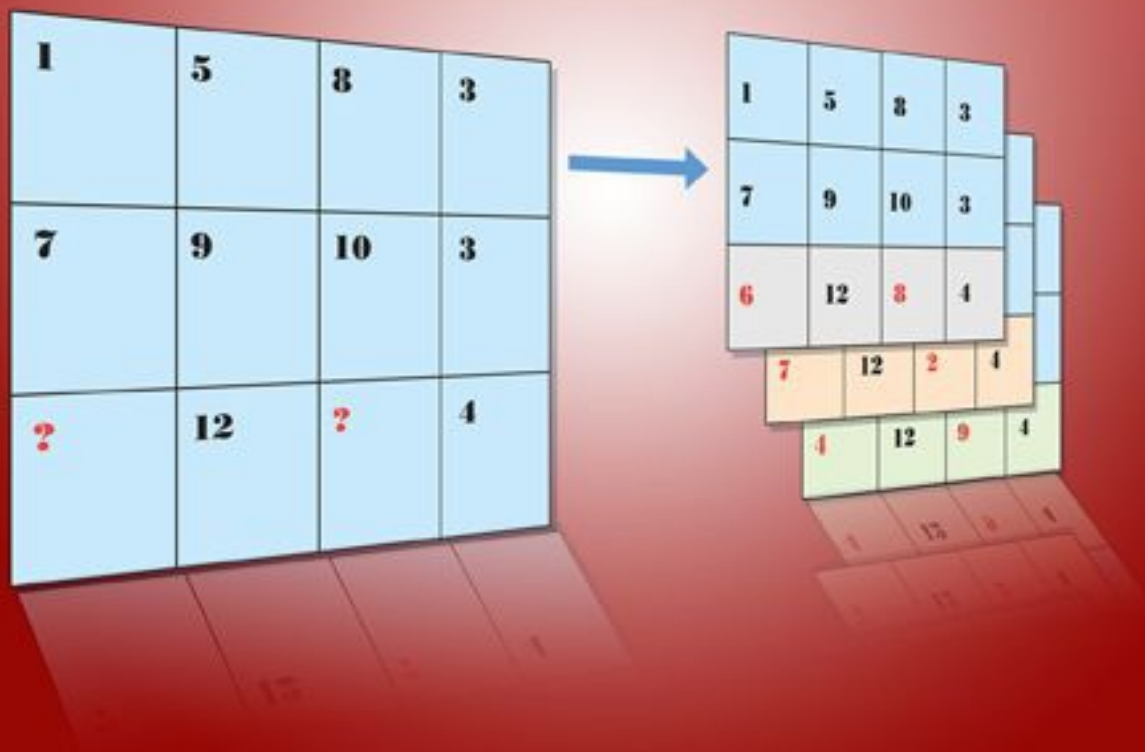
Multiple Imputation of Missing Data Using SAS®



Patricia Berglund and Steven Heeringa

Multiple Imputation of Missing Data Using SAS®

Patricia Berglund and Steven Heeringa

support.sas.com/bookstore

The correct bibliographic citation for this manual is as follows: Berglund, Patricia and Heeringa, Steven, 2014. *Multiple Imputation of Missing Data Using SAS®*. Cary, NC: SAS Institute Inc.

Multiple Imputation of Missing Data Using SAS®

ISBN 978-1-62959-204-6

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

July 2014

SAS provides a complete selection of books and electronic products to help customers use SAS® software to its fullest potential. For more information about our offerings, visit support.sas.com/bookstore or call 1-800-727-3228.

Contents

About This Book

Purpose

Multiple Imputation of Missing Data Using SAS provides both theoretical background and constructive solutions for those working with incomplete data sets in an engaging example-driven format. It offers practical instruction on the use of SAS for multiple imputation and provides numerous examples using a variety of public release data sets.

Is This Book for You?

Written for users with an intermediate background in SAS programming and statistics, this book is an excellent resource for anyone seeking guidance on multiple imputation. The authors cover PROC MI and PROC MIANALYZE in detail along with other procedures used for analysis of complete data sets. They guide analysts through the multiple imputation process, including evaluation of missing data patterns, choice of an imputation method, execution of the process, and interpretation of results.

Prerequisites

An intermediate level of SAS programming and statistics is recommended.

Scope of This Book

The authors cover PROC MI and PROC MIANALYZE in detail along with other procedures used for analysis of complete data sets. They guide analysts through the multiple imputation process, including evaluation of missing data patterns, choice of an imputation method, execution of the process, and interpretation of results. Many applications using real-world data sets are presented.

About the Examples

Software Used to Develop the Book's Content

SAS v9.4 is used throughout the book. PROC MI and PROC MIANALYZE are highlighted along with use of a variety of standard and SAS SURVEY procedures.

Example Code and Data

Example code and data are available from the book website. You can access the

example code and data for this book by linking to the Berglund author page at http://support.sas.com/publishing/authors.

Select the name of the author. Then, look for the cover thumbnail of this book, and select Example Code and Data to display the SAS programs that are included in this book.

For an alphabetical listing of all books for which example code and data is available, see http://support.sas.com/bookcode. Select a title to display the book's example code.

If you are unable to access the code through the website, e-mail saspress@sas.com.

Output and Graphics Used in This Book

All output and graphics were created using SAS v9.4.

Additional Resources

Applied Survey Data Analysis and website: http://www.isr.umich.edu/src/smp/asda/.

IVEware software download and documentation is available at: iveware.org.

Keep in Touch

We look forward to hearing from you. We invite questions, comments, and concerns. If you want to contact us about a specific book, please include the book title in your correspondence to saspress@sas.com.

To Contact the Author through SAS Press

By e-mail: saspress@sas.com

Via the Web: http://support.sas.com/author_feedback

SAS Books

For a complete list of books available through SAS, visit http://support.sas.com/bookstore.

Phone: 1-800-727-3228

Fax: 1-919-677-8166

E-mail: sasbook@sas.com

SAS Book Report

Receive up-to-date information about all new SAS publications via e-mail by subscribing to the SAS Book Report monthly eNewsletter. Visit http://support.sas.com/sbr.

Publish with SAS

SAS is recruiting authors! Are you interested in writing a book? Visit http://support.sas.com/saspress for more information.

About The Authors

Patricia Berglund is a Senior Research Associate in the Survey Methodology Program at the University of Michigan Institute for Social Research (ISR). She has extensive experience in the use of SAS for data management and analysis. She is a faculty member in the ISR's Summer Institute in Survey Research Techniques and also directs the ISR's SAS training programs. Berglund also teaches a SAS Business Knowledge Series class titled "Imputation Techniques in SAS." Her primary research interests are mental health, youth substance issues, and survey methodology.

Steven Heeringa is a Senior Research Scientist at the University of Michigan Institute for Social Research (ISR) where he is Director of the Statistical Design Group. He is a member of the Faculty of the University of Michigan Program in Survey Methods and the Joint Program in Survey Methodology at the University of Maryland. Heeringa is a Fellow of the American Statistical Association and elected member of the International Statistical Institute. He is the author of many publications on statistical design and sampling methods for research in the fields of public health and the social sciences. Heeringa has over 35 years of statistical sampling experience in the development of the ISR's National Sample

design, as well as research designs for the ISR's major longitudinal and cross-sectional survey programs.

Learn more about these authors by visiting their author pages, where you can download free book excerpts, access example code and data, read the latest reviews, get updates, and more: http://support.sas.com/berglund http://support.sas.com/heeringa

Acknowledgements

We gratefully acknowledge the anonymous technical reviewers along with editors and production staff at SAS Institute who made many excellent suggestions. Their help made this a much better book overall.

Special thanks go to our colleagues Dr. Trivellore Raghunathan and Dr. Brady West for their work and leadership in the area of multiple imputation.

Additional thanks to SAS employees Stacey Hamilton, Patsy Poole, Katie Whitley, Rob Agnelli, and Dr. Catherine Truxillo who each supported and encouraged our work on this book.

Chapter 1: Introduction to Missing Data and Methods for Analyzing Data with Missing Values

1.1 Introduction

Over the past half-century, statistical analysts have employed a wide range of techniques to address the theoretical and practical question of "what do I do about missing values?" These techniques have ranged from a simple default of dropping observations with missing data values from analysis (the list-wise deletion default in SAS and most other major software systems) to highly sophisticated methods for modeling the missing data mechanism in order to derive imputed values or to conduct a complex maximum likelihood analysis. There is no single approach that is optimal for all missing data problems—either in theory or in practice. Fortunately for the data analyst, SAS and other major statistical analysis software packages now provide their users with robust procedures tailored to address differing problems of missing data. *Multiple Imputation of Missing Data Using SAS* is written to serve as a practical guide for those dealing with general missing data problems in fields such as the social, biological, and physical sciences; medical and public health research; education; business; and many other scientific and professional disciplines.

Central to this book is the method of multiple imputation (MI) for item missing data. Supported by the SAS PROC MI and PROC MIANALYZE procedures, MI is based on a powerful set of methods for both filling in the missing values in the user data set and for performing robust statistical estimation and inference using the completed data sets.

The combination of basic theoretical background and extensive practical applications using SAS (v9.4) presented in this volume provides a solid foundation for understanding and resolving missing data problems using the multiple imputation method. The applications presented in Chapters 4 through 8 address a number of common missing data problems and imputation approaches using PROC MI and PROC MIANALYZE along with various descriptive and inferential tools for analysis of complete data sets. All examples stress the three-step process of multiple imputation: 1) selection of an appropriate data model for the imputation and application of an appropriate imputation method using PROC MI; 2) analysis of complete data sets using standard or SURVEY procedures; and 3) synthesis of analytic results for statistical inference using PROC MIANALYZE.

1.2 Sources and Patterns of Item Missing Data

Missing data takes many forms and can be attributed to many causes. For example, in data derived from surveys, item missing data occurs when a respondent elects not to answer certain questions, resulting in only a "don't know" or "refused" response. The failure to respond can be driven by the desire to elude answering questions that are sensitive in nature or perceived to be intrusive (e.g., illegal behavior, income-related, or medical history questions) or by a simple lack of knowledge required to answer the questions (e.g., expenditure on durable goods in the past 30 days). More generally in observational and experimental research, missing observations can arise due to missed clinical appointments, equipment failures, or other circumstances that disrupt the intended measurement. For example, a power outage that deactivates environmental data collection instrumentation for a period of time results in a missing data problem. In some research settings such as epidemiology (Schenker et al. 2010) and genetics (Bobb et al. 2011), "missing data" is actually better labeled "nonobserved data" and powerful new statistical tools including MI are used to impute the nonobserved data based on strong associations with variables that were observed in the same or a different study.

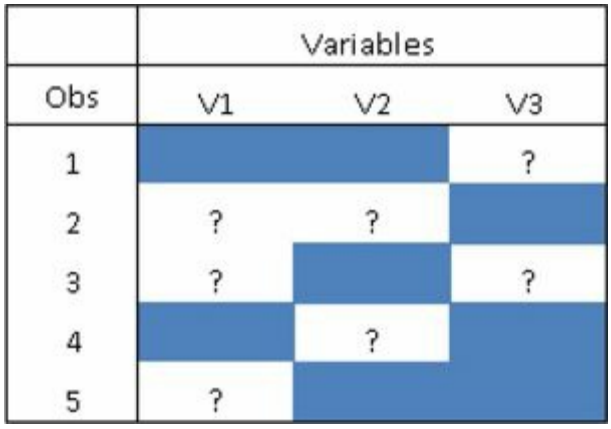
Missing data problems can be classified based on a notation and taxonomy employed by Little and Rubin (2002). In regard to the notation, the data of analytic interest are the true underlying values for an n x p matrix, Y, consisting

of i=1,…,n rows (sample cases) and j=1,… ,p variables, $Y_i = \{Y_{i1}, \ldots, Y_{ip}\}$ for each case. This underlying set of true values for the variables of interest is then decomposed into two subvectors of variables, $Y = \{Y_{obs}, Y_{mis}\}$, where Y_{obs} are the set of variable values that are observed and Y_{mis} are the variable values that are missing and must be imputed (Heeringa, West, and Berglund 2010). The statistical properties of individual variables and their relationships in the underlying data are governed by a distributional model, $f(y|\theta)$. Our analytic interest lies in making inference about the parameters θ or functions of these parameters. Maximum likelihood (ML) methods are often used in analysis, although corresponding Bayesian methods of analysis and inference may also be employed. In addition to the distributional model for the underlying data, the missing data problem also includes a second statistical model, $g(m|\psi)$, that governs the stochastic process which determines whether the true value of a Y_{ij} is missing (M_{ij}=1) or observed (M_{ij}=0). Corresponding to the Y matrix of true data values is a second array of identical dimension, M, of indicator values that flag whether the corresponding element of Y is missing or observed.

To help us better understand our missing data problem and to choose an appropriate imputation strategy, statisticians have created a two-part taxonomy for the most common missing data problems. The first dimension is labeled the *missing data pattern*, which as the name implies defines the particular distribution of the missing observations (represented by the M matrix) across the data cases (i=1,…,n) and variables (j=1,…,p) that comprise our analytical data set.

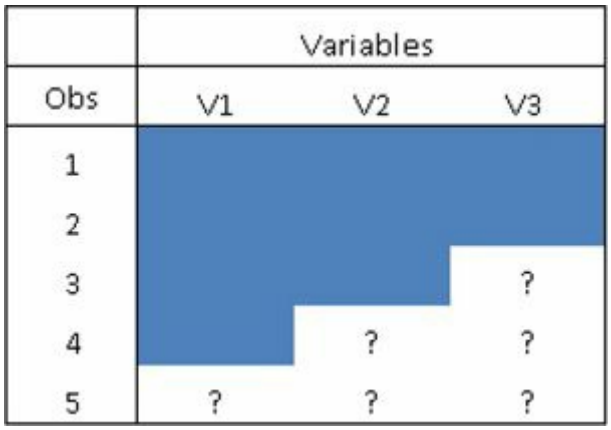
The most common missing data pattern is termed generalized or arbitrary—where there is no particular pattern in the missing data structure. As illustrated by the cells with "?" in the schematic array in Figure 1.1, missing observations are distributed across cases and variables in a nonsystematic fashion. The arbitrary missing data pattern illustrated in Figure 1.1 might be handled with an imputation using a Markov chain Monte Carlo (MCMC) or a fully conditional specification (FCS) method, both available in PROC MI.

Figure 1.1: Generalized Pattern of Missing Data



Some data collection processes produce a more structured or systematic pattern of missingness in the data. For example, in a medical study with multiple phases, missing data can occur when an entire phase of the data collection effort, such as a blood draw or obtaining medical records needed for follow-up, requires special consent of the subject. Without agreement from the study subject for participation in this type of data collection, there is missing data for all variables from the entire phase of the study. The result is a monotonic missing data pattern similar to that illustrated in Figure 1.2. Nonresponse to follow-up waves in a longitudinal survey may also produce a monotonic pattern of item missing data. Monotone patterns of missing data lend themselves to imputation methods that require simpler assumptions than approaches that are required for a general pattern of missing data and are efficiently handled in PROC MI. In fact, in Chapter 6 we will consider a two-step procedure that imputes small amounts of missing data in selected variables in order to transform a problem with a generalized missing data pattern to one that has a monotone pattern for key variables that suffer the highest rates of missing data. We note here that analysts always have the option to address monotone missing data patterns using procedures such as the FCS method that are applicable to the more general case of an arbitrary missing data pattern.

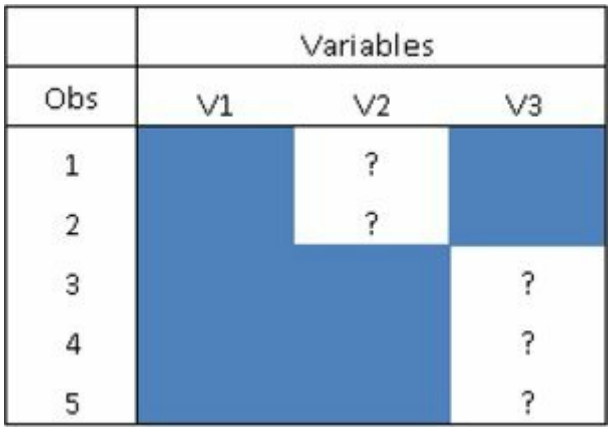
Figure 1.2: Monotone Missing Data Pattern



A third pattern of missing data arises in studies that incorporate randomization procedures to allow item missing data on selected variables for subsets of study

observations. The technique is termed *matrix sampling* or "missing by design" sampling (Thomas et al. 2006) and is often employed with modularized sets of data observations (e.g., physical tests or sets of survey questions). In a survey data collection that employs matrix sampling, designated subsets of core questions are asked of all respondents, with additional modules of more in-depth questions randomly assigned to subsamples of participants. Matrix sampling designs tend to produce a non-monotonic missing data structure such as that illustrated in Figure 1.3.

Figure 1.3 Matrix Sampling (Missing by Design)



For this type of missing data pattern, multiple imputation has typically been the primary tool to analyze these data (Raghunathan and Grizzle 1995). Use of PROC MI with an imputation method such as FCS that is suitable for an arbitrary missing data pattern can be employed. As with any imputation problem, the recommended imputation method depends on the pattern of missing data and the type of variables to be imputed.

1.3 Item Missing Data Mechanisms

The second dimension of statisticians' two-part taxonomy of missing data problems is termed the *missing data mechanism*. Missing data for a single variable is classified into one of three categories: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). Missing data are MCAR if the probability that an item value is missing is completely random and does not depend on the missing values for a case, Y_{mis}, nor does it depend on any of the observed variables for the case, Y_{obs}. A more realistic assumption for item missing data that underlies the procedures covered in this book is that the data are missing at random (MAR). The MAR assumption requires that, conditional on the observed data for the case, Y_{obs}, the probability that a value is missing does not depend on the true values of the missing items, Y_{mis}. For example, the predictive distribution used to draw imputed values for

Y_{mis} may be a regression model in which the predictors are selected from Y_{obs}. In practice, the MAR assumption may not strictly apply for all missing items. If the probability that a variable value is missing depends on the missing value and cannot be fully explained by the remaining observed variables, Y_{obs}, the missing data mechanism is labeled missing not at random (MNAR). For example, if after accounting for observed age, gender, education, and marital status of the household head, the probability of item missing data on a measure of household income depends on the underlying dollar value, then the problem is MNAR.

Little and Rubin (2002) classify a missing data mechanism as "ignorable" for likelihood-based inference if two conditions hold: 1) the missing data are MAR (missingness does not depend on Y_{mis}); and 2) the parameters of the data distribution, $f(y|\theta)$, are distinct from the parameters of the model for the missing data mechanism $g(M|\psi)$. (For Bayesian inference, the second condition requires that the prior distributions for θ and ψ are independent.) From our perspective as analysts of data with missing values, the first of these two conditions is the more important. Likelihood inference remains valid, albeit statistically less efficient, if the parameter spaces of the data distribution and the missing data generating model are not distinct.

A commonly asked question is, "Can the MAR assumption be tested in SAS?" The current version of SAS/STAT software, SAS/STAT 13.1, implements sensitivity analysis for departures from the MAR assumption with the new MNAR statement in PROC MI. See the PROC MI documentation for details. In addition, Little (1988) presents a likelihood ratio test of the null hypothesis that the data are MCAR versus the alternative that they are MAR (conditional on a defined set of observed covariates). Although useful in some special cases, this test is not sufficient to establish that the missing data mechanism is ignorable. The econometric literature on selection bias (e.g., Amemiya, 1985; Heckman, 1976) presents several tests of the null hypothesis that the data are MAR versus the MNAR alternative. As Little (1985) notes, however, these tests are very sensitive to correct model specification. Schafer (1997) describes a number of data collection designs where the missing data mechanism is clearly ignorable (double sampling, medical screening with multiple tests, matrix sampling, etc.) and others where it is not clear whether the missing data mechanism can be ignored (sample surveys where people are not at home, unexpected problems in experiments that prevent data collection, etc.). One approach to address possible departures from the MAR assumption is to increase the number of variables in the imputation model, thus making the assumption more plausible (Schafer 1997). Extreme departures from the MAR assumption may require special methods that require explicit modeling of the missing data mechanism.

Imputation methods for MNAR problems are beyond the scope of this volume, and therefore we refer the reader to the statistical literature on potential methods. Chapter 15 of Little and Rubin (2002) is a good starting point to learn more about methods for MNAR missing data.

1.4 Review of Strategies to Address Item Missing Data

1.4.1 Complete Case Analysis

The first and simplest approach to the problem of missing data is complete case analysis. The majority of SAS analytic procedures default to list-wise deletion and will automatically strike any case with item missing data from the analysis. Many analysts faced with a missing data problem question if imputation (or an alternative missing data procedure) is really necessary when considering the additional effort required to conduct the analysis. While item missing data rates of 1% to 5% for single variables are not likely to produce major biases for univariate estimates based on only the complete cases, list-wise deletion of missing data cases from a complex multivariate analysis can result in a significant loss of statistical information. The analytic implications of item missing data are both practical and statistical. A practical consequence of list-wise deletion is that it permits the size and composition of the analysis sample to vary depending on the variables included in the analysis, making it difficult to maintain a standardized set of inputs across a variety of analytic methods. Other statistical implications of list-wise deletion of cases due to item missing data include reduction of effective or "working" sample size and loss of precision, regardless of the missing data mechanism. If the missing data are MAR, list-wise deletion of cases can result in a biased analysis for the complete cases. Of course, if the aggregate rate of missing data or the individual rate for a single key variable (e.g., household income, diastolic blood pressure) is higher (say, 5% to 10%), the precision losses and instability in case counts for differing analyses and the potential for bias under MAR will increase. In these cases, it is best practice to employ maximum likelihood estimation methods or missing data imputation that maximize the use of the observed data in the complete and incomplete cases.

1.4.2 Complete Case Analysis with Weighting Adjustments

A second option for handling missing data is to analyze complete cases but introduce additional weighting adjustments to compensate for item missing data on a key variable or set of variables. The use of weighting to compensate for missing data is generally limited to monotonic missing data patterns in which large numbers of variables are missing for each case. Unit nonresponse in

surveys or other data collections, phase nonresponse, and longitudinal attrition in panel surveys each produce missing data patterns where a global weighting adjustment—applicable across many forms of analysis of the data—is an appropriate and practical choice to compensate for missing observations. In practice, adjustments to the base weight variables to address item missing data on single variables often lead to difficulties, in that different adjustment factors would be needed for each target variable. Analytically, the weight-by-variable approach would work for univariate analyses, but which of the variable-specific weights would be chosen for a multivariate analysis? Another problem is that access to data used for base weight construction is restricted, making later weight adjustments impossible. For these reasons, imputation is generally considered a better strategy for addressing generalized patterns of item missing data.

1.4.3 Full Information Maximum Likelihood

For some item missing data problems, the preferred approach may be one of several maximum likelihood approaches that are designed to find the parameter estimates of interest, $\hat{\theta}$, that maximize the likelihood, $L(\theta|Y_{obs})$. Full information maximum likelihood (FIML) methods (Enders 2001) directly maximize a likelihood function in which incomplete cases contribute support only to estimation of parameters for which the sufficient statistics are functions of the observed values for the case (e.g., μ_1,σ_1 for observed Y_1; σ_{12} for observed $\{Y_1, Y_2\}$). In the statistical literature, many common applications of FIML are used in analyses such as structural equation modeling (SEM) or other latent variable model analyses in which the full data Y are assumed to follow a multivariate normal distribution with parameter vector $\theta=\{\mu,\Sigma\}$. The FIML method is available in PROC CALIS—a standard SAS procedure for conducting SEM or other related forms of latent variable modeling. FIML methods require that the user define the parametric likelihood for the data. This presents a challenge in FIML applications to complex sample survey data where weighting is required and informative stratification and clustering of observational units make it difficult to specify the true form of the data likelihood (Heeringa, West, and Berglund 2010). We should note here that these same design features also pose challenges in multiple imputation approaches to missing data from complex sample designs (see Chapter 4).

1.4.4 Expectation-Maximization Algorithm

The expectation-maximization (EM) algorithm (Little and Rubin 2002) is another tool that can be used in missing data problems to generate ML estimates. Like FIML methods, the EM method requires a parametric likelihood function

for the complete data. EM employs an iterative two-step (expectation and maximization) approach to numerically derive ML estimates of parameters. The E step of the algorithm replaces the missing values with their expectations under the current iteration's estimates of the distributional model and constructs the corresponding sufficient statistics/complete data log-likelihood function. The M step then maximizes the complete data likelihood to obtain updated estimates of the model parameters. This E and M cycle repeats until the model parameters converge. If the complete data for the analysis problem are assumed to be distributed as multivariate normal, MVN(μ,Σ), the EM statement in PROC MI can be used to compute maximum likelihood estimates of the mean and covariance parameters. The estimated mean vector and covariance matrix can be output to a user designated file with the OUTEM= option of the EM statement in PROC MI. A "completed" data set with missing values replaced by the EM estimates of their expected values can also be output. However, it is important to note that the "imputed" data set so generated does not account for two sources of variability: 1) variability of the true values about their expectations (residual variance) and 2) the imputation variability that is inherent in the estimation of the expected values. These two sources of variability are reflected in a proper MI treatment of the missing data problem. EM does play an important role in PROC MI in that the completed data set generated by EM serves as the matrix of starting values for the iterative MCMC multiple imputation procedure.

1.4.5 Single Imputation of Missing Values

A common approach for handling item missing data prior to analysis is to perform a single imputation of missing values, creating a "complete" data set. In fact, scientific public use data sets are often released with a single imputed value replacing missing data on key variables. Or, data users may also choose to perform their own single imputations using an established stochastic imputation method such as the hot deck, regression imputation, or predictive mean matching (Little and Rubin 2002). Use of procedures such as mean, median, or modal value imputation are not encouraged unless the imputation is simply serving to fill in a small handful of missing values for an otherwise nearly complete variable.

An advantage to a singly imputed data set is that it is "completed" with missing values replaced by imputed values. Provided that the imputation technique is multivariate and retains the stochastic properties in the observed data, a single imputation may address potential bias for MAR missing data. On the other hand, an important disadvantage in the standard analysis of a singly imputed data set is that it precludes estimation and inference that fully reflects the variance attributable to the item missing data imputations. Rao and Shao (1992) have

proposed a technique for estimating variances and developing confidence intervals for estimates based on singly imputed data sets. More recently, Kim (2011) has proposed the method of fractional imputation, which also enables variance estimation and inference from imputed data. At this writing, neither of these methods has been implemented in SAS procedures.

1.4.6 Multiple Imputation

The robust, flexible option in many practical problems and the major focus of this book is to address missing values within the MI framework for estimation and inference. This approach consists of a three-step process: 1) formulation of the imputation model and imputation of missing data using PROC MI, 2) analysis of complete data sets using standard SAS procedures (that assume the data are identically and independently distributed or from a simple random sample) or SURVEY procedures for analysis of data from a complex sample design, and 3) analysis of the output from the two previous steps using PROC MIANALYZE. Many types of missing data patterns and analytic models can be handled within this framework, making it, in our opinion, the preferred option for dealing with most missing data problems.

The many approaches and options available in PROC MI reflect the historical sequence of developments in MI theory and practice. Some approaches were developed specifically for particular patterns of missing data (e.g., MONOTONE), while others assume specific joint distributions for the variables of interest (e.g., MCMC for multivariate normal data). In years past, analysts often faced practical problems where the data patterns or distributions did not conform to the exact theoretical assumptions of the available multiple imputation tools. For example, variables with few missing observations could be initially imputed to convert a generalized pattern of missing data to a monotone missing data problem. The MCMC method, designed for multivariate normal data, would be applied to a generalized pattern of missing data for a vector of categorical variables or a mixture of categorical and continuous variables. While many of these MI practices are still serviceable, it is the case that new developments in PROC MI now provide a simpler, better approach. For example, we view the newly introduced FCS method as preferred over MCMC for multiple imputation of multivariate problems that include mixtures of categorical and continuous variables. To ensure comprehensive treatment of the capabilities of SAS for MI, throughout this text we will attempt to cover all of the common approaches—old and new—that are available in PROC MI, but when multiple alternatives are available we will clearly indicate which approach we judge to be current best practice.

1.5 Outline of Book Chapters

Chapter 1 introduced the topic of imputing missing data and the issues that arise in analysis of data sets with missing data. It provides an overview of how and why missing data occurs along with an introduction to the use of multiple imputation to deal with missing data problems. It also presents an overview of examples to come in later chapters.

Chapter 2 offers a detailed look at imputation with an emphasis on the multiple imputation approach. It addresses how to implement multiple imputation by outlining the general framework of model specification, imputation methods, analysis of MI data sets, and MI estimation and inference. A description of general imputation algorithms and formulae for the MI and MIANALYZE procedures are presented in this chapter. Chapter 2 also includes a brief overview of PROC MI and PROC MIANALYZE along with discussion of common SAS procedures used in the MI process.

Chapter 3 outlines a general step-by-step approach for planning and conducting a multiple imputation analysis in SAS. A simple example of the MI process is presented at the end of this chapter and serves as a prelude to more complex applications described in later chapters.

Chapter 4 provides an overview of the special issues involved in multiple imputation for complex sample design data. It includes a discussion of how things change with complex sample data and provides a comparative example of a naive imputation (ignoring the design in the imputation step) contrasted with imputation that includes the complex sample design features in the imputation model.

Chapter 5 covers imputation of continuous variables. Each example in this chapter includes a brief overview of the statistical foundations of the particular imputation method; demonstration of the three-step process of multiple imputation, including use of applicable options and diagnostics in both PROC MI and PROC MIANALYZE; and interpretation of the output from each step of the process.

Chapter 6 repeats the process and steps described for Chapter 5 but focuses on imputation of classification (categorical) variables.

Chapter 7 presents two case studies typical of "real-world" missing data problems and multiple imputation options for analysis. The first example provides a comparison of a complete case analysis and MI treatment of a missing data problem based on the Health and Retirement Study (HRS). The

second case study demonstrates multiple imputation of longitudinal data based on a clinical trial focused on the impact of anti-epilepsy medication on seizures.

Chapter 8 includes examples of preparation of output data sets from the analysis of complete data sets in formats readable by PROC MIANALYZE. This chapter covers details of the various types of estimates, parameter, and covariance output data sets that can be used by PROC MIANALYZE and includes examples using a variety of regression procedures.

The author page for this book includes the SAS code and SAS data sets used in the application examples in the book as well as FAQs, a bibliography, and other resources and updates for data analysts using the SAS multiple imputation procedures.

1.6 Overview of Analysis Examples

In Chapters 4 through 7 we include a variety of applications of multiple imputation to a wide range of data sets, imputation methods, and analysis techniques. Table 1.1 is a summary grid outlining the examples, including information about the three-step process of multiple imputation. Details are provided about each example, such as chapter/example number, data set name and characteristics, missing data pattern, type of variable(s) imputed, imputation method, type of variables used in the imputation, and analysis procedure used in the second step of the MI process.

Table 1.1: Overview of Complete Multiple Imputation Examples from Chapters 4 through 7

Chapter/ Example Number	Data Set/Design Characteristics	Missing Data Pattern	Type of Variable(s) Imputed	Type of Variables Used in Imputation	Imputation Methods Used	Analy Techn to An Imput Sets
4.1	NHANES 2009–2010/Complex Sample Design	Monotone	Continuous	Mixed	Monotone	Design means subpo (PRO MEAN
5.2	Major League Baseball Players' Salaries (1992)/Standard	Arbitrary	Continuous	Continuous	MCMC	Linear (PRO
5.3						Design

	NHANES 2009–2010/Complex Sample Design	Monotone	Continuous	Mixed	Regression PMM	means for a s subpo (PRO SURV
5.4	NHANES 2009–2010/Complex Sample Design	Arbitrary	Continuous	Mixed	FCS regression	Design linear (PRO SURV
6.2	Myocardial Infarction/Standard	Monotone	Categorical	Mixed	Logistic regression	Freque (PRO
6.3	NCS-R/Complex Survey Design	Arbitrary	Categorical	Mixed	FCS logistic regression FCS discriminant function	Design freque (PRO SURV
6.4	NHANES 2009–2010/Complex Sample Design	Arbitrary	Categorical	Mixed	FCS logistic regression Multistep MCMC monotone with logistic regression	Design logistic (PRO LOGI
7.1	HRS 2006/Complex Sample Design	Arbitrary	Mixed	Mixed	FCS logistic regression, discriminant function, regression	Design logistic (PRO LOGI
7.2	Clinical Trial Seizure Data/Standard Longitudinal data	Arbitrary	Continuous	Mixed	FCS PMM	Poisso Regres Repea Measu GENM

Table 1.2 summarizes the application demonstrated in Chapter 8 where we highlight the process of producing output data sets from step 2 that can be easily input into PROC MIANALYZE.

Table 1.2: Overview of Creating Output Data Sets for PROC MIANALYZE from Chapter 8

						Analysis Procedu

Chapter/ Example Number	Data Set/Design Characteristics	Missing Data Pattern	Type of Variable(s) Imputed	Type of Variables Used in Imputation	Imputation Methods Used	Used to Create Output Set for U PROC MIANA
8.2 and 8.3.1	MLB Baseball Players' Salaries (1992)/Standard	Monotone	Continuous	Mixed	MCMC	Linear regressio (PROC C
8.2 and 8.3.2	MLB Baseball Players' Salaries (1992)/Standard	Monotone	Continuous	Mixed	MCMC	Linear regressio (PROC MIXED)
8.4 and 8.5	NCS-R/Complex Sample Design	Arbitrary	Categorical	Mixed	FCS logistic regression	Design-adjusted Proportic Hazards (PROC SURVEY PHREG)

Chapter 2: Introduction to Multiple Imputation Theory and Methods

2.1 The Origins and Properties of Multiple Imputation Methods for Missing Data

2.1.1 A Short History of Imputation Methods

Item missing data has long been recognized as a problem for data analysts. Early solutions to the problem of missing data were directed to specific distributions for the variables of interest and patterns of missing data. For example, Buck's (1960) method introduced imputations of conditional mean values for each pattern of missing observations in a multivariate normal vector of variables.

Broad, formal recognition of imputation as a statistical technique for dealing with missing data may have originated with the National Research Council's (NRC) Panel on Incomplete Data. Many of the earliest papers on imputation concepts and theory appear in the 1985 three-volume publication produced by the panel (Madow and Olkin 1983). Throughout the 1980s, statisticians continued to conduct research and to publish on imputation methods (Kalton 1983; Rubin 1980; Sande 1983). Of particular relevance to the software and methods described in this volume, the general theory and methods were greatly extended by the introduction of the multiple imputation method (Rubin 1987). Despite these developments, the introduction of imputation methods to statistical practice at that time was a slow process and by no means universal.

Prior to the mid-1980s, the accepted procedure among most data analysts was to explicitly denote values as missing (e.g., the "." symbol for numeric data in SAS) but to take no corrective steps in actual data analysis other than to analyze complete data cases. During the 1980s, major federal survey programs in the United States and Canada took the lead in the development and application of basic imputation methods such as regression imputation and the hot deck imputation method. In the United States, developments in imputation methods were promoted by programs such as the Survey of Income and Program Participation (SIPP), programs that required the collection of many financial variables that were subject to significant rates of item missing data. By the early 1990s, large-scale, general purpose imputation of item-missing values in major survey data sets had become a common and accepted statistical practice.

During the 1990s and continuing to the present, the demand for practical methods to address increasingly large and complex missing data problems in surveys and other statistical investigations led to an explosion of new theoretical work during the next two decades, much of it focused on methods of multiple imputation (van Buuren 2012).

In the wake of these advances in imputation theory, general purpose procedures for multiple imputation of item missing data (PROC MI) and statistical analysis of imputed data sets (PROC MIANALYZE) were introduced to SAS and other major statistical software systems. Today, despite multiple imputation's roots in

problems of missing data for large complex surveys, MI techniques and software are being applied to a wide range of statistical problems that involve missing data (Reiter and Raghunathan 2007).

As introduced in Chapter 1, multiple imputation in SAS is a three-step procedure for the treatment of missing data in statistical analysis. In SAS:

1. The analyst defines a multivariate "imputation model" for the data, and under this model PROC MI is used to independently impute missing values in the original data set m=1,…,M times, generating M complete "repetition" versions of the analysis data set.
2. Standard or SURVEY procedures (e.g., PROC MEANS/SURVEYMEANS, PROC REG/SURVEYREG, etc.) in SAS are then used to analyze each of the M completed data sets and output the results.
3. PROC MIANALYZE inputs the results of the M separate analyses and applies multiple imputation formulae to generate estimates, standard errors, confidence intervals, and test statistics for the descriptive statistics or model parameters of interest.

This chapter provides an intermediate-level introduction to multiple imputation theory and methods that we feel are most relevant to the SAS user. Readers who are interested in a more detailed theoretical review of MI are referred to Chapter 56 (PROC MI) and Chapter 57 (PROC MIANALYZE) of the SAS/STAT documentation or to one of the many excellent published texts on the subject (Allison 2001; Rubin 1987; Schafer 1997; van Buuren 2012).

2.1.2 Why the Multiple Imputation Method?

No imputation method or statistical modeling technique is optimal for all forms of missing data problems. As discussed in Chapter 1, there are at least six general approaches to the treatment of missing data in statistical analyses:

1. Conduct complete case analysis, ignoring cases with missing data (the list-wise deletion default in SAS);
2. Employ weighting of complete cases to compensate for missing data;
3. Employ full information maximum likelihood methods to analyze the data;
4. Analyze the incomplete data using the EM algorithm (Allison 2001; Little and Rubin 2002);
5. Perform single imputation of missing values using deterministic or stochastic approaches such as mean/mode imputation; regression imputation, predictive mean matching, nearest neighbor method, or the hot

deck (see Kalton and Kasprzyk [1986] and Little and Rubin [2002] for a comprehensive review);

6. Develop multiple imputations of the item missing data and employ MI estimation and inference in analysis.

Since the primary purpose of this volume is to instruct the data user in the capabilities of SAS for multiple imputation using the paired PROC MI and PROC MIANALYZE procedures, we will not go into depth on the particular and comparative properties of these missing data methods. Instead, we refer the reader to the many excellent references that are identified in the text or in the extended reference list. It is our view that the strengths of the multiple imputation approach to item missing data rest on the following attributes of properly designed and executed MI methods:

MI is model-based. It ensures statistical transparency and integrity of the imputation process. To ensure robustness in analysis, the *imputation model* should be broader than the *analysis models* that will be analyzed using the imputed data (see Section 2.2). The model that underlies the imputation process is often an explicit distributional model (e.g., multivariate normal), but good results may also be obtained using techniques where the imputation model is implicit (e.g., nearest neighbor imputation).

MI is stochastic. It imputes missing values based on draws of the model parameters and error terms from the predictive distribution of the missing data, Y_{mis}. For example, in linear regression imputation of the missing values of a continuous variable, the conditional predictive distribution may be: $\hat{Y}_{k,mis} = \hat{\beta}_0 + \hat{\beta}_{j \neq k} \cdot y_{j \neq k} + e_k$. In forming the imputed values of $Y_{k,mis}$, the individual predictions incorporate multivariate draws of the $\hat{\beta}$ s and independent draws of e_k from their respective estimated distributions. In a hot deck, predictive mean, or propensity score matching imputation, the donor value for $Y_{k,mis}$ is drawn at random from observed values in the same hot deck cell or in a matched "neighborhood" of the missing data case.

MI is multivariate. It preserves not only the observed distributional properties of each single variable but also the associations among the many variables that may be included in the imputation model. It is important to note that under the assumption that the data are missing at random (MAR), the multivariate relationships that are preserved are those relationships that are reflected in the observed data, $Y_{obs.}$

MI employs multiple independent repetitions of the imputation procedure

that permit the estimation of the uncertainty (the variance) in parameter estimates that is attributable to imputing missing values. This is variability that is in addition to the recognized variability in the underlying data and the variability due to sampling.

MI is robust against minor departures from strict theoretical assumptions. No imputation model or procedure will ever exactly match the true distributional assumptions for the underlying random variables, *Y*, nor the assumed missing data mechanism. Empirical research has demonstrated that if the more demanding theoretical assumptions underlying MI must be relaxed that applications to data can produce estimates and inferences that remain valid and robust (Herzog and Rubin 1983).

MI is very usable in real statistical applications, a practical feature that has been enhanced tremendously in the past ten years through the introduction of MI procedures to SAS and other major statistical software systems.

Setting aside for the moment its theoretical elegance and ties to sophisticated theory of Bayesian inference, the concept of multiple imputation was formulated by Rubin (1987) in large part to address the need for a robust method that could be applied to large data sets with many variable types. Furthermore, he recognized that in most cases the "data imputer" and the "data analyst" might well be different individuals with access to differing levels of information concerning the data collection design and the missing data process. Rubin sought a procedure that would enable the "imputer" to take full advantage of the available data and sophisticated imputation procedures yet leave the "data analyst" with a simple process to analyze the imputed data and obtain robust statistical inferences.

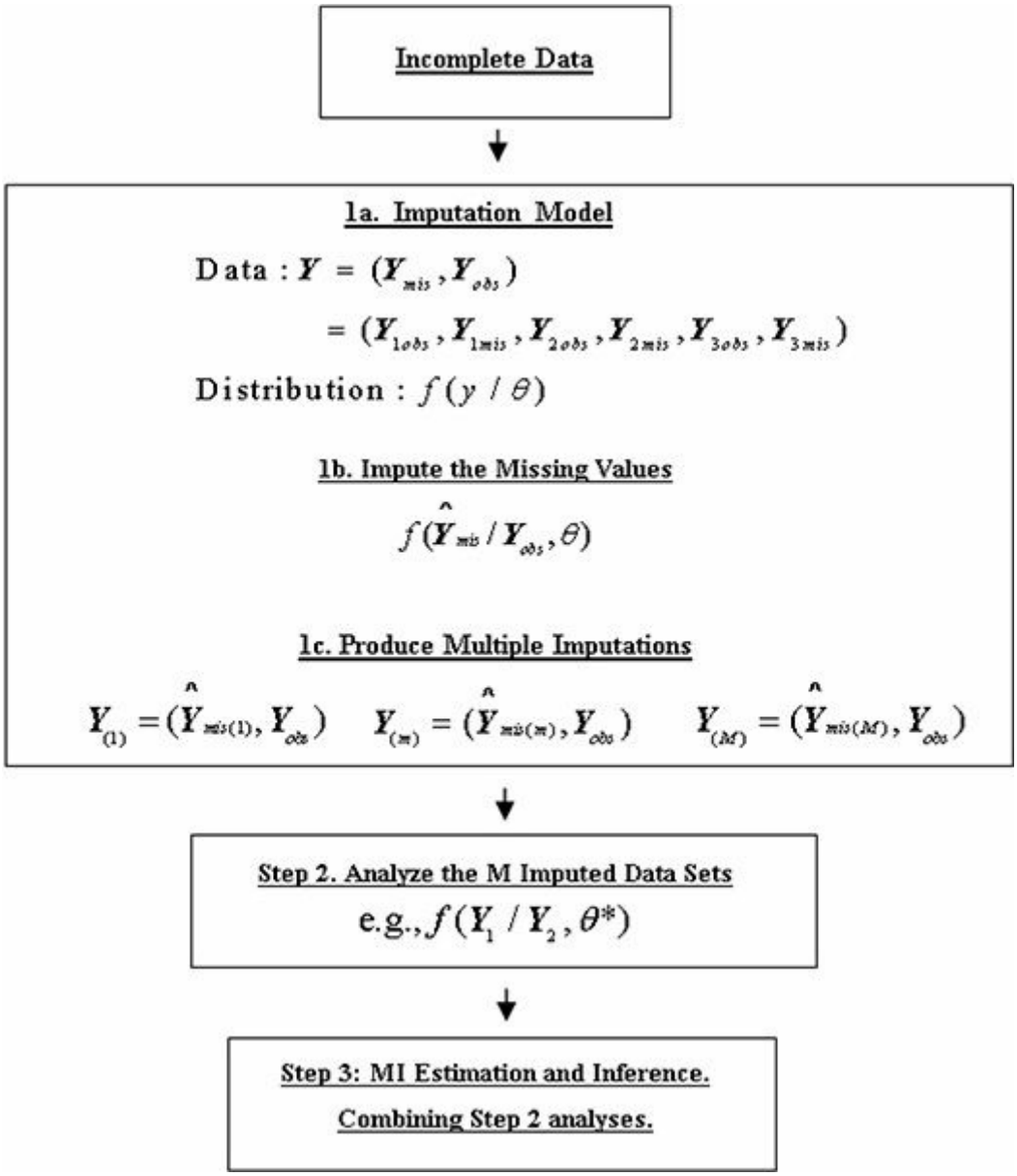
The most valid criticisms of the multiple imputation method (Fay 1996; Kim et al. 2006) have zeroed in on the notion that the imputer's statistical model for the imputation might be very different from the data models of interest to the many data analysts who will subsequently use the multiply imputed data. For example, in developing the multiple imputations of the item missing data, the imputer may fail to include an important analytical variable in her/his imputation model. Or, he/she may impute the missing data assuming a linear relation between two variables (e.g., height and weight) when, in fact, the relationship is nonlinear. The imputer may also fail to incorporate analytically important interactions or moderating effects among variables in the imputation model. Depending on the strength of these omitted effects and the amount of missing data for the variables in question, statistical analysis based on the imputed data may be subject to estimation and inferential bias. In Section 2.2 and throughout this volume, we will provide guidance on how to avoid such known pitfalls in the MI method.

In summary, for general missing data problems of the type covered in this book, PROC MI and PROC MIANALYZE provide a user-friendly platform for conducting a multiple imputation analysis that is theoretically-based, empirically-tested, and capable of generating robust statistical inferences.

2.1.3 Overview of Multiple Imputation Steps

Multiple imputation (MI) is not simply a technique for imputing missing data. It is also a method for obtaining estimates and correct inferences for statistics ranging from simple descriptive statistics to the parameters of complex multivariate models. As illustrated in Figure 2.1, a complete multiple imputation analysis in SAS can be organized into three sequential steps.

Figure 2.1: Three Steps in the Multiple Imputation Approach (adapted with permission from Heeringa, West, and Berglund 2010)



Step 1 defines the variables and distributional assumptions of the imputation

model and applies specific MI algorithms to generate the multiple imputations of the missing values and output a completed data set for each of m=1,…,M repetitions of the imputation process. The definition of the variables and distributional assumptions of the imputation model are primarily the responsibility of you, the user. The guidance and examples provided in this and later chapters are designed to help you make informed decisions on how to select the imputation model for your missing data problem.

SAS users have a wide array of choices for the imputation algorithm. As we will learn, the user-specified choice of the imputation algorithm will depend on the pattern of missing data (monotone, arbitrary), the type (continuous, binary, nominal, ordinal) of the variables to be imputed, and reasonable assumptions about the nature of the multivariate distribution of the variables specified in the imputation model. PROC MI will output the multiple copies of the completed data set as a "stacked" file with the _IMPUTATION_ variable added to distinguish the repetition assignment of each completed case.

Step 2 inputs the imputed data file produced by PROC MI and uses standard or SURVEY procedures in SAS with a BY _IMPUTATION_ statement to independently analyze each of the m=1,…,M repetition data sets. A critical activity at this analysis step is to specify an appropriate output of the estimated statistics and their standard errors from each of the repetitions of the analysis. The output data set of the estimated statistics and standard errors will be the required input for PROC MIANALYZE (step 3).

At Step 3, PROC MIANALYZE inputs the parameter estimates and standard errors from the preceding analysis step and applies the multiple imputation formulae to generate the MI estimate, standard errors, confidence intervals, and hypothesis test statistics for making inferences for descriptive statistics or model parameters.

The following three sections provide an intermediate-level introduction to these three steps of the MI process.

2.2 Step 1—Defining the Imputation Model

The actual process of imputing item missing values is governed by the imputation model, which we define as the set of variables that are available to the imputation process, Y=$\{Y_1,\ldots,Y_p\}$, and the distributional assumptions, $f(Y|\theta)$, for the multivariate relationships among the elements of Y.

Consider a trio of variables, Y= {Y_1=diastolic blood pressure (mm Hg); Y_2=age

(years); and Y_3= Body Mass Index (kg/m^2)} from the National Health and Nutrition Examination Survey (NHANES) 2009–2010 medical examination component (MEC). The data are restricted to NHANES respondents age 20 and older (n=6,059). The unweighted missing data rate is 8.4% for diastolic blood pressure (DBP) and 1% for body mass index (BMI), and following NHANES editing, age is observed for every case. The pattern of missing data is not monotone. One possible imputation model would be to include all three variables in the imputation process and to assume that the joint distribution for these three continuous variables is multivariate normal with mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$, that is, $f(Y|\theta)$=MVN(μ,Σ). Under this distributional model, multiple imputations of the missing data for diastolic BP and BMI are easily performed using the PROC MI Markov chain Monte Carlo (MCMC) method (Schafer 1997). The imputations performed under this imputation model would serve for univariate MI estimation of the mean diastolic BP, μ_1, or mean BMI, μ_3, and they would also serve for the MI estimation of multiple regression parameters, for example, the regression of diastolic BP on age and BMI.

2.2.1 Choosing the Variables to Include in the Imputation Model

The choice of variables to include in the imputation model should not be limited to only variables that have item missing data or variables that are expected to be used in a subsequent analysis. As a general rule of thumb, the set of variables included in the imputation model for an MI analysis should be much larger and broader in scope than the set of variables required for the analytic model. For example, if age and BMI are the chosen predictors in the analytic model for diastolic blood pressure, the imputation of item missing data for diastolic blood pressure and BMI might include many additional variables, such as gender, race/ethnicity, marital status, height, weight, systolic blood pressure, and so on. If the relationship of age to diastolic blood pressure is not linear, the regression model that is used to impute missing blood pressure measurements may include both a linear and quadratic term for age. If gender moderates the effects of age on diastolic blood pressure, an interaction term for age and gender should be included in the imputation model. Obviously, it is not feasible to define an imputation model and perform multiple imputations using every possible variable in the survey data set. Based on recommendations from Schafer (1999) and van Buuren (2012), some practical guidelines for choosing which variables to include in the imputation model are the following:

1. Include all key analysis variables: (dependent: Y_1 and independent: Y_2)

2. Include other variables that are correlated or associated with the analysis variables: (Y_3)
3. Include variables that predict item missing data on the analytic variables: (Z)

Failure to include one or more analysis variables (Y_1 and Y_2) in the imputation model can result in bias in the subsequent MI estimation and inference. Including additional variables, (Y_3), that are good predictors of the analytic variables improves the precision and accuracy of the imputation of item missing data. Under the assumption that item missing data is MAR, incorporating variables (Z) that are correlated with the variables that have missing data and predict the propensity for response will reduce bias associated with the item missing data mechanism.

For multiple imputation, a general piece of advice for practitioners that has come from extensive empirical work and simulation testing is: "When in doubt, including more variables in the imputation model is better."

2.2.2 Distributional Assumptions for the Imputation Model

Statistical models that define the relationships of the variables that are jointly considered in the missing data problem are key to all imputation methods. Under some imputation methods such as hot deck imputation the models are implicit in the mechanics of the procedure. Other imputation methods are based on explicit probability models, $f(Y|\theta)$, for the multivariate relationships among the elements of the complete data Y. In theoretical discussions of multiple imputation methods, convenient choices of a multivariate model for the joint distribution of the broad set of imputation variables might be multivariate normal (continuous) or multinomial (classification). As described in Section 2.3, several of the imputation options in PROC MI explicitly assume that the imputation variables follow one of these standard multivariate data models. Other procedures such as the fully conditional specification (FCS) method will not specify the distributional models for the data but use iterative simulation to approximate the complex and unknown distribution for problems involving a mixture of variable types and arbitrary patterns of missing data.

2.3 Algorithms for the Multiple Imputation of Missing Values

Once the user has defined the imputation model, an imputation algorithm is used to generate m=1,…,M completed data sets in which the missing values, Y_{mis}, have been imputed. In this volume, the m=1,…,M independently imputed versions of the data set are termed repetitions.

2.3.1 General Theory for Multiple Imputation Algorithms

The theoretical development of multiple imputation methods for missing data is rooted in the Bayesian framework for statistical inference. Within this framework, the task of imputing missing values, Y_{mis},in a data set equates to a random draw of an imputed value from the posterior predictive distribution of the missing data which we will denote as $f(Y_{mis}|Y_{obs},\theta)$ where θ is the vector of parameters (e.g., μ,Σ) or functions of these parameters (e.g., regression parameters β, $\sigma_{y.x}$) that uniquely define this predictive distribution for the missing values. Fortunately for most of us, we do not need to be experts in Bayesian inference to apply multiple imputation methods in practice. However, it is useful to have a simple overview to understand how this "posterior predictive distribution" is actually derived or simulated under the various MI techniques that are included in SAS PROC MI.

To avoid confusion in terms, from this point forward we will refer to the posterior predictive distribution as the predictive distribution for the missing data values. Examining the notational symbol for this distribution, $f(Y_{mis}|Y_{obs}, \theta)$, we see that this distribution is a function of our observed data, Y_{obs} and distributional parameters, θ. The process of imputing missing values for Y_{mis} requires that we first derive the predictive distribution, $f(Y_{mis}|Y_{obs}, \theta)$— labeled the "posterior" or "P" step—and then make random draws of imputed values from the predictive distribution, $Y_i^* \approx f(Y_{mis} \mid Y_{obs}, \theta)$. The drawing of random variates from the predictive distribution for Y_{mis} to fill in the missing values in the data is called the imputation or "I" step in the imputation process. In many imputation procedures, the process of generating "draws" from the predictive distribution is based on familiar predictions from regression functions (linear, logistic, discriminant function) that impute the missing value based on its expected relationship to other observed covariates.

We noted above that the predictive distribution $f(Y_{mis}|Y_{obs}, \theta)$ is a function of our observed data, Y_{obs}, and distributional parameters, θ. The data, Y_{obs}, however, as in any problem of statistical inference from a sample of data, has values of the parameters, θ, that must be estimated or derived. The same is true for the P-step in the imputation process. In the Bayesian framework for estimation and inference, these distributional parameters, e.g., $\theta=\{\mu, \Sigma\}$ for the multivariate normal, are assumed to have a prior probability distribution, $g(\theta)$. The observed data are used to update our information about the likely values of the true θ, to produce a new posterior distribution for the parameters, $p(\theta|Y_{obs})$. To execute the P-step in the imputation process, it is necessary to derive this posterior

distribution for θ exactly or to somehow simulate it closely through an iterative computational process. If the form of the complete data distribution, $f(Y|\theta)$, and the prior distribution, $g(\theta)$, are specified, the form of the posterior distribution of the parameters can often be derived through a formula based on Bayes' Rule:

$$p(\theta \mid Y_{obs}) = \frac{f(y \mid \theta) \cdot g(\theta)}{\int_{\theta} f(y \mid \theta) \cdot g(\theta) \cdot d(\theta)}$$

The monotone methods in PROC MI (linear regression for continuous *Y*, logistic regression for binary and ordinal *Y*, discriminant function method for nominal categorical *Y*) impute one variable at a time. The algorithm for each of these three MONOTONE methods imputes missing data based on a known expression for the posterior distribution of the parameters in the predictive distribution of the missing data. See the SAS/STAT PROC MI documentation for details.

In the common situation where the missing data problem is multivariate, has an arbitrary pattern of missing values, and may include variables of differing type (continuous, nominal, binary, ordinal), it is analytically difficult or impossible to evaluate the true expression for the joint posterior distribution, $p(\theta|Y_{obs})$. In such cases, statisticians have devised iterative simulation techniques that permit us to approximate draws from the analytically intractable complex joint posterior. The PROC MI MCMC method is such an algorithm to simulate the joint posterior, $p(\theta|Y_{obs})$, for arbitrary data patterns in which the underlying complete data are assumed to follow a multivariate normal distribution. The FCS method uses an iterative sequence of draws from conditional distributions (linear regression or predictive mean matching for continuous *Y*, logistic regression for binary and ordinal *Y*, discriminant function method for nominal categorical *Y*) to simulate draws from the highly complex joint posterior distribution of parameters for a set of variables of mixed distributional type.

To summarize, depending on the pattern of missing data and variable types, PROC MI provides three primary classes of methods for generating the multiple imputations. If the pattern of missing data is univariate or monotonic (Figure 2.2), the monotone option is the method of choice. For an arbitrary multivariate pattern of missing data (Figure 2.3), the choice is between the MCMC or the FCS methods. Table 2.1 summarizes methods available in SAS (v9.4) according to the pattern of missing data and the type of variable being imputed.

Table 2.1: SAS PROC MI Imputation Methods

Missing Data Pattern	Variable Type	Method

Monotone	Continuous	Linear regression, predictive mean matching, propensity score
	Binary/ordinal	Logistic regression
	Nominal	Discriminant function
Arbitrary	Continuous	With continuous covariates: MCMC monotone method MCMC full-data imputation
	Continuous	With mixed covariates: FCS regression FCS predictive mean matching
	Binary/ordinal	FCS logistic regression
	Nominal	FCS discriminant function

The following paragraphs provide an overview of each of these algorithms as they apply to variables of differing types.

2.3.2 Methods for Monotone Missing Data Patterns

The PROC MI algorithm for the multiple imputation of monotone missing data involves a non-iterative sequence of steps to generate each of the m=1,…,M repetition imputations of Y_{mis}. To outline the steps in the algorithm, we will use the notation and missing data pattern shown in Figure 2.2 with Y={Y_1, Y_2, Y_3, Y_4, Y_5}. In Figure 2.2, Y_1 and Y_2 are shown as fully observed—there are no missing values for these two variables. Moving from left to right in the ordered data array, the remaining variables have increasing amounts of missing data, and the missing data is always nested so that whenever Y_3 is missing, Y_4 and $Y_5$are missing as well. Likewise, any cases that are missing values on $Y_4$are also missing Y_5.

Figure 2.2 Monotone Multivariate Missing Data Pattern

	Variables				
Obs	Y1	Y2	Y3	Y4	Y5
1			?	?	?
2				?	?
3					?
4					?
5					?

In this example, the sequence of imputations in the monotone pattern therefore begins with imputation of missing values of Y_3.

The P-step in the imputation of missing Y_3 will utilize the relationship of the observed values of $Y_3$to the corresponding observed values of Y_1 and Y_2 to estimate the parameters of the predictive distribution, p($Y_{3,mis}|Y_1,Y_2,\theta_3$). The predictive distribution and the parameters to be estimated will depend on the variable type for Y_3.PROC MI will use either linear regression or predictive mean matching (continuous), logistic regression (binary or ordinal categorical), or the discriminant function method (nominal categorical) to estimate the predictive distribution. For example, if Y_3 is a continuous scale variable, the default predictive distribution is the linear regression of $Y_{3,obs}$ on Y_1, Y_2 with parameters, θ_3 ={β-the vector of linear regression coefficients, and $\sigma^2_3$the residual variance}. To ensure that all sources of variability are reflected in the imputation of $Y_{3,mis}$, the values of the parameters for the predictive distribution, p($Y_{3,mis}|Y_1,Y_2,\theta_3$), are randomly drawn from their estimated posterior distribution, p($\theta_3|Y_1,Y_2$).

Linear Regression

In PROC MI, when the monotone (or FCS) imputation method is specified, the default imputation method for continuous variables is linear regression. Here we will illustrate the P-step and I-step for the monotone missing data pattern. In FCS, the P-step and I-step are similar with minor adaptations to account for the iterative cycles of the algorithm.

Assume that Y_3 in our example is a continuous variate with missing values. The MONOTONE method will first regress the observed values of Y_3 on the values of the more fully observed $\{Y_1,Y_2\}$ using the standard model:

$$Y_3 = \beta_0 + \beta_1 Y_1 + \beta_2 Y_2 + \varepsilon$$

The regression will yield current estimates of the regression parameters, $\hat{\beta} = \left\{\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2\right\}$, and, the residual variance $\hat{\sigma}^2_3$, and also V_3, the inverse of the sum of squares and cross products (SSCP) matrix from the regression of Y_3 on Y_1 and Y_2. These regression estimates define the posterior distribution for the regression model parameters.

Based on the estimated posterior for the regression model parameters, the **I-step** imputes "draws," Y_3*. The first step in this imputation is to draw random parameter values from their joint posterior distribution. First, the value for the residual variance is drawn from its posterior (assuming a non-informative prior):

$$\sigma^2_{*j} = \hat{\sigma}^2_j (n_j - k - 1) / g$$

where :

n_j = count of non-missing values for variable being imputed;

k = the number of parameters (excluding intercept) in the model;

g = a random draw from the central Chi-square distribution, $\chi^2_{n_j-k-1}$

In our example of imputing missing values of Y_3, j is 3 and k=2. Conditional on the drawn value of σ^2_{*j}, draws of the regression parameters are made from their conditional posterior distribution:

$$\beta_* = \hat{\beta} + \sigma_{*j} \sqrt{V}' Z$$

where :

$\sqrt{V}$ = the upper triangle in the Cholesky decomposition (square root) of $V = (X'X)^{-1}$; and

Z = k + 1 dimensional vector of independent normal, N(0,1), variates.

Finally, the actual imputations of missing values of Y_3 are derived from the regression equation:

$$Y_{3*} = \beta_{0*} + \beta_{1*} Y_1 + \beta_{2*} Y_2 + z\sigma_{*3}$$

where :

z = a random draw of a standard normal, N(0,1), deviate.

The algorithm will then turn to the P-step for Y_4, using linear or logistic regression or discriminant classification to define $p(Y_{4,mis}|Y_1, Y_2, Y_{3,obs} Y_{4,obs}, \theta_4)$. Draws from the corresponding predictive distribution for Y_4 will be used to create the imputations, Y_{4*}.

The imputation of this monotone sequence will conclude by applying the process described for $Y_{4,mis}$ to the missing values for the fifth and final variable, $Y_{5,mis}$.

Each step in the sequence is a univariate imputation of a single variable that is conditioned on the more completely observed variables in the monotone pattern.

Predictive Mean Matching

Predictive mean matching (PMM) is an option in PROC MI for the imputation of missing values for continuous variables. The PROC MI predictive mean matching method for imputing continuous variables utilizes the same initial steps

as the linear regression method above; however, the I-step differs. Using our example of imputing a missing value for a continuous Y_3, the draws of the regression parameters are used to construct a regression prediction of Y_3 for each case.

The data set is sorted according to regression predictions of Y3 for both the missing and observed cases. A "neighborhood" for each missing value is defined by the set of observed cases that most closely matches on the regression prediction. Each missing value of Y3 is then imputed by drawing and substituting an observed value that is in the "neighborhood" of its predicted value. The size of the neighborhood used to select the random donor is controlled by the imputer. The default in PROC MI is to select the imputation donor from the nearest K=5 cases. Note that the random draw of an actual observation ensures that the imputation will lie within the range of actual observed values. This algorithm for predictive mean matching imputation for a continuous variable is also an option in the FCS method.

Logistic Regression

The logistic regression method is used in the MI monotone and FCS methods to impute binary or ordinal classification variables. The actual process of imputing missing values under this model follows a P-step, I-step sequence that is similar to that detailed above for the linear regression methods. In the P-step, logistic regression is applied to observed values of the dependent variables (e.g., Y_4 in our five-variable example) and observed predictors (e.g., Y_1, Y_2, Y_3), yielding the fitted logistic model for the probability that the missing value belongs to each category of the binary or ordinal classification variable. To illustrate, assume that Y_4 is a binary variable with values (0,1). The fitted logit model with the DESCENDING option models the logit of the probability that Y_4 =1. The fitted model is:

$$\text{logit}(p(Y_4 = 1)) = \log\left(\frac{p}{1-p}\right) = \hat{\beta}_0 + \hat{\beta}_1 Y_1 + \hat{\beta}_2 Y_2 + \hat{\beta}_3 Y_3$$

The fitted logistic regression model yields estimates of the logistic regression parameters, $\boldsymbol{\beta}=\{\beta_0, \beta_1, \beta_2, \beta_3\}$ and the covariance matrix for parameters **V**. As in linear regression, the posterior distribution of the logistic regression parameters is assumed to be multivariate normal and random draws from this posterior based on the following:

$$\beta_* = \hat{\beta} + \sqrt{V}' Z$$

where :

$\sqrt{V}$ = the upper triangle in the Cholesky decomposition (square root) of the Covariance Matrix for the $\hat{\beta}$; and

Z = k + 1 dimensional vector of independent normal, N(0,1), variates.

To complete the P-step for imputing missing values of the binary variable, the inverse logit transform is computed using the drawn values of the β_*s:

$$p(Y_4 = 1) = \frac{\exp(Y_{j<4}\beta_*)}{1 + \exp(Y_{j<4}\beta_*)}$$

In the I-step, a random number drawn from a uniform (0,1) probability distribution is used to determine the imputed category (e.g., Y_4=1 or Y_4=0 for binary Y_4). To understand how this works, consider the example of a binary measure of self-reported type-2 diabetes status. Assume that the P-step logistic model predicts the probability of a "Yes" response for a missing value to be: p(Yes)=0.12. If the drawn $U(0,1)$ random number for the imputation of the missing observation is u=0.73, since the random drawn is greater than the predicted probability (u >p(Yes)), a value of "No" will be imputed to the case. If, instead, the random number draw was u=0.08, the random number is less than or equal to the predicted probability and a value of "Yes" would be imputed. In applications of the MONOTONE and FCS methods (discussed below), the logistic regression imputation method is the technique used to impute missing values of both binary and ordinal variables.

The process illustrated here for a binary variable is easily extended to the ordinal variables with K>2 levels.

Discriminant Function Method

In major software systems, multiple imputation of missing data for a classification variable, Y_j, with nominal category groups, g=1,…,G (e.g., workforce status), is typically performed using one of two methods: a discriminant function method or a method based on the generalized (multinomial) logit model. Under both the monotone and FCS methods, PROC MI uses the discriminant function method to impute missing values for nominal classification variables. The discriminant function method and the generalized logit approach each employ a P-step that simulates the parameters (group probabilities) of the multinomial posterior distribution for Y. Both methods relate these multinomial class probabilities to a vector of observed covariates, X={X_1, …,X_p}. The discriminant function method requires a more rigid set of

distributional assumptions concerning these covariates. Within each category group, g=1,…,G, the vector of observed covariates, X_g, is assumed to be approximately multivariate normal, and furthermore the variance-covariance matrix of these normally distributed covariates is assumed to be constant across groups, that is, $\Sigma_g=\Sigma$ for all groups.

Given a multivariate normal vector of covariates for a case, the predictive distribution of the missing value for the case is the multinomial distribution with posterior probability of belonging to category, g=1,…,G obtained from the following formula:

$$p(Y_{i,mis} = g \mid x_i) = \frac{\exp(-0.5 \times D_g^2(x_i))}{\sum_c \exp(-0.5 \times D_g^2(x_i))}$$

with

$D_g^2(x_i)$ = the squared distance discriminant function value for case i and group g,

$= (x_i - \mu_{*_g})' \Sigma_*^{-1} (x_i - \mu_{*_g}) - 2\log(q_{*_g})$

where :

$x_i = \{ x_{i1},\ldots,x_{ip}\}$ vector of covariates for case i;

μ_{*g} = draw of posterior mean vector for X in group g;

Σ_*^{-1} = draw of posterior covariance matrix for X (common to all groups); and

q_{*g} = draw of prior probability of group g = 1,...,G membership

Before these multinomial probabilities can be evaluated in PROC MI, values of the parameters Σ_*, μ_{*g}, and q_{*g} must be drawn from the appropriate distributions. The sequence of three "draws" begins withΣ_*, the simulated posterior value of the common MVN variance covariance matrix, Σ. Through the PCOV= option, PROC MI provides two options. The default (also PCOV=POSTERIOR) is to draw the value of Σ_* from its posterior distribution, which under the PROC MI assumption of a noninformative prior is the inverted Wishart distribution:

$p(\Sigma \mid X) \sim W^{-1}(n - G,(n - G)S)$,

where :

n = total sample size;

G = total number of groups in nominal classification variable Y; and

S = the fixed estimate of the pooled covariance matrix for the observed Xs,

$$= \frac{1}{(n - 1)} \sum_{g} (n_g - 1) S_g$$

The option, PCOV=FIXED, specifies that the Σ_* is simply set to the fixed value of S estimated from the observed X values.

The second step is to draw the values of the group-specific mean vectors from their respective posterior distributions. PROC MI assumes a noninformative prior for the values of these group mean vectors. The posterior distribution for each group mean is therefore the normal distribution:

$$p(\mu_g \mid \Sigma, \overline{x}_g) \sim N[\overline{x}_g, (1 / n_g)\Sigma_*]$$

Note that these draws condition on the previously drawn value for Σ_*.

Next, PROC MI computes or draws the q values of the prior probabilities of belonging to group g=1,…,G. The PRIOR= option allows the user to control the specification of the prior distribution for these probabilities. The default option is PRIOR=JEFFRIES, a noninformative Dirichlet prior (not shown).

With the Σ_* μ, and q draws completed, the discriminant function imputation method uses the formula given above to estimate the posterior probabilities of group membership of each missing data case. Across the G categories of the variable, the sum of these posterior probabilities will be 1.0 for each case. The remaining step in the imputation process is to use these probabilities to assign the missing Y value to one of the G nominal groups. As is the case for the logistic regression method, the I-step in the discriminant function method uses a random draw from a $U(0,1)$ distribution. This $U(0,1)$ random number draw is then compared to the accumulation of the g=1,…,G category probabilities from the discriminant classification model to determine the category for the imputed value, Y*.

Propensity Score

PROC MI also offers a propensity score option (Schafer 1999) for performing imputation of missing data. This is a univariate method that was developed for use in very specialized missing data applications. It does not incorporate or preserve associations among the variables in the imputation model, and therefore we feel should not be recommended for MI applications where multivariate analysis is the ultimate goal. For this reason, we encourage analysts to use one of the other options available in PROC MI.

2.3.3 Methods for Arbitrary Missing Data Patterns

Here, we consider the more typical data context where the imputation model is multivariate, includes variables of all types, and has an arbitrary pattern of missing data (Figure 2.3). In such cases of a "messy" pattern of missing data where exact methods do not strictly apply, the authors of multiple imputation software have generally followed one of three general approaches. Each of these three approaches to an arbitrary pattern of missing data are available in PROC MI.

Figure 2.3: Arbitrary Multivariate Missing Data Pattern

	Variables				
Obs	Y1	Y2	Y3	Y4	Y5
1			?	?	
2	?	?		?	?
3	?		?		
4		?			
5	?			?	

The Markov chain Monte Carlo (MCMC) Method: Using an Explicit Multivariate Normal Model and Applying Bayesian Posterior Simulation Methods

A first algorithmic approach to generating imputations for an arbitrary pattern of item missing data is to declare an explicit probability model for the data and a prior distribution for the parameters, $g(\theta)$ (Schafer 1997). The earliest versions of multivariate multiple imputation programs for continuous data (Schafer 1999) assumed a multivariate normal distribution for all variables, $f(y|\theta)$=MVN(μ,Σ) and a noninformative or Jeffries prior distribution for the parameters μ and Σ. In the case of complete data, the posterior distribution p(μ,$\Sigma|Y$) can be derived under Bayes' Rule. However, for an arbitrary pattern of missing data where individual cases are missing different combinations of the variables in Y={Y_1, …,Y_p}, the same posterior—now conditional only on the observed data—is difficult or impossible to derive in a closed form. The PROC MI MCMC full-data imputation method uses an iterative Markov chain Monte Carlo method to simulate draws from the posterior, p$(\mu,\Sigma|Y_{obs})$. For the curious reader, Schafer (1999) provides a detailed description of the MCMC algorithm. Here, we will describe the algorithm in general terms. The MCMC algorithm involves an iterative sequence of paired I-steps and P-steps.

I-Step

At each iteration of the simulation (t=1,…,T), the MCMC algorithm draws

imputations from the current iteration's predictive distribution, $f(Y^{(t+1)}_{mis} | Y_{obs}, \mu^{(t)}, \Sigma^{(t)})$. The imputation proceeds case by case, taking into account the pattern of missing variables for the case. For example, the predictive posterior for a case with the observed/missing pattern of $Y_i=(Y_1, . , Y_3, . , Y_5)$ is different from that for $Y_i=(Y_1, . , . , Y_4, Y_5)$. For efficiency, MCMC uses the SWEEP operator (Goodnight 1979)—a computationally convenient way to estimate linear regression parameters from $\Sigma^{(t)}$—to derive the conditional distributions needed to simulate the predictive posterior for each possible pattern of missing data.

P-Step

After each I-step, the parameter values for the predictive distribution are updated by draws from the completed data posterior, $p(\mu,\Sigma|Y_{obs}, Y^{(t+1)}_{mis})$.

In theory, if the chain of MCMC I-step/P-step pairs is allowed to continue for many iterations, the algorithm will converge so that the imputation draws for the missing values will simulate draws from the true joint posterior, $p(Y_{mis}|\mu,\Sigma, Y_{obs})$. Once a sufficient burn in period of iterations has passed, the $m=1,\ldots,M$ repetitions can be taken as a successive series of systematic draws in the single imputation chain. Another option is to use $m=1,\ldots,M$ MCMC runs in parallel chains and obtain each repetition of the multiple imputation as single draws from each of the independent MCMC chains. The single chain option is the default in SAS, but users may optionally request a multiple chain approach in applications of the MCMC method.

There is no exact test to establish that the MCMC algorithm has in fact converged to the joint posterior distribution. Whether single or multiple MCMC chains are used, the length of the burn in period (or initial iteration cycles) that is required to ensure convergence of the MCMC algorithm will depend on the number of variables and the rates and patterns of missing data. PROC MI defaults to NBITER=200, meaning 200 burn-in iterations. In addition, SAS provides several graphic tools to evaluate the convergence of the MCMC algorithm. Trace plots permit the user to plot the value of posterior estimates of means and variances against the iteration number. These plots should show the posterior mean and variance for the M multiple imputation repetitions converging to a stationary distribution as the number of iterations increases. Ultimately, if convergence is reached, the plotted posterior mean and variance should vary randomly. After a sufficient number of burn-in iterations, the trace plots for the posterior mean and variance should not exhibit any patterns or trends either within or across the traces for the independent MI repetitions. Autocorrelation plots graph the autocorrelation in the values of the posterior

parameters as a function of the "lag" in the number of iterations. Autocorrelation plots should show the lagged autocorrelations decline in value, ultimately varying randomly about zero.

The MCMC algorithm makes the assumption that the underlying variables in the imputation model are distributed as a multivariate normal random variable, $Y \sim MVN(\mu,\Sigma)$. In the case of continuous variables that are highly skewed or otherwise non-normal in distribution, PROC MI currently enables the user to specify transformations (e.g., a natural log transformation for a log normally distributed income measure).

PROC MI also allows the user to force the MCMC assumptions on data of mixed type and then use post-imputation rounding to restore the imputed variables to their original measurement scale. For example, a binary variable imputed as 1.15 would be rounded to "1," while a value of 1.73 would be rounded to "2." We agree with Allison (2005) and do not recommend the use of MCMC with the rounding technique for imputing classification variables. With the availability of the FCS method for imputation of mixed variable types, we advise the user to use imputation methods (regression, logistic regression, discriminant function method) that are directly appropriate to the variable type.

Transform the Arbitrary Missing Data Pattern to a Monotonic Missing Data Structure

A second approach to dealing with arbitrary missing data is to transform the pattern of item missing data to a monotonic pattern by first using simple imputation methods or an MCMC posterior simulation approach to fill in the missing values for variables in the model that have very low rates of item missing data. If all variables in the imputation model are assumed to be continuous, the MCMC method with the MONOTONE option can be used to implement this approach. As previously described, imputation of a true monotonic pattern of item missing data is greatly simplified—reduced to a sequence of imputations for single variables. This MCMC monotone approach works best when the generalized pattern of missing data is dominated by missing data for one or two variables.

Consider the missing data for the variables selected for analysis of a Health and Retirement Study (HRS) survey question on serious falls in the past two years shown in Table 2.2. The generalized pattern of missing data is dominated by the missing data on falls (4.5%), with a lesser rate for weight (1.4%), virtually no missing data for arthritis (0.2%), and complete data for age and gender. Under the MCMC MONOTONE option, PROC MI uses MCMC to impute the minimum number of missing data values to transform the problem to a monotone missing

pattern. Noniterative monotone regression or predicted mean matching imputations can then be used to sequentially fill in the remaining missing values. This technique will be illustrated through a worked example in Chapter 6.

Table 2.2: Item Missing Data Rates for Variables Included in the 2006 HRS Falls Model (n = 11,731 Eligible Respondents Age 65 and Older)

Variable	Falls	Age	Gender	Arthritis	Weight
% Missing	4.50%	0.00%	0.00%	0.20%	1.40%

FCS, Sequential Regression, and Chained Regressions

The FCS approach is the third alternative to multiply impute arbitrary missing data for large mixed sets of continuous, nominal, ordinal, count, and semicontinuous variables. The FCS method is also labeled the sequential regression algorithm (Raghunathan et al., 2001) or the "chained equations" approach (Carlin, Galati, and Royston 2008; Royston 2005; van Buuren, Boshuizen, and Knook 1999). Each of these algorithms is based on an iterative algorithm. Each iteration (t=1,…,T) of the algorithm moves one by one through the sequence of variables in the imputation model, for example, Y= $\{Y_1, Y_2, Y_3, Y_4, Y_5\}$ as illustrated in Figure 2.3. At each iteration and for each variable, there is a P-step and an I-step. In the P-step, the current (iteration t) values of the observed and imputed values for the imputation model variables are used to derive the predictive distribution of the missing values for the target variable. To model the conditional predictive distribution of individual Y_k, $f(\hat{Y}_k^{(t)} / \hat{Y}_{j\neq k}^{(t)}, \theta^{(t)})$, PROC MI uses the same regression or discriminant function methods described above for the monotone missing data patterns. That is, linear regression or the regression-based predicted mean matching (PMM) approach is used to impute missing values for continuous variables, ordinal logistic regression to generate imputations for binary or ordinal classification variables, and the discriminant function method for nominal classification variables. Updated imputations, $\hat{Y}_k^{(t)}$, are then generated by stochastic draws from the predictive distribution defined by the updated regression model. When the last variable in the sequence has been imputed, the algorithm cycles again through each variable, repeating the chain of regression estimation and imputation draw steps. The burn-in iteration of the cycles continues until the user-defined algorithm convergence or system default value is met (i.e., NBITER=x iterations or the default of NBITER=10, as specified in PROC MI).

Under an explicitly defined imputation model, $f(Y|\theta)$, and a suitable prior distribution, $g(\theta)$, for the distributional parameters, this FCS algorithm will, in theory, converge to the joint posterior distribution. Since the sequential

regression method never explicitly defines $f(Y|\theta)$, the assumption must be made that the posterior distribution does exist and that the final imputations generated by the iterative algorithm do represent draws from an actual, albeit unknown, joint posterior distribution. Although the exact theoretical properties of the resulting FCS imputations remain somewhat of an unknown, in most applications the sequential regression approach does converge to a stable joint distribution. The multivariate imputations generated by the algorithm show reasonable distributional properties and have empirically been shown to produce results comparable to those for the EM algorithm and exact methods of Bayesian posterior simulation (Heeringa, Little, and Raghunathan 2002).

2.4 Step 2—Analysis of the MI Completed Data Sets

Although there is no requirement to separate the MI analysis step from the estimation and inference step, for most analyses SAS has chosen to split these two steps. This provides the user maximum flexibility to use the wide array of existing SAS standard and SURVEY procedures to conduct analysis. The input to the analysis step is the concatenated or "stacked" data set produced by PROC MI. The analyst then specifies standard or SURVEY procedure statements with a BY _IMPUTATION_ statement to independently analyze each of the m=1,…,M repetition data sets.

A critical activity at this analysis step that will be carefully covered in examples in following chapters is to specify an appropriately structured output data set containing the estimated statistics and their standard errors from each of the M repetitions of the analysis. The output data set of the estimated statistics and standard errors will be the required input for PROC MIANALYZE (step 3). Chapters 4 through 8 provide examples of how to ensure that output of parameters estimates and variance/covariance matrices from SAS analysis procedures is correctly formatted for input to PROC MIANALYZE.

2.5 Step 3—Estimation and Inference for Multiply Imputed Data Sets

Once SAS procedures have been used to individually analyze the multiply imputed data sets and save the results in an output file, the next step in a complete MI analysis is to compute multiple imputation estimates of the descriptive statistics or model parameters and the variance of these MI estimates. The MI estimates and standard errors can then be used to construct confidence intervals for population quantities or model parameters.

Many of the major software systems include a pair of programs to conduct multiple imputation analysis—one program to perform the multiple imputation of item missing data and a second to perform MI estimation and inference. In SAS, as previously discussed, this pair of programs is PROC MI and PROC MIANALYZE. Although SAS supports coordinated processing of both the imputation and estimation/inference phases of an MI analysis, provided some care is taken in choosing the imputation model, the imputation and estimation phases can be performed separately by different persons (i.e., imputation by data producers, analysis by data users) or using separate programs (e.g., imputations in IVEware [Raghunathan, Solenberger, and Van Hoewyk 2002], analysis of completed data repetitions using SAS procedures and MI estimation and inference in PROC MIANALYZE).

2.5.1 Multiple Imputation—Estimators and Variances for Descriptive Statistics and Model Parameters

Following Rubin (1987), multiple imputation estimates of descriptive statistics and model parameters are computed by simply averaging the estimates from the m=1,…,M independent repetitions of the imputation algorithm:

$$\overline{\theta} = \frac{1}{M}\sum_{m=1}^{M}\hat{\theta}_m$$

where $\hat{\theta}_m$ = estimate of θ from the completed data set m=1,...,M.

Rubin (1987) also proves that the corresponding multiple imputation variance for $\overline{\theta}$ is estimated as a function of the average of the estimated sampling variance for each repetition estimate (termed the "within" component) and a between imputation variance component that captures the imputation variability over the repetitions of the imputation process.

The within-imputation variance component is computed as the average of the estimated variances for the $\hat{\theta}_m$ from the m=1,…,M completed data set analyses:

$$\overline{W} = \text{Within - imputation variance} = \frac{1}{M}\sum_{m=1}^{M}\hat{W}_m = \frac{1}{M}\sum_{m=1}^{M} var(\hat{\theta}_m);$$

where : $\text{var}(\hat{\theta}_m)$ is the estimate of the variance of $\hat{\theta}_m$ for MI repetition $m = 1,...,M$.

The between-imputation component of the MI variance is estimated using the formula:

$$B = \text{Between - imputation variance} = \frac{1}{(M-1)} \sum_{m=1}^{M} (\hat{\theta}_m - \bar{\theta})^2 .$$

The total variance of the MI estimate of θ is then computed using Rubin's combining formula:

$$\text{var}(\bar{\theta}) = \bar{W} + \left(\frac{M+1}{M} \right) \times B$$

In applications to complex sample survey data, the analysis step to generate the repetition estimates and standard errors is performed using SAS SURVEY procedures. The estimated sampling variance of each repetition estimate, $\text{var}(\hat{\theta}_m)$, is computed using an appropriate Taylor Series or replication variance estimator.

2.5.2 Multiple Imputation—Confidence Intervals

Based on the MI estimates of the parameter of interest and its variance, Rubin (1987) shows that the statistic:

$$t = \frac{(\theta - \bar{\theta})}{\sqrt{T}}$$

$$\text{where : } T = \text{var}(\bar{\theta})$$

is approximately distributed as a Student *t* with degrees of freedom equal to:

$$\nu_{mi} = (M-1)\left\{1 + r^{-1}\right\}^2$$

where :

$$r = \frac{(1 + M^{-1}) \times B}{\bar{W}}$$

The factor, *r*, in this expression is labeled the "increase in variance due to nonresponse." When there is no missing data for the variables required to estimate θ, both B, the between-imputation variance, and *r* will be zero.

A related measure of the impact of missing data on the variance of parameter estimates is λ, or the "fraction of missing information (FMI)" about θ. No matter how good the imputation model and methods are, the imputed data set will never achieve the same level of "statistical information" that would have existed in the complete, fully observed data. Based on the estimates of the within and between components of the multiple imputation variance estimator, the "fraction of missing information" is computed in SAS as:

$$\hat{\lambda} = \frac{r + 2(v_{mi} + 3)}{r + 1}$$

This statistic measures the proportion of information lost due to imputation relative to the full information that would be present if all data values for all cases had actually been observed. Since the fraction of missing information is a function of the within and between variance of specific estimates, it is specific to that estimate. In a single regression model, the fraction of missing information for one estimated parameter, $\hat{\beta}_j$, may differ from that for another, $\hat{\beta}_k$. If missing data for a single variable, *y*, were imputed based only on the distribution of the observed values of that same variable, the fraction of missing information for the estimated mean, $\overline{y}$, would equal the missing data rate for *y*. However, more generally, when the model for imputing for *y* conditions on observed values of other related variables, the imputation borrows strength from the multivariate relationships and the fraction of information lost will be reduced such that $0 < \hat{\lambda} <$ missing data rate for y

The unique feature of the MI confidence interval for population statistics or parameter values is the degrees of freedom determination for the Student *t* distribution. In practical problems where the degrees of freedom for a complete data analysis, $v_{0,}$ is small and the proportion of missing data is also small, the computed MI degrees of freedom approximation, v_{mi}, may be greater than v_0—a theoretical impossibility. This situation can occur in practice when the number of independent sample observations, *n*, is small or in complex sample data designs where the effective complete data degrees of freedom is determined by the numbers of primary strata and clusters and not simply by the number of unique data observations. To ensure that the MI degrees of freedom are correctly bounded and better approximate the true degrees of freedom for the Student *t* reference distribution, SAS incorporates the "small sample" method of Barnard and Rubin (1999) to determine the degrees of freedom for constructing the confidence interval:

$$v_{mi}^{*} = \left[\frac{1}{v_{mi}} + \frac{1}{\hat{v}_{obs}}\right]^{-1}$$

$$where:$$
$$\hat{v}_{obs} = (1-\gamma)v_0(v_0+1)/(v_0+3);$$

$$v_o = \text{complete data degrees of freedom;}$$
$$\gamma = (1 + M^{-1})B/T.$$

The default value used by PROC MI for the complete case degrees parameter is

v_0=infinite. Under this default, it is clear that the Barnard-Rubin small sample approximation for degrees of freedom reduces to the original MI degrees of freedom, v_{mi}. However, the EDF option for the MI procedure permits the analyst to override the default and specify a finite, known value for the complete data degrees of freedom. As a case in point, the complete case degrees of freedom for data collected under a stratified, clustered complex sample design is generally approximately as v_0=(# PSUs – # Strata). In later chapters, examples based on the NHANES, HRS, or NCS-R survey data sets will employ the EDF option on the PROC MIANALYZE command line to set the complete case degrees of freedom to the appropriate approximation for their complex sample design (e.g., EDF=16 for NHANES 2009–2010).

Therefore, MI confidence intervals for descriptive population statistics or single parameters in models are constructed from the multiple imputation estimate, its standard error, and a critical value from the Student *t* distribution (Rubin and Schenker 1986) with v_{mi}^{*} degrees of freedom:

$$CI_{(1-\alpha)}(\theta) = \overline{\theta} \pm t_{v_{mi}^{*},1-\alpha/2} \cdot \sqrt{T}$$

where :

v_{mi}^{*} = the Barnard-Rubin estimate of the MI degrees of freedom.

MI confidence intervals are routinely reported for individual population statistics or model parameters computed by PROC MI or PROC MIANALYZE. Simulation studies have demonstrated that for large sample sizes, this MI confidence interval provides true coverage of the population value that is very close to the nominal (1-a)% coverage level.

2.6 MI Procedures for Multivariate Inference

2.6.1 Multiple Parameter Hypothesis Tests

In regression and other multivariate statistical procedures, it is common to test multiple parameter hypotheses of the form, H0: β={β1,…,βq}={0,…,0}. MI procedures for making multivariate inferences of this form (i.e., the MULT option in PROC MIANALYZE) are a direct extension of those for constructing a confidence interval or *t* test statistics for single parameters. The MI estimates for the vector of parameters is computed by averaging the *m*=1,…,*M* repetition estimates of the multiparameter vector:

$$\bar{\theta} = \frac{1}{M}\sum_{m=1}^{M}\hat{\theta}_m$$

$\bar{W}$ is a q x q within imputation covariance matrix for the q x 1 parameter vector computed by averaging the covariance matrices from the m=1,…,M repetitions:

$$\bar{W} = \frac{1}{M}\sum_{m=1}^{M}\hat{W}_m$$

The between-imputation covariance matrix is defined as:

$$B = \left(\frac{1}{M-1}\right)\sum_{m=1}^{M}(\hat{\theta}_m - \bar{\theta})(\hat{\theta}_m - \bar{\theta})'$$

T, or the total covariance matrix, is defined as:

$$T = \bar{W} + \left(1 + \tfrac{1}{M}\right)B$$

Unfortunately, in many practical applications where the number (q) of multivariate parameters in θ is large relative to the number of MI repetitions (M), estimates of the between-imputation covariance matrix B are unstable. For this reason, PROC MIANALYZE uses a more stable estimate of T:

$$\tilde{T} = (1+r)\bar{W}$$

where :

$$r = (1 + 1/M) \times trace(B\bar{W}^{-1})/q$$

= the average of the relative increase in variance due to missing data for the $j = 1,...,q$ parameters in θ.

To test the multivariate null hypotheses, $H0: \bar{\theta} = 0$, a special Wald Test F statistic is used:

$$F = (\bar{\theta} - \theta_0)'\tilde{T}^{-1}(\bar{\theta} - \theta_0)/q$$

Under the null hypothesis, this test statistic is referred to a central F distribution with q and

$$v_1 = \frac{1}{2}(p+1)(M-1)(1 + 1/r)^2$$

degrees of freedom, where:

$$r = (1 + \frac{1}{M}) \times trace(B\bar{W}^{-1})/q$$

For situations where M is larger (i.e., $q(M\text{-}1) > 4$), PROC MIANALYZE uses an alternate expression for the denominator degrees of freedom that was

proposed by Li, Raghunathan, and Rubin (1991):

$$v_2 = r + (t-4)\left[1 + \frac{1}{r} \times \left(1 - \frac{2}{t}\right)\right]^2$$

2.6.2 Tests of Linear Hypotheses

In the various forms of "regression analysis," linear tests of hypotheses about the parameters can be expressed in matrix notation as: $H_0 : L\beta = c$ where β is a vector of regression parameters, L is a matrix of coefficients that define the test, and *c* is a vector of constants. In many cases, the hypothesis of interest sets *c*=0. To test such linear hypotheses in PROC MIANALYZE, the standard TEST statement is employed with the MULT option to generate a Wald *F* test statistic to test the linear hypotheses.

2.7 How Many Multiple Imputation Repetitions Are Needed?

Theoretically, the statistical efficiency of multiple imputation methods is maximized when the number of repetitions is infinite, $M=\infty$. Fortunately, the same theory tells us that if we make the practical choice of using only a modest, finite number of repetitions (e.g., *M*=5, 10, or 20), that loss of efficiency compared to the theoretical maximum is relatively small. A measure of relative efficiency reported in SAS outputs from MI analysis is:

$$\mathrm{RE} = \left(1 + \frac{\lambda}{M}\right)^{-1}$$

where :

λ is the fraction of missing information;

and *M* is the number of MI repetitions

If the rates of missing data and therefore fraction of missing information are modest (< 20%), MI analyses based on as few as *M*=5 or *M*=10 repetitions will achieve > 96% of the maximum statistical efficiency. If the fraction of missing information is high (30% to 50%), analysts are advised to specify *M*=20 or *M*=30 to maintain a minimum relative efficiency of 95% or greater. Historically, the rule of thumb for most practical applications of MI was to use *M*=5, and this is the current default in PROC MI. Recent research has shown benefits in using larger numbers of repetitions to achieve better nominal coverage for MI confidence intervals or nominal power levels for MI hypothesis tests. Van Buuren (2012) suggests a practical "stepped" approach in which all initial MI analyses are conducted using *M*=5 repetitions. When the analyses have reached the point where a final model has been identified, the imputation can be repeated

with M=30 or M=50 repetitions to ensure that the final results do not suffer from a relative efficiency loss. In offering this advice, the author also notes that this last confirmation step is unlikely to alter any conclusions that would have been drawn based on the M=5 repetition analyses.

2.8 Summary

This chapter has provided a theoretical foundation underlying the multiple imputation process. With this foundation in place, later chapters cover details of PROC MI and PROC MIANALYZE syntax along with practical applications of the MI process in SAS.

Chapter 3: Preparation for Multiple Imputation

3.1 Planning the Imputation Session

Chapter 3 provides examples of planning and executing a multiple imputation session. The multiple imputation approach introduced in previous chapters is considered in a step-by-step manner, followed by a simple introductory example of the entire process in action. Our examples in this chapter are deliberately simple to provide a foundation from which more complex multiple imputation projects can be planned and executed.

3.2 Choosing the Variables to Include in a Multiple Imputation

From Section 2.2.1, we know that the first step in a multiple imputation analysis is to identify the "Imputation Model" that we will use in the PROC MI step. Strictly speaking, this requires us to do two things: 1) identify the set of variables we wish to include in the imputation run; and 2) make a best possible choice or assumption about the joint distribution of the variables that have been chosen. Here we will focus on choosing the variables to include in the imputation run. The requirement to explicitly or implicitly choose a distributional assumption (e.g., multivariate normal) for the selected variables will depend on the variable types we choose to impute and the pattern of

missing data. Since it is much more straightforward to develop that aspect of imputation model identification using real data and examples, we defer a detailed discussion of this topic to later chapters.

The selection of the variables to be used in the imputation of missing data can be a complex task. If you are new to multiple imputation analysis this may seem more like "art" than "science." Do not let this intimidate you. While it is true that you may gain additional insight and sophistication with increasing experience, if you follow a few basic rules, the SAS procedures described in this book will enable you to perform multiple imputation and analysis that is statistically robust. To start you out, here again are three simple rules (van Buuren 2012):

1. Include all of the key analysis variables, regardless of whether they have missing data;
2. Include other variables that are correlated or associated with the variables you intend to analyze (regardless of whether these variables will ultimately be included in your analysis);
3. Include variables that predict item missing data on the analysis variables (again regardless of whether these variables will ultimately enter your analysis).

Also, remember the advice from Chapter 2, "When in doubt including more variables in the imputation model is better."

In practice, establishing correlation/association with the imputed variable is a simple matter of performing a correlation/regression analysis or a test of association among the variables. Evaluating "related to missingness" can done by regressing (e.g., logit/proibit) a binary indicator of missingness for a variable (1=observed, 0=missing) on the candidate set of covariates and selecting those that are important predictors of the missing outcome. Refer to Chapter 2, Section 2.1, for a discussion of choosing the imputation model variables intelligently or Shafer (1997) or Allison (2001) for more details on this topic.

As an introductory example, consider the data set, **teen_behavior**, which includes a number of variables pertaining to adolescent behavior. We first use PROC CONTENTS to examine variables and number of observations in the data set.

```
proc contents data=teen_behavior;
run;
```

Output 3.1: Partial Output from CONTENTS Listing of the Teen_Behavior Data Set

Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
6	FamilyIncomePastYear	Num	8	BEST8.	F8.	Family Income Past Year
1	ID	Num	8	BEST8.	F8.	ID
4	NumberDrinksPast30Days	Num	8	BEST8.	F8.	# Drinks Past 30 Days
5	NumberDrugsPast30Days	Num	8	BEST8.	F8.	# Drugs Past 30 Days
2	PreTeenIntercourse	Num	8	BEST8.	F8.	Pre-Teen Intercourse
3	STDPast12Months	Num	8	BEST8.	F8.	STD Past 12 Months
7	age	Num	8	BEST8.	F8.	Age

Output 3.1 shows the variables are all Type= Num (numeric) with n=15. For the sake of simplicity and ease of inspection, we use small data sets in this chapter but turn to larger data sets in later chapters. The description of each variable is as follows:

Family Income during the Past Year (FAMILYINCOMEPASTYEAR): continuous numeric variable representing family income during the past year

Case Identifier (ID): continuous numeric variable used as a case identifier

Number of Drinks Past 30 Days (NUMBERDRINKSPAST30DAYS): continuous numeric variable with a count of number of drinks during past 30 days

Number of Drugs Used Past 30 Days (NUMBERDRUGSPAST30DAYS): with a count of number of times used drugs during past 30 days

Sexual Intercourse Prior to Age 13 (PRETEENINTERCOURSE): numeric binary variable indicating if the respondent had sexual intercourse prior to age 13

Sexually Transmitted Disease Past 12 Months (STDPAST12MONTHS): numeric binary indicator indicating if the respondent had sexually transmitted disease during past 12 months

Age Interviewed (AGE): continuous numeric variable representing age at interview

As a prelude to the later sections of this book, we introduce a key method to

examine the missing data pattern and group means: PROC MI with NIMPUTE=0 and a SIMPLE option. This approach to exploring missing data patterns and rates is used extensively because it provides a concise grid display of the missing data pattern, the amount of missing data for each variable and group as well as group means, and univariate and correlation statistics for the specified variables. This example uses a CLASS statement with the FCS statement to illustrate how to incorporate categorical variables such as indicators of having sexual intercourse prior to age 13 and having a sexually transmitted disease during the past 12 months.

```
proc mi data=teen_behavior nimpute=0 simple;
 class preteenintercourse stdpast12months;
 fcs;
 var Id FamilyIncomePastYear NumberDrinksPast30Days
NumberDrugsPast30Days
  PreTeenIntercourse STDPast12Months age;
run;
```

Output 3.2: Missing Data Pattern for Teen Behavior Data Set

Missing Data Patterns

Group	ID	FamilyIncomePastYear	NumberDrinksPast30Days	NumberDrugsPast30Days	PreTeenIntercourse	STDPast12Months	age	Freq	Percent	Group Means				
										ID	FamilyIncomePastYear	NumberDrinksPast30Days	NumberDrugsPast30Days	age
1	X	X	X	X	X	X	X	10	66.67	7.500000	64148	10.100000	5.300000	15.500000
2	X	X	X	X	.	X	X	3	20.00	6.666667	44803	14.666667	2.000000	16.333333
3	X	.	X	X	X	X	X	2	13.33	10.500000	.	11.000000	5.500000	16.000000

Output 3.2 shows an arbitrary missing data pattern where PRETEENINTERCOURSE and FAMILYINCOMEPASTYEAR both have some missing data.

Output 3.3: Descriptive Statistics for the Teen Behavior Data Set

Univariate Statistics							
						Missing Values	
Variable	N	Mean	Std Dev	Minimum	Maximum	Count	Percent
ID	15	8.00000	4.47214	1.00000	15.00000	0	0.00
FamilyIncomePastYear	13	59683	27257	23400	99000	2	13.33
NumberDrinksPast30Days	15	11.13333	8.79015	0	30.00000	0	0.00
NumberDrugsPast30Days	15	4.66667	4.02965	0	13.00000	0	0.00
age	15	15.73333	1.57963	13.00000	18.00000	0	0.00

Pairwise Correlations					
	ID	FamilyIncomePastYear	NumberDrinksPast30Days	NumberDrugsPast30Days	age
ID	1.000000000	0.294723434	0.221677037	0.206107012	0.101111597
FamilyIncomePastYear	0.294723434	1.000000000	-0.378103192	0.174529629	-0.673695185
NumberDrinksPast30Days	0.221677037	-0.378103192	1.000000000	-0.157962843	0.738367633
NumberDrugsPast30Days	0.206107012	0.174529629	-0.157962843	1.000000000	-0.048626221
age	0.101111597	-0.673695185	0.738367633	-0.048626221	1.000000000

In Output 3.3, we see the result of specifying the SIMPLE option on the PROC MI statement. Note that the variables PRETEENINTERCOURSE and STDPAST12MONTHS are both coded as binary indicators with values of 0 or 1and are not treated as continuous variables in this exploratory analysis. The output includes standard univariate statistics as well as a count and percent of missing values for the continuous variables in the example data set. This option also produces estimates of pairwise correlations for the continuous variables.

For the imputation step, the binary indicator variables would be declared as classification variables in the CLASS statement, as demonstrated above. Our data exploration shows that two variables, PRETEENINTERCOURSE and FAMILYINCOMEPASTYEAR, have missing values that need to be imputed. The remaining fully observed variables (with the exception of the case ID) will be included in the multivariate imputation model.

3.3 Amount and Pattern of Missing Data

Evaluation of the amount and pattern of missing data is an important step in our multiple imputation. We briefly introduced this step in Section 3.2 but go into more detail here. This exploratory step will help to determine if the missing data problem can be addressed using PROC MI options for imputing monotone missing data (see Section 2.3.2) or requires one of the MCMC or FCS approaches that are appropriate for imputation of arbitrary patterns of missing data (see Section 2.3.3). Recall from Chapter 2 that in cases where the missing data pattern is arbitrary but the amount of missing data for all but a few variables is small, it is possible in SAS to use the MCMC method and

MONOTONE option to convert the problem to a monotone pattern and then apply the MONOTONE procedure to complete the imputation for the remaining variables.

For now, let's not worry about the exact multiple imputation method that we will use. We will cover that in detail in later chapters and examples. Let's focus here on how to analyze the basic rates and pattern of missingness in our key variables. In practice, these missing data characteristics can be described and/or visualized through study of printouts from PROC PRINT (assuming the data set is small), frequency tables from PROC FREQ (best for classification variables), distribution analysis using PROC MEANS/PROC UNIVARIATE (best for continuous numeric variables), or through the use of PROC MI with the NIMPUTE=0 option.

As previously stated, the latter approach, using PROC MI and NIMPUTE=0 along with selected options, is a good choice for most situations as it provides a concise missing data pattern grid for all numeric variables in the data set. The grid displays the missing data pattern in a manner that allows easy evaluation of the pattern, that is, monotone, nearly monotone, or fully arbitrary. It also provides summary statistics such as variable means and frequency counts for the cases assigned to each unique grouping of cases (row of the grid) defined by patterns of observed and missing values. With use of the SIMPLE option we just introduced, you can also obtain overall univariate statistics such as the mean, standard deviation, minimum, and maximum of each variable.

For example, consider the data set, **sample**, created in the data step using the code below. The variable ID is used solely as a case identifier, HEARTBEAT represents number of heart beats per minute, and AGE contains age measured in years.

```
data sample;
 input id heartbeat age;
 datalines;
  1    32    66
  2    99    68
  3    47    62
  4    43    93
  5    24    100
  6    86    58
  7    40     .
  8    67    38
  9    26     .
 10    45    20
 11    47    58
```

```
 12     83     93
 13     41      .
 14     55     57
 15     51     97
 16     77     80
 17     80     82
 18     63     38
 19     59     26
 20     57     57
 ;
run;
```

Let's examine the amount and pattern of missing data using both PROC PRINT and PROC MI (with the NIMPUTE=0 option).

```
proc print data=sample noobs;
run;
```

Output 3.4: List Output from PROC PRINT

id	heartbeat	age
1	32	66
2	99	68
3	47	62
4	43	93
5	24	100
6	86	58
7	40	.
8	67	38
9	26	.
10	45	20
11	47	58
12	83	93
13	41	.
14	55	57
15	51	97
16	77	80
17	80	82
18	63	38
19	59	26
20	57	57

Output 3.4 shows that both variables, ID and HEARTBEAT, have fully observed data while AGE has missing data for 3 of the 20 records. Given the small size of

the data set, it is easy to examine the missing data pattern visually and determine that this is a monotone missing data pattern with 15% missing data on AGE. Alternatively, we can use PROC MI with the NIMPUTE=0 option to conveniently obtain information on missing data amounts and patterns (without actually imputing any data at this point). Note that any classification variables would be treated as continuous since by default PROC MI assumes the MCMC method will be used when actual imputations are eventually created. (In this example, all variables are continuous.)

```
proc mi data=sample nimpute=0;
run;
```

Output 3.5: Results from PROC MI with NIMPUTE=0 Option

Model Information	
Data Set	WORK.SAMPLE
Method	MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	0
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	215561001

Missing Data Patterns

Group	id	heartbeat	age	Freq	Percent	Group Means		
						id	heartbeat	age
1	X	X	X	17	85.00	10.647059	59.705882	64.294118
2	X	X	.	3	15.00	9.666667	35.666667	.

Output 3.5 reiterates that ID and HEARTBEAT have fully observed data with missing data for AGE on 3 of 20 records in this data set. The additional information in the "Model Information" output header simply lists the basic information about the various PROC MI settings including "Number of

Imputations" set to zero (NIMPUTE=0). The other settings listed here will be discussed in more detail in later chapters.

The "Missing Data Patterns" grid shows two groups of data patterns: Group=1, which consists of those with fully observed data, indicated by an X, for each of the three variables; and Group=2, which indicates complete data on ID and HEARTBEAT but missing data, indicated by a '.', for AGE. The remainder of the table provides summary statistics such as Frequency and Percent of cases for the data pattern and pattern-specific mean values for each variable. Note that the mean value for observed Group 1 values of HEARTBEAT is 59.71 beats/minute, while the mean for Group 2 is a much lower value of 35.67. This missing data pattern is considered univariate monotone.

3.4 Types of Variables to Be Imputed

Once the variables to be included in the imputation model and amount and pattern of missing data have been determined, the type and characteristics of the variable(s) to be imputed are addressed. Variable type in this context means either numeric or character. The variable type can be determined from PROC CONTENTS output. Furthermore, each variable that requires imputation can be characterized as continuous (numeric) or classification (either numeric or character). Note: the MI procedure allows use of either character or numeric classification variables, but continuous variables must be numeric. Determination of variable characteristics/type impacts the selection of an imputation method and model to be used in the imputation.

To illustrate, consider the data set, **income**. We use PROC PRINT to produce a list of this simple data set.

```
proc print noobs data=income;
run;
```

Output 3.6: List Print of Income Data Set

ID	AGE	GENDER	INCOME
1	32	F	10000
2	56	M	20000
3	73		38765
4	25	F	32456
5	76	M	76897
6	49	M	32454
7	41		56453
8	32	F	101345
9	65	F	10356
10	56	F	76547

In Output 3.6, we see that the **income** data set includes four variables: ID=case identifier, AGE=age, GENDER=sex of respondent, and INCOME=respondent annual income. All but GENDER are numeric variables. ID serves as the unique case identifier, AGE and INCOME are fully observed and continuous, and GENDER is a binary, character variable with missing data for 2 of 10 records (represented by the default blank space). For a small data set with a limited number of records and variables, the printout approach is sufficient, but for larger data sets a more general set of tools is required.

Use of PROC CONTENTS to determine the SAS variable type followed by distribution analyses using PROC MEANS for numeric variables and PROC FREQ for classification variables typically serves us well for general data exploration. Let's apply these techniques to the **income** data set.

```
proc contents data=income;
run;
```

Output 3.7: Selected Output from PROC CONTENTS

Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
2	AGE	Num	8	BEST8.	F8.	Age
3	GENDER	Char	8	$8.	$8.	Gender
1	ID	Num	8	BEST8.	F8.	ID
4	INCOME	Num	8	BEST8.	F8.	Income

Output 3.7 illustrates partial output from PROC CONTENTS, including the position, name, type, length, format, and informats for each variable in the data set. In the **income** data set, AGE, GENDER, and INCOME are Type=Numeric

and GENDER is Type=Character.

For examination of variable distributions, PROC MEANS (for continuous variables) and PROC FREQ (for classification variables) are presented. Note that we include the options NMISS (PROC MEANS) and MISSING (on the PROC FREQ TABLES statement) to display the count of missing observations for each variable.

```
proc means data=income n nmiss mean min max;
run;
```

Output 3.8: Means Analysis of Continuous Variables from Income Data Set

Variable	N	N Miss	Mean	Minimum	Maximum
AGE	10	0	50.5000000	25.0000000	76.0000000
INCOME	10	0	45527.30	10000.00	101345.00
ID	10	0	5.5000000	1.0000000	10.0000000

Output 3.8 provides details about each continuous variable, including the number of missing values (N Miss) as well as the Mean, Minimum, and Maximum of the nonmissing values. Here, we note no missing data on ID, AGE, or INCOME.

```
proc freq data=income;
tables gender / missing;
run;
```

Output 3.9: Frequency Table Analysis of Classification Variables from Income Data Set

GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
	2	20.00	2	20.00
F	5	50.00	7	70.00
M	3	30.00	10	100.00

In Output 3.9, the frequency table for GENDER includes missing data represented by a blank space (the SAS default for character variables), “F” for female, and “M” for males. The two cases with missing data on GENDER represent 20% of the total sample. Outputs 3.6-3.9 provide key information such as variable type/characteristics, data values, amount of missing and observed data, and characteristics of variables available for imputation.

Use of PROC MI with a CLASS, VAR, and FCS statement along with NIMPUTE=0 will provide the missing data information/pattern for all variables

in this data set, including the character classification variable GENDER.

```
proc mi data=income nimpute=0;
 class gender;
 fcs;
 var id age income gender;
run;
```

Output 3.10 Missing Data Patterns in the Income Data Set

Missing Data Patterns

Group	ID	AGE	INCOME	GENDER	Freq	Percent	Group Means		
							ID	AGE	INCOME
1	X	X	X	X	8	80.00	5.625000	48.875000	45007
2	X	X	X	.	2	20.00	5.000000	57.000000	47609

Output 3.10 includes FCS model specification information, but given that no imputation is performed, these details are noted but left in the background. Had an imputation been executed, the FCS approach would have used the discriminant function method for imputing the missing values of the classification variable GENDER.

3.5 Imputation Methods

Thus far, we have considered the variables to be used in the imputation model, the amount and pattern of missing data, and the type of variables to be imputed. An important and related concern is what imputation method to use. Because the imputation methods available in SAS v9.4 were discussed in detail in Chapter 2, we will not repeat this information here. Practical guidance on choosing the method that is best suited to your imputation model will be provided for each application presented in later chapters.

3.6 Number of Imputations (MI Repetitions)

The number of MI repetitions needed is a common question. Early research showed that relative to the theoretical optimum where the number of independent MI repetitions is infinite (M=∞), high levels of relative efficiency (RE) could be achieved with as few as *M*=5 or *M*=10 replications. However, the statistic RE reported by SAS measures the proportion of information "recovered" (relative to the optimum) by the multiple imputation repetitions. It is a function of the size of the between (B) and within (W) imputation components of variance. As a brief reminder, in multiple imputation perfect efficiency of 1.0 can only be achieved with an infinite number of imputation repetitions, which is obviously

not possible in practice. In general, as few as 5 to 10 imputations are needed for an acceptable relative efficiency (Rubin 1987). The RE formula is repeated here:

$$RE = (1 + \frac{\lambda}{M})^{-1}$$

where λ is rate of missing information and M is the number of imputations.

For example, reconsider the missing data pattern grid from the **sample** data created in a previous data step.

```
proc mi data=sample nimpute=0;
run;
```

Output 3.11: Missing Data Patterns for the Sample Data Set

Missing Data Patterns								
						Group Means		
Group	id	heartbeat	age	Freq	Percent	id	heartbeat	age
1	X	X	X	17	85.00	10.647059	59.705882	64.294118
2	X	X	.	3	15.00	9.666667	35.666667	.

We observe that AGE has 15% missing data (3 of 20 records). Based on Table 61.7 from the PROC MI documentation, with M=5, relative efficiency of about .97 would be achieved and increasing the number of imputations will result in only modest incremental gains in relative efficiency. (Please refer to Table 61.7 in the PROC MI documentation for an overview of common M and rates of missing information.) Bear in mind that the rate of missing information is generally not equal to the missing data rate, as the missing information is a function of the within and between imputation variance for the imputation of AGE.

More recently, research into the relationship between the number of MI repetitions, M, and the achieved "coverage" of the 95% MI confidence intervals for parameter estimates has led to a recommendation to use larger numbers of MI repetitions to ensure that inferences remain unbiased for the target parameters. For example, Allison (http://www.statisticalhorizons.com/more-imputations) and Bodner (2008) suggest that a higher number of imputations will often result in more stable and accurate SEs, CIs, and p values. Given that the only penalty to requesting a higher number of replications in a SAS MI analysis (e.g., M=3 versus M=20) is increased computational processing, generating more imputation repetitions in return for more accuracy is a potentially

beneficial approach. In this volume, most examples utilize examples based on a minimum of *M*=5 MI repetitions. As a matter of best practice in every data analysis, we recommend using a smaller number of MI repetitions (*M*=5 or *M*=10) in the exploratory phase of a data analysis with a repeat of the final analyses at a higher repetition count (e.g., *M*=30 to *M*=100). This "sensitivity analysis" will identify for the analyst whether key inferential statistics (e.g., CI half-widths, *p*-values for hypothesis tests) change in any statistically meaningful way. If significant change occurs when analyses are repeated at the higher repetition count, final results based on the more computationally intensive approach should be reported.

3.7 Overview of Multiple Imputation Procedures

PROC MI is designed to fill in missing data using one of several optional multiple imputation methods. This is step 1 in the three-step multiple imputation process. The MI statement is required and executes the procedure either using all defaults or with user-specified options to override default settings. As already demonstrated, this statement can be used to examine the missing data pattern through use of PROC MI with an optional NIMPUTE=0 specification for no imputations performed. After the initial examination of the missing data pattern and consideration of several issues that influence the imputation process, PROC MI is used to impute and create *M* imputed data sets.

The second step of the multiple imputation process is analysis of complete (imputed) data sets using standard SAS procedures or SAS SURVEY procedures. By default, PROC MI reports the multiple imputation estimates and inferential statistics for the unweighted mean of each continuous variable in the imputation model. However, MI estimates for many other univariate and multivariate statistics require the additional steps in which multiply imputed data generated by PROC MI are analyzed by SAS procedures and then combined for MI estimation and inference using PROC MIANALYZE. Typical examples of standard and corresponding SURVEY procedures are PROC MEANS (with a BY/DOMAIN group or similar technique since PROC MI provides means analyses by default)/PROC SURVEYMEANS for analysis of means, PROC FREQ/PROC SURVEYFREQ for analysis of frequency tables, PROC REG/PROC SURVEYREG/PROC GLM/PROC GENMOD, and so on for linear regression models, PROC PHREG/PROC SURVEYPHREG for survival analysis using Cox proportional hazards regression models, and PROC LOGISTIC/PROC SURVEYLOGISTIC for logistic regression with a variety of dependent variable types (binary, multinomial, ordinal) and link functions (e.g., logit, probit, complementary log-log options).

In step 2, SAS procedures are used to output a data set of statistics estimated for each MI repetition and organized for input into PROC MIANALYZE. For example, we use a BY _IMPUTATION_ statement to repeat the analysis for each of the multiple imputation repetitions in the input data set. The format of the output from the analysis of the *m*=1....*M* repetitions depends on the procedure used in the second step as well as the intended analysis from PROC MIANALYZE.

In the third and final step, PROC MIANALYZE combines results from step 2 to generate multiple imputation estimates and variances of estimates that permit the user to make statistical inferences from the MI data set. PROC MIANALYZE is considered a companion procedure to PROC MI in that it completes the three-step process.

These procedures and a variety of options will be demonstrated through many example applications presented in later chapters.

3.8 Multiple Imputation Example

This section presents an introductory multiple imputation example using the data set, **housesprice3**, which includes continuous and categorical variables and exhibits an arbitrary missing data pattern. In this example, we demonstrate the entire three-step multiple imputation process.

Our analysis goal is to perform a linear regression of house price (PRICE) on the square footage of the house (SQFEET) and number of bedrooms in the home (BEDRM).

We begin by using PROC CONTENTS and PROC MI without imputation to determine the type of variables that are to be included in the imputation model and the amount and pattern of missing data. We then proceed to multiply impute the missing data values, followed by regression analysis of the MI repetition data sets and use of PROC MIANALYZE to generate MI estimates and the corresponding inferential statistics for the regression model.

```
proc contents data=houseprice3;
run;
```

Output 3.12: Partial Output from Contents Listing of Houseprice3 Data Set

Alphabetic List of Variables and Attributes				
#	Variable	Type	Len	Label
2	bedrm	Num	8	Number of Bedrooms
3	price	Num	8	Price of Home
1	sqfeet	Num	8	Square Footage of Home

Output 3.12 consists of selected output from PROC CONTENTS and indicates that all variables are type=numeric with a length of 8.

We next use PROC MI without imputation (NIMPUTE=0) to examine the missing data pattern.

```
proc mi data=houseprice3 simple nimpute=0;
grun;
```

Output 3.13: PROC MI without Imputations—Missing Data Patterns and Simple Statistics

Missing Data Patterns								
						Group Means		
Group	sqfeet	bedrm	price	Freq	Percent	sqfeet	bedrm	price
1	X	X	X	17	60.71	1532.647059	2.882353	85974
2	X	X	.	7	25.00	1229.285714	3.142857	.
3	.	X	X	4	14.29	.	3.250000	83980

Univariate Statistics							
						Missing Values	
Variable	N	Mean	Std Dev	Minimum	Maximum	Count	Percent
sqfeet	24	1444	336.71581	720.00000	2105	4	14.29
bedrm	28	3.00000	0.60858	2.00000	4.00000	0	0.00
price	21	85594	22229	30000	125000	7	25.00

Output 3.13 reveals that the missing data pattern for the three variables is arbitrary, the two variables to be imputed (PRICE and SQFEET) are continuous, and the third variable, number of bedrooms (BEDRM), is fully observed. Before

performing the actual imputation, it is useful to examine the marginal distribution of each variable to determine if the sample values are highly skewed or if there are extreme outlier values that could influence PROC MI's estimation of the predictive distribution of the missing values.

Here, we use PROC SGPLOT to generate histograms with a superimposed normal curve and a boxplot for each of the two continuous variables, PRICE and SQFEET.

```
proc sgplot data=houseprice3;
 histogram sqfeet; density sqfeet;
run;

proc sgplot data=houseprice3;
 histogram price; density price;
run;
```

Figures 3.1: Histogram of Square Footage

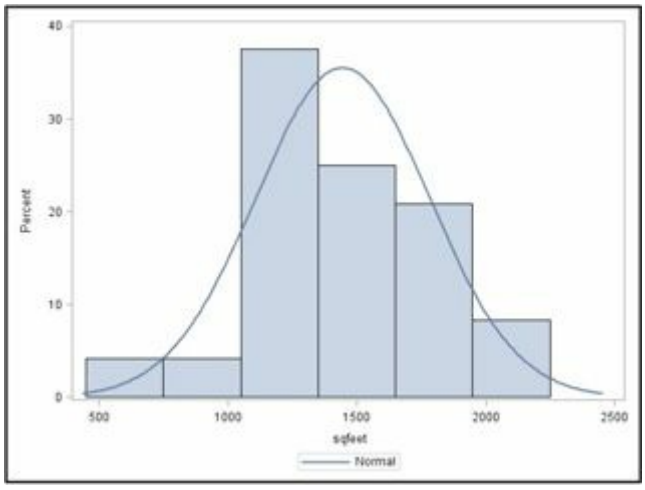
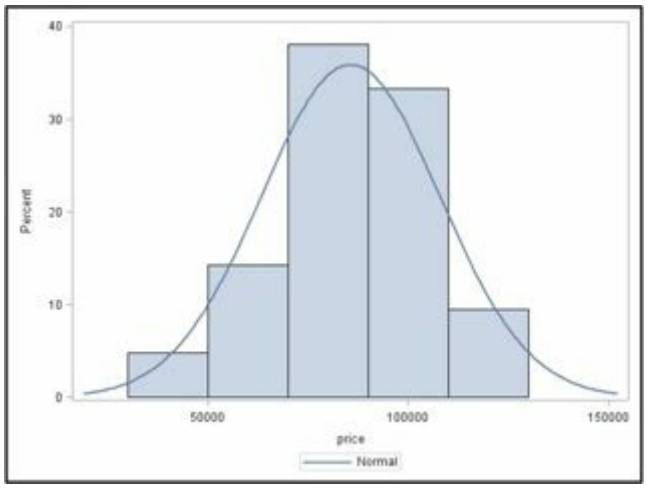


Figure 3.2: Histogram of Price



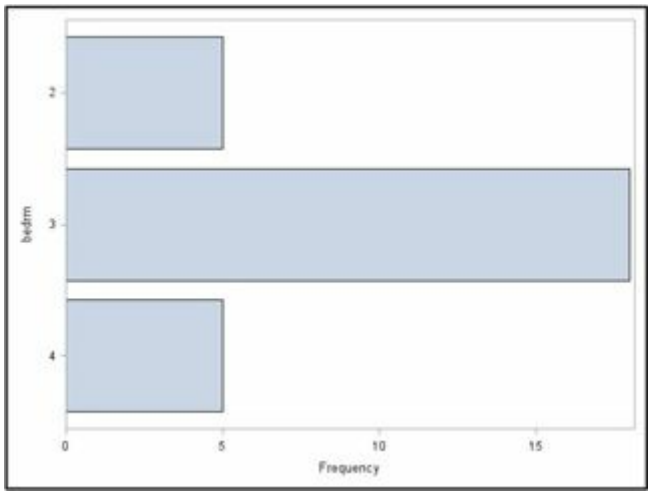
Figures 3.1 and 3.2 present histograms with superimposed normal curves. They provide an informal check of the normality assumption for the two continuous

variables used in the imputation: SQFEET and PRICE. Although the small sample size for this example data set limits the ability to graphically analyze the properties of the data distributions, each variable appears to be relatively symmetrically, if not normally, distributed.

For the number of bedrooms in the home, BEDRM, we again use PROC SGPLOT to produce a horizontal bar graph.

```
proc sgplot data=houseprice3;
 hbar bedrm;
run;
```

Figure 3.3: Horizontal Bar Chart of Number of Bedrooms in Home



The VBOX statement produces the vertical boxplots for SQFEET and PRICE.

```
proc sgplot data=houseprice3;
 vbox sqfeet;
run;

proc sgplot data=houseprice3;
 vbox price;
run;
```

Figure 3.4: Boxplot of Square Footage

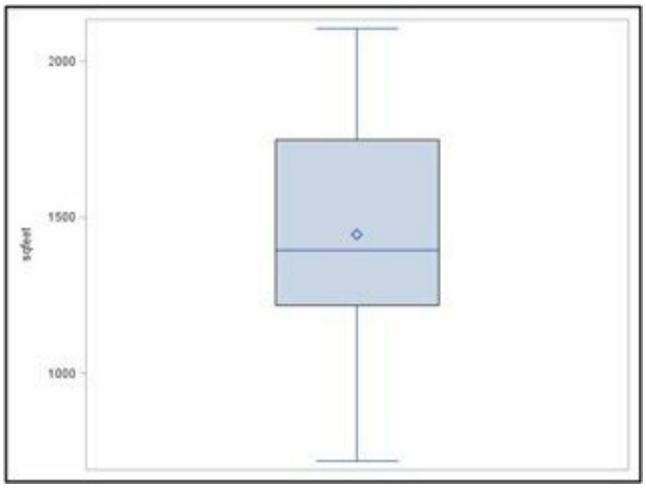
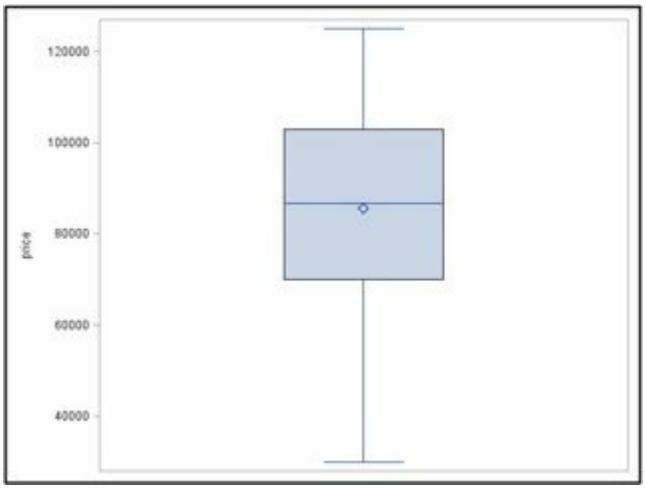


Figure 3.5: Boxplot of Asking Price



Figures 3.4 and 3.5 present boxplots that provide a visual check of the distributions of the two continuous variables used in the imputation: SQFEET and PRICE. In these plots, we are looking for extreme outliers as well as the overall distribution of each variable. Neither Figures 3.1–3.2 nor Figures 3.4–3.5 provides strong evidence of a need for variable transformations in the imputation model. Because number of bedrooms is a categorical variable, it is not included in this evaluation.

Since the data exploration steps show that the variables to be imputed are both continuous with a fully observed categorical covariate and the data set has an arbitrary missing data pattern, the FCS imputation method is one appropriate choice. For this simple example, we use mainly default options but do specify a SEED value. The seed is used to initiate the generation of random numbers that are used in the imputation process. The specification of a fixed SEED value enables the analyst to exactly replicate the random number generation sequence and reproduce the results of a multiple imputation analysis run. For other settings, the defaults are used: M=5 repetitions and 20 burn-in iterations for the FCS method.

The imputation is executed using the code below.

```
proc mi data=houseprice3 out=out_imputed seed=678;
 fcs regression (price sqfeet);
 var sqfeet bedrm price;
run;
```

The VAR statement lists the variables to be used in the imputation, and the OUT= option specifies creation of the output data set **out_imputed** with individual data records identified by the automatic SAS variable called _IMPUTATION_ with values of 1–5 corresponding to the five repetitions of the imputation process.

Output 3.14: Example Output from PROC MI

Model Information	
Data Set	WORK.HOUSEPRICE3
Method	FCS
Number of Imputations	5
Number of Burn-in Iterations	20
Seed for random number generator	678

FCS Model Specification	
Method	Imputed Variables
Regression	sqfeet bedrm price

Missing Data Patterns								
						Group Means		
Group	sqfeet	bedrm	price	Freq	Percent	sqfeet	bedrm	price
1	X	X	X	17	60.71	1532.647059	2.882353	85974
2	X	X	.	7	25.00	1229.285714	3.142857	.
3	.	X	X	4	14.29	.	3.250000	83980

Variance Information							
	Variance				Relative	Fraction	
Variable	Between	Within	Total	DF	Increase in Variance	Missing Information	Relative Efficiency
sqfeet	345.999487	4007.593378	4422.792762	21.741	0.103603	0.097844	0.980807
price	4514725	19145698	24563368	15.855	0.282971	0.238850	0.954408

Parameter Estimates										
			95% Confidence Limits						t for H0:	
Variable	Mean	Std Error			DF	Minimum	Maximum	Mu0	Mean=Mu0	Pr > |t|
sqfeet	1442.402987	66.504081	1304.39	1580.42	21.741	1417.955986	1464.298930	0	21.69	<.0001
price	84254	4956.144484	73739.96	94768.74	15.855	81177	86238	0	17.00	<.0001

Output 3.14 displays the standard output from PROC MI for our imputation problem. The Variance Information included in this output pertains to the multiple imputation variance of the estimated means for the PRICE and SQFEET variables for which missing values have now been imputed.

The Parameter Estimates output also pertains to the estimates of the mean of PRICE and SQFEET and includes the combined MI standard error and MI 95% confidence limits for each mean. The MI degrees of freedom used to select the critical value of the Student *t* distribution for constructing the 95% CI for the mean of SQFEET is 21.741. As described in Section 2.5.3, PROC MI uses a special formula derived by Barnard and Rubin (1999) to estimate the degrees of freedom for the construction of the MI confidence interval for the true mean value of SQFEET and PRICE.

We next examine the output data set (by repetition) means for PRICE and SQFEET as an informal check of the imputation results. The five imputed data sets are contained in the **out_imputed** data set and are identified by the _IMPUTATION_ variable with values of 1–5. We expect the five means to be similar but not exactly the same given the variability of the MI process.

```
proc means data=out_imputed mean;
 class _imputation_;
 var price sqfeet;
run;
```

Output 3.15: Means Analysis of Price by Imputation

Imputation Number	N Obs	Variable	Mean
1	28	price	85050.25
		sqfeet	1464.30
2	28	price	86238.16
		sqfeet	1429.05
3	28	price	81176.79
		sqfeet	1449.29
4	28	price	82993.62
		sqfeet	1417.96
5	28	price	85812.93
		sqfeet	1451.42

Each of the *M*=5 imputed data sets has a slightly different estimate of the mean value for SQFEET and PRICE due to the differing values imputed for the missing data (Output 3.15).

Although PROC MI has by default already provided MI estimates and CIs for the means of the SQFEET and PRICE variables, our assumed analytic objective in this exercise is to estimate the regression of PRICE on the square footage of the house and the number of bedrooms in the home. Therefore, we next use the PROC REG command with a BY statement to execute five separate regressions, one for each imputation repetition data set contained in the **out_imputed** data set.

The following PROC REG code shows how to create an output data set consisting of the regression model parameter estimates and their standard errors along with covariance information for subsequent use with PROC MIANALYZE. PROC REG produces an estimates (EST) data set that includes the regression parameter estimates and the corresponding estimated variance/covariance matrix for each of the five replications of the regression analysis. (Note: In this model we are treating the categorical variable BEDRM as continuous.) The printout of **out_est** in Output 3.16 highlights the structure and content of the output data set.

```
proc reg data=out_imputed outest=out_est covout;
 model price=sqfeet bedrm ;
 by _imputation_;
```

```
run;

proc print data=out_est noobs;
run;
```

Output 3.16: Printout of out_est Data Set

Imputation	_MODEL_	_TYPE_	_NAME_	_DEPVAR_	_RMSE_	Intercept	sqfeet	bedrm	price
1	MODEL1	PARMS		price	14522.58	-10808.46	36.11	14327.12	-1
1	MODEL1	COV	Intercept	price	14522.58	223501660.32	-51390.99	-46905851.98	.
1	MODEL1	COV	sqfeet	price	14522.58	-51390.99	100.98	-32156.92	.
1	MODEL1	COV	bedrm	price	14522.58	-46905851.98	-32156.92	31331066.78	.
2	MODEL1	PARMS		price	11703.55	62.78	24.13	17230.59	-1
2	MODEL1	COV	Intercept	price	11703.55	149802816.01	-35285.25	-31495476.35	.
2	MODEL1	COV	sqfeet	price	11703.55	-35285.25	57.55	-15651.18	.
2	MODEL1	COV	bedrm	price	11703.55	-31495476.35	-15651.18	17953947.96	.
3	MODEL1	PARMS		price	11300.02	-20228.70	53.61	7905.12	-1
3	MODEL1	COV	Intercept	price	11300.02	140871642.76	-38427.78	-26872765.51	.
3	MODEL1	COV	sqfeet	price	11300.02	-38427.78	69.04	-20542.36	.
3	MODEL1	COV	bedrm	price	11300.02	-26872765.51	-20542.36	18881527.76	.
4	MODEL1	PARMS		price	10738.47	-29010.83	52.71	12419.83	-1
4	MODEL1	COV	Intercept	price	10738.47	124939291.43	-27962.63	-27057041.91	.
4	MODEL1	COV	sqfeet	price	10738.47	-27962.63	45.89	-12370.44	.
4	MODEL1	COV	bedrm	price	10738.47	-27057041.91	-12370.44	14865929.19	.
5	MODEL1	PARMS		price	16425.38	-11052.24	33.01	16317.03	-1
5	MODEL1	COV	Intercept	price	16425.38	286040233.29	-68640.78	-58926105.94	.
5	MODEL1	COV	sqfeet	price	16425.38	-68640.78	140.26	-44979.81	.
5	MODEL1	COV	bedrm	price	16425.38	-58926105.94	-44979.81	41403534.08	.

From Output 3.16, first note that the _TYPE_=PARMS row includes the estimated regression parameters for each repetition's covariates. Also note that the estimated parameters from the five regression models differ in value. For example, for repetition 1 (_IMPUTATION_=1), the parameter estimate for SQFEET=36.11, repetition 2=24.13, repetition 3=53.61, repetition 4=52.71, and repetition 5=33.01. The other estimates (BEDRM) show similar fluctuations across the *M*=5 repetitions. The additional rows contain the covariance matrix information for the regression parameter estimates (in this example a 3 x 3 matrix), identified by _TYPE_=COV for the five MI repetitions. The covariance information will ultimately be used in PROC MIANALYZE for estimation of the variances.

The third and final step in our analysis plan is to use PROC MIANALYZE to

combine these five sets of regression estimates to produce final MI estimates of the regression model parameters, their standard errors, and 95% CIs for the true parameter values. For this step, the **out_est** data set from the PROC REG analysis of the MI repetition data sets (second step) is used as input to PROC MIANALYZE.

The following code illustrates use of PROC MIANALYZE to combine the MI repetition estimates to produce the final MI estimates, standard errors, and CIs for the regression model parameters. Use of the MODELEFFECTS statement, including the intercept followed by the model predictors, declares the effects used in the model. The EST (estimate) type of data set (here named **out_est**) includes both the needed parameter estimates (_TYPE_=PARMS) and the covariance matrices for the regression parameters (_TYPE_=COV) required by PROC MIANALYZE to complete the MI analysis for this linear regression model. In this example, we use an estimate type of output data set for use in PROC MIANALYZE, but many other types of data sets are also appropriate as input to PROC MIANALYZE. There are numerous examples of this process in action in Chapters 4 through 7 as well as in Chapter 8, which provides more detail on this particular issue.

```
proc mianalyze data=out_est;
 modeleffects intercept sqfeet bedrm;
run;
```

Output 3.17: Variance Information and Parameter Estimates from PROC MIANALYZE

Model Information

Data Set	WORK.OUT_EST
Number of Imputations	5

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
intercept	120135888	185031129	329194195	20.857	0.779129	0.485047	0.911569
sqfeet	165.626240	82.743719	281.495207	8.0238	2.402013	0.759385	0.868148
bedrm	13727171	24887201	41359806	25.217	0.661891	0.440926	0.918961

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept	-14207	18144	-51955.1	23540.16	20.857	-29011	62.783534	0	-0.78	0.4424
sqfeet	39.914437	16.777819	1.2	78.58	8.0238	24.130384	53.605680	0	2.38	0.0445
bedrm	13640	6431.158980	400.5	26879.38	25.217	7905.117794	17231	0	2.12	0.0439

Output 3.17 includes standard output from PROC MIANALYZE. Some of the key estimates included for each parameter are the between, within, and total variance; the relative increase in variance (due to the imputation); relative efficiency; and the fraction of missing information. Note that regression parameters for each predictor variable have a value for RE and fraction of missing information, even if there is no missing data on that particular variable. This is due to the fact that estimation of the regression coefficient includes the covariance of that variable with PRICE and SQFEET, which were subject to missing data.

The parameter estimates from the five regressions are averaged and the associated standard errors account for the imputation variability. The estimates would be interpreted in the usual manner for linear regression, keeping in mind that the estimates are averaged across the *M* imputed data sets. The estimated coefficient for SQFEET, a measure of a house's square footage, implies that one additional square foot of house size results in an increase of about $39.91 in house price. And, one additional bedroom results in an increase of about $13,640 in the home price, always holding all other covariates constant. Both predictors are statistically significant at the alpha=.05 level.

3.9 Summary

This chapter has provided a checklist of preliminary data checks and the sequence of MI steps that are required when planning and executing a multiple imputation session in SAS. With this background and introductory example under our belts, we can now turn to more complex applications of the MI method for estimation and inference.

Chapter 4: Multiple Imputation for the Analysis of Complex Sample Survey Data

4.1 Multiple Imputation and Informative Data Collection Designs

The general introductions to multiple imputation using SAS (Chapter 2) and preparation for an actual MI analysis in SAS (Chapter 3) have focused on the types of variables to be imputed, the patterns and amount of missing data, and the statistical distribution assumptions that are explicit or implicit in the imputation model. To this point, we have not really considered how that data was originally collected or whether information on the data collection design (e.g., sample survey, randomized clinical trial, or genetic assay) should be incorporated in the imputation or analysis steps. The simplest experiments or trials will have randomized assignment of subjects to treatment(s) and control conditions. Case/control study observations will be stratified by an outcome and will attempt to establish association of the outcome with prior exposures and risk factors. A genome-wide association study (GWAS) in genetics is based on observed single nucleotide polymorphisms (SNPs) that are sampled in a structured fashion across the genome. Multilevel data collections, randomized cluster trials, and many population surveys collect data on clusters of observational units with intracluster (intraclass) correlations for the variables of interest. Complex sample survey designs that incorporate features such as stratification and clustering may require case-specific weighting to account for differences in the sample selection and response probability for the observed

units.

It is beyond the scope of this book to address all of the many applications of the MI technique to data from such potentially "informative" data collection designs. However, we do want the reader to recognize that features of the data collection design often are important, and in many cases it is appropriate to incorporate these design features into the three MI imputation, estimation, and inference steps. We refer the reader to the many excellent review articles that are available in the statistical literature on special methods for applying MI to data from randomized trials (Taylor and Zhou 2009), randomized cluster trials (Andridge and Little 2011), multilevel data collections (van Buuren 2011), and genetic studies (e.g., Guex, Migliavacca, and Xenarios 2010; Marchini et al. 2007).

In this chapter, we place a special focus on MI for data from complex sample surveys—observational data collection designs that are widely used in descriptive and analytical studies in the fields of social science, education, health, agriculture, natural resources, and business.

In approaching MI analysis of data from surveys and other types of studies, it is important to distinguish two ways in which the data collection design may be "informative" in estimation and inference. First, information about the design of the study may be needed to correctly analyze the complete sample data. For example, unbiased estimation of a population mean from a complex sample survey will typically require weighted estimation. To obtain robust estimates of standard errors and confidence intervals for this estimate of the mean, information on the design stratification and clustering is a necessary input to a Taylor Series or repeated replication variance estimation method. In the survey-based examples presented in this text, we assume that the complex sample designs are "informative" and must be incorporated in the MI approach to analysis of sample survey data. For this reason, at step 2 of the MI process, SAS SURVEY procedures are used to analyze the completed data sets produced by PROC MI. In addition, in the step 3 MI "combining" step, the EDF= option is used to reflect the approximate complete data degrees of freedom for each MI repetition.

A related issue is whether conditional on the underlying data, the study design is also "informative for nonresponse"—that is, does the design itself play a role in the missing data mechanism and the resultant missing data patterns? For example, if the mechanism that produces item missing data for income variables in the Health and Retirement Survey differs for strata of sample subjects in New York and Wisconsin, do we need to include indicators of New York or

Wisconsin stratum residency in the imputation model to correctly capture the stratum-specific effects? Does the analysis weight assigned to each case in the data set also predict the missing values of the income variable? These questions are difficult if not impossible to answer solely based on the observed study data. The answers may well differ depending on the individual variables that are being considered. Unlike design-based estimation for complete data, where there is long-standing theory and practice to guide us on how design features should be incorporated in estimation and inference (Heeringa, West, and Berglund 2010), there is only a limited amount of empirical research (Reiter, Raghunathan, and Kinney 2006; Schenker et al. 2006) to guide us on how design features should be incorporated in the imputation of missing data. In this chapter and throughout this text, we assume that study design features are in fact "informative" for the mechanism that generates the missing data in study measurements. As a result, we feel that current best practice is to include these design variables in the imputation process itself (step 1 of the MI process). The approach outlined in Section 4.3 and illustrated in the example in Section 4.4 is based on the current literature and the recommendations of leading experts on the topic of MI. As time progresses, we expect there to be new work in this area, and we encourage the reader to check the companion website for this volume periodically for updates.

4.2 Complex Sample Surveys

Most surveys are based on sample designs with one or more complex features, including stratification, clustering of sampled elements, and weighting to compensate for varying probabilities of sample inclusion or differential response (Heeringa, West, and Berglund 2010). To illustrate approaches for including features of the complex sample design in the MI model, consider the National Health and Nutrition Examination Survey (NHANES) 2009–2010—one of the most widely analyzed sources of information on the health and health-related characteristics of the U.S. household population. NHANES is based on a multistage stratified, clustered, probability sample design. For each NHANES two-year data collection cycle, the first stage of the NHANES sample assigns all U.S. households to 1 of 15 distinct, nonoverlapping strata. The NHANES strata are defined based on a fixed set of broad characteristics for U.S. geographic areas, including region of country, urban/rural character, and size and characteristics of the resident population. From each of the 15 strata, 2 (3 in one 2009–2010 stratum) primary stage units (PSUs) are selected to the biennial sample. Each PSU is a geographically defined grouping (e.g., U.S. county) of households. The "ultimate cluster" sample of selected households for each PSU is identified through two additional nested levels of clustered sampling (e.g.,

sampling of blocks within selected PSUs and households within sampled blocks). Within sample households, NHANES oversamples designated types of individuals (e.g., persons with disability) to ensure minimum sample sizes for standalone analysis of important subpopulations. The NHANES data sets include analysis weight variables that are designed to compensate for these differential inclusion probabilities for household members as well as to adjust for differential nonresponse to either the basic health interview or the medical examination center (MEC) phase of the data collection. Although the percent of missing observations for key NHANES demographic survey variables such as age, gender, and educational attainment is typically low (<1–2%), rates of missing data for important substantive variables such as measures of income or physiological measurements can be significant.

In the following sections, we will use data from the NHANES 2009–2010 to illustrate what we currently believe to be the best practical approach to multiple imputation analysis of a complex sample survey data set in SAS. As noted in Section 4.1, in doing so, we acknowledge that this is an area that is still undergoing substantial development in both theory and practice.

4.3 Incorporating the Complex Sample Design in the MI Imputation Step

Don Rubin offered the following guidance on MI for complex samples: "Minimally, major clustering and stratification indicators and sample design weights (or estimated propensity scores of being in the sample) should be included in the imputation models. The possible lost precision when including unimportant predictors is usually a small price to pay for the general validity of the resultant multiply imputed data base" (1996, 478–479).

So how do we set about including the "design" in the MI imputation model?

Of course, the simplest approach to multiple imputation of complex sample survey data would be to ignore the sample design features. Omitting design properties such as stratification, clustering, and weights from the imputation model assumes that these features are not informative regarding the response mechanisms that produced the missing values. Unfortunately, these are not testable assumptions in an observational survey data set, especially if we consider the many survey variables, statistical estimates, and domains of analysis that data users may consider in analysis of large multipurpose data sets such as the NHANES.

Temporarily setting aside the question of the analysis weights, the natural

approach to modeling the relationship of the complex sample stratification and clustering to a specific outcome variable of interest, *y*, is through a generalized linear mixed model or more specifically a "multilevel" model. In this framework, the value of *y* (or function of *y*) for a surveyed individual (e.g., a NHANES diastolic blood pressure measurement) is regressed on fixed effects for covariates of substantive interest (x), a fixed effect (γ_{str}) for the design stratum to which the individual respondent belongs, and a random effect ($u_{cl(str)}$) for the PSU cluster, nested within stratum. Observational units, indexed by "*i*" are nested within a cluster (*cl*) and a stratum (*str*):

$$y_i = \beta x_i + \gamma_{str,i} + u_{cl(str),i} + \varepsilon_{i(str,cl)}$$

Theoretically, a hierarchical Bayes approach would be a flexible choice of method to multiply impute missing values of *y* under this mixed effects regression model. Research on these methods is ongoing (Gelman et al. 1995; van Buuren, 2011), but to date empirical and simulation studies have not provided us with clear guidance on how this approach could be applied in general practice, nor is PROC MI able to implement this highly sophisticated approach to multiple imputation of complex sample survey data.

Following Reiter, Raghunathan, and Kinney (2006), Schenker et al. (2006), and Rubin (1996), the practice that we will follow in all survey data examples in this text is to include the complex sample design strata, clusters, and weights as fixed effects in the imputation model. Unfortunately, for complex sample designs with many strata and two or more clusters per stratum, a full "fixed effects" specification with indicator variables for strata and the interaction of stratum and cluster indicators (to reflect the nesting of clusters within strata) may require estimation of too many parameters and result in an unstable model fit or convergence failure. This approach may work for NHANES with its small number of H=15 primary stage strata. However, when this fully specified fixed effects model for complex sample design features is applied to MI examples based on the National Comorbidity Survey–Replications (NCS-R, H= 42 primary stage strata) or the Health and Retirement Survey (HRS, H=56), the model fitting is more likely to become unstable or fail.

Empirically, the following reduced model (with only a fixed effect for the weight and the stratum X cluster interaction) performs well and generally avoids the instability problems sometimes encountered in estimating the full fixed effects model for design parameters:

$$y_i = \beta x_i + \delta w_i + \gamma_{\text{strata x cluster},i} + \varepsilon_i$$

where :

W_i = the survey analysis weight for case i = 1,..., n;

$\gamma_{\text{strata x cluster,i}}$ = fixed effect corresponding to the stratum / cluster to which the case belongs.

The above expression represents the general form of the design variable indicator variable coding that we will use in this text for all PROC MI examples that are based on complex sample data. Implementation of this approach to incorporating the complex design in the SAS MI procedure is straightforward. We will now illustrate it here using the NHANES 2009–2010 data set.

In this example, our analysis objective is to estimate the mean of the household-income-to-poverty-line ratio (INDFMPIR) for U.S. males and females of all ages based on data for respondents in the 2009–2010 NHANES household interview sample. Additional NHANES covariates included in the imputation model are: Gender (RIAGENDR), Age in Years (RIDAGEYR), and Race/Ethnicity (RIDRETH1). We first outline and execute the steps in the analysis based on the preceding recommendation for including the design and analysis weight variables in the imputation model. We then replicate the MI analysis twice, once ignoring the design variables and weights in specifying the imputation model and a second re-analysis that includes the stratum and cluster in the imputation model but uses the FREQ statement in PROC MI to incorporate the weights into the estimation formulae used to derive the regression and discriminant classification parameters for the predictive distributions for the item missing values. The results of these three approaches to treatment of the design variables are compared in a demonstration of the example-specific sensitivity of the final results to the treatment of the design variables in the imputation process.

We begin the MI analysis of the NHANES data by creating a new categorical variable that is the combination of the NHANES primary-stage stratum and cluster codes provided in the public user data set. The statement below is part of a larger data step but shows how the combined variable can be easily created:

```
descode=sdmvstra*10 + sdmvpsu;
```

We illustrate an approach that concatenates the two-digit stratum code (SDMVSTRA) and the one-digit PSU or cluster code (SDMVPSU) to create a new variable (DESCODE) that has one categorical level for each stratum and PSU code combination. (Any comparable procedure would suffice provided it generates a unique category for each stratum and cluster combination.) This approach works well provided there is no missing data on the complex design

variables. If you find missing data on either of these key variables or the survey weight, this **must** be resolved before proceeding with multiple imputation.

In this example, the monotone regression method of PROC MI is employed to impute missing values for the household-income-to-poverty-ratio variable, INDFMPIR. Along with other categorical predictors, the constructed DESCODE is declared in the CLASS statement, ensuring that SAS will generate the required indicator variables for each combination of stratum and cluster. The DESCODE variable and the continuous NHANES interview weight variable (WTINT2YR) are included in the PROC MI VAR statement, ensuring that they will be used as predictors in the regression imputation of missing values for the income to poverty ratio.

As usual, we first check the missing data patterns for our selected variables using PROC MI with the NIMPUTE=0 option and a VAR statement.

```
proc mi data=c4_ex1 nimpute=0;
 var riagendr ridreth1 ridageyr descode wtint2yr
indfmpir;
run;
```

Output 4.1: Missing Data Patterns for NHANES 2009–2010 Subset

Missing Data Patterns

									Group Means					
Group	RIAGENDR	RIDRETH1	RIDAGEYR	descode	WTINT2YR	INDFMPIR	Freq	Percent	RIAGENDR	RIDRETH1	RIDAGEYR	descode	WTINT2YR	INDFMPIR
1	X	X	X	X	X	X	9541	90.55	1.501520	2.773923	32.324914	815.404675	29077	2.228627
2	X	X	X	X	X	.	996	9.45	1.529116	2.496988	35.234940	822.373494	24623	.

Since the Missing Data Patterns display (Output 4.1) shows a monotone missing data structure, PROC MI with the MONOTONE regression method is used to impute missing data on the INDFMPIR variable. Included in the imputation model are gender, race/ethnicity, age, the combined design variable, and the two-year interview weight.

```
proc mi data=c4_ex1 nimpute=5 seed=41
out=nhanes0910_fullcomplex_reg;
 class riagendr ridreth1 descode;
 monotone regression (indfmpir);
 var riagendr ridreth1 ridageyr descode wtint1yr
indfmpir;
run;
```

Five (NIMPUTE=5) repetitions of the imputed NHANES data set are written to the working output file, nhanes0910_fullcomplex_reg.

4.4 Incorporating the Complex Sample Design in the MI Analysis and Inference Steps

Complex sample data impose two special requirements to ensure that the estimates and associated inferences generated in the MI analysis step are robust and correctly reflect the effects of stratification, clustering, and weighting. These two requirements are: 1) the appropriate SAS SURVEY procedure should be used to generate the desired estimates and standard errors for each of the m=1, …,M multiply imputed data sets; and 2) the EDF option must be used in PROC MIANALYZE to ensure a correct determination of the complete data degrees of freedom used in the construction of MI confidence intervals for the population statistics of interest.

We demonstrate use of the SURVEYMEANS procedure to generate gender-specific estimates of the mean and standard error for the ratio of family income to poverty threshold for each of the five imputed data sets produced by PROC MI above. Through use of the SURVEYMEANS procedure, we ensure that the "within imputation" estimates and standard errors that are generated (and subsequently input to PROC MIANALYZE) are consistent and correctly reflect the effect of the complex sample design features on the precision of these estimates.

The interview weight is used in the WEIGHT statement to correctly weight the population estimates. The BY _IMPUTATION_ statement requests a separate analysis of each imputation repetition data set. The subpopulation variable, gender (RIAGENDR), is coded 1=Male and 2=Female and is specified in the DOMAIN statement to ensure that an unconditional as opposed to conditional approach to subpopulation estimation is used (West, Berglund, and Heeringa, 2008). Finally, the ODS OUTPUT DOMAIN statement produces a data set containing estimates for each imputation repetition m=1,…,5 and the two subpopulation domains (men, women) of interest.

```
proc surveymeans data=nhanes0910_fullcomplex_reg;
  strata sdmvstra; cluster sdmvpsu; weight wtint2yr;
  by _imputation_;
  domain riagendr;
  var indfmpir;
  ods output domain=nhanes0910_fullcomplex_reg_m;
run;
```

After running the above code, a warning appears in the SAS log that merits further explanation:

```
NOTE: The BY statement provides completely separate
```

```
analyses of the BY groups. It does not provide a
statistically valid subpopulation or domain analysis,
where the total number of units in the subpopulation is
not known with certainty. If you want a domain
analysis, you should include the domain variables in a
DOMAIN statement.
```

In this SURVEYMEANS analysis, the BY statement is used to request separate analyses of each MI _IMPUTATION_ repetition. The DOMAIN statement requests that for each repetition, PROC SURVEYMEANS generate correct design-based subpopulation estimates for women and men. Though the warning may seem alarming, we are analyzing each repetition of the MI data set and correctly using a DOMAIN statement to specify the subpopulation estimates for each replication of the analysis. Therefore, the warning is important but not of concern in this particular example.

To view the output data set from PROC SURVEYMEANS, a listing is produced with PROC PRINT.

```
proc print data=nhanes0910_fullcomplex_reg_m;
run;
```

Output 4.2: Listing of the Nhanes0910_fullcomplex_reg_m Data Set

Obs	_Imputation_	DomainLabel	RIAGENDR	VarName	VarLabel	N	Mean	StdErr	LowerCLMean	UpperCLMean
1	1	Gender	1	INDFMPIR	Ratio of family income to poverty	5225	2.951138	0.041166	2.86387110	3.03840552
2	1	Gender	2	INDFMPIR	Ratio of family income to poverty	5312	2.747058	0.048043	2.64521061	2.84890554
3	2	Gender	1	INDFMPIR	Ratio of family income to poverty	5225	2.952598	0.040709	2.86629860	3.03889699
4	2	Gender	2	INDFMPIR	Ratio of family income to poverty	5312	2.741644	0.048040	2.63980326	2.84348418
5	3	Gender	1	INDFMPIR	Ratio of family income to poverty	5225	2.942628	0.039605	2.85866918	3.02658636
6	3	Gender	2	INDFMPIR	Ratio of family income to poverty	5312	2.746046	0.050023	2.64000289	2.85208890
7	4	Gender	1	INDFMPIR	Ratio of family income to poverty	5225	2.947302	0.039129	2.86435358	3.03025108
8	4	Gender	2	INDFMPIR	Ratio of family income to poverty	5312	2.745225	0.044322	2.65126627	2.83918351
9	5	Gender	1	INDFMPIR	Ratio of family income to poverty	5225	2.950041	0.040782	2.86358891	3.03649501
10	5	Gender	2	INDFMPIR	Ratio of family income to poverty	5312	2.762410	0.047758	2.66116835	2.86365241

Output 4.2 highlights how the output data set is organized. The variable VARNAME contains the name of our analysis variable INDFMPIR while other variables needed for MI step 3 are RIAGENDR and _IMPUTATION_. To ensure that estimates are in domain and MI repetition order we sort by RIAGENDR and _IMPUTATION_. This step is required for PROC MIANALYZE to correctly combine the results within these groupings.

```
proc sort data=nhanes0910_fullcomplex_reg_m;
 by riagendr _imputation_;
run;
```

At this stage, we are ready to proceed with the PROC MIANALYZE step. Here, we use the EDF option to supply the complete data degrees of freedom for the NHANES 2009–2010 complex sample design. The approximate value of the complete data degrees of freedom is computed using the following expression: $df_{complex}$=#PSUs-#Strata. The design has 31 PSU clusters assigned to 15 primary-stage strata, that is, df_{NHANES}=31-15=16. This approach follows the "fixed rule" of calculating complex sample degrees of freedom for multistage complex sample designs (Korn and Graubard, 1999) and is the default method for df approximation used by SAS in its SURVEY procedures. In specifying EDF=16 in the example code below, PROC MIANALYZE will use the design-corrected complete degrees of freedom when it applies the Barnard and Rubin formula (see Section 2.5.2) to determine the appropriate degrees of freedom for constructing MI confidence intervals for the population statistics of interest.

```
proc mianalyze data=nhanes0910_fullcomplex_reg_m
edf=16;
 by riagendr; modeleffects mean; stderr stderr;
run;
```

The PROC MIANALYZE results for the MI estimates of gender-specific means, standard errors, and confidence intervals for the household-income-to-poverty-ratio variable is summarized in the initial columns of Output 4.3.

Next, for purposes of a comparison to the recommended technique for including the design variables as fixed effects in the imputation model, we repeat the imputation and analyses but omit the complex sample design variables and weight from the imputation model. We still incorporate these design variables in the MI estimation and inference steps (steps 2 and 3) by including the STRATA, CLUSTER, and WEIGHT statements in the PROC SURVEYMEANS analysis and use of EDF=16 option in the PROC MIANALYZE syntax. This strategy allows evaluation of the impact of the design and weights in the imputation process. The SAS code for this analysis is presented below. The estimates, SEs, and CIs are presented in Output 4.3.

```
proc mi data=c4_ex1 nimpute=5 seed=41
out=nhanes0910_nocomplex_reg;
 class riagendr ridreth1;
 monotone regression (indfmpir);
 var riagendr ridreth1 ridageyr indfmpir;
run;

proc surveymeans data=nhanes0910_nocomplex_reg;
 strata sdmvstra; cluster sdmvpsu; weight wtint2yr;
 by _imputation_;
```

```
 domain riagendr;
 var indfmpir;
 ods output domain=nhanes0910_nocomplex_reg_m;
run;

proc sort data=nhanes0910_nocomplex_reg_m;
 by varname riagendr _imputation_;
run;

proc mianalyze data=nhanes0910_nocomplex_reg_m edf=16;
 by varname riagendr;
 modeleffects mean;
 stderr stderr;
run;
```

Finally, extending our comparative or "sensitivity" analysis, we test a third approach, using the DESCODE variable as a categorical predictor in the imputation model but use the interview weight in the FREQ statement of PROC MI rather than as a covariate in the imputation model. This approach is effectively "weighting" the estimation of the regression or discriminant classification models that are fitted in the P-Step of the MI process. Because the FREQ statement allows only integer weights, we inflate the WTINT2YR variable by a multiplicative factor of 1000 to create WTINT2YR_INT. (This linear scaling of the weight increases the relative weight values by the multiplicative constant but will have no impact on the resulting estimates of regression or classification model parameters.)

```
wtint2yr_int=wtint2yr*1000;
```

The remaining code for MI steps 2 and 3 is repeated. Results of this additional comparative analysis are also presented in Output 4.3.

```
proc mi data=c4_ex1 nimpute=5 seed=41
out=nhanes0910_complex_freq_reg;
 class riagendr ridreth1 descode;
 monotone regression (indfmpir);
 freq wtint2yr_int;
 var riagendr ridreth1 ridageyr descode indfmpir;
run;

proc surveymeans data=nhanes0910_complex_freq_reg;
 strata sdmvstra; cluster sdmvpsu; weight wtint2yr;
 by _imputation_;
 domain riagendr;
 var indfmpir;
 ods output domain=nhanes0910_complex_freq_reg_m;
run;
```

```
proc sort data=nhanes0910_complex_freq_reg_m;
 by riagendr _imputation_;
run;

proc mianalyze data=nhanes0910_complex_freq_reg_m
edf=16;
 by riagendr;
 modeleffects mean;
 stderr stderr;
run;
```

Output 4.3: Comparison of Gender-Specific Estimates of Mean-Family-Income-to-Poverty Ratio with and without Design Representation in the Imputation Model, and with Use of the FREQ Statement

	Model with Complex Sample Design Variables and Weight in Imputation Model					Model without Complex Sample Design Variables and Weight in Imputation Model					Model with Complex Sample Design Variables in Imputation Model and Weight in FREQ Statement				
Family Income to Poverty Ratio	Estimate	Std Error	95% Confidence Limits		DF	Estimate	Std Error	95% Confidence Limits		DF	Estimate	Std Error	95% Confidence Limits		DF
Male	2.95	0.04	2.86	3.04	14.15	2.90	0.04	2.81	2.99	14.12	2.94	0.04	2.85	3.03	14.32
Female	2.75	0.05	2.64	2.85	13.79	2.71	0.05	2.61	2.81	14.02	2.74	0.05	2.64	2.85	14.32

Output 4.3 presents the results of the three MI analyses of the NHANES 2009–2010 complex sample data. Estimates of the gender-specific mean of the household-income-to-poverty-line ratio, their standard errors, and 95% CIs for the three different approaches to the complex sample MI analysis are presented. Compared to the recommended approach, ignoring the complex sample design in specifying the imputation model has a noticeable (albeit not significant) impact on the point estimates and confidence intervals of the NHANES estimates of male and female mean-family-income-to-poverty ratio. The second alternative approach, including the stratum and cluster design variables as fixed effects in the imputation model along with the FREQ statement (for weighted estimation), produces estimates and 95% CIs that are very similar to those for the recommended method. Although inclusion of the NHANES complex sample design variables in the MI model for this example problem does not result in statistically significant differences in point estimates or inferential statistics, it is incorrect to conclude that these design characteristics are "non-informative for the missing data mechanism" in all MI analyses of the NHANES survey data. Reiter and Raghunathan (2007) demonstrate that especially in smaller sample problems where intraclass correlations for the variables of interest are high (such as may be the case in education or housing studies) that the design variables should not be omitted from the imputation model. Our recommendation to data analysts is to incorporate the design variables in the imputation model but also replicate the simple "sensitivity analysis" illustrated in the results in Output 4.3 to determine if the exclusion of the design variables from the imputation model would significantly alter the estimates (i.e., suggest bias) or inferences that are drawn from the MI analysis.

4.5 MI Imputation and Analysis for Subpopulations of Complex Sample Design Data Sets

Subpopulation analysis of complex sample data requires special procedures to ensure that the distribution of the subpopulation observations across design strata and clusters is correctly reflected in the estimation of sampling variances and confidence intervals for the population statistics or model parameters of interest (West, Berglund, and Heeringa, 2008). Intuitively, the analyst might simply use a WHERE statement in the SAS procedure (e.g., WHERE AGE >= 8;) to include only the desired subpopulation cases from the larger data set as input to a SAS procedure. Depending on the distribution of subpopulation observations across the sample design strata and clusters, this method of simply "subsetting" the complex sample data can result in biased estimates of variance and confidence intervals. Correct subpopulation analysis using SAS SURVEY procedures for estimating means or regression models requires the specification of the subpopulation in a DOMAIN statement (e.g., DOMAIN RIAGENDR;), while correct subpopulation analysis in PROC SURVEYFREQ requires that a classification variable that defines the subpopulation levels be included as the first variable in the TABLES statement (e.g., TABLES SUBPOP *AGECAT*RIAGENDR;).

For most multiple imputation analyses of subpopulation data, multiple imputation will be performed using all sample data, and analysis will be performed on the subpopulation observations for the multiply imputed data. Under this scenario, the correct approach is to proceed with the usual MI imputation step as described in Section 4.3 and then incorporate the subpopulation specification (DOMAIN statement or modified TABLES statement) in the SAS SURVEY procedure command syntax. For example, the PROC SURVEYMEANS example code in Section 4.4 uses the DOMAIN statement to request separate estimates of the NHANES mean-family-income-poverty-line ratio separately for men and women. A feature of the DOMAIN statement is that output is produced for each level of the categorical variable used to define the subpopulations of interest. In situations where the analysis is focused on only one subpopulation level (e.g., women only), the analyst has two options: 1) perform the step 3 MI estimation step for both women and men—ignoring results for the latter; or 2) use a DATA step to remove the estimates for men from the estimates output file before it is input to the MI estimation step. The second approach is demonstrated in the PROC SURVEYREG example in Section 5.4.1.

On occasion, and for various reasons, the imputation model and the PROC MI imputations (step 1) must be restricted to a subset of cases from the full complex

sample data set. The primary reason for restricting the MI imputation step to a subset of cases is that the variables to be imputed are not applicable for the excluded cases. For example, it would be inappropriate to include male subjects in the imputation of missing data for variables that assess pregnancy experience for women. In the NHANES 2009–2010 Medical Exam Center (MEC) studies, blood pressure is not measured for children under the age of eight. While it would be computationally feasible under the imputation model to "impute" the blood pressures for the young children, these children are by no means missing at random from the observation process. Thus, the values assigned in the imputation would be an extrapolation based on the relationships actually observed in the sample of persons age eight and older.

In such situations where imputation is restricted to a subset of a complex sample data set, we recommend that the following modified procedure be used to ensure that subsequent analysis (step 2) and MI estimation (step 3) for subpopulations correctly reflects the full design stratification and clustering in the estimation of variances and CIs:

Pre-Imputation: Using the full sample weight, stratum code, and cluster code variables provided with the complex sample data set, conduct a full sample PROC SURVEYMEANS analysis of an arbitrary variable (e.g., age of subject). On the PROC SURVEYMEANS command line specify the VARMETHOD=JK and (OUTWEIGHTS=) options to generate jackknife repeated replication (JRR) weights for the full sample data. With the (OUTWEIGHTS=) option, SAS will append the JRR weights to the permanent or working SAS data set that you will be using as input to the PROC MI imputation step. The sole purpose in conducting the PROC SURVEYMEANS analysis of the full complex sample is to generate and add the JRR replicate weights to the SAS data set.

PROC MI Imputation (Step 1): Specify the PROC MI procedure for complex samples as illustrated in Section 4.3. Include in the PROC MI command sequence a WHERE statement to restrict the imputation to the desired subpopulation (e.g., WHERE AGE >= 8;). The full-sample JRR replicate weights generated in the pre-imputation step will be included in the PROC MI output data set.

SURVEY Procedure Analysis of MI Repetitions (Step 2): Use the appropriate SURVEY procedure to specify the desired analysis of the multiply imputed data set. Override the default Taylor Series variance estimation method by specifying VARMETHOD=JK in the PROC statement. Insert both a REPWEIGHTS statement (e.g., REPWEIGHTS REPWT_1-REPWGT_31;) and a full sample WEIGHT statement (e.g.,

WEIGHT WTMEC2YR) in the command syntax. The replicate weights specified in the REPWEIGHTS statement are the full-sample JRR replicate weight variables generated in the pre-imputation step (above). If an additional level of subpopulation analysis (e.g., by male and female) is desired within the subset of cases (e.g., persons age eight and older) that were imputed, follow the standard procedure and include a DOMAIN statement (e.g., DOMAIN RIAGENDR;) in the SURVEY procedure.

PROC MIANALYZE Estimation and Inference (Step 3): Follow standard PROC MIANALYZE procedures for survey data described in Section 4.4 above.

This four-step procedure to correctly impute and analyze data for restricted subsets of a complex sample requires only the one-time addition of the pre-imputation step and minor modifications to PROC MI and SURVEY analysis procedures. We illustrate this sequence of steps in detail in the example analyses presented in sections 5.3.1, 5.3.2, 5.3.4, 6.4.1, and 6.4.2.

4.6 Summary

This chapter has focused on the role that an experimental or observation data collection design may play in multiple imputation of missing observations in the study data set. A special emphasis has been placed on practical methods of incorporating features of complex sample survey designs in a SAS MI analysis conducted using the MI and MIANALYZE procedures. In the chapters to follow, the procedures described in this chapter will be used for all examples that are based on a complex sample survey data set such as the NHANES, NCS-R, or HRS.

Chapter 5: Multiple Imputation of Continuous Variables

5.1 Introduction to Multiple Imputation of Continuous Variables

Chapter 5 includes applications of the concepts and guidelines presented in the first four chapters of this book. The focus in this chapter is on the applications of multiple imputation to missing data for continuous variables. As described in Chapter 2, SAS PROC MI provides three general approaches to multiple imputation of item missing data for continuous variables. For a true monotone missing data pattern, the monotone method with linear regression or the predictive mean matching technique is the best approach to impute missing values for continuous variables with missing data. In cases where the missing data pattern is technically arbitrary (but nearly monotone), the MCMC monotone method can be first used to "fill in" the missing values for variables that have very low rates of missing data, effectively transforming the problem to a monotone pattern. The monotone method can then be applied to impute the

remaining variables in the filled-in monotone missing data pattern. If the variables in the imputation model are all continuous and approximately distributed as multivariate normal, $Y \sim$ MVN (μ,Σ), MI draws of missing values can be obtained using the MCMC method for posterior simulation. If the pattern of missing data is arbitrary and the imputation model includes a mixture of continuous and categorical variables with missing values, we recommend the fully conditional specification (FCS) method to impute the missing data. In FCS, the analyst can specify either linear regression or predictive mean matching as the technique for generating the missing value imputations for continuous variables. As covered in Chapter 6, the FCS and MONTONE methods use logistic regression (binary, ordinal) or discriminant techniques (nominal classes) to impute the missing values for any classification (categorical) type variables in the imputation model.

Our goal is to present SAS solutions to a range of example missing data problems that involve continuous variables. Each example works through the three-step process of MI and includes discussion of the relevant syntax, output, and interpretation of results. Section 5.2 uses the MCMC procedure to illustrate the full multiple imputation analysis of a data set that has a simple but arbitrary pattern of missing data for an imputation model that includes only continuous variables. Using both the regression and the predictive mean matching techniques, Section 5.3 illustrates the steps in the monotone imputation of missing values for a continuous variable. The FCS approach to imputation of missing values for continuous variables in an arbitrary missing data pattern is illustrated by example in Section 5.4.

5.2 Imputation of Continuous Variables with Arbitrary Missing Data

Our first example of multiple imputation for continuous variables is based on a Major League Baseball (MLB) salaries data set from 1992, downloaded from KEEL (Knowledge Extraction Based on Evolutionary Learning) (sci2s.ugr.es/keel/datasets.php 1991–1992) under the regression data sets repository. The original data set is modified for this example to include missing data values for the variables SALARY, a measure of 1992 player salaries and STRIKE_OUTS, number of strike outs during 1992.

Highlights of this example include use of the MCMC method for imputation of continuous variables (step 1), Trace and ACF plots to assess MCMC convergence, use of PROC REG for a linear regression analysis of the multiply imputed data sets (step 2) and PROC MIANALYZE to combine the results to

form multiple imputation estimates and inferential statistics (step 3).

The data set, **c5_ex1,** contains information for 1992 Major League Baseball players (excluding pitchers) including salary, a variety of performance statistics, and arbitration eligibility and participation. The data set contains both continuous and classification numeric variables. For this example, only numeric, continuous variables are used.

The overall analysis goal in this exercise is to estimate the regression of players' salaries (in $1000's) on a number of performance measures such as runs batted in, stolen bases, strikeouts and errors. We begin with an outline of the variables that we choose to include in the imputation model. Recall that the set of variables used in the imputation model will generally be larger than the variables that will actually be used in a specific analysis of the imputed data set.

BATTING_AVERAGE: Batting average, continuous and fully observed

ON_BASE_PERCENTAGE: On Base %, continuous and fully observed

RUNS: Number of runs, continuous and fully observed

HITS: Number of hits, continuous and fully observed

DOUBLES: Number of doubles, continuous and fully observed

TRIPLES: Number of triples, continuous and fully observed

HOMERUNS: Number of homeruns, continuous and fully observed

RUNS_BATTED_IN: Number of runs batted in, continuous and fully observed

WALKS: Number of walks, continuous and fully observed

STRIKE_OUTS: Number of strike outs, continuous with some missing data

STOLEN_BASES: Number of stolen bases, continuous and fully observed

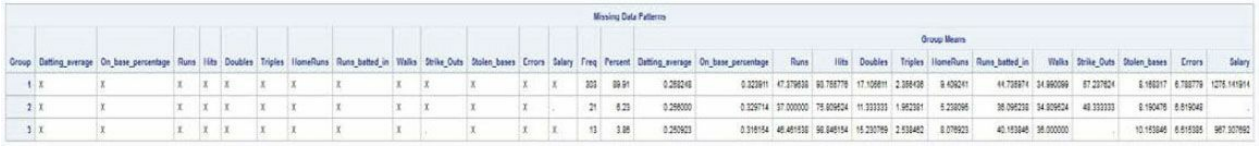
ERRORS: Number of Errors, continuous with fully observed data

SALARY: 1992 salary in thousands of dollars, continuous with some missing data

An initial exploratory analysis of the baseball data is performed using PROC MI with the NIMPUTE=0 and SIMPLE options. The exploratory step provides us a summary of the amount and pattern of missing data.

```
proc mi data=c5_ex1 nimpute=0 simple;
run;
```

Output 5.1: Missing Data Patterns and Group Means for the Baseball Data Set



Output 5.1 includes the grid showing the missing data patterns across the variables included in the imputation model. The GROUP variable is created automatically by PROC MI. It includes a value that indexes each unique missing data pattern. In this example, there are three unique groups. Group 1 consists of 303 (89.91%) records with fully observed data for all variables, Group 2 has 21 (6.23%) records with fully observed data on all variables except SALARY, while Group 3 has 13 (3.86%) records with missing data on STRIKE_OUTS. The Group Means section of the exploratory output includes estimated means for each variable by missing data group. This output verifies that the baseball data set has two variables with missing data and an arbitrary missing data pattern.

The SAS PROC MI code below demonstrates use of the MCMC method with a variety of options and diagnostic tools. The command statements for the MCMC imputation include specification of a seed value, 10 imputations, use of the ROUND, MIN, and MAX options, and selected diagnostic plots that can be used to assess MCMC convergence. We set bounds on the imputed values for salary and number of strike outs using the MIN= and MAX= options. For example, we use a salary (in thousands of $) minimum of 50 and a maximum set to the greatest observed value of 6100. For number of strike outs, a minimum of 1 and max of 100 provide logical bounds. Restricting the imputations to “bounds” derived from actual observed values does limit the imputation draws to a range that may be somewhat smaller than is theoretically possible given the estimated predictive distribution. For example, it is quite possible that the true salary for a player with a missing value is as high as 8000 (i.e., $8,000,000). This practice of bounding the imputations to the range of observed values may therefore eliminate the possibility of an imputation draw producing a value (e.g., $20 or $200,000,000) that, while statistically probable, is most unlikely in the real world (at least for 1992 MLB salaries!). We should note here that when the predictive mean matching technique is used in the MONTONE and FCS methods to impute continuous variables, the actual imputations are “draws” from a neighboring observed value, and by default are restricted to the observed range of variable values. The ROUND=1 option rounds imputed values to match the number of decimal places for the observed values of salary and strike outs in the original data set. We evaluate convergence of the single MCMC chain by plotting several statistics against the MCMC iteration number. When the MCMC algorithm has converged to the desired posterior distribution, these plots should display a random fluctuation about an average value for the statistic as the number of iterations advances. At convergence, the draws of posterior

parameters (i.e., means) at successive MCMC iterations should not exhibit patterns (increasing or decreasing trends). We investigate convergence of the MCMC algorithm with trace plots of the simulated posterior mean of SALARY and STRIKE_OUTS. Likewise, if posterior convergence of the MCMC algorithm has been achieved, there should not be correlations among the successive simulated parameter values. The MCMC ACF() option plots the lagged autocorrelation of the current iteration and prior iteration values against the length of the lag period (the default is up to a lag of 20 iterations). The ODS GRAPHICS ON statement enables use of the built-in plotting tools in SAS to display these plots.

```
ods graphics on;
proc mi data=c5_ex1 out=outc5ex1 seed=192 nimpute=10
 min=. . . . . . . . . . . . 50 1
 max=. . . . . . . . . . . . 6100 100
 round =1;
 var Batting_average On_base_percentage Runs Hits
Doubles Triples HomeRuns
   Runs_batted_in Walks Stolen_bases Errors salary
Strike_Outs;
 mcmc plots=(trace(mean(salary) mean(strike_outs))
acf(mean(salary)
   mean(strike_outs)));
run;
```

Output 5.2: Model Information from PROC MI

Model Information	
Data Set	WORK.C5_EX1
Method	MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	10
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	192

Output 5.3: Variance Information and Parameter Estimates from PROC MI

Variance Information							
Variable	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
Salary	142.365112	4488.135059	4642.736682	310.1	0.034908	0.033975	0.996614
Strike_Outs	0.038066	3.316086	3.357959	327.98	0.012627	0.012504	0.998751

Parameter Estimates									
Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Salary	1267.427596	68.137630	1133.357	1401.498	310.1	1249.563798	1291.682493	0	18.60	<.0001
Strike_Outs	56.674481	1.832473	53.070	60.279	327.98	56.436202	57.041543	0	30.93	<.0001

Outputs 5.2 and 5.3 display the Model Information, Variance Information, and Parameter Estimates tables produced by PROC MI.

The Model Information table details the method used (MCMC with a single chain), use of the EM algorithm for initial estimates for the chain, the default Jeffreys prior for the multivariate normal parameters, 200 burn-in iterations, 100 iterations separating each repetition imputation in the single chain, a seed value of 192, and number of imputation repetitions (10).

The Variance Information table provides information on the estimates of the MI variance components for the estimated mean for player salaries and number of strike outs. The initial columns provide the estimated between imputation, within imputation, and total variance for an MI estimate of the mean of player salaries. Additional MI statistics included in this output table are the Relative Efficiency (RE= 0.99 for both imputed variables), the Relative Increase in Variance (RIV=0.03 and 0.01) and the Fraction of Missing Information (FMI=0.03 and 0.01).

The Parameter Estimates table in the PROC MI output provides the MI estimates of the mean for SALARY (1267.42) and STRIKE_OUTS (56.67), the standard errors and 95% confidence limits along with other descriptive statistics from the multiply imputed data set. The variability or uncertainty introduced by the imputation process is factored into these estimates through use of the within and between imputation variances. The default Mu0=0 and Student *t* statistics for testing H_0: Mean=Mu0, along with the *p* values, specify and test the null hypothesis that the means for salary and strike outs are equal to 0. In both cases, they are significantly different from 0 at *p* values of <.0001.

Figure 5.1: Trace Plot for Salary

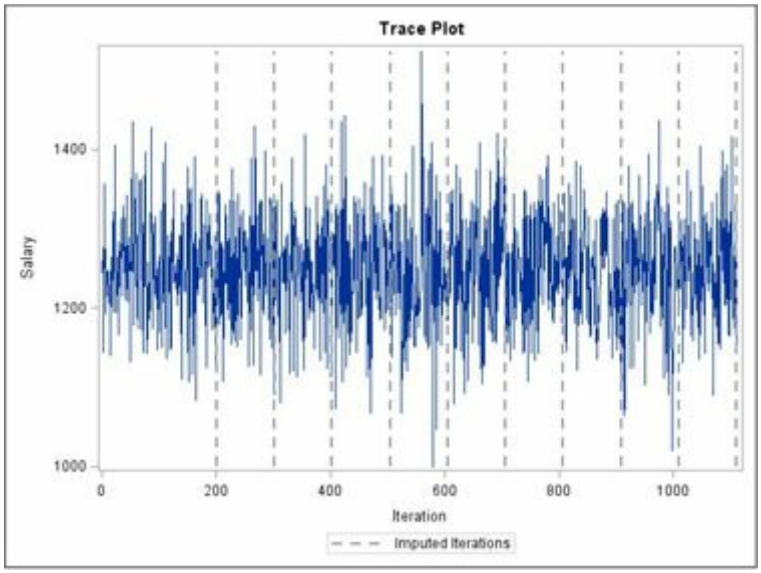
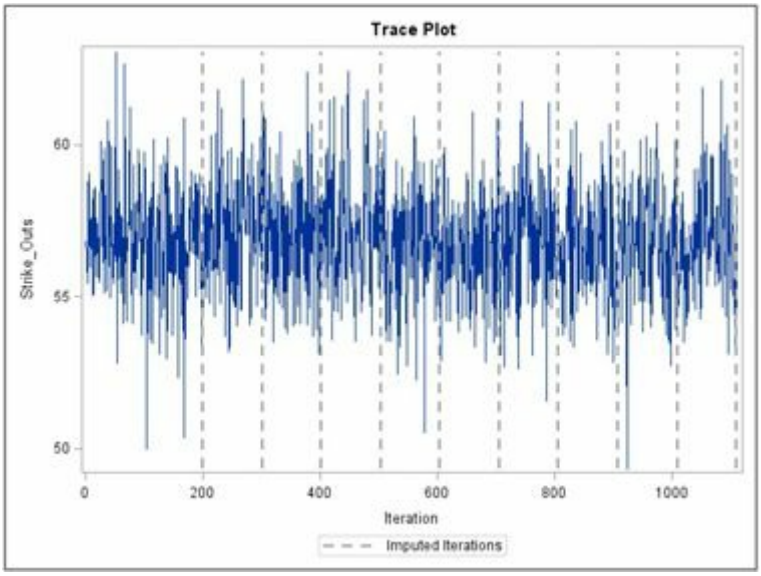


Figure 5.2: Trace Plot for Strike Outs



Figures 5.1–5.2 evaluate MCMC convergence through use of trace plots of the posterior mean of the variables SALARY and STRIKE_OUTS. In each plot, the values of these means are plotted against the iteration number for the 200 burn-in iterations and the 100 iterations that separate each of ten imputations, presented on the x axis. Neither Figure 5.1 nor 5.2 show any systematic pattern in these plots. The randomness to the plotted values of the posterior mean suggest that there should be no major concerns regarding convergence of the default single MCMC chain used in this imputation or the independence of draws for the individual imputation repetitions.

Figure 5.3: Autocorrelation Plot for Salary

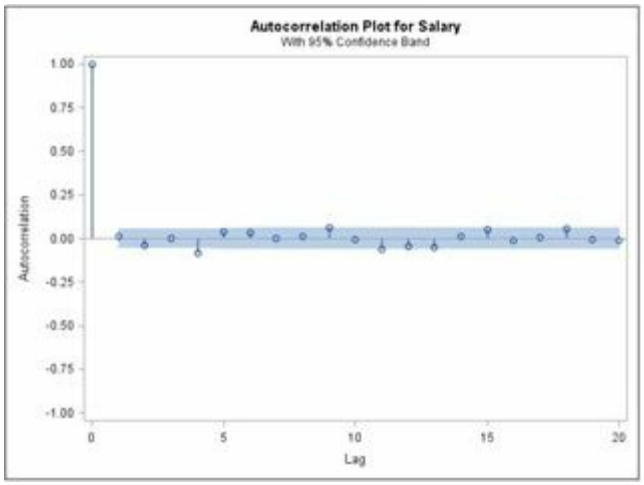
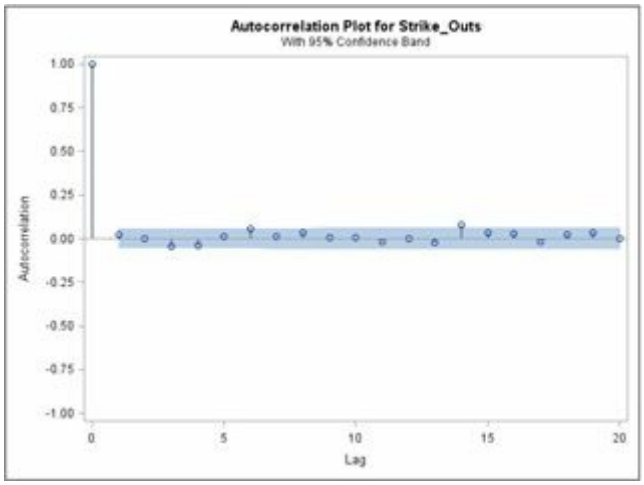


Figure 5.4: Autocorrelation Plot for Strike Outs



Figures 5.3 and 5.4 present the lagged autocorrelations among successive estimates of the posterior means of SALARY and STRIKE_OUTS. Specifically, the autocorrelations plotted on the Y-axis represent the relationships of successive parameter estimates for each variable at a defined number of lagged intervals (length of lag on the X-axis). Here, we use the default of 20 lagged autocorrelations, though this can be changed through use of the NLAG= option. Other plot options are set to the procedure defaults. Both figures show that the autocorrelations of the posterior means at 1,2,…,20 length lags in the cycle of iterations are all negligible, varying randomly about zero. These graphical results also reveal no problems with the convergence of the MCMC chain or the independence of repetition draws of imputed values after the initial burn in period is completed.

As previously stated, the analytic goal in this exercise is to estimate a linear regression model to predict player SALARY based on selected covariates including STRIKE_OUTS, RUNS_BATTED_IN, WALKS, STOLEN_BASES and ERRORS. We now use PROC REG to estimate the separate linear regression models for each of M=10 imputed data sets. We use the BY _IMPUTATION_ statement to request separate regressions for each imputation

repetition and create an output data set through use of the OUTEST=OUT_EST_C5EX1 with the COV option on the PROC statement. This output data set contains the regression parameter estimates and covariance information for each of the ten regression analyses. The statistics from the individual MI repetition regression analyses can be identified by the _IMPUTATION_ variable included in the output data set.

```
proc reg data=outc5ex1 outest=out_est_c5ex1 covout;
 model salary =strike_outs runs_batted_in walks
stolen_bases errors;
 by _imputation_;
run;
```

Output 5.4: Regression Output for the First Imputation Repetition Data Set

The REG Procedure
Model: MODEL1
Dependent Variable: Salary

Imputation Number=1

Number of Observations Read	337
Number of Observations Used	337

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	253049493	50609899	66.23	<.0001
Error	331	252953271	764209		
Corrected Total	336	506002765			

Root MSE	874.19064	R-Square	0.5001
Dependent Mean	1260.39763	Adj R-Sq	0.4925
Coeff Var	69.35832		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	188.69012	100.53166	1.88	0.0614
Strike_Outs	1	-9.85895	2.24949	-4.38	<.0001
Runs_batted_in	1	30.43928	2.77329	10.98	<.0001
Walks	1	8.94663	3.01463	2.97	0.0032
Stolen_bases	1	13.65555	4.44081	3.08	0.0023
Errors	1	-20.23032	8.52049	-2.37	0.0182

Output 5.4 illustrates the standard linear regression output for only the first imputation repetition. As explained above, use of the BY _IMPUTATION_ statement produces similar output for each of the imputed data sets (results for 2-10 not shown here).

The PROC PRINT syntax below produces a list report and displays the structure and actual values of the parameter estimates and covariance estimates that PROC REG has output to the file out_est_c5ex1.

```
proc print data=out_est_c5ex1;
run;
```

Output 5.5: Listing of Estimate Output Data Set from PROC REG (First MI Repetition Only)

Obs	_Imputation_	_MODEL_	_TYPE_	_NAME_	_DEPVAR_	_RMSE_	Intercept	Strike_Outs	Runs_batted_in	Walks	Stolen_bases	Errors	Salary
1	1	MODEL1	PARMS		Salary	874.191	188.69	-9.8589	30.4393	8.9466	13.6556	-20.230	-1
2	1	MODEL1	COV	Intercept	Salary	874.191	10106.62	-71.7614	-19.3834	-28.8774	-26.4562	-251.169	.
3	1	MODEL1	COV	Strike_Outs	Salary	874.191	-71.76	5.0602	-3.1320	-1.5350	-1.1026	-1.992	.
4	1	MODEL1	COV	Runs_batted_in	Salary	874.191	-19.38	-3.1320	7.6911	-3.9850	1.7324	-2.499	.
5	1	MODEL1	COV	Walks	Salary	874.191	-28.88	-1.5350	-3.9850	9.0880	-3.5834	0.348	.
6	1	MODEL1	COV	Stolen_bases	Salary	874.191	-26.46	-1.1026	1.7324	-3.5834	19.7208	-3.638	.
7	1	MODEL1	COV	Errors	Salary	874.191	-251.17	-1.9922	-2.4994	0.3478	-3.6380	72.599	.

Output 5.5 provides a partial listing of the out_est_c5ex1 data set including parameter estimates and covariances. To save space, the displayed output is again limited only to that portion that applies to the first of our 10 imputed data sets. The key variables are _IMPUTATION_ with values of 1–10, _DEPVAR_ with the dependent variable SALARY specified, along with the predictor variables in columns to the right of _RMSE_. Use of the OUTEST option with a COVOUT option ensures that both estimates of the regression parameters, residual variance (_RMSE_) and covariance information for the regression parameter estimates is available to PROC MIANALYZE.

In the third and final step of our analysis of the MLB baseball salaries data, we use PROC MIANALYZE with a DATA=out_est_c5ex1 statement and list the model predictors including the intercept in the MODELEFFECTS statement. This instructs the procedure to calculate MI estimates of the regression model parameters for our analytic model, along with their standard errors and confidence intervals. The order of the variables in the MODELEFFECTS statement must match the order used in the PROC REG analysis with the intercept always listed first.

```
proc mianalyze data=out_est_c5ex1;
 modeleffects intercept strike_outs runs_batted_in
walks stolen_bases errors;
run;
```

Output 5.6: Model and Variance Information and Parameter Estimates

The MIANALYZE Procedure

Model Information	
Data Set	WORK.OUT_EST_C5EX1
Number of Imputations	10

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
intercept	159.668190	10363	10538	32401	0.016949	0.016727	0.998330
strike_outs	0.291767	5.151487	5.472431	2616.6	0.062301	0.059366	0.994098
runs_batted_in	0.283135	7.898331	8.209780	6253.6	0.039432	0.038244	0.996190
walks	1.034493	9.229291	10.367234	747.01	0.123297	0.112137	0.988911
stolen_bases	5.504655	20.143218	26.198339	168.48	0.300603	0.240094	0.976554
errors	3.535091	74.354427	78.243027	3643.7	0.052298	0.050220	0.995003

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept	214.226920	102.655701	13.0179	415.4359	32401	188.690124	231.686974	0	2.09	0.0369
strike_outs	-10.667717	2.339323	-15.2548	-6.0806	2616.6	-11.561531	-9.858949	0	-4.56	<.0001
runs_batted_in	30.576828	2.865271	24.9599	36.1937	6253.6	29.855707	31.582397	0	10.67	<.0001
walks	9.114325	3.219819	2.7934	15.4353	747.01	7.452839	10.579511	0	2.83	0.0048
stolen_bases	12.862005	5.118431	2.7575	22.9665	168.48	9.189991	17.479949	0	2.51	0.0129
errors	-16.753901	8.845509	-34.0965	0.5887	3643.7	-20.230324	-14.661094	0	-1.89	0.0583

Output 5.6 includes the PROC MIANALYZE Model and Variance Information along with the Parameter Estimates table for the MI regression of salary on strike outs, runs batted in, walks, stolen bases, and errors. The MI parameter estimates and *t* tests suggest that runs batted in, walks, and stolen bases have a significant and positive effect on player salaries. Each additional error predicts a loss of about 16.75*1000 = $16,750 in salary and each strike out results in an estimated salary loss of about $10,667 (holding all other predictors in the model constant). The MI Student *t* test suggests that all predictors in the regression model are significantly different from zero (or nearly so) at the α=0.05 level.

The Variance Information table in the output summarizes the MI estimates of the within, between and total variance of the estimated linear regression parameters. This table also provides the PROC MI variance ratio type statistics: RE that measures the efficiency of using the finite (M=10) vs. an infinite number of MI repetitions; and the RIV and FMI statistics that measure the increase in variance of estimates or "information loss" in the MI analysis due to the missing data (relative to analysis of a completely observed data set).

5.3 Imputation of Continuous Variables with Mixed Covariates and a Monotone Missing Data Pattern Using the Regression and Predictive Mean Matching Methods

This section presents an example that illustrates and compares the regression and predictive mean matching (PMM) imputation methods for imputation of continuous variables with a monotone missing data pattern and predictors that include a mix of continuous and classification type variables. Section 5.3.1 uses the regression method, while 5.3.2 repeats the example imputation and analysis using the PMM method.

Data from the NHANES 2009–2010 survey are used in these example applications (stored in **c5_ex2)**. The imputation focuses on NHANES 2009–2010 interviewed persons who were age 8 and older and participated in the Medical Examination Component (MEC) examination phase of the study. The PROC MI imputation of missing data will apply to all eligible cases. However, the analytic goal in this exercise is to estimate mean blood pressure and mean pulse rates by gender among those 20 years of age or older. To restrict the final analysis to these age 20+ subpopulations of men and women, we will use an estimation approach that is appropriate for subpopulation analysis of complex sample data (i.e., subpopulations declared in a DOMAIN statement).

These examples demonstrate use of two appropriate methods for imputation of continuous variables with a monotone missing data pattern. We also demonstrate our recommended procedure (see Chapter 4) for incorporating the NHANES analysis weights and design variables with a WHERE statement in the imputation model. The CLASS statement will be used to declare classification variables; the SURVEYMEANS procedure with a DOMAIN statement will be used for a complex sample design corrected analysis of mean blood pressure and pulse by gender and age group of interest; and the EDF option will be used in PROC MIANALYZE to specify the approximate complete data degrees of freedom for the NHANES complex sample design.

5.3.1 Imputation of Continuous Variables with Mixed Covariates and a Monotone Missing Data Pattern Using the Regression Method

NHANES 2009–2010 variables used in this example are:

RIDRETH1: Race/Ethnicity, categorical variable with no missing data, categories are 1=Mexican American, 2=Other Hispanic 3=Non-Hispanic White, 4=Non-Hispanic Black 5=Other / Multiracial

RIDAGEYR: Age in Years, continuous with no missing data, values range

from 0-80 with 80+ topcoded as 80 years of age

RIAGENDR: Gender, categorical with no missing data, 1=Male, 2=Female

WTMEC2YR: Weight used for analysis of the MEC data for the 2 year period, continuous with no missing data

SDMVSTRA: Complex Sample Design Strata variable, categorical with no missing data

SDMVPSU: Complex Sample Design PSU or Cluster variable, categorical with no missing data

SDMVSTRA_SDMVPSU: Combined Complex Sample Variable, categorical with no missing data

RIDSTATR: Interview Status, categorical variable with no missing data, 1=Interviewed Only, 2=Interviewed and Medical Exam

IMPUTE_BPXPLS: Imputation Flag variable, 1=imputed value, 0=observed value

IMPUTE_BPXDI1_1: Imputation Flag variable, 1=imputed value, 0=observed value

BPXPLS: Pulse Rate measured in beats per minute, continuous with some missing data

BPXDI1_1: Recoded diastolic blood pressure with a value of 0 set to missing, continuous with some missing data

Following our standard procedure, we start with an analysis of the rates and pattern of missing data.

```
proc mi nimpute=0 data=c5_ex2 simple;
 where ridstatr=2 and ridageyr >=8;
 var riagendr ridreth1 ridageyr wtmec2yr
sdmvstra_sdmvpsu bpxpls bpxdi1_1;
run;
```

Before discussing the missing data pattern and rates, the use of the WHERE statement in the command syntax merits further explanation. Like many complex sample surveys, the NHANES 2009-2010 survey was collected in defined parts such as the general health history interview, the MEC examination, laboratory components, dietary section, and limited access questions. When using data from various parts of the survey, the decision about which cases ought to be imputed or left as valid missing becomes more complex. In this example, we intend to impute missing data only for respondents that should have provided measurements for pulse rate and blood pressure, that is, those that were age 8 and older and also participated in the MEC portion of the NHANES survey. We

use the WHERE statement in the following code to define a working data set of these cases (n=8,185) and also to avoid imputation of pulse and blood pressure for those under age 8 or who did not participate in the medical examination.

Since we will be imputing missing data for only a subset of the full complex sample data set for NHANES 2009–2010, we will use the modified approach outlined in Section 4.5. We employ a strategy in which the restricted PROC MI output data set will include jackknife repeated replication (JRR) replicate weights based on the full-sample NHANES 2009–2010 data set. This ensures that the complex sample design representation of the NHANES 2009–2010 is not distorted by the WHERE statement and thus, JRR variance estimation for the SURVEY procedure analysis will accurately reflect the full sample design. At the same time, it allows use of a subset of data records during the imputation without sacrificing correct variance estimation. Note that these steps are needed only when a subset of data records are used in the imputation of missing data derived from a complex sample design survey. The third step of the MI process is not affected by this process and as usual, combines results from the SURVEY procedure analyses to produce the desired MI estimates and CIs.

Output 5.7: Missing Data Patterns

Missing Data Patterns

Group	RIAGENDR	RIDRETH1	RIDAGEYR	WTMEC2YR	sdmvstra_sdmvpsu	BPXPLS	bpxdi1_1	Freq	Percent	Group Means						
										RIAGENDR	RIDRETH1	RIDAGEYR	WTMEC2YR	sdmvstra_sdmvpsu	BPXPLS	bpxdi1_1
1	X	X	X	X	X	X	X	7459	91.13	1.500201	2.778925	40.163293	33043	814.948519	73.792734	66.810296
2	X	X	X	X	X	X	.	377	4.61	1.562334	2.737401	40.058355	30294	818.705570	76.175066	.
3	X	X	X	X	X	.	.	349	4.26	1.587393	3.063037	37.871060	30446	821.048711	.	.

Univariate Statistics

Variable	N	Mean	Std Dev	Minimum	Maximum	Missing Values	
						Count	Percent
RIAGENDR	8185	1.50678	0.49998	1.00000	2.00000	0	0.00
RIDRETH1	8185	2.78913	1.14493	1.00000	5.00000	0	0.00
RIDAGEYR	8185	40.06072	22.18483	8.00000	80.00000	0	0.00
WTMEC2YR	8185	32806	25176	4292	158147	0	0.00
sdmvstra_sdmvpsu	8185	815.38167	41.49375	751.00000	892.00000	0	0.00
BPXPLS	7836	73.90735	12.51089	40.00000	126.00000	349	4.26
bpxdi1_1	7459	66.81030	13.43366	2.00000	134.00000	726	8.87

For the age-restricted data set, Output 5.7 shows missing data on both BPXPLS (Pulse Rate) and BPXDI1_1 (Diastolic Blood Pressure) with all other variables fully observed (n=8,185). This output presents a monotone missing data pattern with missing data on the two continuous variables just mentioned. Note from the Univariate Statistics output that although RIAGENDR, RIDRETH1, and SDMVSTRA_SDMVPSU are classification variables, the univariate statistics are generated. In the actual PROC MI run below with the monotone method, these variables are declared as CLASS variables in the command syntax and are handled accordingly in the regression modeling that generates the imputations for the pulse rate and diastolic blood pressure missing data.

One step that is often overlooked in regression imputation of continuous variables is a standard check of the linear regression model assumptions including: normality, model specification, and regression diagnostics. Of course, any preliminary diagnostics are limited to the observed data. But even with this limitation, it is important to conduct some preliminary investigation of variable and model properties to ensure that the estimated regression that is the basis for subsequent imputations conforms in a reasonable way to the underlying assumptions of the model.

We will not illustrate a comprehensive diagnostic investigation of candidate regression models for pulse rate and diastolic blood pressure here. For demonstration purposes, we will examine the distributional properties of the observed measurements of these two variables to establish that the normality (or symmetry) assumption for the regression model is not seriously violated. Graphical displays are good options for examining these distributions, therefore we use PROC SGPLOT to produce unweighted histograms with a normal density curve.

```
title "Pulse Rate";
proc sgplot data=c5_ex2;
 where ridstatr=2 and ridageyr >=8;
 histogram bpxpls / nbins=100;
 density bpxpls;
run;

title "Diastolic Blood Pressure";
proc sgplot data=c5_ex2;
 where ridstatr=2 and ridageyr >=8;
 histogram bpxdi1_1/ nbins=100;
 density bpxdi1_1;
run;
```

Figure 5.5: Histogram of Pulse Rate

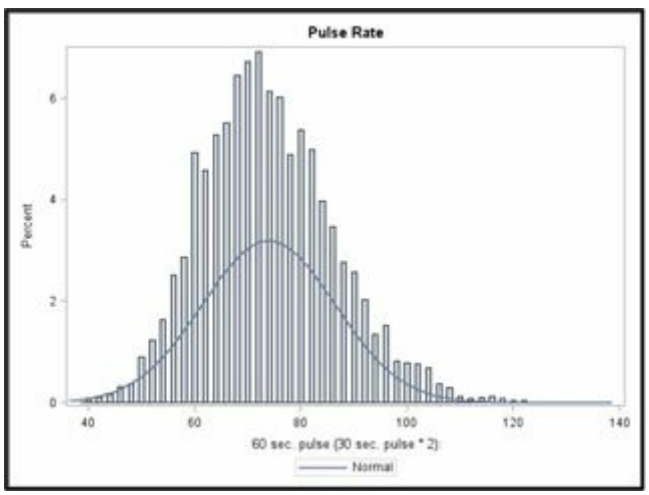
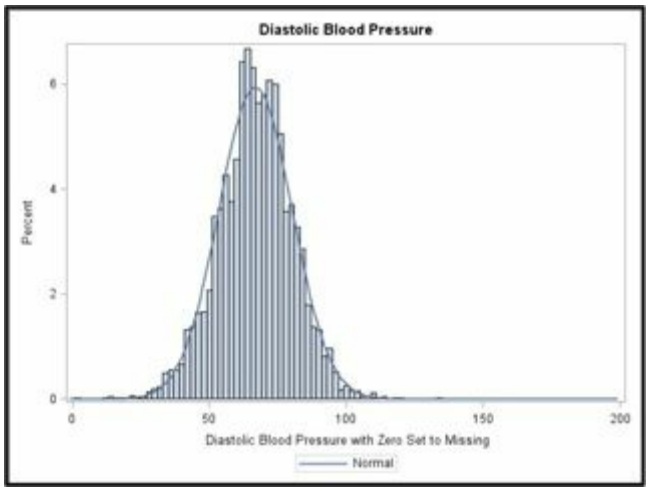


Figure 5.6: Histogram of Diastolic Blood Pressure



Figures 5.5 and 5.6 present histograms of the observed pulse rate and diastolic blood pressure measurements. Although neither variable has a perfectly normal distribution, the distributions of observations are not highly asymmetric and no transformations will be used in the monotone regression imputation process.

We should note here that these exploratory graphics are unweighted and reflect the distributions of pulse rate and diastolic blood pressure among the observed sample cases aged 8+ in the NHANES 2009–2010 MEC data set. Due to differential sample selection probabilities for respondents in differing demographic groups, they are not equivalent to graphs that depict the estimated distributions of these physiological measurements in the U.S. household population that the weighted NHANES 2009–2010 sample represents.

As described above, we next create (from PROC SURVEYMEANS, though any SURVEY procedure will do) replicate weights based on the entire data set using the strata and cluster variables (SDMVSTRA and SDMVPSU) and the MEC 2-year weight (WTMEC2YR). PROC SURVEYMEANS with a VARMETHOD=JACKKNIFE (OUTWEIGHTS=) option is used to create an output data set with replicate weights to be used in subsequent analysis of completed data sets with the jackknife repeated replication method rather than the default Taylor Series Linearization method. We also request an output data set of jackknife coefficients for use in subsequent analyses with the OUTJKCOEFS=REPWT_COEF_C5_EX2 option of the SURVEYMEANS statement. These coefficients are used to correctly adjust the JRR variance estimates for the count of "delete one" replicates created for each design stratum. For example, the NHANES 2009–2010 data set has 2 or 3 clusters per stratum and use of the coefficients ensures that the variances are not overestimated as JRR variance estimation (VARMETHOD=JK) will default to a value of 1.0 for the JRR coefficients if replicate weights are provided without the JKCOEFS option in the REPWEIGHTS statement.

The following syntax creates a data set called repwt_c5_ex2 which contains all of the variables already present in the data set along with replicate weights named REPWT_1-REPWT_31. These weights represent the stratification and clustering of the NHANES data and are designed for use with the jackknife repeated replication method for variance estimation. By default, the SURVEYMEANS procedure without a VAR statement will perform a means analysis for each numeric variable in the data set along with creation of the replicate weights (results not shown here). Because we are primarily using the procedure to generate replicate weights for later use, we are not interested in the estimated means of any particular variables. As a check, we use PROC MEANS to examine the results from a means analysis of the 31 replicate weights. We also include a printout of the JK coefficients and then use the values directly in later analyses.

```
proc surveymeans data=c5_ex2 varmethod=jk
(outweights=repwt_c5_ex2
       outjkcoefs=repwt_coef_c5_ex2);
 strata sdmvstra; cluster sdmvpsu; weight wtmec2yr;
 var ridageyr;
run;

proc means data=repwt_c5_ex2;
 var repwt_1-repwt_31;
run;

proc print data=repwt_coef_c5_ex2;
run;
```

Output 5.8: Distributions of Jackknife Replicate Weights and Listing of Jackknife Coefficients

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
RepWt_1	Replicate Weight	10253	30175.67	27634.79	0	250641.69
RepWt_2	Replicate Weight	10253	28722.94	24450.60	0	243500.27
RepWt_3	Replicate Weight	10253	29698.16	27734.13	0	216097.02
RepWt_4	Replicate Weight	10253	29300.45	26717.12	0	218383.42
RepWt_5	Replicate Weight	10253	28961.36	27179.80	0	217518.30
RepWt_6	Replicate Weight	10253	29917.25	28339.60	0	250634.34
RepWt_7	Replicate Weight	10253	30059.69	26605.70	0	221308.53
RepWt_8	Replicate Weight	10253	28838.92	24813.10	0	164026.58
RepWt_9	Replicate Weight	10253	29615.31	28950.30	0	316293.84
RepWt_10	Replicate Weight	10253	29083.30	26510.66	0	190376.41
RepWt_11	Replicate Weight	10253	29564.72	26775.46	0	224849.62
RepWt_12	Replicate Weight	10253	29333.89	25791.86	0	159443.08
RepWt_13	Replicate Weight	10253	30224.28	27336.92	0	215092.26
RepWt_14	Replicate Weight	10253	28674.33	24418.54	0	190803.73
RepWt_15	Replicate Weight	10253	29440.93	25211.80	0	229237.26
RepWt_16	Replicate Weight	10253	29457.68	25339.57	0	190660.66
RepWt_17	Replicate Weight	10253	29320.32	26051.88	0	222826.59
RepWt_18	Replicate Weight	10253	29578.29	27023.78	0	226705.10
RepWt_19	Replicate Weight	10253	29363.90	24937.37	0	173156.65
RepWt_20	Replicate Weight	10253	29534.71	25141.36	0	159251.20
RepWt_21	Replicate Weight	10253	29457.07	25548.71	0	189908.14
RepWt_22	Replicate Weight	10253	29441.54	25385.09	0	170647.66
RepWt_23	Replicate Weight	10253	29229.77	24778.36	0	158146.92
RepWt_24	Replicate Weight	10253	29336.54	25056.82	0	158146.92
RepWt_25	Replicate Weight	10253	29781.60	25214.15	0	158146.92
RepWt_26	Replicate Weight	10253	29579.07	25360.87	0	191206.32
RepWt_27	Replicate Weight	10253	29319.54	25204.20	0	221647.48
RepWt_28	Replicate Weight	10253	29403.00	24785.94	0	173419.73
RepWt_29	Replicate Weight	10253	29495.61	24671.83	0	173726.30
RepWt_30	Replicate Weight	10253	29579.09	24476.93	0	158146.92
RepWt_31	Replicate Weight	10253	29319.51	24229.20	0	158146.92

Obs	Replicate	Donor Stratum	JKCoefficient
1	1	1	0.50000
2	2	1	0.50000
3	3	2	0.50000
4	4	2	0.50000
5	5	3	0.50000
6	6	3	0.50000
7	7	4	0.50000
8	8	4	0.50000
9	9	5	0.50000
10	10	5	0.50000
11	11	6	0.50000
12	12	6	0.50000
13	13	7	0.50000
14	14	7	0.50000
15	15	8	0.50000
16	16	8	0.50000
17	17	9	0.50000
18	18	9	0.50000
19	19	10	0.50000
20	20	10	0.50000
21	21	11	0.50000
22	22	11	0.50000
23	23	12	0.66667
24	24	12	0.66667
25	25	12	0.66667
26	26	13	0.50000
27	27	13	0.50000
28	28	14	0.50000
29	29	14	0.50000
30	30	15	0.50000
31	31	15	0.50000

Output 5.8 provides a check of the distributions of the replicate weights and a listing of the JK coefficient values, replicates, and donor stratum. The results indicate that the full number of respondents who completed both the general interview and the MEC portion of the NHANES survey (n=10,253) are included. As expected for the JRR replicate weights, values range from a minimum of 0 to a maximum of 316293.84. The JK coefficients are 0.50 for all strata with 2 PSUs and 0.66667 for the one stratum with 3 PSUs.

The PROC MI imputation syntax detailed below includes use of a SEED value,

a CLASS statement to declare categorical variables in the imputation model, a MONOTONE REGRESSION statement to request the linear regression imputation method appropriate for continuous variables with a monotone missing data pattern, and a VAR statement to list the variables to be used in the imputation. As described above and in Section 4.5, the WHERE statement restricts the imputation to the 8,185 respondents 8+ years old and MEC participants. The VAR statement specifies the variables to be used in the imputation with the BPXPLS and BPXDI1_1 variables listed in the next to last and last positions. This order reflects the monotone missing data pattern of the data. The SDMVSTRA_SDMVPSU and WTMEC2YR variables serve as fixed effect covariates in the imputation model and account for the complex sample design and weighting during the imputation. Five repetitions of the imputed NHANES data set are stored in an output data set named c5ex2imp_reg. Finally, the REGRESSION statement includes an option to list imputation details for both imputed variables (DETAILS).

```
proc mi data=repwt_c5_ex2 nimpute=5 seed=41
out=c5ex2imp_reg;
 class sdmvstra_sdmvpsu ridreth1 riagendr;
 where ridageyr >=8 and ridstatr=2;
 monotone regression (bpxpls bpxdi1_1/details);
 var riagendr ridreth1 ridageyr sdmvstra_sdmvpsu
wtmec2yr bpxpls bpxdi1_1;
run;
```

Output 5.9: Regression Model Details for Diastolic Blood Pressure

Regression Models for Monotone Method										
Imputed Variable	Effect	RIAGENDR	RIDRETH1	sdmvstra_sdmvpsu	Obs-Data	Imputation				
						1	2	3	4	5
bpxdi1_1	Intercept	.	.	.	0.05817	0.050746	0.045522	0.031294	0.048095	0.067818
bpxdi1_1	RIAGENDR	1.000000	.	.	0.10214	0.106121	0.090949	0.103857	0.088446	0.101756
bpxdi1_1	RIDRETH1	.	1.000000	.	0.03525	0.041936	0.011407	0.056590	0.037533	0.031108
bpxdi1_1	RIDRETH1	.	2.000000	.	0.12524	0.167504	0.094849	0.101877	0.067808	0.172978
bpxdi1_1	RIDRETH1	.	3.000000	.	-0.22586	-0.230046	-0.235210	-0.191930	-0.233806	-0.223593
bpxdi1_1	RIDRETH1	.	4.000000	.	0.17376	0.148831	0.172818	0.142018	0.175245	0.180577
bpxdi1_1	RIDAGEYR	.	.	.	0.32967	0.338203	0.337904	0.336335	0.334381	0.338286
bpxdi1_1	sdmvstra_sdmvpsu	.	.	751.000000	0.12446	0.071271	0.145016	0.078665	0.087905	0.143535
bpxdi1_1	sdmvstra_sdmvpsu	.	.	752.000000	-0.08760	-0.085747	0.007233	-0.108451	-0.139482	-0.047740
bpxdi1_1	sdmvstra_sdmvpsu	.	.	761.000000	-0.15192	-0.107266	-0.107643	-0.148978	-0.177399	-0.152496
bpxdi1_1	sdmvstra_sdmvpsu	.	.	762.000000	-0.17480	-0.242590	-0.282320	-0.218285	-0.119665	-0.154322
bpxdi1_1	sdmvstra_sdmvpsu	.	.	771.000000	-0.01360	0.088927	0.014850	-0.050086	0.038237	-0.032924
bpxdi1_1	sdmvstra_sdmvpsu	.	.	772.000000	0.00904	-0.093454	0.034715	0.101192	-0.044724	-0.065947
bpxdi1_1	sdmvstra_sdmvpsu	.	.	781.000000	0.22099	0.273059	0.173974	0.216892	0.261614	0.246764
bpxdi1_1	sdmvstra_sdmvpsu	.	.	782.000000	-0.52748	-0.633524	-0.497957	-0.504819	-0.442573	-0.585657
bpxdi1_1	sdmvstra_sdmvpsu	.	.	791.000000	0.39569	0.325020	0.334508	0.359273	0.446173	0.424001
bpxdi1_1	sdmvstra_sdmvpsu	.	.	792.000000	-0.27756	-0.308579	-0.284162	-0.351211	-0.142934	-0.335694
bpxdi1_1	sdmvstra_sdmvpsu	.	.	801.000000	0.22604	0.219802	0.266052	0.250550	0.211821	0.246730
bpxdi1_1	sdmvstra_sdmvpsu	.	.	802.000000	0.13406	0.184131	0.132235	0.099126	0.075719	0.155399
bpxdi1_1	sdmvstra_sdmvpsu	.	.	811.000000	0.12352	0.129462	0.085934	0.134654	0.139707	0.070972
bpxdi1_1	sdmvstra_sdmvpsu	.	.	812.000000	-0.15442	-0.162753	-0.116607	-0.114287	-0.181090	-0.169583
bpxdi1_1	sdmvstra_sdmvpsu	.	.	821.000000	-0.04672	0.025123	0.065668	0.047115	-0.036416	-0.095370
bpxdi1_1	sdmvstra_sdmvpsu	.	.	822.000000	0.28744	0.215214	0.377625	0.264907	0.309689	0.323267
bpxdi1_1	sdmvstra_sdmvpsu	.	.	831.000000	0.24819	0.152197	0.291122	0.237876	0.235160	0.292221
bpxdi1_1	sdmvstra_sdmvpsu	.	.	832.000000	-0.11746	-0.102978	-0.125652	-0.119469	-0.052669	-0.062835
bpxdi1_1	sdmvstra_sdmvpsu	.	.	841.000000	0.00522	0.081026	0.032853	-0.005203	-0.051420	-0.126626
bpxdi1_1	sdmvstra_sdmvpsu	.	.	842.000000	0.16411	0.275240	0.113719	0.044848	0.133778	0.235921
bpxdi1_1	sdmvstra_sdmvpsu	.	.	851.000000	-0.37862	-0.363803	-0.350520	-0.352681	-0.359030	-0.321910
bpxdi1_1	sdmvstra_sdmvpsu	.	.	852.000000	-0.38956	-0.460195	-0.418189	-0.401144	-0.428264	-0.495171
bpxdi1_1	sdmvstra_sdmvpsu	.	.	861.000000	0.42564	0.349748	0.413097	0.496742	0.486363	0.513234
bpxdi1_1	sdmvstra_sdmvpsu	.	.	862.000000	-0.05444	0.053915	0.028054	-0.014217	-0.039854	-0.019461
bpxdi1_1	sdmvstra_sdmvpsu	.	.	863.000000	0.23367	0.139140	0.296472	0.193447	0.243586	0.248233
bpxdi1_1	sdmvstra_sdmvpsu	.	.	871.000000	-0.10503	-0.074334	-0.118628	-0.175149	-0.186073	-0.103365
bpxdi1_1	sdmvstra_sdmvpsu	.	.	872.000000	0.21561	0.245138	0.227286	0.227686	0.086788	0.233163
bpxdi1_1	sdmvstra_sdmvpsu	.	.	881.000000	-0.06357	-0.163006	-0.094259	-0.101987	-0.050906	-0.105392
bpxdi1_1	sdmvstra_sdmvpsu	.	.	882.000000	-0.17554	-0.162757	-0.277040	-0.148753	-0.143785	-0.123518
bpxdi1_1	sdmvstra_sdmvpsu	.	.	891.000000	-0.09679	0.065677	-0.262350	0.002706	-0.118295	-0.075426
bpxdi1_1	WTMEC2YR	.	.	.	0.21580	0.209152	0.199365	0.207639	0.205666	0.221398
bpxdi1_1	BPXPLS	.	.	.	0.09624	0.100870	0.109593	0.101394	0.100505	0.099007

Output 5.10: Variance Information and Parameter Estimates for Diastolic Blood Pressure and Pulse Rate

Variance Information

Variable	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
BPXPLS	0.001775	0.019089	0.021219	376.74	0.111567	0.104866	0.979458
bpxdi1_1	0.000695	0.022122	0.022956	2188.6	0.037706	0.036972	0.992660

Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
BPXPLS	73.933400	0.145668	73.64698	74.21982	376.74	73.859955	73.964890	0	507.55	<.0001
bpxdi1_1	66.846870	0.151512	66.54975	67.14399	2188.6	66.806541	66.877386	0	441.20	<.0001

Output 5.9 includes the detailed regression model output for the imputed variable BPXDI1_1 requested by adding the DETAILS option to the MONOTONE statement. The output presents estimated parameters for each continuous predictor and each level (except the reference category) of the classification variables.

In Output 5.10, the PROC MI Variance Information table shows small increases in variance and fraction of missing information for both imputed variables. The Parameter Estimates table presents the standard MI output for the mean pulse rate and diastolic blood pressure. Unfortunately, these default MI estimates for means produced in PROC MI are not the correct weighted means and the corresponding estimates of variances and standard errors, and inferential statistics are not correctly adjusted for the complex sample design of the NHANES. We will need to use SURVEY procedures in the MI repetition estimation (step 2) and the EDF= option in PROC MIANALYZE in the MI "combining" step (step 3) to correctly account for the complex sample design and obtain unbiased estimates of the population means and correct estimates of standard errors, confidence intervals and test statistics.

In multiple imputation analysis it is useful to create an "imputation flag" variable that can be used to later discriminate which variable values in the MI data set were observed and which were missing and subsequently imputed. In this example, two binary "imputation flags" for the pulse rate and diastolic blood pressure variables were created in a preliminary data step and named IMPUTE_BPXPLS and IMPUTE_BPXDI1_1. Each binary indicator is set to 1 if the value of that variable has been imputed and 0 if the value was observed and therefore not imputed. After running PROC MI to create the imputations, it is useful to compare the distributions (e.g., means, standard deviations, percentiles) of the imputed and observed values. To demonstrate, we perform a

means analysis of BPXPLS with IMPUTE_BPXPLS used as a CLASS variable. This check is primarily to ensure that the imputed means are not extremely different from the observed values. Here again, note that this simple quality check does not use the NHANES analysis weight so these estimates should not be interpreted as unbiased estimates of the population values.

```
proc means data=c5ex2imp_reg;
 class _imputation_ impute_bpxpls;
 var bpxpls;
run;
```

Output 5.11: Means of Imputed and Observed Pulse Rate by Imputation

Analysis Variable : BPXPLS 60 sec. pulse (30 sec. pulse * 2):							
Imputation Number	impute_bpxpls	N Obs	N	Mean	Std Dev	Minimum	Maximum
1	0	7836	7836	73.9073507	12.5108899	40.0000000	126.0000000
	1	349	349	74.7342272	12.6208907	39.3485965	113.8488622
2	0	7836	7836	73.9073507	12.5108899	40.0000000	126.0000000
	1	349	349	72.7958003	11.9594451	38.4251623	113.8987509
3	0	7836	7836	73.9073507	12.5108899	40.0000000	126.0000000
	1	349	349	75.0617496	12.3086266	41.7801505	102.3872466
4	0	7836	7836	73.9073507	12.5108899	40.0000000	126.0000000
	1	349	349	75.2568020	12.0942161	32.4567069	104.6681103
5	0	7836	7836	73.9073507	12.5108899	40.0000000	126.0000000
	1	349	349	74.7427787	12.1189954	41.5390764	107.4729607

Output 5.11 includes estimated descriptive statistics for pulse rate by imputation repetition number and imputation flag (IMPUTE_BPXPLS). Within each imputation repetition, we see only minor differences between imputed mean pulse rates versus observed mean pulse rates. In general, the imputation methods available in PROC MI assume that the missing data are generated by a missing at random (MAR) mechanism. The rates of missing data and underlying true values may well differ across key subgroups (e.g., men and women, young and old) in the study population. Therefore, it is reasonable to expect some differences in the distributions of the observed and the imputed values. In general though, these distributions of observed and imputed values for any variables should not be extremely different. If simple exploratory analysis of the form illustrated in Output 5.11 shows large differences (observed versus imputed) in the measures of central tendency (means, medians) or the extremes of the distributions (e.g., 95th percentile, maximum), there may be a problem

with poor fit or a missing variable in the imputation model. If this is the case, more in-depth investigation of the set of variables included in the imputation model or the specific form of the regression or discriminant classification model used to impute the specific variable is warranted. For example, if the monotone linear regression model for imputing the NHANES diastolic blood pressure variable, BPXDI1_1, includes only a linear term for age (RIDAGEYR) and the relationship with age was actually convex quadratic in form, the imputed values might overestimate the true distribution at older ages. To reemphasize our previous point, the imputation models available in PROC MI are powerful tools but it remains the responsibility of the analyst to choose both the set of variables included in the imputation model and to specify the correct functional form of the model for each variable in that model.

Now that the multiple imputation step is complete, we turn to our analysis goal which is to estimate mean pulse rate and diastolic blood pressure by gender in the subpopulation of NHANES subjects aged 20 and older who participated in the MEC. Since the NHANES is a complex sample survey data set, PROC SURVEYMEANS is used to generate the estimates of subpopulation means and their standard errors. In this step, we use the replicate weights (REPWT_1-REPWT_31) generated prior to imputation with the jackknife repeated replication method for variance estimation along with the 2-year MEC weight (WTMEC2YR). The jackknife coefficient values are inserted directly in the REPWEIGHTS statement to highlight the actual values and with a small number of replicates, this is a viable option. For higher numbers of replicates, use of an output data set is an equally effective method of supplying the coefficients to the SURVEY procedure. The specification (AGE20P*RIAGENDR) in the DOMAIN statement instructs PROC SURVEYMEANS to generate separate estimates for the age and gender subpopulations. In addition, the BY _IMPUTATION_ statement is used to produce separate SURVEYMEANS analyses within each imputed data set. This is an important distinction. The BY statement is used to process entire repetition data sets while the DOMAIN statement defines subpopulations within the repetition data sets.

In the following code, we use ODS OUTPUT DOMAIN=c5ex2imp_reg_m to create an output data set containing domain statistics, that is, estimated means and design-corrected standard errors for each level of the domain categories for each the five imputed data sets. The Output 5.12 listing of the output data set from PROC PRINT statement in the command syntax displays the contents and structure of **c5ex2imp_reg_m**.

```
proc surveymeans data=c5ex2imp_reg varmethod=jk;
 repweights repwt_1-repwt_31 / jkcoefs= .5 .5 .5 .5 .5
```

```
.5 .5 .5 .5 .5 .5 .5
    .5 .5 .5 .5 .5 .5 .5 .5 .5 .5 .66667 .66667 .66667
.5 .5 .5 .5 .5 .5 ;
 weight wtmec2yr;
 by _imputation_;
 domain age20p*riagendr;
 var bpxpls bpxdi1_1;
 ods output domain=c5ex2imp_reg_m;
run;

proc print noobs data=c5ex2imp_reg_m;
 var _imputation_ age20p riagendr varname varlabel mean
stderr;
run;
```

Output 5.12: Listing of the Output Domain Data Set from PROC SURVEYMEANS

Imputation	age20p	RIAGENDR	VarName	VarLabel	Mean	StdErr
1	0	1	BPXPLS	60 sec. pulse (30 sec. pulse * 2):	74.995621	0.408191
1	0	1	bpxdi1_1	Diastolic Blood Pressure with Zero Set to Missing	57.569387	0.935145
1	0	2	BPXPLS	60 sec. pulse (30 sec. pulse * 2):	80.926681	0.473966
1	0	2	bpxdi1_1	Diastolic Blood Pressure with Zero Set to Missing	58.109117	0.943203
1	1	1	BPXPLS	60 sec. pulse (30 sec. pulse * 2):	70.607455	0.306198
1	1	1	bpxdi1_1	Diastolic Blood Pressure with Zero Set to Missing	71.919643	0.433193
1	1	2	BPXPLS	60 sec. pulse (30 sec. pulse * 2):	74.532655	0.377329
1	1	2	bpxdi1_1	Diastolic Blood Pressure with Zero Set to Missing	68.819827	0.565795

Output 5.12 details the estimated male and female means of pulse rate and diastolic blood pressure by AGE20P, and _IMPUTATION_ (for the first MI repetition only). The standard errors of the estimated means for each age X gender domain produced by PROC SURVEYMEANS correctly account for the complex sample design. It is important to take note of the variable names in the output DOMAIN data set for later use in PROC MIANALYZE.

Next, the MI data set is sorted by using the names derived from the output data set produced by the statement, ODS OUTPUT DOMAIN=c5ex2imp_reg_m. This sort is needed for PROC MIANALYZE to correctly identify the estimated means and standard errors for each combination of the values of the VARNAME, AGE20P, RIAGENDR (gender), and _IMPUTATION_ variables.

```
proc sort data=c5ex2imp_reg_m;
 by varname age20p riagendr _imputation_;
run;
```

To complete the MI process, PROC MIANALYZE is used to combine the estimates saved in the newly sorted output data set from step 2. This step generates the final MI estimates, standard errors and confidence intervals for the estimates of the subpopulation means. The code below uses EDF=16 option to define the complete design-based degrees of freedom for the NHANES analysis. Again, this is approximated as 31 clusters–15 strata=16 "complete" degrees of freedom. We also use a BY statement to estimate means and variances for pulse rate and diastolic blood pressure for men and women. Note that the order in which VARNAME, AGEP, and RIAGENDR are listed in this BY statement corresponds to the sort order in the preceding PROC SORT command. The MODELEFFECTS and STDERR statements refer to the variable names in the data set produced by the PROC SURVEYMEANS analysis in step 2.

```
proc mianalyze data=c5ex2imp_reg_m edf=16;
 by varname age20p riagendr;
 modeleffects mean;
 stderr stderr;
run;
```

Output 5.13: Parameter Estimates for Pulse by Gender, among Adults 20+

The MIANALYZE Procedure

Variable Name=BPXPLS Age 20+ Indicator=1 Gender=1

Model Information	
Data Set	WORK.C5EX2IMP_REG_M
Number of Imputations	5

Variance Information							
Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
mean	0.001390	0.094950	0.096618	14.054	0.017569	0.017412	0.996530

Parameter Estimates										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
mean	70.630741	0.310835	69.96431	71.29718	14.054	70.606193	70.694325	0	227.23	<.0001

The MIANALYZE Procedure

Variable Name=BPXPLS Age 20+ Indicator=1 Gender=2

Model Information	
Data Set	WORK.C5EX2IMP_REG_M
Number of Imputations	5

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
mean	0.004382	0.144974	0.150232	13.756	0.036272	0.035593	0.992932

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
mean	74.509966	0.387598	73.67727	75.34266	13.756	74.407466	74.563866	0	192.24	<.0001

Output 5.14: Parameter Estimates for Diastolic Blood Pressure by Gender, among Adults 20+

The MIANALYZE Procedure

Variable Name=bpxdi1_1 Age 20+ Indicator=1 Gender=1

Model Information	
Data Set	WORK.C5EX2IMP_REG_M
Number of Imputations	5

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
mean	0.009450	0.193398	0.204739	13.384	0.058838	0.056836	0.988761

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
mean	72.077638	0.452481	71.10296	73.05232	13.384	71.919643	72.154740	0	159.29	<.0001

The MIANALYZE Procedure

Variable Name=bpxdi1_1 Age 20+ Indicator=1 Gender=2

Model Information	
Data Set	WORK.C5EX2IMP_REG_M
Number of Imputations	5

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
mean	0.014344	0.360835	0.378048	13.568	0.047703	0.046519	0.990782

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
mean	68.664729	0.614856	67.34204	69.98741	13.568	68.495705	68.819827	0	111.68	<.0001

Outputs 5.13 and 5.14 present selected MI statistics and estimates (for those age 20+ only) of the subpopulation means, standard errors and 95% confidence intervals for pulse rate and diastolic blood pressure for the male and female subpopulations of interest. All statistics account for the complex sample design features and the variability introduced by the imputation of the missing values. The results suggest that for U.S. adults age 20 and older, the estimated mean pulse rate and 95% CIs for the population mean is 74.5 (73.7, 75.3) for women and 70.6 (70.0, 71.3) for men while the estimated mean and 95% CI for diastolic blood pressure for men is 72.1 (71.1, 73.1) versus 68.7 (67.3, 70.0) for women.

5.3.2 Imputation of Continuous Variables with Mixed Covariates and a Monotone Missing Data Pattern Using the Predictive Mean Matching Method

In this section, the example from Section 5.3.1 is repeated using PROC MI's predictive mean matching option rather than the linear regression method to impute the missing values of the NHANES 2009–2010 continuous measures of pulse rate and diastolic blood pressure. Predictive mean matching imputes a randomly selected value from a set of neighboring observed "donor" values whose regression-predicted values are closest to the predicted value for the missing case. For a review of the predictive mean matching imputation technique, see Chapter 2, Section 2.3.2.

Note that only the imputation step differs in this comparative example. In the code that follows, the MONOTONE REGPMM statement requests use of the monotone regression Predictive Mean Matching method but the remainder of the SAS code is basically unchanged. As explained previously, with use of the

PMM method, there is no need for use of the MIN, MAX, or ROUND statements. PROC MI's K option allows the user to specify the number of closest observed values to be considered as "donors" in the imputation of the missing data value. Here, we use the default of K=5. Once again, we use the WHERE statement to impute missing data only for NHANES 2009–2010 sample individuals who were 8+ years of age and participated in the MEC examination. The replicate weights and output data set plus the JK coefficients generated for the previous example are used for the JRR variance estimation in PROC SURVEYMEANS.

```
proc mi data=repwt_c5_ex2 nimpute=5 seed=41
out=c5ex2imp_regpmm;
 where ridstatr=2 and ridageyr >= 8;
 class riagendr ridreth1 sdmvstra_sdmvpsu;
 monotone regpmm (bpxpls bpxdi1_1);
 var riagendr ridreth1 ridageyr sdmvstra_sdmvpsu
wtmec2yr bpxpls bpxdi1_1;
run;
```

The remainder of the code for MI steps 2 and 3 follows.

```
proc surveymeans data=c5ex2imp_regpmm varmethod=jk;
  repweights repwt_1-repwt_31 / jkcoefs= .5 .5 .5 .5 .5
.5 .5 .5 .5 .5 .5
       .5 .5 .5 .5 .5 .5 .5 .5 .5 .5 .66667 .66667
.66667 .5 .5 .5 .5 .5 .5;
  weight wtmec2yr;
 by _imputation_;
 domain age20p*riagendr;
 var bpxpls bpxdi1_1;
 ods output domain=c5ex2imp_regpmm_m;
run;

proc sort data=c5ex2imp_regpmm_m;
 by varname age20p riagendr;
run;

ods select parameterestimates;
proc mianalyze data=c5ex2imp_regpmm_m edf=16;
 by varname age20p riagendr;
 modeleffects mean;
 stderr stderr;
run;
```

Output 5.15: Parameter Estimates for Pulse by Gender, among Adults Age 20+

The MIANALYZE Procedure

Variable Name=BPXPLS Age 20+ Indicator=1 Gender=1

Model Information	
Data Set	WORK.C5EX2IMP_REGPMM_M
Number of Imputations	5

Variance Information							
Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
mean	0.002503	0.092135	0.095138	13.816	0.032593	0.032047	0.993631

Parameter Estimates										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
mean	70.561473	0.308445	69.89910	71.22385	13.816	70.484081	70.605219	0	228.77	<.0001

The MIANALYZE Procedure

Variable Name=BPXPLS Age 20+ Indicator=1 Gender=2

Model Information	
Data Set	WORK.C5EX2IMP_REGPMM_M
Number of Imputations	5

Variance Information							
Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
mean	0.006198	0.139731	0.147169	13.475	0.053231	0.051751	0.989756

Parameter Estimates										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
mean	74.480677	0.383626	73.65486	75.30649	13.475	74.365920	74.569508	0	194.15	<.0001

Output 5.16: Parameter Estimates for Diastolic Blood Pressure by Gender, among Adults Age 20+

The MIANALYZE Procedure

Variable Name=bpxdi1_1 Age 20+ Indicator=1 Gender=1

Model Information	
Data Set	WORK.C5EX2IMP_REGPMM_M
Number of Imputations	5

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
mean	0.001841	0.195603	0.197812	14.15	0.011293	0.011228	0.997759

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
mean	72.082077	0.444760	71.12911	73.03505	14.15	72.030043	72.134197	0	162.07	<.0001

The MIANALYZE Procedure

Variable Name=bpxdi1_1 Age 20+ Indicator=1 Gender=2

Model Information	
Data Set	WORK.C5EX2IMP_REGPMM_M
Number of Imputations	5

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
mean	0.003617	0.340852	0.345193	14.128	0.012735	0.012653	0.997476

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
mean	68.616939	0.587531	67.35788	69.87600	14.128	68.524345	68.675101	0	116.79	<.0001

Although the PMM approach uses imputation of missing values from matched cases, Outputs 5.15 and 5.16 show similar results to those in Section 5.3.1 where the monotone linear regression method produced imputations via stochastic draws from a predictive distribution. Our overall conclusions about mean pulse and blood pressure rates by gender and age do not change under the two different imputation methods. In general, we should not expect significant differences in analysis results between imputations of continuous variables performed by the linear regression and the predictive mean matching alternatives. Some analysts favor the predictive mean matching technique since it constrains draws of imputed values to the range of observed values (van Buuren 2012). As noted above, this implicit "bounding" of the imputation draws may introduce a small and mostly negligible bias into the process of simulating

draws from the posterior predictive distribution. On the plus side, it is robust in that it protects against extreme draws that while "probable" in a probability distribution sense are unlikely to be observed in the real world.

5.4 Imputation of Continuous Variables with an Arbitrary Missing Data Pattern and Mixed Covariates Using the FCS Method

Example 5.4.1 presents an application of the FCS method to impute missing data for continuous variables. As described in Chapter 2, the MCMC and FCS methods use iterative algorithms designed to simulate draws from a joint multivariate predictive posterior distribution. The MCMC algorithm (Schafer 1997) was one of the first MI procedures implemented in SAS. In theory, it is designed to impute missing values in a vector of continuous variables that are assumed to be jointly distributed as multivariate normal, $Y \sim \text{MVN}(\mu, \Sigma)$. The MCMC option in PROC MI is flexible in that it allows the use of default Jeffreys (non-informative) Bayesian priors for μ and Σ or user-specified alternative priors for these parameters. Like MCMC, the FCS method is an iterative algorithm designed to simulate the joint predictive posterior for a multivariate set of variables and an arbitrary missing data pattern. But, unlike MCMC, the FCS method makes no strong distributional assumptions about the nature of the joint distribution of the variables in the imputation model. It easily handles mixtures of categorical and continuous variables. The presumption of the FCS method is that even though the exact form of the posterior distribution is unknown, the iterative algorithm will converge to the correct predictive posterior and that imputation draws created by FCS will, in fact, correctly simulate draws from the correct, but unknown posterior distribution.

The imputation model for the missing data problem considered in this section involves a mixture of continuous and classification variables. The missing data pattern is arbitrary but only continuous variables in the imputation model are subject to missing data. The classification variables included in the imputation model are fully observed. Although the example MI analysis presented below might be performed using an MCMC algorithm, the inclusion of the classification variables in the multivariate imputation model clearly stretches MCMC's theoretical assumption that $Y \sim \text{MVN}(\mu, \Sigma)$. Schafer (1997) does discuss exact Bayesian methods for such problems that involve multivariate mixtures of continuous and classification variables. Before the FCS-type algorithms were available in major software systems, it was also common practice to impute the continuous/categorical variable mixtures as though the data were multivariate normal. As noted in Chapter 2, continuously imputed

values for the categorical variables were then "rounded" to integer category values for the final imputation. Allison (2005) and others have recently shown that this practice can be problematic and it is better to choose an imputation method that better matches the properties of the variable that is being imputed (e.g., logistic regression for a binary variable). Although no categorical variables are actually imputed in the following example, we still endorse the use of the FCS method to tackle this problem.

5.4.1 Imputation of Continuous Variables with an Arbitrary Missing Data Pattern and Mixed Covariates Using the FCS Method

Example 5.4.1 uses selected variables from the NHANES 2009–2010 stored in a data subset named **c5_ex3**. As in Examples 5.3.1 and 5.3.2, we create pre-imputation replicate weights, a combined strata and PSU variable and the 2 year MEC weight and WHERE statement are included in the imputation model, SURVEY procedures are used for analysis of completed data sets, and the EDF= option is specified in PROC MIANALYZE to declare the approximate complete data degree of freedom for the complex sample variance estimates.

Variables included in this example:

MEXAM, OTHHISP, WHITE, BLACK, OTHER: Indicators we created in a DATA step (not shown here) based on RIDRETH1 (Race/Ethnicity, a categorical variable with no missing data). The indicator variable names are MEXAM=Mexican American, OTHHISP=Other Hispanic, WHITE=Non-Hispanic White, BLACK=Non-Hispanic Black, and OTHER=Other/Multiracial, all are binary indicators coded 1=Yes 0=No

RIDAGEYR: Age in Years, continuous with no missing data, values range from 0–80 with 80+ coded as 80 years of age

RIAGENDR: Gender, categorical with no missing data, 1=Male, 2=Female

WTMEC2YR: Weight used for analysis of the MEC data for the 2 year period, continuous with no missing data

SDMVSTRA: Complex Sample Design Strata variable, categorical with no missing data

SDMVPSU: Complex Sample Design PSU or Cluster variable, categorical with no missing data

SDMVSTRA_SDMVPSU: Combined Complex Sample Variable, categorical with no missing data

RIDSTATR: Interview Status, categorical variable with no missing data, 1=Interviewed Only, 2=Interviewed and Medical Exam

IMPUTE_LBXTC: Imputation Flag variable for Total Cholesterol, 1=imputed value, 0=observed value

IMPUTE_BPXSY1: Imputation Flag variable for Systolic Blood Pressure, 1=imputed value, 0=observed value

LBXTC: Total Cholesterol (mg/Dl), continuous with some missing data

BPXSY1: Systolic Blood Pressure Measurement Number 1, continuous with some missing data

Our analytic goal is to perform a linear regression of total cholesterol on a continuous measure of systolic blood pressure and indicator variables for being male and of Mexican-American ethnicity. The analysis is restricted to the subpopulation of NHANES 2009–2010 respondents that were 25–35 years of age at interview and participated in the MEC examination. We first examine the missing data pattern and amount of missing data, restricting the summary to respondents that were age eligible (8+ years of age) to provide blood pressure and cholesterol measurements and participated in the NHANES health history interview, MEC examination, and laboratory measurements.

```
proc mi nimpute=0 data=c5_ex3;
 where ridstatr=2 and ridageyr >=8;
 var male mexam othhisp white black other ridageyr
sdmvstra_sdmvpsu wtmec2yr lbxtc bpxsy1;
run;
```

Output 5.17: Missing Data Pattern Grid

Missing Data Patterns

														Group Means									
Group	male	mexam	othhisp	white	black	other	RIDAGEYR	sdmvstra_sdmvpsu	WTMEC2YR	LBXTC	BPXSY1	Freq	Percent	male	mexam	othhisp	white	black	other	RIDAGEYR	sdmvstra_sdmvpsu	WTMEC2YR	LBXTC	BPXSY1
1	X	X	X	X	X	X	X	X	X	X	X	6970	85.18	0.502582	0.213630	0.106169	0.449211	0.174749	0.068241	40.541607	814.599687	33260	186.476763	119.502726
2	X	X	X	X	X	X	X	X	X	X	.	570	6.96	0.396491	0.187719	0.114035	0.391228	0.229070	0.078947	41.625316	820.854386	30895	189.763158	.
3	X	X	X	X	X	X	X	X	X	.	X	569	6.83	0.481216	0.150268	0.080501	0.384515	0.320215	0.064401	34.053867	819.871199	29443	.	116.236136
4	X	X	X	X	X	X	X	X	X	.	.	98	1.05	0.453498	0.127907	0.069767	0.430233	0.290698	0.061395	29.755611	813.127907	30512	.	.

Output 5.17 shows two variables with missing data, LBXTC and BPXSY1, while all other variables are fully observed. Furthermore, the Missing Data Patterns grid reveals that we have an arbitrary missing data pattern with 4 distinct missing data groups. Each of the variables to be imputed, LBXTC and BPXSY1, is numeric and continuous.

We next create an output data set called repwt_c5_ex3 containing all variables in our working data set plus the 31 replicate weights needed for MI step 2. The code below reads in the c5_ex3 work data set, creates replicate weights for the JRR variance estimation method and stores the replicate weights in the variables REPWT_1- REPWT_31.

```
proc surveymeans data=c5_ex3 varmethod=jk
```

```
(outweights=repwt_c5_ex3);
 strata sdmvstra; cluster sdmvpsu; weight wtmec2yr;
run ;
```

The PROC MI code below includes a number of options including a SEED= option, use of the complex sample combined strata and cluster variable and the appropriate weight in the VAR statement, a WHERE statement, an ordered list of variables in the VAR statement (from no missing to most missing), and a CLASS statement to define categorical variables. The FCS regression method is used to impute each of the variables with missing data. The number of burn-in iterations of the FCS algorithm is set to NBITER=10. We request regression model details from each iteration (DETAILS) for the imputation model predicting total cholesterol. The minimum and maximum values for total cholesterol are set to 66 and 305 respectively and 72 and 232 for systolic blood pressure.

```
proc mi data=repwt_c5_ex3 nimpute=10 seed=891
out=c5ex3_imp
 min= . . . . . 66 72
 max= . . . . . 305 232
 round= . . . . . 1.0 1.0;
 where ridstatr=2 and ridageyr >=8;
 var male mexam othhisp white black ridageyr
sdmvstra_sdmvpsu wtmec2yr lbxtc
  bpxsy1;
 class sdmvstra_sdmvpsu;
 fcs nbiter=10 regression (lbxtc/details) regression
(bpxsy1);
run;
```

Output 5.18: FCS Method Details for Regression Model Predicting Total Cholesterol (Partial Output)

Regression Models for FCS Method											
Imputed Variable	Effect	sdmvstra_sdmvpsu	Imputation									
			1	2	3	4	5	6	7	8	9	10
LBXTC	Intercept	.	-0.010448	-0.018945	-0.036740	-0.013214	0.005669	-0.007758	-0.016359	-0.007996	-0.021202	-0.025061
LBXTC	male	.	-0.049285	-0.057126	-0.054287	-0.059027	-0.030108	-0.048454	-0.042082	-0.061795	-0.044139	-0.040701
LBXTC	mexam	.	0.064604	0.072345	0.065574	0.062178	0.044378	0.051575	0.046908	0.084166	0.086170	0.056200
LBXTC	othhisp	.	0.070552	0.068980	0.060406	0.055719	0.042851	0.050842	0.027616	0.091322	0.073519	0.055331
LBXTC	white	.	-0.009336	-0.055273	-0.048563	-0.058308	-0.051492	-0.057939	-0.085554	-0.046337	-0.023069	-0.060340
LBXTC	black	.	0.020671	0.021774	-0.020761	-0.004333	-0.027811	-0.016920	-0.039016	0.018220	-0.019632	-0.009547
LBXTC	RIDAGEYR	.	0.262149	0.258709	0.256966	0.276314	0.267018	0.273046	0.280287	0.285989	0.273809	0.273183
LBXTC	sdmvstra_sdmvpsu	751.000000	-0.131748	-0.093474	-0.133297	-0.035006	-0.136267	-0.077392	-0.121081	0.073223	-0.130663	-0.043387
LBXTC	sdmvstra_sdmvpsu	752.000000	-0.107053	-0.025957	0.023340	-0.008102	0.004129	-0.038352	-0.060717	0.014337	0.067934	-0.017829
LBXTC	sdmvstra_sdmvpsu	761.000000	0.116249	0.239779	0.317509	0.245173	0.177051	0.238785	0.187912	0.241532	0.234864	0.267493
LBXTC	sdmvstra_sdmvpsu	762.000000	-0.109515	-0.098953	-0.203989	-0.115286	-0.163367	-0.225054	-0.105496	-0.264471	-0.089698	-0.191040
LBXTC	sdmvstra_sdmvpsu	771.000000	0.106550	0.131080	0.102199	0.073514	0.039303	0.065133	0.088877	0.117724	0.057621	0.109590
LBXTC	sdmvstra_sdmvpsu	772.000000	0.063861	0.022681	0.124442	0.032787	0.061658	0.013540	0.146742	0.025495	0.122006	0.076559

As a reminder, the DETAILS option for the outcome, total cholesterol (LBXTC), requests a list of parameter estimates for each predictor by MI repetition. In Output 5.19 (partial output), we observe imputation model parameter estimates that are very similar over the ten MI repetitions, suggesting stability across the imputations. In this output, note the distinct levels of the combined strata and PSU variable which is treated as a classification variable in this model. The full output is not shown here due to space considerations.

After imputation, we examine observed and imputed means by imputation repetition and age groups for systolic blood pressure (BPXSY1) and total cholesterol (LBXTC). We use a created variable called AGEGP to examine the observed versus imputed values across 5 age groups (age <=18, 19-29, 30-39, 40-49, 50+). Because we expect the impact of age on observed and imputed blood pressure and cholesterol to differ, the analysis by age groups highlights changes in these measurements over the life course. The age group variable and imputation indicators (IMPUTE_BPXSY1 and IMPUTE_LBXTC) were created in a preliminary data step (not shown).

```
proc means data=c5ex3_imp;
 class _imputation_ agegp impute_lbxtc;
 var lbxtc;
run;

proc means data=c5ex3_imp;
 class _imputation_ agegp impute_bpxsy1;
 var bpxsy1;
run;
```

Output 5.19: Imputed and Observed Means by Imputation and Age Groups, Total Cholesterol

Analysis Variable : LBXTC Total Cholesterol (mg/dL)								
Imputation Number	Age in Groups: 1: <=18, 2: 19-29, 3: 30-39, 4: 40-49, 5:50+	Impute Flag for Total Cholesterol	N Obs	N	Mean	Std Dev	Minimum	Maximum
1	Age <=18	No	1698	1698	159.0530035	27.3876688	66.0000000	300.0000000
		Yes	268	268	164.0597015	39.3003914	67.0000000	266.0000000
	Age 19-29	No	1105	1105	176.2597285	35.9675082	92.0000000	320.0000000
		Yes	77	77	189.1298701	36.3301204	92.0000000	285.0000000
	Age 30-39	No	946	946	194.4249471	38.3635239	107.0000000	357.0000000
		Yes	65	65	181.0000000	43.0472125	101.0000000	280.0000000
	Age 40-49	No	1040	1040	205.2298077	40.0886179	93.0000000	528.0000000
		Yes	47	47	183.0638298	34.4480732	97.0000000	263.0000000
	Age 50+	No	2751	2751	198.3631407	42.8886263	90.0000000	380.0000000
		Yes	188	188	200.9893617	41.3331307	111.0000000	304.0000000
2	Age <=18	No	1698	1698	159.0530035	27.3876688	66.0000000	300.0000000
		Yes	268	268	165.8246269	38.0806732	74.0000000	300.0000000
	Age 19-29	No	1105	1105	176.2597285	35.9675082	92.0000000	320.0000000
		Yes	77	77	173.8701299	43.2423862	86.0000000	296.0000000
	Age 30-39	No	946	946	194.4249471	38.3635239	107.0000000	357.0000000
		Yes	65	65	183.8923077	41.8894091	80.0000000	297.0000000
	Age 40-49	No	1040	1040	205.2298077	40.0886179	93.0000000	528.0000000
		Yes	47	47	195.0000000	35.2173376	115.0000000	270.0000000
	Age 50+	No	2751	2751	198.3631407	42.8886263	90.0000000	380.0000000
		Yes	188	188	199.7074468	35.7249527	105.0000000	304.0000000

Output 5.20: Imputed and Observed Means by Imputation and Age Groups, Systolic Blood Pressure

Analysis Variable : BPXSY1 Systolic: Blood pres (1st rdg) mm Hg								
Imputation Number	Age in Groups: 1: <=18, 2: 19-29, 3: 30-39, 4: 40-49, 5:50+	Impute Flag for Systolic Blood Pressure	N Obs	N	Mean	Std Dev	Minimum	Maximum
1	Age <=18	No	1810	1810	105.4441989	11.0342164	76.0000000	146.0000000
		Yes	156	156	103.5000000	15.5125340	74.0000000	144.0000000
	Age 19-29	No	1099	1099	113.0755232	11.4512005	82.0000000	178.0000000
		Yes	83	83	111.7951807	14.3830515	79.0000000	148.0000000
	Age 30-39	No	917	917	115.9018539	13.1213336	80.0000000	184.0000000
		Yes	94	94	116.0957447	15.5798803	77.0000000	167.0000000
	Age 40-49	No	989	989	119.6258847	15.3507433	82.0000000	190.0000000
		Yes	98	98	122.2755102	16.2098230	73.0000000	166.0000000
	Age 50+	No	2714	2714	131.9801032	20.2756068	72.0000000	232.0000000
		Yes	225	225	131.6311111	15.7828996	96.0000000	175.0000000
2	Age <=18	No	1810	1810	105.4441989	11.0342164	76.0000000	146.0000000
		Yes	156	156	106.0448718	14.6874284	77.0000000	146.0000000
	Age 19-29	No	1099	1099	113.0755232	11.4512005	82.0000000	178.0000000
		Yes	83	83	111.0361446	14.3556276	78.0000000	156.0000000
	Age 30-39	No	917	917	115.9018539	13.1213336	80.0000000	184.0000000
		Yes	94	94	114.6595745	15.3745728	74.0000000	149.0000000
	Age 40-49	No	989	989	119.6258847	15.3507433	82.0000000	190.0000000
		Yes	98	98	119.9693878	14.8327126	74.0000000	157.0000000
	Age 50+	No	2714	2714	131.9801032	20.2756068	72.0000000	232.0000000
		Yes	225	225	131.1688889	14.4641772	86.0000000	176.0000000

Outputs 5.19 and 5.20 present estimated mean values for imputed and observed values for total cholesterol and systolic blood pressure by age groups, for two of the ten imputation repetitions. The results show some variability across imputations with small differences between observed and imputed means. In general, we see observed means for total cholesterol and blood pressure at their highest as individuals enter middle age (40–49) and move into their 50s and older. We also observe similar patterns among imputed means. This analysis serves to informally evaluate the performance of the FCS imputation and to ensure that none of the imputed means within repetitions and age groups differ dramatically from the observed mean within those groupings.

Of note is the use of indicator variables in the imputation model in MI steps 2 and 3. This direct parameterization of the gender and race/ethnicity variables using indicator variables created in a DATA step simplifies the post-estimation processing required to input the results of this analysis to PROC MIANALYZE.

In the next block of code we use the ten repetition data sets output by PROC MI

and analyze each separately using PROC SURVEYREG with the VARMETHOD=JK option. Use of the REPWEIGHTS statement with the JRR replicate weights and the JK coefficients option plus the full sample weight (WTMEC2YR) account for the complex sample design features. The analysis focuses on total cholesterol regressed on systolic blood pressure, an indicator of being Mexican-American (reference group is all other race/ethnicities) and male (reference group is female), among the subpopulation of adults age 25–35. The DOMAIN statement is used for our subpopulation of interest along with the BY statement for analysis of each MI repetition data set. The ODS OUTPUT statement outputs a data set containing parameter estimates and standard errors (c5ex3imp_regparms).

```
proc surveyreg data=c5ex3_imp varmethod=jk;
 repweights repwt_1-repwt_31 / jkcoefs=.5 .5 .5 .5 .5
.5 .5 .5 .5 .5 .5 .5 .5
  .5 .5 .5 .5 .5 .5 .5 .5 .5 .66667 .66667 .66667 .5 .5
.5 .5 .5 .5 ;
 weight wtmec2yr;
 by _imputation_;
 domain age25_35;
 model lbxtc = bpxsy1 mexam male / solution;
 ods output parameterestimates=c5ex3imp_regparms;
run;

data c5ex3imp_regparms_1;
 set c5ex3imp_regparms;
 where domain eq 'Age 25-35=1';
run;

proc print;
run;
```

Output 5.21: Listing of Output Parameter Data Set from PROC SURVEYREG (Partial Output)

Obs	_Imputation_	Parameter	Estimate	StdErr	DenDF	tValue	Probt	Domain	age25_35
1	1	Intercept	157.180905	11.6243456	31	13.52	<.0001	Age 25-35=1	1
2	1	BPXSY1	0.218806	0.1088445	31	2.01	0.0532	Age 25-35=1	1
3	1	mexam	9.094150	2.5821067	31	3.52	0.0014	Age 25-35=1	1
4	1	male	7.914728	2.5913846	31	3.05	0.0046	Age 25-35=1	1

Output 5.21 provides a listing of the output data set, c5ex3imp_regparms_1, produced by PROC SURVEYREG (for the first repetition only). The values stored in the variables _IMPUTATION_, PARAMETER, ESTIMATE, AND STDERR will each be used either directly or indirectly in MI step 3. From the

full output produced by PROC SURVEYREG, a DATA step is used to save only the records where the domain variable is equal to 'Age25-35=1' as this is our analysis subpopulation of interest. Because we correctly used a DOMAIN statement in PROC SURVEYREG, there is no issue with use of the conditional WHERE statement here. To complete the MI process, we again use the EDF=16 option to specify the approximate complete data degrees of freedom for the NHANES 2009–2010 complex sample design.

```
proc mianalyze parms=c5ex3imp_regparms_1 edf=16;
 modeleffects intercept bpxsy1 mexam male;
run;
```

Output 5.22: Model and Variance Information and Parameter Estimates

The MIANALYZE Procedure

Model Information	
PARMS Data Set	WORK.C5EX3IMP_REGPARMS_1
Number of Imputations	10

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
intercept	29.075837	106.804903	138.788324	10.344	0.299456	0.239371	0.976622
bpxsy1	0.002366	0.009439	0.012042	10.604	0.275741	0.224155	0.978076
mexam	1.107682	7.298856	8.517306	11.935	0.166937	0.146927	0.985520
male	0.658591	5.585059	6.309509	12.441	0.129712	0.117401	0.988396

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept	154.882706	11.780846	128.7513	181.0141	10.344	144.578749	161.571197	0	13.15	<.0001
bpxsy1	0.237371	0.109734	-0.0053	0.4800	10.604	0.171935	0.328006	0	2.16	0.0543
mexam	10.608648	2.918442	4.2461	16.9712	11.935	9.094150	12.250182	0	3.64	0.0034
male	6.738244	2.511874	1.2868	12.1897	12.441	5.472065	7.914728	0	2.68	0.0194

Output 5.22 includes selected output from PROC MIANALYZE. The Model Information table lists the data set used in the MIANALYZE analysis and verifies that M=10 imputation repetitions were used. The Variance Information shows the highest FMI for the systolic blood pressure variable with smaller impacts for Mexican-Americans and males. This is expected since there was missing data only for the blood pressure predictor and the dependent variable, total cholesterol.

Based on the Parameter Estimates table of MI regression coefficient estimates, we conclude that all else being equal, a one unit increase in systolic blood pressure has a positive and marginally significant (at the α= 0.05 level) effect on

total cholesterol while being male or Mexican/American results in positive and significant estimated increases in total cholesterol. These MIANALYZE results incorporate the increased variability in the estimated parameters due to both the imputation of missing data and the NHANES complex sample design.

5.5 Summary

Chapter 5 has presented common methods for imputation of continuous variables with either a monotone or an arbitrary missing data pattern. Numerous options in both PROC MI/PROC MIANALYZE are demonstrated through applications to standard and complex sample design data sets. We also provide comparisons of imputation methods applied to the same missing data problem.

Chapter 6: Multiple Imputation of Classification Variables

6.1 Introduction to Multiple Imputation of Classification Variables

In this chapter we focus on examples of the multiple imputation approach to estimation and inference for classification variables (also called "categorical variables" throughout this book) that are subject to missing data. As in Chapter 5, each example works through the three-step MI process with discussion of the relevant command syntax and interpretation of output.

PROC MI provides the user several choices for the imputation of missing values for classification variables. The recommended choice of the method will depend on the pattern of missing data and the nature of the classification variable. If the missing data pattern is monotone and the imputation model includes solely classification variables or a mixture of classification and continuous variables, PROC MI monotone is the logical choice of method. The monotone method will

use logistic regression to impute missing values for binary and ordinal classification variables and the discriminant function to impute missing values for nominal classification variables (see Chapter 2 for a review of methods).

For arbitrary missing data patterns, including imputation models with mixtures of classification and continuous variables, the FCS method is most often the method of choice. The iterative cycles of the FCS algorithm will individually impute missing values for each classification variable using the same techniques as the monotone method: logistic regression for binary and ordinal-valued variables and the discriminant function method for nominal classification variables. For situations where the missing data pattern is "nearly" monotone and some variables require imputations of a small number of missing values to complete the monotone pattern, the MCMC method with the MONOTONE modifier can first be used to complete the monotone pattern, followed by standard application of the MONOTONE method to impute missing data for the remaining variables in the filled in monotone pattern.

We do not provide an example of the well-studied practice in which the MCMC algorithm and post-imputation rounding are used to impute missing values for classification variables. As noted in Chapter 2, simulation studies published in the statistical literature (Allison 2005) have shown that this procedure can, in special cases, produce biased results (e.g., application to binary variables in which the proportion p is nearer to the extremes of 0 or 1). Although this method may perform satisfactorily for some classification variable missing data problems, our view is the logistic or discriminant imputations available in the monotone or FCS methods are a better match of the imputation technique to the classification nature of the variable being imputed.

The examples presented in this chapter will illustrate the application of these methods (MONTONE, FCS, and MCMC/monotone) to real-world problems of classification variable missing data. Section 6.2 illustrates the use of the monotone method to impute missing classification response in a monotone missing data pattern. This example is followed in Section 6.3 with an example of FCS imputation for an arbitrary missing data problem involving both classification and continuous variables. The chapter concludes in Section 6.4 with a comparative application of both the FCS and MCMC/monotone methods to a common missing data problem that includes missing values for classification variables.

6.2 Imputation of a Classification Variable with a Monotone Missing Data Pattern Using the Logistic Method

The first example of MI for classification data is based on a health data set, c6_ex1, related to myocardial infarction that we modified for purposes of the example exercise. The original data set is from a data analysis class taught at the Johns Hopkins School of Public Health. The data and variable information can be obtained from http://www.biostat.jhsph.edu/~fdominic/teaching/LDA/lda.html#data. This site includes the raw data and column information needed to read the data into SAS. For this example, we modified the data to include missing data values on the variable INFARC (a binary measure of experiencing a myocardial infarction) and also created a new variable named SELF-HEALTH that simulates measurements of self-rated health status on a five point ordinal scale (1=Excellent,…,5=Poor).

The analytic goals of this example are to estimate overall prevalence of myocardial infarction in the study population represented by this data set and to estimate category percentages for the five-category self-rated health status variable. This example covers a number of key techniques, including use of logistic regression for imputation of binary and ordinal classification variables, the CLASS statement in PROC MI to declare classification variables, PROC FREQ to obtain percentages and standard errors formatted for input to PROC MIANALYZE, and a BY statement in PROC MIANALYZE in order to obtain estimates of statistics for selected BY groups.

Variables used in this example:

ID: Case identifier; continuous and fully observed, not used in the imputations

ORAL_CON: Binary indicator of use of oral contraceptives, fully observed

AGE: Age in years, continuous, fully observed

SMOKE: Binary indicator of whether subject smokes or not, fully observed

INFARC: Binary indicator of whether subject experienced myocardial infarction with some missing data

SELF_HEALTH: Ordinal classification variable with values 1=Excellent, 2=Very Good, 3=Good, 4=Fair, 5=Poor, with some missing data

IMPUTE_INFARC: Imputation flag variable for INFARC, 1=Imputed value 0=Observed value

IMPUTE_SELF_HEALTH: Imputation flag variable for SELF_HEALTH, 1=Imputed value 0=Observed value

Before performing the imputation, we begin (as always!) by exploring the missing data rates and pattern.

```
proc mi nimpute=0 data=c6_ex1 simple;
run;
```

Output 6.1: Missing Data Patterns

Missing Data Patterns

Group	id	oral_con	age	smoke	infarc	self_health	impute_self_health	impute_infarc	Freq	Percent	Group Means							
											id	oral_con	age	smoke	infarc	self_health	impute_self_health	impute_infarc
1	X	X	X	X	X	X	X	X	180	90.00	99.250000	0.494444	34.850000	0.322222	0.216667	3.088889	0	0
2	X	X	X	X	.	X	X	X	9	4.50	96.666667	0.666667	37.777778	0.333333	.	3.000000	0	1.000000
3	X	X	X	X	.	.	X	X	11	5.50	124.090909	0.272727	34.000000	0.454545	.	.	1.000000	1.000000

Output 6.1 identifies three distinct missing data groups: Group 1 with 180 records (90.0%) fully observed, Group 2 with 9 records (4.5%) with missing data on just INFARC, and Group 3 with 11 records (5.5%) missing on both INFARC and SELF_HEALTH. The missing data pattern is monotone with the greatest percentage of missing values for the binary variable, INFARC, a lesser missing data count for SELF_HEALTH, and complete observations for ORAL_CON, AGE, and SMOKE. The imputation model for our multiple imputation will include these five variables. Several of the variables shown in the Missing Data Patterns grid will not be included in the imputation model (ID, IMPUTE_SELF_HEALTH, and IMPUTE_INFARC). Recall from previous chapters that the latter two variables are "imputation flag" variables that are used in later analyses to indicate which values of INFARC and SELF_HEALTH have been imputed and which are observed values.

The following block of SAS code performs the multiple imputation with NIMPUTE=5 to produce five imputation repetitions. A SEED value is specified to ensure that the results of this PROC MI imputation can be exactly repeated at a future time. The five repetitions of the imputed data set will be output to the SAS file, c6_ex1out. The CLASS statement is used to declare ORAL_CON, SMOKE, INFARC, and SELF_HEALTH as classification variables. The monotone order is specified in the VAR statement by listing all of the fully observed variables first followed in order of increasing percentage of missing data by SELF_HEALTH and INFARC. The MONOTONE LOGISTIC statement defines the imputation method for the binary and ordinal variables with missing data, INFARC and SELF_HEALTH. The DETAILS option requests output showing the estimated logistic regression model parameters in the predictive equation for SELF-HEALTH, based on the original observed data and each completed imputation repetition.

```
proc mi data=c6_ex1 nimpute=5 seed=608 out=c6_ex1out;
 class oral_con smoke infarc self_health;
 var oral_con age smoke self_health infarc;
 monotone logistic (self_health/details) logistic
(infarc);
```

```
run;
```

Output 6.2: Model Information and Specification, Missing Data Patterns, and Logistic Model Details from PROC MI

The MI Procedure

Model Information	
Data Set	WORK.C6_EX1
Method	Monotone
Number of Imputations	5
Seed for random number generator	608

Monotone Model Specification	
Method	Imputed Variables
Regression	age
Logistic Regression	self_health infarc
Discriminant Function	smoke

Missing Data Patterns

Group	oral_con	age	smoke	self_health	infarc	Freq	Percent	Group Means age
1	X	X	X	X	X	180	90.00	34.850000
2	X	X	X	X	.	9	4.50	37.777778
3	X	X	X	.	.	11	5.50	34.000000

Logistic Models for Monotone Method

Imputed Variable	Effect	self_health	oral_con	smoke	Obs-Data	Imputation 1	2	3	4	5
self_health	Intercept	1.000000	.	.	-1.44246	-1.349461	-1.470081	-1.202385	-1.279440	-1.610085
self_health	Intercept	2.000000	.	.	-0.45874	-0.468591	-0.435112	-0.323697	-0.342802	-0.359912
self_health	Intercept	3.000000	.	.	0.29273	0.217929	0.182744	0.125230	0.296288	0.469261
self_health	Intercept	4.000000	.	.	1.19830	1.110450	1.134556	0.967135	1.179838	1.381989
self_health	oral_con	.	0	.	0.02648	-0.179208	0.149640	-0.106983	-0.066914	-0.034715
self_health	age	.	.	.	0.01100	-0.262868	-0.105733	0.226115	-0.007475	-0.109410
self_health	smoke	.	.	0	-0.01692	0.022631	-0.029578	-0.146962	-0.105328	-0.026459

Output 6.2 provides information about the imputation and the logistic regression models used to impute the missing values of the self-rated health and myocardial infarction variables. The Model Information table details that the monotone logistic method is used for both of these variables. Since the DETAILS option was not specified for INFARC, only the parameter output for the model used to impute draws of SELF_HEALTH is included in PROC MI output listing. Note from the final subtable in Output 6.2 that PROC MI has used the cumulative logit model with the 4 (i.e., 5–1) category specific intercepts to generate the posterior

predictive distribution and the imputation draws. In SAS, this is the default model for logistic regression. By default, the highest numbered category is used as the basis for estimating the intercepts or "cut points" and the cumulative logit values. For a binary variable such as INFARC, which has only two categories, the same form of the logit model will be used and the highest numbered category (e.g., 1 for a variable coded 0,1) will be used as the baseline category for estimation of the logit regression function. This table also lists the default methods that would be used for the other variables included in the imputation model (regression for AGE and the discriminant function for SMOKE). However, since these are fully observed variables in this example, no imputation for these variables has actually occurred.

After the monotone imputation of SELF_HEALTH and INFARC, we use PROC TABULATE to examine the frequency counts and percentages of myocardial infarction for each imputation repetition and imputation flag value.

```
proc format ;
 value ynf 1='Yes' 0='No' ;
 value shf 1='Excellent' 2='Very Good' 3='Neutral'
4='Fair' 5='Poor' ;
run ;

proc tabulate data=c6_ex1out;
 class _imputation_ impute_infarc infarc;
 table _imputation_ * impute_infarc='Imputed',
 infarc='Infarction' *(n rowpctn='Row %' ) all /
rts=40;
 format impute_infarc infarc ynf. ; keylabel
all='Total';
run;
```

Output 6.3: Tabulation of Frequency of Infarction by Imputation and Imputation Status

		Infarction				Total
		No		Yes		
		N	Row %	N	Row %	N
Imputation Number	Imputed					
1	No	141	78.33	39	21.67	180
	Yes	14	70.00	6	30.00	20
2	No	141	78.33	39	21.67	180
	Yes	17	85.00	3	15.00	20
3	No	141	78.33	39	21.67	180
	Yes	14	70.00	6	30.00	20
4	No	141	78.33	39	21.67	180
	Yes	15	75.00	5	25.00	20
5	No	141	78.33	39	21.67	180
	Yes	12	60.00	8	40.00	20

The second step in the MI analysis of the INFARC and SELF_HELP variables involves estimation of category percentages and standard errors for each MI repetition data set. In this example, we use PROC FREQ with user-defined formats to estimate the percentages for each level of both variables. Use of ODS OUTPUT ONEWAY= produces the output data set of one-way percentage estimates for INFARC and SELF_HEALTH. We then employ a DATA step approach to calculate standard errors for each level per variable. Both the percentages and associated standard errors are needed for processing in PROC MIANALYZE. Note that an alternative approach would be use of the BINOMIAL option on the TABLES statement, however this would require separate TABLE statements for each level of the categorical variables and additional data processing to prepare the data set for PROC MIANALYZE (see Example 40.4 of the 9.4 SAS/STAT PROC FREQ documentation for details). Therefore, we choose to directly calculate the estimate standard errors in one DATA step using the simple random sampling formula (ignoring any finite population correction) of se(p) ~ sqrt[p(1-p)/(n-1)].

```
proc freq data=c6_ex1out;
 by _imputation_ ;
```

```
 tables infarc self_health / nofreq ;
 format infarc ynf. self_health shf.;
 ods output onewayfreqs=c6_ex1outfreqs;
run;

data c6_ex1outfreqs_se;
 set c6_ex1outfreqs;
 stderr=sqrt(percent*(100-percent)/199);
run;

proc print data=c6_ex1outfreqs_se;
run;
```

Output 6.4: Percentages and Standard Errors for Self-Rated Health and Infarction

Obs	_Imputation_	Table	F_infarc	infarc	Frequency	Percent	CumFrequency	CumPercent	F_self_health	self_health	stderr
1	1	Table self_health		.	36	18.00	36	18.00	Excellent	Excellent	2.72343
2	2	Table self_health		.	39	19.50	39	19.50	Excellent	Excellent	2.80859
3	3	Table self_health		.	38	19.00	38	19.00	Excellent	Excellent	2.78095
4	4	Table self_health		.	37	18.50	37	18.50	Excellent	Excellent	2.75257
5	5	Table self_health		.	36	18.00	36	18.00	Excellent	Excellent	2.72343
6	1	Table self_health		.	38	19.00	74	37.00	Very Good	Very Good	2.78095
7	2	Table self_health		.	40	20.00	79	39.50	Very Good	Very Good	2.83552
8	3	Table self_health		.	39	19.50	77	38.50	Very Good	Very Good	2.80859
9	4	Table self_health		.	39	19.50	76	38.00	Very Good	Very Good	2.80859
10	5	Table self_health		.	38	19.00	74	37.00	Very Good	Very Good	2.78095
11	1	Table self_health		.	38	19.00	112	56.00	Neutral	Neutral	2.78095
12	2	Table self_health		.	35	17.50	114	57.00	Neutral	Neutral	2.69352
13	3	Table self_health		.	37	18.50	114	57.00	Neutral	Neutral	2.75257
14	4	Table self_health		.	37	18.50	113	56.50	Neutral	Neutral	2.75257
15	5	Table self_health		.	38	19.00	112	56.00	Neutral	Neutral	2.78095
16	1	Table self_health		.	41	20.50	153	76.50	Fair	Fair	2.86176
17	2	Table self_health		.	38	19.00	152	76.00	Fair	Fair	2.78095
18	3	Table self_health		.	38	19.00	152	76.00	Fair	Fair	2.78095
19	4	Table self_health		.	39	19.50	152	76.00	Fair	Fair	2.80859
20	5	Table self_health		.	40	20.00	152	76.00	Fair	Fair	2.83552
21	1	Table self_health		.	47	23.50	200	100.00	Poor	Poor	3.00565
22	2	Table self_health		.	48	24.00	200	100.00	Poor	Poor	3.02751
23	3	Table self_health		.	48	24.00	200	100.00	Poor	Poor	3.02751
24	4	Table self_health		.	48	24.00	200	100.00	Poor	Poor	3.02751
25	5	Table self_health		.	48	24.00	200	100.00	Poor	Poor	3.02751
26	1	Table infarc	No	No	155	77.50	155	77.50		.	2.96016
27	2	Table infarc	No	No	158	79.00	158	79.00		.	2.88733
28	3	Table infarc	No	No	155	77.50	155	77.50		.	2.96016
29	4	Table infarc	No	No	156	78.00	156	78.00		.	2.93651
30	5	Table infarc	No	No	153	76.50	153	76.50		.	3.00565
31	1	Table infarc	Yes	Yes	45	22.50	200	100.00		.	2.96016
32	2	Table infarc	Yes	Yes	42	21.00	200	100.00		.	2.88733
33	3	Table infarc	Yes	Yes	45	22.50	200	100.00		.	2.96016
34	4	Table infarc	Yes	Yes	44	22.00	200	100.00		.	2.93651
35	5	Table infarc	Yes	Yes	47	23.50	200	100.00		.	3.00565

Output 6.4 is a listing of self-rated health and infarction from the working data set that is output from the PROC FREQ and DATA step manipulation of the multiply imputed data set. The output contains estimated percentages and standard errors for each level of the INFARC and SELF_HEALTH variables per repetition along with other variables produced by default by PROC FREQ.

Finally, we first sort the output data set by INFARC, SELF_HEALTH, and _IMPUTATION_ and use PROC MIANALYZE to process c6_ex1outfreqs_se and produce MI estimated percentages and standard errors for each level of the myocardial infarction and self-rated health variables. We choose to use ODS OUTPUT to save a data set of parameter estimates, outmi_c6_ex1, and produce a compact listing of the estimated percentages and standard errors with PROC

PRINT. This approach is illustrated here only as a way to save space in the output listings. The standard output produced directly from PROC MIANALYZE would also suffice.

```
proc sort data=c6_ex1outfreqs_se;
 by infarc self_health _imputation_ ;
run ;

proc mianalyze data=c6_ex1outfreqs_se;
 ods output parameterestimates=outmi_c6_ex1;
 by infarc self_health;
 modeleffects percent;
 stderr stderr;
run;

proc print noobs data=outmi_c6_ex1;
 var infarc self_health estimate stderr;
run;
```

Output 6.5: Estimated Category Percentages for Myocardial Infarction and Self-Rated Health

infarc	self_health	Estimate	StdErr
.	Excellent	18.600000	2.848953
.	Very Good	19.400000	2.840208
.	Neutral	18.500000	2.832865
.	Fair	19.600000	2.902944
.	Poor	23.900000	3.033059
No	.	77.700000	3.113482
Yes	.	22.300000	3.113482

The PROC MIANALYZE results in Output 6.5 show that an estimated 22.3% (se=3.1%) of individuals represented by this data set experienced a myocardial infarction. Regarding self-rated health, an estimated 18.6% (2.8%) would rate their health as Excellent, 19.4% (2.8%) Very Good, 18.5% (2.8%) as Good, 19.6% (2.9%) as Fair, and 23.9% (3.0%) as Poor.

6.3 Imputation of Classification Variables with an Arbitrary Missing Data Pattern and Mixed Covariates Using the FCS

Discriminant Function and the FCS Logistic Regression Method

In the second example of this chapter, we demonstrate imputation of missing data for classification variables with an arbitrary missing data pattern. The FCS discriminant function is used to impute missing data for a nominal variable (WKSTAT3C), and FCS logistic regression is employed to impute missing values for an ordinal variable (OBESE6CA). This example features use of complex sample design data from the National Comorbidity Survey-Replication (NCS-R; Kessler et al. 2004). As described in Chapter 4 and illustrated in previous NHANES data examples, the NCS-R's complex sample design triggers the following special actions in the three-step MI process: incorporation of complex design variables and weights in the imputation model and step-1 multiple imputations; use of the appropriate SURVEY procedures in MI step 2; and specification of the adjusted complete data degrees of freedom in using the PROC MIANALYZE EDF= option. Other features highlighted in this example are the DETAILS option in PROC MI and the BY statement in PROC MIANALYZE.

Data from Part 2 of the NCS-R survey are used in this example and stored in a dataset named c6_ex2. Part 2 of the NCS-R data set (n=5,692) consists of respondents who completed NCS-R Part 1 (n=9,282) and were also selected to complete additional, more detailed questionnaire sections on various aspects of their personal mental health. See www.hcp.med.harvard.edu/ncs for more information on the structure and sections of this survey.

The analytic goal of this example is to estimate the percentage distributions of two classification variables, obesity status (OBESE6CA) and work status (WKSTAT3C), accounting for both the added variability due to the imputation of missing data and the NCS-R complex sample design effects.

Variables used in this example:

AGE: Age in years, continuous and fully observed

MALE: Male, indicator variable, fully observed (1=Male, 0=Female)

RACECAT: Race/Ethnicity, classification variable and fully observed (1=Other, 2=Hispanic, 3=Black, 4=White). We also created (not shown) a binary indicator variable for each level of Race/Ethnicity, coded as 1=Yes and 0=No. These indicator variables are OTHER, HISPANIC, BLACK, and WHITE.

POVINDEX: Poverty Index, continuous and fully observed

DSM_SO: Social Phobia (DSM), classification indicator of diagnosis and fully observed (1=Yes, 0=No)

MDE: Major Depressive Episode (DSM), classification indicator of diagnosis and fully observed (1=Yes, 0=No)

DSM_ALA: Alcohol Abuse (DSM), classification indicator of diagnosis and fully observed (1=Yes, 0=No)

OBESE6CA: Obesity/BMI status, classification variable with some missing data (1 = BMI <=18.99, 2=19–24.99, 3=25–29.99, 4=30–34.99, 5=35–39.99, 6=40+)

WKSTAT3C: Work Status, classification variable with some missing data (1=Employed, 2=Previously Employed, 3=Out of the labor force [OOLF])

NCSRWTLG: NCS-R Part 2 weight, continuous, fully observed

SESTRAT: Strata variable, classification with values from 1–42, fully observed

SECLUSTR: Cluster variable, classification with values of 1 or 2, fully observed

SESTRAT_SECLUSTR: Combined strata and cluster variable, classification, fully observed with 84 values, created variable for use in imputation, a concatenation of 42 strata plus values of 1 or 2 on the SECLUSTR variable

The missing data pattern in this data set is first explored using PROC MI.

```
proc mi nimpute=0 data=c6_ex2;
 var age male racecat povindex ncsrwtlg
sestrat_seclustr dsm_so mde dsm_ala
  obese6ca wkstat3c;
run;
```

Output 6.6: Missing Data Patterns Grid

Missing Data Patterns

Group	AGE	male	racecat	POVINDEX	NCSRWTLG	sestrat_seclustr	DSM_SO	mde	DSM_ALA	OBESE6CA	WKSTAT3C	Freq	Percent	Group Means									
														AGE	male	racecat	POVINDEX	NCSRWTLG	sestrat_seclustr	DSM_SO	mde	DSM_ALA	OBESE6CA	WKSTAT3C
1	X	X	X	X	X	X	X	X	X	X	X	5581	98.05	43.383444	0.419280	3.548110	4.611714	0.998736	265.109299	0.194230	0.315893	0.181150	2.968823	1.625157
2	X	X	X	X	X	X	X	X	X	X	.	13	0.23	43.076923	0.230769	3.461538	2.425583	1.908883	279.153846	0	0.076923	0	3.384615	.
3	X	X	X	X	X	X	X	X	X	.	X	98	1.72	43.122449	0.397959	3.530612	4.516199	0.951436	273.918367	0.255102	0.326531	0.183265	.	1.551020

Output 6.6 presents the Missing Data Patterns Grid and details the missing data on the two classification variables of interest: obesity status (OBESE6CA) and work status (WKSTAT3C). Obesity status is an ordinal variable with 6 categories ranging from 1. BMI <= 18 or “underweight” to 6. BMI >=40 or “grossly overweight” while work status is a nominal classification variable

with three categories. From the Missing Data Patterns grid, we observe 3 distinct groups: Group 1 (98.1%), which is fully observed on each variable; Group 2 (0.2%), missing only work status; and Group 3 (1.7%), with missing data on the obesity variable only. Although the overall pattern is simple, it is still considered arbitrary. Therefore, the FCS option will be used to impute the missing values for OBESE6CA and WKSTAT3C.

The SAS code below illustrates the PROC MI syntax that is used to generate *M*=3 imputation repetitions and create an output data set called c6_ex2_imp. The CLASS statement defines WKSTAT3C, OBESE6CA, and SESTRAT_SECLUSTR as classification variables. The FCS DISCRIM (WKSTAT3C) and LOGISTIC (OBESE6CA) syntax instructs PROC MI to use the FCS discriminant function to impute the missing values of the multinomial classification variable and work status, and the logistic regression method to impute the ordinal obesity status variable.

To illustrate an alternative to specifying race as a CLASS variable in this example, we use indicator variables for the four levels of RACECAT: White, Black, Hispanic, and Other. Three of the four indicators for race/ethnicity are specified in the VAR statement (with Hispanic as the deleted reference category). This approach is not required for the imputation but makes the declaration of covariates for the FCS DISCRIM function a bit simpler, as we can then use the indicators directly in the model specification.

By default, the DISCRIM technique in PROC MI will use only continuous variables (or those not specified in a CLASS statement) to estimate the classification function. Though the combined strata/cluster variable is treated as a classification variable in the logistic imputation model and included in our CLASS statement, we choose to omit this variable from the work status imputation model only. This exclusion is done to avoid use of a discriminant function model with a large number of classification levels used as predictors for a method that is designed primarily for continuous variables. Therefore, we directly specify the model predictors for the work status variable (WKSTAT3C) through use of the (WKSTAT3C=) syntax and omit the SESTRAT_SECLUSTR variable. Note that in this example we are forcing PROC MI to use binary indicators for categories of race/ethnicity and so on as continuous variables as it estimates the distance function used to define posterior classification probabilities. In contrast, the logistic regression imputation model for OBESE6CA does not require any exclusions and will be estimated using standard parameterization for continuous and classification variables.

```
proc mi nimpute=3 data=c6_ex2 seed=987 out=c6_ex2_imp;
```

```
 class wkstat3c obese6ca sestrat_seclustr;
 fcs discrim(wkstat3c=age male white black other
povindex ncsrwtlg dsm_so mde
   dsm_ala) logistic (obese6ca/details);
 var age male white black other povindex ncsrwtlg
sestrat_seclustr dsm_so mde
   dsm_ala wkstat3c obese6ca;
run;
```

Output 6.7: Estimated Regression Parameters from the FCS Imputation of Obesity Status (Partial Output)

Logistic Models for FCS Method

Imputed Variable	Effect	OBESE6CA	sestrat_seclustr	WKSTAT3C	Imputation 1	Imputation 2	Imputation 3
OBESE6CA	Intercept	1.000000	.	.	-3.518321	-3.383443	-3.519725
OBESE6CA	Intercept	2.000000	.	.	-0.329332	-0.354200	-0.413277
OBESE6CA	Intercept	3.000000	.	.	1.155158	1.124631	1.003041
OBESE6CA	Intercept	4.000000	.	.	2.282149	2.253789	2.229130
OBESE6CA	Intercept	5.000000	.	.	3.251483	3.267530	3.258137
OBESE6CA	AGE	.	.	.	-0.247099	-0.266548	-0.235976
OBESE6CA	male	.	.	.	-0.193974	-0.194558	-0.131521
OBESE6CA	white	.	.	.	0.071522	0.095907	0.231053
OBESE6CA	black	.	.	.	-0.120750	-0.141082	-0.022184
OBESE6CA	other	.	.	.	0.077571	0.063268	0.122774
OBESE6CA	POVINDEX	.	.	.	0.139350	0.105088	0.063899
OBESE6CA	NCSRWTLG	.	.	.	0.078217	0.104649	0.110788
OBESE6CA	sestrat_seclustr	.	11.000000	.	0.567077	0.084310	0.847423
OBESE6CA	sestrat_seclustr	.	12.000000	.	0.093297	0.152075	0.197871
OBESE6CA	sestrat_seclustr	.	21.000000	.	-0.746318	-0.548825	-0.285338
OBESE6CA	sestrat_seclustr	.	22.000000	.	0.965340	0.537221	0.083109
OBESE6CA	sestrat_seclustr	.	31.000000	.	0.917327	1.177620	0.899843
OBESE6CA	sestrat_seclustr	.	32.000000	.	-0.527405	-0.707147	-0.646280

Output 6.7 provides partial output from PROC MI, highlighting the results of the FCS Logistic regression model imputation of missing values for OBESE6CA. We request model details for the imputation of obesity status and include output to evaluate the estimated regression parameters for each imputation repetition.

The following code uses the completed data sets produced by PROC MI and performs a design-adjusted frequency table analysis with PROC SURVEYFREQ. This procedure is used to account for the complex sample

design and generates design-adjusted standard errors. In this syntax, we declare the complex sample variables and weight in the STRATA, CLUSTER, and WEIGHT statements and use a TABLES statement to analyze OBESE6CA and WKSTAT3C while omitting frequency and total counts. As usual, we request separate analyses of each imputation repetition via the BY statement and create an output data set, c6_ex2_freq, for subsequent use in PROC MIANALYZE. Finally, PROC PRINT is used to generate a listing of the output data set of the estimated percentages and standard errors for the three imputation repetitions. Formats are defined using PROC FORMAT and applied in the PROC PRINT code.

```
proc surveyfreq data=c6_ex2_imp;
 strata sestrat; cluster seclustr; weight ncsrwtlg;
 tables obese6ca wkstat3c / row nofreq nototal;
 by _imputation_;
 ods output oneway=c6_ex2_freq;
run;

proc format ;
 value obf 1='Underweight' 2='Healthy Weight'
3='Overweight' 4='Obese Class
   I' 5='Obese       Class II' 6='Obese Class III';
 value wkf 1='Employed' 2='Unemployed' 3='OOLF';
run;

proc print data=c6_ex2_freq;
 var _imputation_ obese6ca wkstat3c percent stderr;
 format obese6ca obf. wkstat3c wkf. ;
run;
```

Output 6.8: Listing of PROC SURVEYFREQ Output Data Set

Obs	_Imputation_	OBESE6CA	WKSTAT3C	Percent	StdErr
1	1	Underweight	.	3.8025	0.3898
2	1	Healthy Weight	.	37.1507	0.8928
3	1	Overweight	.	33.3408	0.7270
4	1	Obese Class I	.	15.9510	0.7924
5	1	Obese Class II	.	6.0784	0.4797
6	1	Obese Class III	.	3.6767	0.3929
7	1	.	Employed	64.7133	1.0082
8	1	.	Unemployed	5.1326	0.5044
9	1	.	OOLF	30.1541	0.9280
10	2	Underweight	.	3.7216	0.3702
11	2	Healthy Weight	.	37.4222	0.8769
12	2	Overweight	.	33.2007	0.6995
13	2	Obese Class I	.	15.9786	0.7791
14	2	Obese Class II	.	6.0481	0.4971
15	2	Obese Class III	.	3.6289	0.3952
16	2	.	Employed	64.6660	1.0102
17	2	.	Unemployed	5.1276	0.5043
18	2	.	OOLF	30.2064	0.9303
19	3	Underweight	.	3.6878	0.3646
20	3	Healthy Weight	.	37.1242	0.8860
21	3	Overweight	.	33.4689	0.6863
22	3	Obese Class I	.	15.9606	0.7854
23	3	Obese Class II	.	6.0515	0.5017
24	3	Obese Class III	.	3.7070	0.4005
25	3	.	Employed	64.8638	1.0292
26	3	.	Unemployed	5.1047	0.5142
27	3	.	OOLF	30.0315	0.9207

Output 6.8 provides estimates of category percentages and SEs for each imputed variable and data set (for _IMPUTATION_=1,2,3). In this output, across the three MI repetitions we observe similar but slightly different percentages and standard errors for each level of the two classification variables. These differences reflect the random draws of values used to replace the missing data

in the three independent MI repetitions.

The next set of syntax first sorts the c6_ex2_freq data set by obesity status, work status, and imputation number. As described in previous examples, this step is needed for correct processing with the BY statement variables in PROC MIANALYZE. The PROC MIANALYZE code features use of the EDF=42 option to set the correct value for the NCS-R complete degrees of freedom (84 clusters–42 strata=42 df), a BY statement for separate analyses of the two variables of interest, and an ODS OUTPUT statement to create output parameter estimates and variance information data sets. In our final step, we request PROC PRINT listings of the two output data sets created from MIANALYZE and present the results in Output 6.9. Again, this ODS OUTPUT and PROC PRINT approach is not required but produces more compact tables for presentation.

```
proc sort data=c6_ex2_freq;
 by obese6ca wkstat3c _imputation_;
run;
proc mianalyze data=c6_ex2_freq edf=42;
 by obese6ca wkstat3c; modeleffects percent; stderr
stderr;
 ods output parameterestimates=c6_ex2_mianalyze_parms
            varianceinfo=c6_ex2_mianalyze_varinfo;
run;

proc print data=c6_ex2_mianalyze_varinfo;run;
proc print data=c6_ex2_mianalyze_parms; run;
```

Output 6.9: Variance Information and Parameter Estimates

Obs	OBESE6CA	WKSTAT3C	Parm	BetVar	WinVar	TotVar	DF	RIVar	FracMiss	RelEff
1	.	Employed	percent	0.010666	1.032168	1.046389	39.444	0.013778	0.013773	0.995430
2	.	Unemployed	percent	0.000221	0.257728	0.258023	40.086	0.001143	0.001143	0.999619
3	.	OOLF	percent	0.008059	0.858089	0.866834	39.517	0.012522	0.012518	0.995845
4	Underweight	.	percent	0.003470	0.140633	0.145260	38.104	0.032898	0.032831	0.989175
5	Healthy Weight	.	percent	0.027214	0.783694	0.819979	36.969	0.046301	0.046118	0.984860
6	Overweight	.	percent	0.018000	0.496301	0.520300	36.784	0.048357	0.048149	0.984204
7	Obese Class I	.	percent	0.000196	0.617237	0.617498	40.116	0.000424	0.000424	0.999859
8	Obese Class II	.	percent	0.000276	0.242964	0.243332	40.071	0.001514	0.001514	0.999495
9	Obese Class III	.	percent	0.001551	0.156950	0.159018	39.479	0.013177	0.013172	0.995628

Obs	OBESE6CA	WKSTAT3C	Parm	Estimate	StdErr	LCLMean	UCLMean	DF	Min	Max	Theta0	tValue	Probt
1	.	Employed	percent	64.747704	1.022932	62.67937	66.81603	39.444	64.665994	64.863783	0	63.30	<.0001
2	.	Unemployed	percent	5.121603	0.507960	4.09505	6.14816	40.086	5.104683	5.132553	0	10.08	<.0001
3	.	OOLF	percent	30.130693	0.932112	28.24611	32.01528	39.517	30.031533	30.206433	0	32.33	<.0001
4	Underweight	.	percent	3.737295	0.381130	2.96581	4.50878	38.104	3.687843	3.802466	0	9.81	<.0001
5	Healthy Weight	.	percent	37.232353	0.905527	35.39753	39.06718	36.969	37.124174	37.422226	0	41.12	<.0001
6	Overweight	.	percent	33.336789	0.721318	31.87497	34.79861	36.784	33.200683	33.468919	0	46.22	<.0001
7	Obese Class I	.	percent	15.963401	0.785811	14.37536	17.55144	40.116	15.951002	15.978598	0	20.31	<.0001
8	Obese Class II	.	percent	6.059309	0.493287	5.06239	7.05622	40.071	6.048051	6.078388	0	12.28	<.0001
9	Obese Class III	.	percent	3.670854	0.398771	2.86458	4.47713	39.479	3.628865	3.706975	0	9.21	<.0001

The Variance Information table shows use of the complex sample degrees of freedom (42 in the NCS-R data set) with low RIV and FMI and very high RE values. In this table, BetVar means between variance, WinVar means within variance, and TotVar stands for total variance. The other variables are named in a similar shorthand.

The Parameter Estimates section of Output 6.9 includes MI estimates of percentages and standard errors from PROC MIANALYZE. These are calculated for each level of the obesity status and work status variables. Based on these results, we see that, after imputing missing values, an estimated 41% of the NCS-R adult population is underweight or normal weight, with the remaining 59% overweight or defined as obese class I, II, or III. We also observe that an estimated 64.7% of adults are employed, 5.1% unemployed, and 30.1% are not in the labor force.

The MI estimates of standard errors and the corresponding Student *t* tests are adjusted for both imputation variability and the NCS-R complex sample design. In this example, we use the default *t* test of the simple null hypothesis that the category percentages are equal to zero. Not surprisingly, these *t* tests suggest that percentages of cases at each level of the two analysis variables are significantly different from zero. The null value for this *t* test value can be changed to something other than the simple default through use of the MU0= option in the PROC MI statement.

6.4 Imputation of Classification Variables with an Arbitrary Missing Data Pattern and Mixed Covariates: A Comparison of the FCS and MCMC/Monotone Methods

6.4.1 Imputation of Classification Variables with Mixed Covariates and an Arbitrary Missing Data Pattern Using the

FCS Method

In example 6.4, we use selected NHANES 2009–2010 variables and impute classification variables using the FCS method with a mix of continuous and classification covariates. The data set used is c6_ex3.

The NHANES data set used in this example is limited to respondents two years of age and older that completed the MEC examination. This decision to use respondents that participated in the MEC and are at least two years of age is motivated by a desire not to include infants in the imputations and analyses. As previously discussed, the NHANES survey is based on a complex sample design, and use of the SURVEY procedures and other features is recommended.

Our analytic goal is to use logistic regression to model the probability of being obese as a function of age, gender, race/ethnicity, and a binary indicator of irregular pulse rate. We begin with the FCS method in Section 6.4.1 and then repeat the entire analysis with use of the MCMC/monotone technique with a multistep imputation approach in Section 6.4.2.

The variables used in this example are:

- RIDRETH1: Race/Ethnicity, classification variable with no missing data, categories are 1=Mexican-American, 2=Other Hispanic, 3=Non-Hispanic White, 4=Non-Hispanic Black, 5=Other/Multiracial
- RIDAGEYR: Age in Years, continuous with no missing data, values range from 0–80 with 80+ top-coded as 80 years of age
- RIAGENDR: Gender, classification with no missing data, 1=Male, 2=Female
- WTMEC2YR: Weight used for analysis of the MEC data for the two-year period, continuous with no missing data
- SDMVSTRA: Complex Sample Design Strata variable, classification with no missing data
- SDMVPSU: Complex Sample Design PSU or Cluster variable, classification with no missing data
- SDMVSTRA_SDMVPSU: Combined Complex Sample Variable, classification with no missing data
- RIDSTATR: Interview Status, classification variable with no missing data, 1=Interviewed Only, 2=Interviewed and Medical Exam
- IMPUTE_OBESE: Classification variable indicating imputed value (1) or observed value (0)
- IMPUTE_IRRPULSE: Classification variable indicating imputed value (1) or observed value (0)

IRRPULSE: Classification, binary indicator of having an irregular pulse, created variable with some missing data, 1=Yes, Irregular Pulse, 0=No, Not Irregular Pulse

OBESE: Classification, binary indicator of being obese (BMI >= 30), created variable with some missing data, 1=Obese 0=Not Obese

The following PROC MI code uses the NIMPUTE=0 option to analyze the missing data patterns for this set of variables.

```
proc mi nimpute=0 data=c6_ex3;
 where ridstatr=2 and ridageyr >=2;
 var riagendr ridreth1 ridageyr wtmec2yr
sdmvstra_sdmvpsu irrpulse obese;
run;
```

Output 6.10: Missing Data Patterns, Group Means, and Univariate Statistics

Missing Data Patterns

										Group Means						
Group	RIAGENDR	RIDRETH1	RIDAGEYR	WTMEC2YR	sdmvstra_sdmvpsu	irrpulse	obese	Freq	Percent	RIAGENDR	RIDRETH1	RIDAGEYR	WTMEC2YR	sdmvstra_sdmvpsu	irrpulse	obese
1	X	X	X	X	X	X	X	9069	95.24	1.500386	2.760172	34.892270	30900	815.590032	0.018966	0.266733
2	X	X	X	X	X	X	.	95	1.00	1.442105	2.778947	39.115789	23424	808.905263	0.063158	.
3	X	X	X	X	X	.	X	343	3.60	1.586006	3.055394	37.326531	30204	821.448980	.	0.291545
4	X	X	X	X	X	.	.	15	0.16	1.533333	3.000000	45.533333	30598	806.666667	.	.

Output 6.10 reveals missing data for the two binary indicator variables: irregular pulse (IRRPULSE) and obesity (OBESE). We observe 3.60% missing data on irregular pulse only, 1.00% missing on just the obese indicator, and 0.16% missing on both variables. Therefore, the grid indicates an arbitrary missing data pattern with two classification variables to be imputed.

Prior to imputation, full sample replicate weights are generated and stored in an output data set called repwgts_c6_ex3. This process follows the approach outlined in Section 4.5 and is needed with a complex sample data set when a WHERE statement is used to subset the data records processed during the imputation. The replicate weights will be used in MI step 2 with the jackknife repeated replication method for variance estimation.

```
proc surveymeans data=c6_ex3 varmethod=jk
(outweights=repwgts_c6_ex3);
strata sdmvstra; cluster sdmvpsu; weight wtmec2yr;
run;
```

In the first step of the PROC MI imputations, the FCS logistic method is used to impute missing values for IRRPULSE and OBESE. The repetition data sets are output to c6ex3_1_imp_fcs. The CLASS statement is used to declare gender,

race, combined strata/PSU, irregular pulse, and obese as classification variables. The FCS LOGISTIC statement requests use of the FCS logistic regression model to impute both IRRPULSE and OBESE. The VAR statement lists the variables to be used in the imputation in the desired order. Use of the WHERE statement subsets the imputation to those 2+ years of age and interviewed and MEC participants.

The default for the order in which the iterative FCS algorithm will impute missing data for the variables in the imputation model is the ORDER=FREQ option. At each iteration, the FCS algorithm will impute missing data for variables included in the imputation model beginning with the variable with the highest count of observed values and proceeding in sequence to the variable with the fewest observed values. Without the DETAILS option, the default output is limited to Model Information and Specification along with the Missing Data Patterns grid (not shown here). Because we want to evaluate the imputation process, we use PROC FREQ to obtain a cross-tabulation of the IMPUTE_OBESE and OBESE variables, by imputation.

```
proc mi data=repwgts_c6_ex3 nimpute=5 seed=2013
out=c6ex3_1_imp_fcs;
 where ridstatr=2 and ridageyr >=2;
 class riagendr ridreth1 sdmvstra_sdmvpsu irrpulse
obese;
 fcs logistic (obese irrpulse);
 var riagendr ridreth1 ridageyr wtmec2yr
sdmvstra_sdmvpsu obese irrpulse;
run;

proc freq data=c6ex3_1_imp_fcs;
 by _imputation_;
 tables impute_obese*obese;
run;
```

Output 6.11: Imputation Results for MI Repetitions # 1 and 2

Imputation Number=1

Frequency Percent Row Pct Col Pct	Table of impute_obese by obese		
	obese(Obese Indicator)		
impute_obese	0	1	Total
0	6893 72.39 73.24 98.77	2519 26.45 26.76 99.06	9412 98.84
1	86 0.90 78.18 1.23	24 0.25 21.82 0.94	110 1.16
Total	6979 73.29	2543 26.71	9522 100.00

The FREQ Procedure

Imputation Number=2

Frequency Percent Row Pct Col Pct	Table of impute_obese by obese		
	obese(Obese Indicator)		
impute_obese	0	1	Total
0	6893 72.39 73.24 98.81	2519 26.45 26.76 98.94	9412 98.84
1	83 0.87 75.45 1.19	27 0.28 24.55 1.06	110 1.16
Total	6976 73.26	2546 26.74	9522 100.00

Output 6.11 presents the PROC FREQ tabulations of the imputation flag for OBESE (IMPUTE_OBESE) with the imputed OBESE variable by the imputation repetition number (_IMPUTATION_). The output displays the cross-tabulation results for the first two of the five MI repetitions produced by PROC MI. In each of the MI repetitions, 110 missing values for the OBESE variable

were imputed (i.e. IMPUTE_OBESE=1). Examining the results for MI repetition #1, the prevalence of obesity among the observed cases is denoted by the row % value of 26.76%. In this same MI repetition, the corresponding row % for imputed cases is 21.82%. For repetition #2, the observed percentage remains 26.76% but shows 24.55% for the imputed obese cases. This output shows no apparent issues with quality of the imputation process and results.

As discussed previously in this book, in conducting simple checks of this type, we do not expect the distributions of imputed values to conform exactly to that of the observed cases. Such checks are primarily intended to detect either errors in our specification of the imputation model or the PROC MI command syntax. In addition, if there are major differences in the distributions of observed and imputed values it may signal a need to more closely examine and improve the imputation models specified in PROC MI—either adding additional variables or possibly specifying an improved functional form for the model that PROC MI uses to draw the imputed values.

In step 2, we use PROC SURVEYLOGISTIC to regress the logit of the probability of being obese on gender, race/ethnicity, age, and irregular pulse for each of five complete/imputed data sets. The REPWEIGHTS statement with the JK coefficients option and WEIGHT statement allow us to incorporate the full NHANES sample design features into the analysis. Because we have already determined the values of the JK coefficients, we simply insert the same values into the REPWEIGHTS statement (see Chapter 5 for details). The SURVEYLOGISTIC statement includes VARMETHOD=JACKKNIFE to request a repeated replication variance estimation method rather than the default Taylor Series Linearization approach. Classification variables are defined in the CLASS statement with the PARAM=REFERENCE option to use reference group parameterization when these variables are used as covariates in FCS regression models. We use ODS OUTPUT to create an output data set of parameter estimates (c6ex3imp_fcs_est) for use in PROC MIANALYZE. Prior to the step-3 analysis, we inspect the contents and structure of the output data set using PROC PRINT (here for just the first imputation repetition).

```
proc surveylogistic data=c6ex3_1_imp_fcs varmethod=jk ;
 by _imputation_ ;
 repweights repwt_1-repwt_31 / jkcoefs= .5 .5 .5 .5 .5
.5 .5 .5 .5 .5 .5 .5
   .5 .5 .5 .5 .5 .5 .5 .5 .5 .5 .66667 .66667 .66667
.5 .5 .5 .5 .5 .5 ;
 weight wtmec2yr;
 class riagendr ridreth1 / param=reference ;
 model obese (event='1')=riagendr ridreth1 ridageyr
```

```
irrpulse ;
 format ridreth1 racef. riagendr sexf. ;
 ods output parameterestimates=c6ex3imp_fcs_est ;
run;

proc print data=c6ex3imp_fcs_est ;
run ;
```

Output 6.12: Partial Listing of Parameter Estimates Output Data Set from PROC SURVEYLOGISTIC

Obs	_Imputation_	Variable	ClassVal0	DF	Estimate	StdErr	WaldChiSq	ProbChiSq
1	1	Intercept		1	-2.7695	0.1514	334.4881	<.0001
2	1	RIAGENDR	1=Male	1	-0.0190	0.0866	0.0482	0.8262
3	1	RIDRETH1	1=MexAm	1	0.9526	0.1735	30.1605	<.0001
4	1	RIDRETH1	2=Other Hispanic	1	0.7796	0.1661	22.0328	<.0001
5	1	RIDRETH1	3=Non-Hisp White	1	0.6267	0.1337	21.9582	<.0001
6	1	RIDRETH1	4=Non-Hisp Black	1	1.3124	0.1775	54.6549	<.0001
7	1	RIDAGEYR		1	0.0283	0.00124	522.4675	<.0001
8	1	irrpulse		1	-0.4096	0.2346	3.0491	0.0808

Output 6.12 includes a print-out for the first imputation repetition of the output data set to be used in PROC MIANALYZE. In addition to providing MI repetition estimates of the logistic regression parameters and their standard errors, this PROC PRINT output displays the structure of the output data set from PROC SURVEYLOGISTIC (including variable names) that serves as input to the PROC MIANALYZE estimation and inference step.

MI step 3 combines the repetition results from the previous PROC SURVEYLOGISTIC analysis using PROC MIANALYZE. The code below uses the PARMS(CLASSVAR=CLASSVAL) statement to declare the input data set as a parameter type with the class variable CLASSVAL0 containing the levels of the gender and race variables. (As a reminder, PROC MIANALYZE accepts a variety of data set types such as PARMS, DATA, COVB, and so on; see Chapter 8 and the SAS/STAT PROC MIANALYZE documentation for details.) We also use the EDF=16 option in the PROC statement to set the complete data degrees of freedom to 16 to account for the complex sample design of the NHANES data set.

Gender and race/ethnicity are declared as classification variables in the CLASS statement. Irregular pulse is coded as a binary indicator with 1=yes and 0=no and can be used directly in the MODELEFFECTS statement without declaring it

as a CLASS variable.

```
proc mianalyze
parms(classvar=classval)=c6ex3imp_fcs_est edf=16;
 class riagendr ridreth1;
 modeleffects intercept riagendr ridreth1 ridageyr
irrpulse;
run;
```

Output 6.13: Model Information, Variance Information, and Parameter Estimates from PROC MIANALYZE

The MIANALYZE Procedure

Model Information	
PARMS Data Set	WORK.C6EX3IMP_FCS_EST
Number of Imputations	5

Variance Information

Parameter	riagendr	ridreth1	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
			Between	Within	Total				
intercept			0.000232	0.022871	0.023150	14.136	0.012180	0.012105	0.997585
riagendr	1=Male		0.000008865	0.007796	0.007806	14.296	0.001365	0.001364	0.999727
ridreth1		1=MexAm	0.000073567	0.029841	0.029930	14.273	0.002958	0.002954	0.999410
ridreth1		2=Other Hispanic	0.000102	0.027871	0.027992	14.253	0.004372	0.004363	0.999128
ridreth1		3=Non-Hisp White	0.000054899	0.018552	0.018618	14.264	0.003551	0.003545	0.999292
ridreth1		4=Non-Hisp Black	0.000010925	0.031220	0.031233	14.31	0.000420	0.000420	0.999916
ridageyr			4.4956543E-8	0.000001499	0.000001553	13.761	0.035984	0.035316	0.992986
irrpulse			0.001390	0.058114	0.059782	13.879	0.028696	0.028273	0.994377

Parameter Estimates

Parameter	riagendr	ridreth1	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept			-2.766373	0.152151	-3.09241	-2.44034	14.136	-2.783901	-2.742341	0	-18.18	<.0001
riagendr	1=Male		-0.021146	0.088353	-0.21028	0.16798	14.296	-0.025947	-0.018342	0	-0.24	0.8142
ridreth1		1=MexAm	0.943018	0.173002	0.57263	1.31340	14.273	0.931247	0.952565	0	5.45	<.0001
ridreth1		2=Other Hispanic	0.775461	0.167310	0.41721	1.13371	14.253	0.761031	0.788676	0	4.63	0.0004
ridreth1		3=Non-Hisp White	0.626718	0.136449	0.33457	0.91886	14.264	0.613956	0.632149	0	4.59	0.0004
ridreth1		4=Non-Hisp Black	1.311175	0.176728	0.93290	1.68945	14.31	1.306087	1.314412	0	7.42	<.0001
ridageyr			0.028348	0.001246	0.02567	0.03103	13.761	0.028093	0.028680	0	22.75	<.0001
irrpulse			-0.372211	0.244504	-0.89705	0.15263	13.879	-0.409608	-0.330273	0	-1.52	0.1504

Output 6.13 includes Model and Variance Information and the Parameter Estimates table from PROC MIANALYZE. We observe that the Relative Increase in Variance statistics are all quite low, with Relative Efficiencies all close to 1.0, indicating small increases in variance due to missing information and a sufficient number of imputation repetitions to achieve good efficiency (relative to a maximum of 1.0 for an infinite count of repetitions). The parameter estimates and standard errors correctly account for both the complex sample design of the NHANES survey and the imputation variability from the MI process.

Based on the Parameter Estimates results, we conclude that after adjusting for other factors in the model, the estimated log odds of men being obese are not significantly less than those for women. Compared to the Other race/ethnicity category, Mexican-Americans, Other Hispanics, Whites, and Blacks all have significantly elevated log-odds of being obese. In addition, obesity increases with age, but an irregular pulse is not significantly associated with obesity status. Note the degrees of freedom for the Student *t* tests of the hypotheses, H0: βj=0. The relatively small (compared to *n*-1) and noninteger values are the result of our specification of the approximate complete degrees of freedom for the NHANES complex sample design (EDF=16) and the default PROC MIANALYZE use of the Barnard-Rubin formula for computing the degrees of freedom for MI inferences based on the imputed data.

6.4.2 Imputation of Classification Variables with Mixed Covariates and an Arbitrary Missing Data Pattern Using the MCMC/Monotone and Monotone Logistic Methods with a Multistep Approach

Section 6.4.2 presents an alternative to the FCS method employed in Section 6.4.1. In this section, we repeat the example of the previous section but use a multistep imputation approach with the MCMC monotone and monotone LOGISTIC methods. The multistep approach illustrated first employs the MCMC monotone method to impute just enough missing data to produce monotone missing data for the variables in the imputation model and then finishes the imputation using a monotone LOGISTIC regression technique.

Historically, this method has often been used in practice to impute classification variables with an arbitrary missing data pattern, especially when the FCS method was either not available or not a feasible option. Here, we demonstrate the two-step approach and present final parameter and standard error estimates as a general comparison to the FCS based results from Section 6.4.1.

Recall from Output 6.10 that the missing data pattern is arbitrary and that there are four distinct groups of cases: 1. fully observed cases (95.24%); 2. cases missing only OBESE (1.00%), 3. cases missing only IRRPULSE (3.60%); and 4. cases missing both OBESE and IRRPULSE (0.16%). Also, note from Output 6.10 that if we are able to first impute the missing values for the 95 cases that have missing data only for the OBESE variable that the missing data problem will be "converted" to a monotone pattern.

In the first imputation step, we use the MCMC method with the optional IMPUTE=MONOTONE statement to impute just enough missing data to produce

a monotone missing data pattern. In addition, we use ROUND, MIN, and MAX options to ensure imputed values fall within the realistic ranges defined by the observed values, a WHERE statement to select those NHANES respondents who are age two or older and completed the MEC interview, and a VAR statement to use only selected variables in the imputation. The ROUND statement is used because the MCMC method assumes all variables are continuous, and there is the possibility of imputing extreme values that fall outside a realistic range such as defined by the range of the MIN/MAX values. The ROUND, MIN, and MAX options address this issue by controlling the imputed values to an integer with minimum and maximum values of 0 and 1, respectively. Variables included in the imputation model that have no missing data are assigned a '.' in the ROUND, MIN, and MAX options with the implied list corresponding to the order assigned in the VAR statement.

```
proc mi data=repwgts_c6_ex3 nimpute=5 seed=2013
out=c6ex3_2_imp_1ststep
 round= . . . . . 1 1
 min= . . . . . 0 0
 max= . . . . . 1 1;
 where ridstatr=2 and ridageyr >= 2;
 mcmc impute=monotone;
 var riagendr ridreth1 ridageyr wtmec2yr
sdmvstra_sdmvpsu obese irrpulse;
run;
```

Note here that we are allowing the limited use of the MCMC algorithm to impute a small amount of missing data for the binary variable OBESE. As noted in preceding chapters we do not recommend the use of MCMC for large-scale imputation of classification type variables.

Output 6.14: Model Information and Missing Data Patterns for the First MCMC Monotone Imputation of the OBESE variable

Model Information	
Data Set	WORK.C6_EX3
Method	Monotone-data MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	5
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	2013

Missing Data Patterns

Group	RIAGENDR	RIDRETH1	RIDAGEYR	WTMEC2YR	sdmvstra_sdmvpsu	obese	irrpulse	Freq	Percent	Group Means RIAGENDR	RIDRETH1	RIDAGEYR	WTMEC2YR	sdmvstra_sdmvpsu	obese	irrpulse
1	X	X	X	X	X	X	X	9069	95.24	1.500386	2.760172	34.892270	30900	815.590032	0.266733	0.018966
2	X	X	X	X	X	X	O	343	3.60	1.586006	3.055394	37.326531	30204	821.448980	0.291545	.
3	X	X	X	X	X	.	X	95	1.00	1.442105	2.778947	39.115789	23424	808.905263	.	0.063158
4	X	X	X	X	X	O	O	15	0.16	1.533333	3.000000	45.533333	30598	806.666667	.	.

Output 6.14 includes Model Information and a Missing Data Patterns grid which reflects the first imputation details. In the Model Information table, we observe that 5 imputation repetitions were created with 200 burn-in iterations and 100 iterations per imputation using a single MCMC chain. The initial starting values for the estimates in the MCMC chain are based on the EM posterior mode values.

The Missing Data Patterns table has a series of 'X' '.'or 'O' entries for each variable where X means observed data, '.'indicates missing data that was imputed in this MCMC monotone step, and 'O' means missing data that was not imputed in this step. We note that 95 records with missing data on the obesity status variable are imputed, and this in turn produces a monotone pattern. Because this imputation imputes just enough data to create the desired monotone pattern, no additional output such as variance information or parameter estimates tables are produced.

We next impute the remaining missing data. Given that we already have five MI repetitions, we perform only a single second-step imputation on each of the MI repetitions generated by the initial MCMC monotone step to fill in any remaining missing data in the c6ex3_2_imp_1ststep output data set. For completeness, a BY statement is used to ensure that imputation variability from imputation #1 is incorporated in this second imputation.

In the second imputation step, use of the monotone LOGISTIC method, the preferred method for imputation of the binary classification variables, OBESE and IRRPULSE, is specified. In this monotone imputation step, PROC MI creates an output data set containing the final repetitions called c6ex3_2_imp_2ndstep for subsequent use in MI steps 2 and 3.

```
proc mi data=c6ex3_2_imp_1ststep nimpute=1 seed=2013
out=c6ex3_2_imp_2ndstep;
 where ridstatr=2 and ridageyr >=2;
 by _imputation_ ;
 class riagendr ridreth1 sdmvstra_sdmvpsu irrpulse
obese;
 var riagendr ridreth1 ridageyr wtmec2yr
sdmvstra_sdmvpsu obese irrpulse;
 monotone logistic (obese/details) logistic
```

```
(irrpulse/details);
run;
```

Output 6.15: Model Information and Missing Data Patterns for Second Imputation

Imputation Number=5

Model Information	
Data Set	WORK.C6EX3_2_IMP_1STSTEP
Method	Monotone
Number of Imputations	1
Seed for random number generator	1553256718

Monotone Model Specification	
Method	Imputed Variables
Regression	RIDAGEYR WTMEC2YR
Logistic Regression	obese irrpulse
Discriminant Function	sdmvstra_sdmvpsu

Missing Data Patterns

Group	RIAGENDR	RIDRETH1	RIDAGEYR	WTMEC2YR	sdmvstra_sdmvpsu	obese	irrpulse	Freq	Percent	Group Means	
										RIDAGEYR	WTMEC2YR
1	X	X	X	X	X	X	X	9164	96.24	34.936054	30822
2	X	X	X	X	X	X	.	343	3.60	37.326531	30204
3	X	X	X	X	X	.	.	15	0.16	45.533333	30598

Output 6.15 details the missing data pattern as monotone and presents the number of records left to impute for the two classification variables. The output above is for just one of the five imputed data sets (repetition #5), but this pattern holds for each of the individual repetitions. In this second imputation step, we impute a total of 5*343=1715 missing data values on the irregular pulse variable and 5*15=75 missing data values on both obesity status and irregular pulse.

With the multistep imputation complete, we present the code for the PROC SURVEYLOGISTIC analysis of repetitions (results not shown here) followed by the PROC MIANALYZE commands to combine the results from steps 1 and 2 of the MI process. Note that in this analysis we use the replicate weights created prior to imputation with the JKCOEFS option and the jackknife repeated replication variance estimation method in PROC SURVEYLOGISTIC.

```
proc surveylogistic data=c6ex3_2_imp_2ndstep
varmethod=jk;
 repweights repwt_1-repwt_31 / jkcoefs= .5 .5 .5 .5 .5
.5 .5 .5 .5 .5 .5
    .5 .5 .5 .5 .5 .5 .5 .5 .5 .5 .66667 .66667 .66667
.5 .5 .5 .5 .5 .5;
```

```
 weight wtmec2yr;
 by _imputation_ ;
 class riagendr ridreth1 / param=reference;
 model obese (event='1')=riagendr ridreth1 ridageyr
irrpulse;
 format ridreth1 racef. riagendr sexf.;
 ods output parameterestimates=c6ex3imp_2steps_est;
run;

proc mianalyze
parms(classvar=classval)=c6ex3imp_2steps_est edf=16;
 class riagendr ridreth1;
 modeleffects intercept riagendr ridreth1 ridageyr
irrpulse;
run ;
```

Output 6.16: Model Information, Variance Information, and Parameter Estimates from PROC MIANALYZE (Based on Multistep Imputation)

The MIANALYZE Procedure

Model Information	
PARMS Data Set	WORK.C6EX3IMP_2STEPS_EST
Number of Imputations	5

Variance Information

Parameter	riagendr	ridreth1	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
			Between	Within	Total				
intercept			0.000043186	0.022423	0.022475	14.283	0.002311	0.002308	0.999539
riagendr	1=Male		0.000011182	0.007725	0.007739	14.291	0.001737	0.001735	0.999653
ridreth1		1=MexAm	0.000021002	0.029885	0.029910	14.304	0.000843	0.000843	0.999831
ridreth1		2=Other Hispanic	0.000068388	0.027585	0.027667	14.273	0.002975	0.002971	0.999406
ridreth1		3=Non-Hisp White	0.000056813	0.018534	0.018602	14.263	0.003678	0.003672	0.999266
ridreth1		4=Non-Hisp Black	0.000008797	0.031333	0.031344	14.311	0.000337	0.000337	0.999933
ridageyr			2.6859734E-8	0.000001517	0.000001550	13.997	0.021242	0.021012	0.995815
irrpulse			0.002634	0.061129	0.064289	13.501	0.051705	0.050310	0.990038

Parameter Estimates

Parameter	riagendr	ridreth1	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept			-2.771789	0.149916	-3.09273	-2.45085	14.283	-2.780873	-2.762593	0	-18.49	<.0001
riagendr	1=Male		-0.021555	0.087969	-0.20987	0.16676	14.291	-0.025635	-0.017094	0	-0.25	0.8099
ridreth1		1=MexAm	0.948526	0.172946	0.57833	1.31872	14.304	0.943074	0.953624	0	5.48	<.0001
ridreth1		2=Other Hispanic	0.782431	0.166335	0.42632	1.13855	14.273	0.773812	0.794657	0	4.70	0.0003
ridreth1		3=Non-Hisp White	0.632062	0.136388	0.34004	0.92408	14.263	0.619284	0.637875	0	4.63	0.0004
ridreth1		4=Non-Hisp Black	1.313459	0.177042	0.93451	1.69240	14.311	1.309437	1.317433	0	7.42	<.0001
ridageyr			0.028406	0.001245	0.02574	0.03108	13.997	0.028142	0.028543	0	22.82	<.0001
irrpulse			-0.390340	0.253553	-0.93605	0.15537	13.501	-0.477005	-0.340953	0	-1.54	0.1468

A comparison of parameter estimates and *t* tests from the MCMC/monotone multistep method (Output 6.16) with the same statistics based on the FCS imputation method (Output 6.13) reveals slight differences in the estimated

parameters of the final logistic model predicting the probability of being obese. However, the differences are so small that both approaches would lead to the same inferences about the relationship of obesity to the predictors: age, gender, race/ethnicity, and irregular pulse rate.

These results suggest that either the FCS or multistep approach would be good options for imputation of the NHANES classification variables with an arbitrary missing data pattern. In general, though, the comparability of the FCS and MCMC/monotone methods will depend on the variables included in the imputation model and the rates and specific arbitrary pattern for the missing data. Given the flexibility and ease of use, our personal preference for most problems of this type is to use the FCS method for imputation of classification variables with an arbitrary missing data pattern.

6.5 Summary

This chapter provides examples of imputation for classification variables with both monotone and arbitrary missing data patterns and a mix of continuous and classification covariates. Both standard and complex sample design data sets are used along with consideration of correct imputation methods, standard and SURVEY procedures for analysis of imputed data sets, and many optional features of PROC MI and PROC MIANALYZE.

Chapter 7: Multiple Imputation Case Studies

7.1 Multiple Imputation Case Studies

Chapter 7 presents two case studies typical of "real world" data analysis projects. Data from the 2006 Health and Retirement Study (HRS; available at http://hrsonline.isr.umich.edu/) and a longitudinal seizure counts data set downloaded from Johns Hopkins University (available at http://www.biostat.jhsph.edu/~fdominic/teaching/LDA/lda.html; Thall and Vail [1990]) are used in the examples.

7.2 Comparative Analysis of HRS 2006 Data Using Complete Case Analysis and Multiple Imputation of Missing Data

This first case study is based on data from the 2006 HRS. The HRS is a longitudinal panel study that surveys a representative sample of Americans over the age of 50 every 2 years. Although by design HRS is a longitudinal survey, we use data from only the 2006 data collection wave and treat the analysis as cross-sectional in time rather than longitudinal (over time).

Our analysis example focuses on the association between a diagnosis of diabetes and a set of sociodemographic and health measures collected in the HRS interview. The analysis is restricted to respondents age 50 or older with non-zero weights. The 50-plus and non-zero weight restriction sweeps out a few cases that were either nonsample or in nursing homes or younger than the age group of interest in the HRS study (see the HRS documentation for details). Because the HRS data set, c7_ex1, is based on a complex sample design, use of SURVEY procedures is required (Heeringa, West, and Berglund 2010). In this example, we compare two approaches to the treatment of missing data in the analysis: 1) complete case analysis—using only cases with non-missing values for all variables—and 2) a multiple imputation approach to estimation and inference.

Variables used in this example include:

STRATUM_SECU: Combined stratum and SECU variable, a categorical variable representing the complex sample design, no missing data

KWGTR: HRS respondent weight for 2006, a continuous variable with no missing data, range is 930–16,532

KAGE: Age in years in 2006, a continuous variable with no missing data, range is 50–105

GENDER: Gender, a categorical variable coded 1=Male, 2=Female, no missing data

RACECAT: Race/ethnicity, a categorical variable coded 1=Hispanic, 2=White, 3=Black, 4=Other, some missing data

EDCAT: Education, a categorical variable coded 1=0–11, 2=12, 3=13–15, 4=16+, some missing data

DIABETES: Indicator of having diabetes, coded 0=No, 1=Yes, some missing data

ARTHRITIS: Indicator of having arthritis, coded 0=No, 1=Yes, some missing

data

BODYWGT: Body weight in pounds, a continuous variable, some missing data, range is 73–400 lbs

SELFRHEALTH: Self-rated health, a categorical variable, coded 1=Excellent, 2=Very Good, 3=Good, 4=Fair, 5=Poor, some missing data

Our analysis plan includes application of the logistic regression model to study associations between a diabetes diagnosis and gender, race, self-rated health status, and body weight of HRS respondents. We account for the complex sample design effects throughout the three-step MI process. In addition, we compare the results of the MI logistic regression analysis to those obtained from a complete case analysis, which excludes cases with missing values for one or more variables.

7.2.1 Exploration of Missing Data

The first step is to examine the missing data pattern and univariate statistics via use of PROC MI with the NIMPUTE=0 option.

```
proc mi nimpute=0 data=c7_ex1;
run;
```

Output 7.1: Missing Data Patterns and Group Means for 2006 HRS Data Set

Missing Data Patterns

Group	KWGTR	stratum_secu	KAGE	GENDER	RACECAT	EDCAT	DIABETES	ARTHRITIS	BODYWGT	SELFRHEALTH	Freq	Percent	Group Means								
													KWGTR	stratum_secu	KAGE	GENDER	RACECAT	EDCAT	DIABETES	ARTHRITIS	BODYWGT	SELFRHEALTH
1	X	X	X	X	X	X	X	X	X	X	16592	97.86	4514.447364	311.166044	69.241622	1.570878	2.094021	2.396155	0.206123	0.614091	174.536041	2.875121
2	X	X	X	X	X	X	X	X	X	.	19	0.11	4065.578947	334.473684	72.473684	1.736842	2.157895	2.157895	0.315789	0.736842	178.631579	.
3	X	X	X	X	X	X	X	X	.	X	235	1.39	4305.566957	312.791702	68.306383	1.672340	2.059574	2.272340	0.187234	0.600000	.	2.965957
4	X	X	X	X	X	X	X	X	.	.	1	0.01	4579.000000	271.000000	88.000000	1.000000	2.000000	2.000000	0	1.000000	.	.
5	X	X	X	X	X	X	X	.	X	X	8	0.05	3957.500000	325.000000	79.625000	1.500000	2.000000	1.750000	0.500000	.	152.750000	3.500000
6	X	X	X	X	X	X	X	.	X	.	1	0.01	4396.000000	291.000000	85.000000	1.000000	2.000000	2.000000	0	.	150.000000	.
7	X	X	X	X	X	X	.	X	X	X	10	0.06	6216.800000	303.300000	73.000000	1.800000	2.100000	2.400000	.	0.800000	169.500000	3.500000
8	X	X	X	X	X	X	.	X	X	.	1	0.01	4023.000000	321.000000	69.000000	2.000000	2.000000	1.000000	.	0	196.000000	.
9	X	X	X	X	X	X	.	X	.	X	1	0.01	4996.000000	512.000000	58.000000	1.000000	3.000000	1.000000	.	0	.	5.000000
10	X	X	X	X	X	X	.	.	X	X	2	0.01	4723.000000	271.500000	66.000000	1.500000	2.000000	4.000000	.	.	174.000000	3.500000
11	X	X	X	X	X	X	.	.	X	.	1	0.01	3870.000000	302.000000	65.000000	1.000000	2.000000	3.000000	.	.	150.000000	.
12	X	X	X	X	X	X	.	.	.	.	1	0.01	1612.000000	341.000000	69.000000	2.000000	1.000000	1.000000	.	.	.	.
13	X	X	X	X	X	.	X	X	X	X	79	0.47	5130.202532	298.379747	61.594937	1.430380	2.101266	.	0.126582	0.379747	173.303797	2.784810
14	X	X	X	X	.	X	X	X	X	X	3	0.02	5626.000000	307.666667	64.333333	1.333333	.	1.333333	0.333333	0.666667	169.666667	3.333333

Output 7.1 indicates an arbitrary missing data pattern with varying amounts of missing data on the variables EDCAT, RACECAT, DIABETES, ARTHRITIS, SELFRHEALTH, and BODYWGT. The variables to be imputed include a mixture of variable types with body weight (BODYWGT) continuous while the other variables are classification (categorical). There are 14 distinct groups represented in the Missing Data Patterns grid. Missing data percentages for the individual variables range from 0.01% to 1.39%. The number of records with fully observed data is 16,592, or 97.86%.

7.2.2 Complete Case Analysis Using PROC

SURVEYLOGISTIC

Prior to imputation, we perform a complete case analysis using logistic regression with the binary outcome of diabetes regressed on race, gender, self-rated health, and body weight. Here, nothing is done about the missing data.

In the following code, use of PROC SURVEYLOGISTIC correctly accounts for the HRS complex sample design and individual weight from 2006. Though this handles the complex sample design and weight correctly, any cases with missing data are excluded from the analysis. The SURVEYLOGISTIC output below indicates that our working number of cases is 16,679, (16,954–275 with missing data on the variables included in the model =16,679).

```
Number of Observations Read 16954
Number of Observations Used 16679
Note: 275 observations were deleted due to missing
values for the response or explanatory variables.
```

We model the probability of having diabetes through use of DIABETES (EVENT='1') in the model statement along with use of the STRATA, CLUSTER, WEIGHT, and CLASS statements to set up the design-adjusted logistic regression. Reference group rather than effects coding parameterization is used for the variables requested in the CLASS statement (PARAM=REF). The covariates used are gender, race, self-rated health, and body weight.

```
proc surveylogistic data=c7_ex1;
 weight kwgtr; strata stratum; cluster secu;
 class racecat gender selfrhealth / param=ref;
 model diabetes (event='1')=gender racecat selfrhealth
bodywgt;
run;
```

Output 7.2: Logistic Regression Model for Diabetes—Complete Case Analysis (HRS Respondents 50 Years of Age and Older)

Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-2.2443	0.2011	124.5321	<.0001
GENDER	1	1	-0.1348	0.0579	5.4292	0.0198
RACECAT	1	1	-0.1313	0.1488	0.7787	0.3775
RACECAT	2	1	-0.6133	0.1347	20.7445	<.0001
RACECAT	3	1	-0.1629	0.1542	1.1161	0.2908
SELFRHEALTH	1	1	-2.2541	0.1312	295.2350	<.0001
SELFRHEALTH	2	1	-1.5216	0.1043	212.9046	<.0001
SELFRHEALTH	3	1	-0.7708	0.0708	118.5522	<.0001
SELFRHEALTH	4	1	-0.2585	0.0869	14.9070	0.0001
BODYWGT		1	0.0120	0.000728	272.3493	<.0001

Output 7.2 presents weighted and design-adjusted results for HRS sample-eligible respondents age 50 and older. These results suggest that, holding all other predictors constant, men in the HRS study population are significantly less likely than women to experience diabetes. Respondents who self-report good-to-excellent physical health are significantly less likely than those in poor health to have diabetes. Greater body weight has a significant and positive association with a diagnosis of diabetes. Compared to the other race category (category 4), whites (category 2) are significantly less likely to be diagnosed with diabetes when compared to the other race/ethnicity group. These results will serve as a comparison to results based on the same logistic regression model estimated using multiply imputed data sets (see below).

7.2.3 Multiple Imputation of Missing Data with an Arbitrary Missing Data Pattern Using the FCS Method with Diagnostic Trace Plots

The FCS method is the most flexible procedure for multiply imputing missing data for a mixture of categorical and continuous variables with an arbitrary missing data pattern. In this particular example, use of logistic regression for binary/ordinal variables, the discriminant function for nominal variables, and linear regression for continuous variables is demonstrated.

Highlights of the PROC MI code below include *M*=3 repetitions, use of a trace plot of estimated means for body weight, NBITER=20 for 20 "burn-in" iterations, and a SEED=55 to ensure the ability to replicate these imputation results. As in previous examples, we use a combined strata/cluster variable and the 2006 respondent weight to enrich the imputation model and reflect the complex sample design characteristics. The use of the CLASS statement instructs PROC MI to treat race, gender, education, self-rated health, diabetes, arthritis, and the combined stratum/SECU variable as classification. For the DISCRIM function imputation, we omit the combined stratum and SECU variable from the imputation model but include it in the logistic and regression imputation models. This is to avoid "stretching the limits" of the DISCRIM function as is designed for use with continuous rather than classification covariates. In this data set, with 56 strata (STRATUM) and values of 1 or 2 in the cluster variable (SECU), adding 56*2=112 predictors to each level of the nominal race/ethnicity outcome would severely test the assumptions of the DISCRIM function and model stability.

```
proc mi nimpute=3 data=c7_ex1 seed=55 out=c7_ex1_imp;
 class racecat gender edcat selfrhealth diabetes
arthritis stratum_secu;
 fcs plots=trace(mean(bodywgt)) nbiter=20 logistic
(edcat diabetes arthritis)
   discrim(racecat=gender edcat selfrhealth diabetes
arthritis
   /classeffect=include) regression (bodywgt);
 var stratum_secu kwgtr kage gender racecat edcat
diabetes arthritis bodywgt
   selfrhealth;
run;
```

Output 7.3: FCS Variance Information and Parameter Estimates for Body Weight

Variance Information

Variable	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
BODYWGT	0.001598	0.099324	0.101455	3560.2	0.021457	0.021437	0.992905

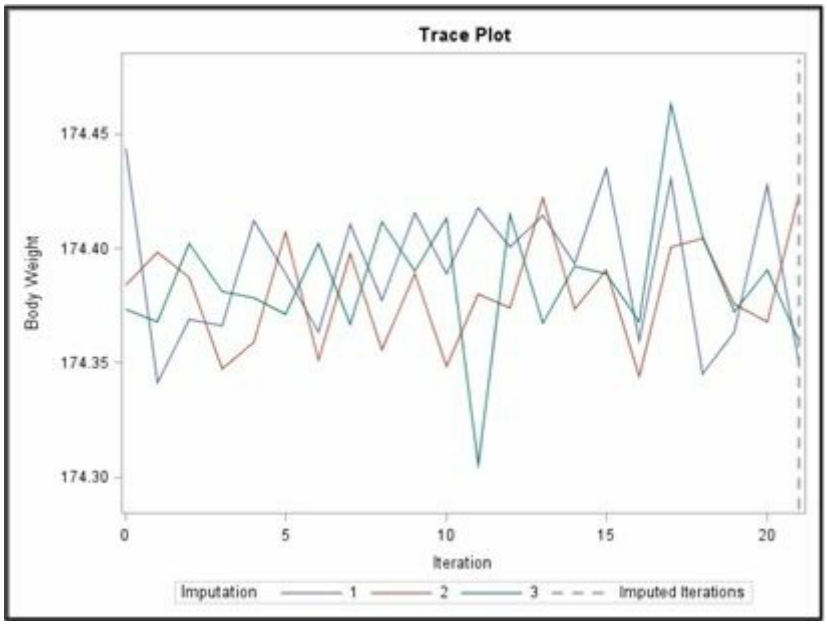
Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
BODYWGT	174.377744	0.318521	173.7532	175.0022	3560.2	174.349962	174.423564	0	547.46	<.0001

Output 7.3 includes Variance Information and Parameter Estimates tables with

information for just the continuous variable BODYWGT. By default, only continuous variable output is included, and given that the other variables to be imputed are categorical, no descriptive output is automatically generated by PROC MI.

Figure 7.1: Trace Plot for Mean of Body Weight



The Trace Plot (Figure 7.1) includes the mean estimates of body weight for the FCS algorithm at each sequential "burn-in" iteration (iterations 1–20) and the actual imputations 1–3 (iterations 21, 22, and 23). The narrow ranges (about 174.3–174.5 pounds) of the MI repetition mean estimates and absence of systematic trends illustrated in the plot suggest that by the end of the "burn-in" sequence of iterations the FCS imputation algorithm has likely converged to a reasonable predictive distribution for imputation of the missing values of the body weight variable.

7.2.4 Logistic Regression Analysis of Imputed Data Sets Using PROC SURVEYLOGISTIC

Our analytic plan includes use of logistic regression to study the relationship of a diabetes diagnosis with gender, race, self-rated health, and body weight in the HRS population age 50-plus. We now use the imputed data sets from step 1 to perform a design-adjusted logistic regression using PROC SURVEYLOGISTIC with the CLASS, BY, and ODS OUTPUT statements. With the ODS OUTPUT statement, we create a data set of parameter estimates (PARAMETERESTIMATES=) for input into PROC MIANALYZE. Use of the BY _IMPUTATION_ statement produces separate model fit and parameter estimate outputs for each of the three imputation repetition data sets generated by PROC MI.

We recommend a careful check of the contents and structure of the parameter estimate output data sets from PROC SURVEYLOGISTIC to understand the variable naming conventions and how CLASS variable levels are handled. For this task, PROC PRINT is used to list the regression parameter output dataset, c7_ex1_est.

```
proc surveylogistic data=c7_ex1_imp;
 weight kwgtr; strata stratum; cluster secu;
 by _imputation_;
 class racecat gender edcat selfrhealth / param=ref;
 model diabetes (event='1')=gender racecat selfrhealth
bodywgt;
 ods output parameterestimates=c7_ex1_est;
run;

proc print data=c7_ex1_est;
run ;
```

Output 7.4: Listing of Parameter Estimates Output Data Set from PROC SURVEYLOGISTIC

Obs	_Imputation_	Variable	ClassVal0	DF	Estimate	StdErr	WaldChiSq	ProbChiSq
1	1	Intercept		1	-2.2746	0.2060	121.8810	<.0001
2	1	GENDER	1	1	-0.1346	0.0569	5.5849	0.0181
3	1	RACECAT	1	1	-0.1471	0.1565	0.8837	0.3472
4	1	RACECAT	2	1	-0.6187	0.1347	21.0873	<.0001
5	1	RACECAT	3	1	-0.1840	0.1528	1.4500	0.2285
6	1	SELFRHEALTH	1	1	-2.2243	0.1265	308.9557	<.0001
7	1	SELFRHEALTH	2	1	-1.4980	0.1006	221.5312	<.0001
8	1	SELFRHEALTH	3	1	-0.7383	0.0664	123.4479	<.0001
9	1	SELFRHEALTH	4	1	-0.2348	0.0646	13.2147	0.0003
10	1	BODYWGT		1	0.0121	0.000718	283.4194	<.0001
11	2	Intercept		1	-2.2671	0.2072	119.7220	<.0001
12	2	GENDER	1	1	-0.1326	0.0568	5.4481	0.0196
13	2	RACECAT	1	1	-0.1309	0.1579	0.6879	0.4069
14	2	RACECAT	2	1	-0.6065	0.1363	19.7917	<.0001
15	2	RACECAT	3	1	-0.1627	0.1542	1.1139	0.2912
16	2	SELFRHEALTH	1	1	-2.2409	0.1295	299.4416	<.0001
17	2	SELFRHEALTH	2	1	-1.4947	0.1008	219.9494	<.0001
18	2	SELFRHEALTH	3	1	-0.7396	0.0661	125.2934	<.0001
19	2	SELFRHEALTH	4	1	-0.2365	0.0644	13.4966	0.0002
20	2	BODYWGT		1	0.0120	0.000707	286.6881	<.0001
21	3	Intercept		1	-2.2610	0.2087	117.3303	<.0001
22	3	GENDER	1	1	-0.1351	0.0565	5.7098	0.0169
23	3	RACECAT	1	1	-0.1309	0.1587	0.6804	0.4094
24	3	RACECAT	2	1	-0.6049	0.1379	19.2425	<.0001
25	3	RACECAT	3	1	-0.1613	0.1549	1.0847	0.2976
26	3	SELFRHEALTH	1	1	-2.2563	0.1312	295.7101	<.0001
27	3	SELFRHEALTH	2	1	-1.5218	0.1022	221.9118	<.0001
28	3	SELFRHEALTH	3	1	-0.7665	0.0679	127.3913	<.0001
29	3	SELFRHEALTH	4	1	-0.2609	0.0659	15.6902	<.0001
30	3	BODYWGT		1	0.0121	0.000718	282.2751	<.0001

Output 7.4 is a listing of the output estimates data set for the three repetitions

from PROC SURVEYLOGISTIC.

7.2.5 Use of PROC MIANALYZE with Logistic Regression Output

We next use PROC MIANALYZE to combine the MI repetition analysis results to generate final MI estimates of parameters and standard errors. A number of options are illustrated, including a CLASS statement, PARMS= (CLASSVAR=CLASSVAL), and EDF=56. The MODELEFFECTS statement uses the information in the VARIABLE and CLASSVAL0 variables (created automatically by SAS with use of the ODS OUTPUT statement above) such that distinct levels of each CLASS variable are recognized. Use of the EDF=56 option ensures that the correct complete complex sample design degrees of freedom for the HRS data are used in significance tests.

```
proc mianalyze parms(classvar=classval)=c7_ex1_est
edf=56;
 class racecat gender selfrhealth;
 modeleffects intercept gender racecat selfrhealth
bodywgt;
run;
```

Output 7.5: Variance Information and Parameter Estimates from PROC MIANALYZE

The MIANALYZE Procedure

Model Information	
PARMS Data Set	WORK.C7_EX1_EST
Number of Imputations	3

Variance Information

Parameter	racecat	gender	selfrhealth	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
				Between	Within	Total				
intercept				0.000046694	0.042984	0.043046	54.02	0.001448	0.001448	0.999517
gender		1		0.000001726	0.003222	0.003224	54.062	0.000714	0.000714	0.999762
racecat	1			0.000087208	0.024860	0.024976	53.818	0.004677	0.004677	0.998443
racecat	2			0.000056692	0.018584	0.018660	53.859	0.004067	0.004067	0.998646
racecat	3			0.000161	0.023705	0.023920	53.499	0.009073	0.009071	0.996985
selfrhealth			1	0.000256	0.016666	0.017007	52.456	0.020477	0.020461	0.993226
selfrhealth			2	0.000219	0.010241	0.010533	51.554	0.028557	0.028513	0.990585
selfrhealth			3	0.000253	0.004465	0.004801	44.765	0.075445	0.074695	0.975706
selfrhealth			4	0.000213	0.004219	0.004503	46.057	0.067254	0.066714	0.978246
bodywgt				3.7749051E-9	0.000000510	0.000000515	53.436	0.009867	0.009865	0.996722

Parameter Estimates												
Parameter	racecat	gender	selfrhealth	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept				-2.267573	0.207475	-2.68353	-1.85161	54.02	-2.274621	-2.260977	0	-10.93	<.0001
gender		1		-0.134082	0.056781	-0.24792	-0.02025	54.062	-0.135082	-0.132594	0	-2.36	0.0218
racecat	1			-0.136304	0.158039	-0.45318	0.18057	53.818	-0.147087	-0.130897	0	-0.86	0.3923
racecat	2			-0.610026	0.136601	-0.88391	-0.33614	53.859	-0.618666	-0.604865	0	-4.47	<.0001
racecat	3			-0.169350	0.154661	-0.47949	0.14079	53.499	-0.183993	-0.161332	0	-1.09	0.2784
selfrhealth			1	-2.240476	0.130412	-2.50211	-1.97884	52.456	-2.256270	-2.224280	0	-17.18	<.0001
selfrhealth			2	-1.504807	0.102631	-1.71079	-1.29882	51.554	-1.521801	-1.494655	0	-14.66	<.0001
selfrhealth			3	-0.748144	0.069292	-0.88773	-0.60856	44.765	-0.766480	-0.738301	0	-10.80	<.0001
selfrhealth			4	-0.244085	0.067101	-0.37915	-0.10902	46.057	-0.260900	-0.234828	0	-3.64	0.0007
bodywgt				0.012038	0.000718	0.01060	0.01348	53.436	0.011969	0.012085	0	16.77	<.0001

Output 7.5 contains Model Information, Variance Information, and the Parameter Estimates table. The Parameter Estimates results suggest that, all else held constant, men (GENDER=1) have significantly lower odds than women of being diagnosed with diabetes; that better self-rated health (SELF_HEALTH=1,2) is associated with lower odds of being diabetic; and that increased body weight has a significant, positive association with diabetes. Comparing individuals by race/ethnicity, Whites (RACECAT=2) have significantly lower odds of diabetes when compared to persons of Other race/ethnicity; however, the odds of diabetes among Hispanic and Black (RACECAT=1,3) adults in the HRS population do not differ significantly from that for Other subpopulation.

7.2.6 Comparison of Complete Case Analysis and Multiply Imputed Analysis

A comparison of the results from complete case analysis (Output 7.2) and MI analysis (Output 7.5) shows little difference between parameter estimates, standard errors, and therefore the nature of the final inferences that we would draw from either approach. In this case, the small difference in results and associated inferences is likely due to relatively low amounts of missing data in the analysis variables. It will not always be true that results from a complete case analysis and a multiple imputation treatment of the data will lead to the same results and inferences. A simple sensitivity test—comparing results of complete case and MI analysis—is a useful tool to help analysts interpret the potential effects (presumably null or positive!) that a correct MI treatment of missing value is having on the analysis and interpretation of their data.

7.3 Imputation and Analysis of Longitudinal Seizure Data

7.3.1 Introduction to the Seizure Data

In our second case study, a data set (c7_ex2) from a study of epileptic seizures

is used to demonstrate methods for multiple imputation and analysis of longitudinal/repeated measures data. The data set analyzed in this application consists of counts of the number of seizures experienced in each of four two-week periods during the trial; the baseline number of seizures for the eight-week period prior to the start of the clinical trial; subject ID; age in years; an indicator of whether the subject received an anti-epilepsy drug treatment or placebo (control); and a time variable representing the four two-week periods. For this example, we modified the original data such that there is missing data for one or more of the four seizure count variables. There were 59 unique individuals in the study. For each subject there are four data records, one for each of the four two-week observation periods. The structure of the modified data set is a multiple-records-per-respondent rectangular data array.

The intended analysis focuses on a regression of number of seizures experienced during the trial on the covariates: baseline seizures, age, treatment status, and time (representing the four measurements during the trial). We use a generalized estimating equation (Liang and Zeger 1986) approach with a Poisson model and a repeated statement to account for the lack of independence among two-week seizure counts for each respondent.

Variables used in this analysis are:

ID: Unique numeric id variable, no missing data, 59 unique respondents and 236 person measurements

TIME: A numeric variable ranging from 1–4, corresponding to four two-week periods of the trial, no missing data

AGE: Age in years, numeric variable with no missing data, range is 18–42

TX: Treatment indicator, randomized clinical trial using Progabide, coded 1=Yes, 0=No, no missing data

BASELINE: Number seizures in the baseline eight-week period prior to the start of the trial divided by four to represent seizures per two-week period, numeric, no missing data, range is 1–38 (rounded)

COUNT: Count of seizures experienced during each of four consecutive two-week periods following the eight weeks of pre-trial baseline, numeric, some missing data, range is 0–76

The variables of interest are number of seizures during each two-week period, age, treatment status, and the baseline measurement of seizures. We intend to use a repeated measures Poisson model to regress the count of seizures on age, treatment status, time period of seizure measurements, and baseline number of

seizures.

7.3.2 Exploratory Analysis of Seizure Data

After data download, we first explore the variable distributions and missing data patterns in the c7_ex2 data set. This analysis is done once for the full data set and again for the count of seizures during each of four sequential two-week observation periods of the study.

```
proc means data=c7_ex2 n nmiss mean min max;
 var id count baseline age tx time;
run;

proc means data=c7_ex2 n nmiss mean min max;
 class time;
 var count;
run;
```

Output 7.6: Exploratory Analysis of Seizure Data

The MEANS Procedure

Variable	N	N Miss	Mean	Minimum	Maximum
ID	236	0	168.3559322	101.0000000	238.0000000
count	217	19	7.8617512	0	76.0000000
baseline	236	0	7.8050847	1.5000000	37.7500000
age	236	0	28.3389831	18.0000000	42.0000000
tx	236	0	0.5254237	0	1.0000000
time	236	0	2.5000000	1.0000000	4.0000000

The MEANS Procedure

Analysis Variable : count						
time	N Obs	N	N Miss	Mean	Minimum	Maximum
1	59	53	6	7.3396226	0	40.0000000
2	59	53	6	8.0188679	0	65.0000000
3	59	57	2	8.5263158	0	76.0000000
4	59	54	5	7.5185185	0	63.0000000

Output 7.6 shows 19 person measurement records with missing data on at least 1

of the 4 seizure counts. For example, the first two-week measurement (TIME=1) has six records missing data, the second measurement has six records without data, the third has two records with missing data, and the fourth has five records with missing data. Given that the data set is structured in a long or multiple records per station format with missing data on some measurements, the ability to account for the dependence within each subject during the imputation is important.

One general method to incorporate the information from multiple records within each person's data array is to convert the data set from a multiple-records-per-respondent format to a one-record-per-respondent structure with differently named variables for each time point's measurement. Without restructuring from multiple records per person to one record per person, the relationships within respondents will not be captured in the imputations. Conversion of the data set is recommended for the multiple imputation step, but for subsequent analysis of the imputed data, conversion back to the multiple-records-per-respondent format is recommended (Allison 2001). This approach works well for a variety of missing data problems with some observed data on each of the multiple measurements and a small number of time points or repeated measurements.

We demonstrate the entire process, including data conversion from a multiple- to a single-record structure, multiple imputation of missing values, conversion back to a multiple record data set to analyze imputed data sets, and combining results from the first two MI steps with PROC MIANALYZE.

7.3.3 Conversion of Multiple-Record to Single-Record Data

Each of the 59 respondents has multiple measurement records containing the number of seizures for each period (TIME=1,2,3,4). The listing below contains data records for three respondents. Respondent IDs 101 and 103 have some missing data on the COUNT variable while ID=102 has fully observed data for the variable COUNT.

```
proc print data=c7_ex2 (obs=12);
run;
```

Output 7.7: Listing of Seizure Measurements for Three Respondents

Obs	ID	time	tx	baseline	age	count
1	101	1	1	19.00	18	11
2	101	2	1	19.00	18	14
3	101	3	1	19.00	18	9
4	101	4	1	19.00	18	.
5	102	1	1	9.50	32	8
6	102	2	1	9.50	32	7
7	102	3	1	9.50	32	9
8	102	4	1	9.50	32	4
9	103	1	1	4.75	20	.
10	103	2	1	4.75	20	4
11	103	3	1	4.75	20	3
12	103	4	1	4.75	20	0

Prior to imputation of missing data, the c7_ex2 data set is converted to a one-record-per-individual data set using the SAS code below. We employ array processing in the data step, although PROC TRANSPOSE is another good option. The data set has been previously sorted by ID.

The data step code performs a number of important steps:

1. Produces one record per unique respondent with four variables representing the seizure counts for the four biweekly observation periods (COUNTN1-COUNTN4);
2. Creates an imputation indicator "flag" variable for each of the four count variables (imputation flag variables are named COUNTNI1–COUNTNI4);
3. Outputs the full data vector for each respondent. Because we need one "summary" record for each person, we output only the last record, which contains the full data array.

Note that the last record per respondent contains all of the needed variables and information for the imputation step.

```
proc sort data=c7_ex2;
 by id;
run;

data onerec;
 array countn [4];
 retain countn1-countn4;
 array countni [4];
 set c7_ex2;
 by id;
 countn(time) = count;

 do i=1 to 4;
 if countn[i] eq . then countni [i]=1; else
countni[i]=0;
 end;
 if last.id then output;
run;

proc print data=onerec (obs=3);
run;
```

Output 7.8: Listing of One Record per Person Data Format

Obs	countn1	countn2	countn3	countn4	countni1	countni2	countni3	countni4	ID	time	tx	baseline	age	count	i
1	11	14	9	.	0	0	0	1	101	4	1	19.00	18	.	5
2	8	7	9	4	0	0	0	0	102	4	1	9.50	32	4	5
3	.	4	3	0	1	0	0	0	103	4	1	4.75	20	0	5

Output 7.8 lists records for three respondents in the one-record-per-ID format. The output shows that ID=101 has missing data on the fourth measurement (COUNTN4). The next record (ID=102) has fully observed data, and ID=103 is missing a value for the COUNT1 variable only. The COUNTNI1–COUNTNI4 imputation flags are set to one if imputation is needed and zero otherwise.

We now use PROC MI without imputation to examine the missing data pattern. Since a single record has been created for each subject with repeated measures for the time-dependent counts of seizures, we can no longer use the time variable itself in the imputation model. Note that this approach to multiple imputation of the longitudinal missing data does not permit us to fully capture the time-specific ordering of the counts in the imputation model—only the dependencies that exist among the four repeated measures. Each of the other variables used in the imputation are time-invariant and can be used directly in the imputation model.

```
proc mi nimpute=0 data=onerec;
  var tx baseline age countn1-countn4;
run;
```

Output 7.9: Missing Data Patterns for Seizure Measurements

Missing Data Patterns

Group	tx	baseline	age	countn1	countn2	countn3	countn4	Freq	Percent	Group Means tx	baseline	age	countn1	countn2	countn3	countn4
1	X	X	X	X	X	X	X	41	69.49	0.536585	6.945122	29.292683	7.439024	6.951220	6.756098	6.024390
2	X	X	X	X	X	X	.	4	6.78	0.750000	8.562500	23.000000	6.750000	7.000000	4.750000	.
3	X	X	X	X	X	.	X	2	3.39	1.000000	6.875000	30.000000	6.500000	5.000000	.	5.500000
4	X	X	X	X	.	X	X	5	8.47	0.200000	9.950000	27.800000	7.600000	.	4.600000	9.000000
5	X	X	X	X	.	X	.	1	1.69	0	3.000000	31.000000	6.000000	.	0	.
6	X	X	X	.	X	X	X	6	10.17	0.500000	12.500000	24.833333	.	17.000000	27.833333	17.166667

Output 7.9 indicates an arbitrary missing data pattern. The seizure count variables have a moderate amount of missing data, with about 30.5% of cases missing on one or more counts. Given the moderately high percentage of missing data on seizure counts, we set the number of imputations to M=10.

7.3.4 Multiple Imputation of Missing Data

With the restructured data set ready for imputation, we use the FCS PMM method to impute the missing data. Highlights of the PROC MI code are: NIMPUTE=10 to produce ten repetition data sets; SEED=45 for potential future replication of the imputation; K=6 option to instruct SAS to use the six closest observed values or "neighbors" for the random draws of a donor value for the missing data imputation.

```
proc mi nimpute=10 data=onerec seed=45 out=c7_ex2_imp;
 var tx baseline age countn1-countn4;
 fcs regpmm (countn1-countn4 / k=6);
run ;
```

Output 7.10: Variance Information and Parameter Estimates

Variance Information

Variable	Variance Between	Variance Within	Variance Total	DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
countn1	0.152973	1.328409	1.496680	46.537	0.126671	0.114912	0.988639
countn2	0.014555	1.702109	1.718120	55.546	0.009406	0.009338	0.999067
countn3	0.006129	3.406976	3.413717	55.986	0.001979	0.001976	0.999802
countn4	0.004533	1.592823	1.597809	55.92	0.003130	0.003123	0.999688

Parameter Estimates										
Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
countn1	7.601695	1.223389	5.139908	10.06348	46.537	6.864407	8.152542	0	6.21	<.0001
countn2	7.898305	1.310771	5.272042	10.52457	55.546	7.711864	8.084746	0	6.03	<.0001
countn3	8.372881	1.847625	4.671624	12.07414	55.986	8.288136	8.508475	0	4.53	<.0001
countn4	7.322034	1.264045	4.789768	9.85430	55.92	7.220339	7.406780	0	5.79	<.0001

Relative Efficiency of the *M*=10 imputations is quite high overall (> 0.98 for each imputed variable). The FMI is highest on the first seizure count variable (.1149), reflecting the higher amounts of missing information on the COUNTN1 variable.

With imputation complete, use of PROC MEANS with a CLASS statement allows examination of means for each seizure count variable by _IMPUTATION_ and corresponding imputation flag variable. This set of statements is used within a SAS macro DO loop to reduce the code burden and allow for easy execution of the four cycles of the PROC MEANS analysis.

```
%macro c (v , iv );
%do i=1 %to 4;
 proc means mean min max data=c7_ex2_imp;
 var &v.&i;
 class &iv.&i;
 run;
%end;
%mend;

%c(countn, countni)
```

Output 7.11: Means for Imputed and Observed Seizure Counts

The MEANS Procedure

Analysis Variable : countn1				
countni1	N Obs	Mean	Minimum	Maximum
0	530	7.3396226	0	40.0000000
1	60	9.9166667	0	40.0000000

The MEANS Procedure

Analysis Variable : countn2				
countni2	N Obs	Mean	Minimum	Maximum
0	530	8.0188679	0	65.0000000
1	60	6.8333333	0	23.0000000

The MEANS Procedure

Analysis Variable : countn3				
countni3	N Obs	Mean	Minimum	Maximum
0	570	8.5263158	0	76.0000000
1	20	4.0000000	0	12.0000000

The MEANS Procedure

Analysis Variable : countn4				
countni4	N Obs	Mean	Minimum	Maximum
0	540	7.5185185	0	63.0000000
1	50	5.2000000	8.881784E-16	15.0000000

The imputed versus observed value means illustrate how the overall statistics vary. For example, for the seizure count measurements from time 2, 3 and 4, the means of imputed values are lower than the mean for observed cases. The

opposite is true for time 1, where the mean of imputed values exceeds the mean for actual observations (9.91 versus 7.33). These differences are expected and reflect the stochastic nature of the imputations and the associated uncertainty that is inherent in the imputation process—uncertainty that MI also ensures is reflected in estimates and inferences. The minimum and maximum values provide a check of the range of each variable. Note that since the PMM method was used to impute missing values for seizure counts, the minima and maxima never exceed the observed minimum and maximum values of the four seizure count variables. Upon review, these results alone do not signal any apparent problems with the imputation process.

7.3.5 Conversion Back to Multiple Record Data for Analysis of Imputed Data Sets

Prior to further analysis, the data set is returned to a multiple-records-per-respondent structure. Longitudinal data analysis in SAS (typically performed using PROC MIXED, PROC GLIMMIX, PROC NLMIXED, or PROC GENMOD) is generally performed with a "long" or multiple records per unit of analysis data set. Therefore, the process we used for imputation is reversed, again with array processing. Use of a DO loop with an output statement outputs 4 records per respondent, per imputation so our data set now consists of 10 imputations*59 unique people*4 seizure measurements=2,360 records.

The variables ID, TX, BASELINE, AGE, TIME, _IMPUTATION_, and the restructured variables COUNT_IMP and COUNT_IMPF are kept for use in our planned analyses.

```
data c7_ex2_imp_long
 (keep=id tx baseline age time count_imp
count_impf_imputation_);
 set c7_ex2_imp;
 by _imputation_ id;
 array mon [4] countn1-countn4;
 array imp [4] countni1-countni4;

do i=1 to 4;
 count_imp = mon [i];
 count_impf= imp [i];
 time=i;
output;
end;
run;

proc sort; by _imputation_ id time; run;

proc print data=c7_ex2_imp_long (obs=12);
```

```
id _imputation_ id;
run;
```

Output 7.12: Listing of First Three Respondents for First Imputation

Imputation	ID	time	tx	baseline	age	count_imp	count_impf
1	101	1	1	19.00	18	11	0
1	101	2	1	19.00	18	14	0
1	101	3	1	19.00	18	9	0
1	101	4	1	19.00	18	12	1
1	102	1	1	9.50	32	8	0
1	102	2	1	9.50	32	7	0
1	102	3	1	9.50	32	9	0
1	102	4	1	9.50	32	4	0
1	103	1	1	4.75	20	2	1
1	103	2	1	4.75	20	4	0
1	103	3	1	4.75	20	3	0
1	103	4	1	4.75	20	0	0

Output 7.12 highlights the values for IDs 101, 102, and 103 in the first (_Imputation_=1) of the 10 MI repetitions included in the new data set. The TIME variable returns to values of 1 to 4, TX is either 1 or 0 for every record per individual, while the baseline seizure count is repeated 4 times as is the age variable (AGE).

7.3.6 Regression Analysis of Imputed Data Sets

We analyze the imputed data using Poisson regression with the count of seizures regressed on time, treatment status, baseline seizure count, and age. Due to the repeated measurements of seizure counts per individual and inherent dependency among these counts, we use PROC GENMOD with a REPEATED statement. The CLASS statement declares TIME and ID as classification variables. We specify the covariance matrix as autoregressive for one time period (AR=(1)) with LINK=LOG and DIST=POISSON to request Poisson regression. The output data set named c7_ex2_outgenmod is saved via the ODS OUTPUT GEEEMPPEST= statement. Here, the ODS table name corresponds to GEE empirical estimates and robust standard errors by default.

```
proc genmod data=c7_ex2_imp_long;
 by _imputation_ ; class time id;
 model count_imp=time age tx baseline / dist=poisson
link=log;
 repeated subject=id / type=ar(1);
 ods output GEEEmpPEst=c7_ex2_out_genmod;
```

```
run;
proc print data=c7_ex2_out_genmod;
run;
```

Output 7.13: Listing of Poisson Regression Estimates by Imputation

Obs	_Imputation_	Parm	Level1	Estimate	Stderr	LowerCL	UpperCL	Z	ProbZ
1	1	Intercept		0.4023	0.3949	-0.3718	1.1763	1.02	0.3084
2	1	time	1	-0.0506	0.1907	-0.4242	0.3231	-0.27	0.7909
3	1	time	2	0.0847	0.0736	-0.0596	0.2289	1.15	0.2499
4	1	time	3	0.1562	0.1225	-0.0839	0.3962	1.28	0.2023
5	1	time	4	0.0000	0.0000	0.0000	0.0000	.	.
6	1	age		0.0286	0.0121	0.0048	0.0524	2.36	0.0184
7	1	tx		-0.1862	0.1606	-0.5009	0.1285	-1.16	0.2461
8	1	baseline		0.0822	0.0052	0.0721	0.0923	15.94	<.0001
9	2	Intercept		0.4596	0.3748	-0.2750	1.1942	1.23	0.2201
10	2	time	1	0.0469	0.1465	-0.2402	0.3340	0.32	0.7486
11	2	time	2	0.0685	0.0705	-0.0697	0.2067	0.97	0.3311
12	2	time	3	0.1124	0.1267	-0.1360	0.3608	0.89	0.3750
13	2	time	4	0.0000	0.0000	0.0000	0.0000	.	.
14	2	age		0.0278	0.0115	0.0051	0.0504	2.41	0.0162
15	2	tx		-0.2479	0.1794	-0.5994	0.1036	-1.38	0.1669
16	2	baseline		0.0838	0.0052	0.0736	0.0939	16.16	<.0001

Output 7.13 lists the output data set produced by PROC GENMOD (for the first two repetitions only). Note how the parameter estimates and corresponding levels are organized. Because the levels of the TIME variable are contained fully in the LEVEL1 variable, use of the PARMS(CLASSVAR=LEVEL) syntax in PROC MIANALYZE is appropriate.

```
proc mianalyze parms(classvar=level)=c7_ex2_out_genmod;
 class time;
 modeleffects intercept time age tx baseline;
run;
```

Output 7.14: Model, Variance, and Parameter Information for Number of Seizures Model

The MIANALYZE Procedure

Model Information	
PARMS Data Set	WORK.C7_EX2_OUT_GENMOD
Number of Imputations	10

Variance Information								
		Variance				Relative Increase	Fraction Missing	Relative
Parameter	time	Between	Within	Total	DF	in Variance	Information	Efficiency
intercept		0.001330	0.143244	0.144706	88089	0.010211	0.010130	0.998988
time	1	0.002382	0.023146	0.025767	870.01	0.113225	0.103767	0.989730
time	2	0.000330	0.005536	0.005899	2372.6	0.065632	0.062380	0.993801
time	3	0.000221	0.015514	0.015758	37691	0.015695	0.015505	0.998452
time	4	0	0	0	.	.	.	.
age		0.000000883	0.000137	0.000138	180511	0.007111	0.007072	0.999293
tx		0.000651	0.030454	0.031169	17068	0.023503	0.023077	0.997698
baseline		0.000000925	0.000026299	0.000027317	6486	0.038692	0.037547	0.996259

Parameter Estimates											
Parameter	time	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept		0.441011	0.380403	-0.30458	1.186597	88089	0.402258	0.527320	0	1.16	0.2463
time	1	0.038407	0.160520	-0.27664	0.353458	870.01	-0.050552	0.122375	0	0.24	0.8110
time	2	0.078298	0.076807	-0.07232	0.228914	2372.6	0.044951	0.102838	0	1.02	0.3081
time	3	0.134109	0.125529	-0.11193	0.380150	37691	0.112429	0.156161	0	1.07	0.2854
time	4	0	0	.	.	.	0	0	0	.	.
age		0.027852	0.011730	0.00486	0.050843	180511	0.025845	0.028882	0	2.37	0.0176
tx		-0.238449	0.176549	-0.58450	0.107604	17068	-0.279950	-0.186240	0	-1.35	0.1768
baseline		0.083764	0.005227	0.07352	0.094009	6486	0.082227	0.085938	0	16.03	<.0001

Based on Output 7.14, age and number of seizures at baseline are positive and significant predictors of seizures during the clinical trial, holding all other covariates constant. Controlling for other factors in the Poisson regression model, neither the trial treatment for epilepsy (TX) nor the time period of observation during the trial results in a significant change in seizure episodes. The variances and standard errors in this analysis are adjusted for both the repeated measures for individuals and the imputation variability through use of the REPEATED statement in PROC GENMOD and the pooled estimates from PROC MIANALYZE.

7.4 Summary

Chapter 7 has illustrated applications of multiple imputation to two case studies. In the first case study, we compare design-adjusted logistic regression results from a complete case analysis with the same design-adjusted logistic regression results from a multiply imputed HRS 2006 data set. The comparison investigates

the impact of doing nothing about missing data versus using multiple imputation to deal with missing data problems in a complex sample design data set.

The second case study presents multiple imputation of missing data and subsequent analysis of longitudinal observations on seizures and the impact of treatment during a clinical trial. We present these case studies to provide practical guidance for analysts dealing with similar complex missing data problems in their daily work.

Chapter 8: Preparation of Data Sets for PROC MIANALYZE

8.1 Preparation of Data Sets for Use in PROC MIANALYZE

Chapter 8 presents two analyses based on the MLB Baseball Players' Salary data set and a third based on the National Comorbidity Survey-Replication (NCS-R) data set. The focus of this chapter is to demonstrate preparation of output data sets from MI step 2 for subsequent input to the PROC MIANALYZE "combining" step (step 3).

PROC MIANALYZE may be used to conduct multiple imputation estimation and inference for a variety of SAS standard and SURVEY procedures. However, depending on the type of output produced in either standard or SURVEY procedures, preparation of output data sets for subsequent input to PROC MIANALYZE can differ. The case studies in this chapter will highlight the special preparations of the output data sets for a number of commonly used SAS analysis procedures.

The types of data sets that can be read into PROC MIANALYZE are detailed in the procedure documentation under the "Input Data Sets" section. As a review, the following text repeats selected information about the types of input data set types for univariate and multivariate inference from the SAS/STAT MIANALYZE documentation.

You can specify input data sets based on the type of inference you requested. For univariate inference, you can use one of the following options:

DATA= data set

DATA=EST, COV, or CORR data set

PARMS= data set

For multivariate inference, which includes the testing of linear hypotheses about parameters, you can use one of the following option combinations:

DATA=EST, COV, or CORR data set

PARMS= and COVB= data sets

PARMS=, COVB=, and PARMINFO= data sets

PARMS= and XPXI= data sets

For univariate inference based on confidence intervals or Student *t* tests (see Section 2.5.2), the needed statistics for input into PROC MIANALYZE are parameter estimates, and either explicit or derived standard errors, that is, standard errors (explicit) or a covariance matrix from which standard errors can be derived. For multivariate inference involving Wald *F* tests of multiparameter hypotheses (see Section 2.6), the parameter estimates and the covariance matrix for the parameter estimates are needed to generate linear hypothesis tests. The combination of data sets outlined in Table 8.1 shows how these pieces of information can be organized.

8.2 Imputation of Major League Baseball Players' Salaries

For the two regression examples presented in Sections 8.3.1 and 8.3.2, we use the MLB salary data set initially presented in Chapter 5. For these two examples, only player salary (SALARY) requires imputation.

We use the following variables in these examples:

BATTING_AVERAGE: Batting average, continuous and fully observed

ON_BASE_PERCENTAGE: On-base %, continuous and fully observed

HOMERUNS: Number of homeruns, continuous and fully observed

RUNS_BATTED_IN: Number of runs batted in, continuous and fully observed

WALKS: Number of walks, continuous and fully observed

STRIKE_OUTS: Number of strike outs, continuous and fully observed

STOLEN_BASES: Number of stolen bases, continuous and fully observed

ERRORS: Number of Errors, continuous and fully observed

ARBITRATION: Indicator of player in arbitration process, binary

(0=No,1=Yes) and fully observed

IMPUTE_SALARY: Indicator of imputed value for salary, binary variable created in data step

SALARY: 1992 salary in thousands of dollars, continuous with some missing data

The first step is to impute missing data for the SALARY variable using SEED, ROUND, MIN, and MAX options with the default MCMC method and save the output data set to outc8ex1. Next, a classification variable named ERRORS_CAT with values of 1=9+ errors and 2=0-8 errors in 1992 is created. This new data set is stored as outc8ex1_1 and will be used in Sections 8.3.1 and 8.3.2.

```
proc mi data=c8_ex1 out=outc8ex1 seed=2012 nimpute=5
 round=.01
 min= 109
 max= 6100;
 var Batting_average On_base_percentage HomeRuns
Runs_batted_in Walks Strike_Outs Stolen_bases Errors
arbitration salary;
run;

data outc8ex1_1;
 set outc8ex1;
 if errors >= 0 and errors <=8 then errors_cat=2;
 else if 9 <= errors then errors_cat=1;
run;
```

8.3.1 PROC GLM Output Data Set for Use in PROC MIANALYZE

Example 8.3.1 inputs the outc8ex1_1 data set generated above by PROC MI and performs a linear regression analysis using PROC GLM. This analysis estimates the linear regression of a player's salary (SALARY) on the on-base percentage (ON_BASE_PERCENTAGE), number of home runs (HOMERUNS), number of errors made (ERRORS_CAT), and if the player participated in arbitration (ARBITRATION). ERRORS_CAT is declared as classification in the CLASS statement. Note that the SOLUTIONS option must be included on the MODEL statement in PROC GLM to obtain fixed effects regression parameter estimates that are based on a reference category approach for the ERRORS_CAT variable used in the CLASS statement. By default, when PROC GLM outputs the parameter estimates, the highest number category, ERRORS_CAT=2, will be the reference category.

The ODS OUTPUT statement creates a parameter estimates data set named glmest. This data set contains the regression parameter estimates and standard errors that are the necessary inputs for use by PROC MIANALYZE in the MI "combining step," or step 3 as we have referred to it in this book.

```
proc glm data=outc8ex1_1;
 class errors_cat;
 model salary = on_base_percentage homeruns errors_cat
arbitration /solution;
 ods output parameterestimates=glmest;
 by _imputation_ ;
run;

proc print data=glmest;
run;
```

Output 8.1: Partial Printout (Imputation Repetitions 1 and 2 of *M*=5) of the "glmest" Data Set

Obs	_Imputation_	Dependent	Parameter	Estimate	Biased	StdErr	tValue	Probt
1	1	Salary	Intercept	-858.386390	1	359.447219	-2.39	0.0175
2	1	Salary	On_base_percentage	4695.492116	0	1139.509632	4.12	<.0001
3	1	Salary	HomeRuns	74.812057	0	5.777662	12.95	<.0001
4	1	Salary	errors_cat 1	-48.798042	1	113.552840	-0.43	0.6677
5	1	Salary	errors_cat 2	0.000000	1	.	.	.
6	1	Salary	Arbitration	492.279183	0	302.237607	1.63	0.1043
7	2	Salary	Intercept	-959.625521	1	365.713107	-2.62	0.0091
8	2	Salary	On_base_percentage	4872.315755	0	1159.373579	4.20	<.0001
9	2	Salary	HomeRuns	77.184899	0	5.878378	13.13	<.0001
10	2	Salary	errors_cat 1	-15.669939	1	115.532295	-0.14	0.8922
11	2	Salary	errors_cat 2	0.000000	1	.	.	.
12	2	Salary	Arbitration	487.766396	0	307.506217	1.59	0.1136

Output 8.1 details the structure and content of the regression output for the first two of the M=5 imputation repetition data sets. The variable PARAMETER contains labels for the intercept term and each independent variable of the linear regression model. Note how the ERRORS_CAT variable is represented in the GLM output for the regression model by 2 indicator variables. Category 2 is the reference category for modeling the effect of this categorical predictor, hence the 0.0 value for its estimate and missing values for the standard error and corresponding Student *t* statistic.

The values of the PARAMETER variable consist of the root predictor name, ERRORS_CAT, followed by white space and the number of the indicator variable (1or 2). Because “errors_cat 1” and “errors_cat 2” are not valid SAS variable names, they must be processed to create a valid SAS name prior to use in PROC MIANALYZE. This can be accomplished through use of SAS functions as demonstrated below. Refer to http://support.sas.com/kb/48/700.html for more details on this approach.

The following DATA step uses an IF statement to select just the ERRORS_CAT (with values of 1 and 2) parameter rows (only variable treated as a CLASS variable). Then, the SCAN function picks out the first level or second word in the PARAMETER variable. In this case, this will be the number 1 or 2. The INDEXW function determines the position of the first level, and the SUBSTR function selects information from that position to the end of the string. This process provides a list of the levels while the COMPRESS function removes blanks from the levels list and uses the same variable prefix of ‘errors_cat’ to ensure that the final value in PARAMETER is a valid SAS name (when levels are numeric). This approach can be generalized to other situations involving interactions and valid SAS naming problems. In this case, we have only one CLASS variable to process, but similar coding could be used for more complex situations with multiple CLASS variables.

```
data glmest;
 set glmest;
 if parameter in ('errors_cat   1', 'errors_cat   2')
then do;
 FirstLevel = scan(parameter,2,' ');
 LevelsPos = indexw(parameter,FirstLevel);
 LevelsList = substr(parameter,LevelsPos);
 parameter = 'errors_cat'||(compress(LevelsList));
 end;
run;

proc print;
run;
```

Output 8.2: Partial Listing of Processed glmest Data Set

Obs	_Imputation_	Dependent	Parameter	Estimate	Biased	StdErr	tValue	Probt	FirstLevel	LevelsPos	LevelsList
1	1	Salary	Intercept	-858.386390	1	359.447219	-2.39	0.0175		.	
2	1	Salary	On_base_percentage	4695.492116	0	1139.509632	4.12	<.0001		.	
3	1	Salary	HomeRuns	74.812057	0	5.777662	12.95	<.0001		.	
4	1	Salary	errors_cat1	-48.798042	1	113.552840	-0.43	0.6677	1	20	1
5	1	Salary	errors_cat2	0.000000	1	.	.	.	2	20	2
6	1	Salary	Arbitration	492.279183	0	302.237607	1.63	0.1043		.	
7	2	Salary	Intercept	-959.625521	1	365.713107	-2.62	0.0091		.	
8	2	Salary	On_base_percentage	4872.315755	0	1159.373579	4.20	<.0001		.	
9	2	Salary	HomeRuns	77.184899	0	5.878378	13.13	<.0001		.	
10	2	Salary	errors_cat1	-15.669939	1	115.532295	-0.14	0.8922	1	20	1
11	2	Salary	errors_cat2	0.000000	1	.	.	.	2	20	2
12	2	Salary	Arbitration	487.766396	0	307.506217	1.59	0.1136		.	

Output 8.2 shows that the ERRORS_CAT parameter is now in a valid SAS variable name format. The data set is next sorted by _IMPUTATION_ and finally provided as input to PROC MIANALYZE. In the following MIANALYZE code, ERRORS_CAT1 is used directly in the MODELEFFECTS statement along with the INTERCEPT and other model covariates. ERRORS_CAT2 is not included in the MODELEFFECTS statement since it corresponds to the reference category in the PROC GLM regression model, and parameter and standard errors for this level of the effect are therefore output by PROC GLM as "0" or missing.

```
proc sort data=glmest;
 by _imputation_;
run;
proc mianalyze parms=glmest;
 modeleffects intercept on_base_percentage homeruns
errors_cat1 arbitration;
run;
```

Output 8.3: Variance Information and Parameter Estimates from PROC MIANALYZE

The MIANALYZE Procedure

Model Information	
PARMS Data Set	WORK.GLMEST
Number of Imputations	5

Variance Information							
	Variance				Relative	Fraction	
Parameter	Between	Within	Total	DF	Increase in Variance	Missing Information	Relative Efficiency
intercept	11855	129014	143240	405.51	0.110270	0.103728	0.979676
on_base_percentage	130268	1296585	1452907	345.54	0.120564	0.112713	0.977954
homeruns	2.072394	33.332813	35.819486	829.83	0.074608	0.071663	0.985870
errors_cat1	516.725638	12875	13495	1894.8	0.048159	0.046952	0.990697
arbitration	496.191035	91214	91810	95099	0.006528	0.006506	0.998700

Parameter Estimates										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept	-878.634718	378.470472	-1622.64	-134.626	405.51	-967.990301	-700.623348	0	-2.32	0.0208
on_base_percentage	4632.210888	1205.365916	2261.43	7002.989	345.54	4041.355586	4961.671187	0	3.84	0.0001
homeruns	77.209731	5.984938	65.46	88.957	829.83	74.812057	78.626025	0	12.90	<.0001
errors_cat1	-50.434995	116.170056	-278.27	177.400	1894.8	-73.508336	-15.669939	0	-0.43	0.6642
arbitration	508.202296	303.000919	-85.68	1102.081	95099	487.766396	532.728284	0	1.68	0.0935

Output 8.3 includes the usual Variance Information and Parameter Estimates from the PROC MIANALYZE step.

8.3.2 PROC MIXED Output Data Set for Use in PROC MIANALYZE

Example 8.3.2 repeats the regression analysis of 8.3.1 using PROC MIXED rather than PROC GLM. This example illustrates special handling that is needed to prepare a PROC MIXED parameter output data set along with a COVB data set for input into PROC MIANALYZE. We demonstrate use of a multivariate test in MIANALYZE and show use of a binary indicator of a high number (9+) of errors in the PROC MIXED model rather than the ERRORS_CAT variable in a CLASS statement. These steps are required due to the restriction that a CLASS statement may not be used with the TEST statement in PROC MIANALYZE.

We first create the indicator of 9+ errors (HIGHERRORS) in a DATA STEP. Then, in PROC MIXED, use of the SOLUTION option on the model statement requests a fixed effects solution. The ODS OUTPUT statement creates a data set called mixedest containing fixed effects parameter estimates and standard errors as well as a COVB data set named mixedcovb.

```
data outc8ex1_1;
 set outc8ex1_1;
```

```
 higherrors=(errors_cat=1) ;
run;

proc mixed data=outc8ex1_1;
 model salary = on_base_percentage homeruns higherrors
arbitration / solution
   covb;
 ods output solutionf=mixedest covb=mixedcovb;
 by _imputation_;
run ;

proc print data=mixedest noobs;
run ;
```

Output 8.4: Partial Listing of Fixed Effects from the Mixedest Data Set (MI Repetitions 1 and 2)

Imputation	Effect	Estimate	StdErr	DF	tValue	Probt
1	Intercept	-858.39	359.45	332	-2.39	0.0175
1	On_base_percentage	4695.49	1139.51	332	4.12	<.0001
1	HomeRuns	74.8121	5.7777	332	12.95	<.0001
1	higherrors	-48.7980	113.55	332	-0.43	0.6677
1	Arbitration	492.28	302.24	332	1.63	0.1043
2	Intercept	-959.63	365.71	332	-2.62	0.0091
2	On_base_percentage	4872.32	1159.37	332	4.20	<.0001
2	HomeRuns	77.1849	5.8784	332	13.13	<.0001
2	higherrors	-15.6699	115.53	332	-0.14	0.8922
2	Arbitration	487.77	307.51	332	1.59	0.1136

A listing of unique parameter solutions (for imputation repetitions #1 and #2 only) from the mixedest data set is presented in Output 8.4. The output shows how the parameter estimates and standard errors from the PROC MIXED linear model are stored in the variables named EFFECT, ESTIMATE, and STDERR. Because we also saved a covariance data set called mixedcovb, use of PARMS=MIXEDEST and COVB(EFFECTVAR=ROWCOL)=MIXCOVB is required on the PROC MIANALYZE statement. Both of these data sets will be needed for the multivariate tests performed by the TEST statement. The MODELEFFECTS statement includes the INTERCEPT, and other covariates with the order established in PROC MIXED. The TEST statement with the MULT option requests a joint *F* test of two parameters, home runs and on-base percentage.

```
proc mianalyze parms=mixedest
covb(effectvar=rowcol)=mixedcovb;
```

```
 modeleffects intercept on_base_percentage homeruns
higherrors arbitration;
 test homeruns,on_base_percentage / mult;
run;
```

Output 8.5: Parameter Estimates and Multivariate Test Results from PROC MIANALYZE

The MIANALYZE Procedure

Model Information	
PARMS Data Set	WORK.MIXEDEST
COVB Data Set	WORK.MIXEDCOV
Number of Imputations	5

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
intercept	11855	129014	143240	405.51	0.110270	0.103728	0.979676
on_base_percentage	130268	1296585	1452907	345.54	0.120564	0.112713	0.977954
homeruns	2.072394	33.332613	35.819486	829.83	0.074608	0.071663	0.985870
higherrors	516.725638	12875	13495	1894.8	0.048159	0.046952	0.990697
arbitration	496.191035	91214	91810	95099	0.006528	0.006506	0.998700

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
intercept	-878.634718	378.470472	-1622.64	-134.626	405.51	-967.990301	-700.623348	0	-2.32	0.0208
on_base_percentage	4632.210888	1205.365916	2261.43	7002.989	345.54	4041.355586	4961.671187	0	3.84	0.0001
homeruns	77.209731	5.984938	65.46	88.957	829.83	74.812057	78.626025	0	12.90	<.0001
higherrors	-50.434995	116.170056	-278.27	177.400	1894.8	-73.508336	-15.669939	0	-0.43	0.6642
arbitration	508.202296	303.000919	-85.68	1102.081	95099	487.766396	532.728284	0	1.68	0.0935

Output 8.6: Parameter Estimates and Multivariate Test Results from PROC MIANALYZE

The MIANALYZE Procedure
Test: Test 1

Test Specification

Parameter	L Matrix					C
	intercept	on_base_percentage	homeruns	higherrors	arbitration	
TestPrm1	0	0	1.000000	0	0	0
TestPrm2	0	1.000000	0	0	0	0

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
TestPrm1	2.072394	33.332613	35.819486	829.83	0.074608	0.071663	0.985870
TestPrm2	130268	1296585	1452907	345.54	0.120564	0.112713	0.977954

Parameter Estimates										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	C	t for H0: Parameter=C	Pr > |t|
TestPrm1	77.209731	5.984938	65.462	88.957	829.83	74.812057	78.626025	0	12.90	<.0001
TestPrm2	4632.210888	1205.365916	2261.433	7002.989	345.54	4041.355586	4961.671187	0	3.84	0.0001

Multivariate Inference Assuming Proportionality of Between/Within Covariance Matrices				
Avg Relative Increase in Variance	Num DF	Den DF	F for H0: Parameter=Theta0	Pr > F
0.100660	2	289.67	114.74	<.0001

Parameter Estimates from PROC MIANALYZE (Outputs 8.5 and 8.6) match those from the PROC GLM analysis presented in Section 8.3.1, and the same inferences would be made using either procedure—PROC MIXED or PROC GLM. The Multivariate Inference table shows the Wald F statistic to be $F_{df=2,289}$= 114.74, p <.0001, suggesting that the regression coefficients for home runs and on-base percentage are jointly both significantly different from zero.

8.4 Imputation of NCS-R Data

The next application uses data from Part 2 of the National Comorbidity Survey-Replication (NCS-R). The working data set is called c8_ex2 and includes n=5,692 individual records. All variables are fully observed with the exception of MDE (a binary indicator of a report of a lifetime Major Depressive Episode, 1=Yes, 0=No) and EDUCAT (1=0–11 Yrs of Education, 2=12 Yrs of Education, 3=13–15 Yrs. of Education, and 4=16+ Yrs. of Education). Missing data values were arbitrarily generated in the original NCS-R data for purposes of this demonstration.

Variables used in this example:

INC_RSP: Respondent income, a continuous variable with no missing data

SEX: Gender (0=Male, 1=Female), a categorical variable with no missing data

AGE: Age at interview, a continuous variable with no missing data

SECU: Complex sample design cluster variable, a categorical variable (with values of 1 or 2) with no missing data

STR: Complex sample design stratum variable, a categorical variable with no missing data

STR_SECU: Combined strata and SECU variable, a categorical variable with no missing data

FINALP2WT: Final part 2 weight, a continuous variable with no missing data

RACECAT_: Race (1=Hispanic, 2=Non-Hispanic Black, 3=Other, 4=White), a categorical variable with no missing data

DSM_GAD: Binary indicator of DSM-IV General Anxiety Disorder (1=Yes, 5=No), a categorical variable with no missing data

GAD_OND: Age of onset for those with a diagnosis of GAD (DSM_GAD=1), a continuous variable with age of onset or missing data if no onset of GAD. There is no missing data on onset if diagnosed with the GAD disorder

EDUCAT: Education in 4 categories (1=0–11 Yrs of Education, 2=12 Yrs of Education, 3=13–15 Yrs. of Education, and 4=16+ Yrs. of Education), with some missing data

MDE: Major depressive episode indicator, a categorical variable (1=Yes, 0=No) with some missing data

As usual, evaluation of the missing data pattern and variable-specific missing data rates is an important first step. The following code produces a missing data pattern grid and selected univariate statistics for each variable.

```
proc mi data=c8_ex2 nimpute=0 simple;
 var inc_rsp sex age str_secu finalp2wt racecat_
dsm_gad educat mde;
run;
```

Output 8.7: Missing Data Patterns and Univariate Statistics for NCS-R Data Set

Missing Data Patterns

Group	inc_rsp	sex	age	str_secu	finalp2wt	racecat_	DSM_GAD	educat	mde	Freq	Percent	Group Means				
												inc_rsp	age	str_secu	finalp2wt	DSM_GAD
1	X	X	X	X	X	X	X	X	X	5292	92.97	24805	43.352419	265.268519	0.992335	4.467120
2	X	X	X	X	X	X	X	X	.	165	2.90	27214	41.690909	271.103030	1.084961	4.515152
3	X	X	X	X	X	X	X	.	X	235	4.13	23170	45.140426	261.765957	1.112958	4.642553

Univariate Statistics

Variable	N	Mean	Std Dev	Minimum	Maximum	Missing Values	
						Count	Percent
inc_rsp	5692	24807	26677	0	125000	0	0.00
age	5692	43.37807	16.57940	17.00000	98.00000	0	0.00
str_secu	5692	265.29304	111.54969	11.00000	422.00000	0	0.00
finalp2wt	5692	1.00000	0.95823	0.11441	10.10207	0	0.00
DSM_GAD	5692	4.47576	1.34999	1.00000	5.00000	0	0.00

Output 8.7 details the amount and pattern of missing data in the c8_ex2 data set.

The variables EDUCAT and MDE each have missing data with all other variables fully observed. The Missing Data Patterns grid produced by PROC MI indicates an arbitrary missing data pattern with about 7% of cases having one or more missing values.

The PROC MI code below multiply imputes the MDE and EDUCAT variables using the FCS method, which is appropriate for the arbitrary missing data pattern identified in this data set. Logistic regression is chosen to model the predictive distribution for missing values of the binary (MDE) and ordinal (EDUCAT) classification variables.

```
proc mi data=c8_ex2 nimpute=5 seed=21 out=c8_imp_fcs;
 class sex racecat_ educat str_secu mde dsm_gad;
 fcs nbiter=20 logistic (educat)logistic (mde);
 var inc_rsp sex age str_secu finalp2wt racecat_
dsm_gad educat mde;
run;
```

The CLASS statement declares a number of variables in the imputation model as classification type and SAS will automatically generate indicator variables to represent the levels of these variables in the predictive regression model. The VAR statement lists the variables included in the imputation model. With the FCS method, the order in which variables are imputed at each iteration defaults to ORDER=FREQ, from most observed to least observed.

Output 8.8: Model Information and FCS Model Specification

The MI Procedure

Model Information	
Data Set	WORK.C8_EX2
Method	FCS
Number of Imputations	5
Number of Burn-in Iterations	20
Seed for random number generator	21

FCS Model Specification	
Method	Imputed Variables
Regression	inc_rsp sex age finalp2wt
Logistic Regression	educat mde
Discriminant Function	str_secu racecat_ DSM_GAD

Output 8.8 provides information about the imputation, including *M*=5 imputations and the regression models used to model the predictive distribution used to impute MDE and EDUCAT. Though only EDUCAT and MDE require imputation, had the other variables needed imputation, they would have been modeled with the default regression or discriminant function methods, depending on variable type. The multiply imputed data set, c8_imp_fcs, with five complete MI repetitions, will be used in the next example once a final data management step is complete.

In the following data step, we create a new data set called c8_imp_fcs_1, including a series of indicator variables for male and the four categories of newly imputed EDUCAT. This approach is an alternative to use of the CLASS statement and is another way to input classification variables to PROC MIANALYZE.

```
data c8_imp_fcs_1;
 set c8_imp_fcs;
 ed1=educat=1;
 ed2=educat=2;
 ed3=educat=3;
 ed4=educat=4;
 male=(sex=1);
run;
```

8.5 PROC SURVEYPHREG Output Data Set for Use in PROC MIANALYZE

Section 8.5 presents use of PROC SURVEYPHREG to fit a design-adjusted Cox Proportional Hazards (PH) model (Cox 1972) to the NCS-R data imputed in the previous section. The Cox model is typically used for survival analysis where time is measured on a continuous scale. This model uses a dependent variable representing time to an event of interest or time to censor, if no event occurred during the observation window (Heeringa, West, and Berglund 2010).

The primary analytic goal is to analyze time from birth to onset of a general anxiety disorder diagnosis or age at censor, predicted by gender and controlling for education. In the NCS-R, with its retrospective measurement of diagnostic symptom onset, censoring for symptom-free individuals occurs at the time point at which she or he completed the NCS-R survey interview. We use AGEEVENT as the dependent variable set to either the age of onset of GAD or—if no general anxiety disorder was diagnosed—age at interview.

PROC SURVEYPHREG is used to analyze survival to onset of GAD or censor using a Cox model approach, which also accounts for the complex sample design through the use of the STRATA, CLUSTER, and WEIGHT statements. The dependent variable syntax declares AGEEVENT*DSM_GAD(5), meaning age of either the event of interest or censor is crossed with an indicator of being censored. In this case censored cases are identified as a value of 5 or “No” on the indicator variable for GAD diagnosis.

```
proc surveyphreg data=c8_imp_fcs_1;
 strata str; cluster secu; weight finalp2wt;
 model ageevent*dsm_gad(5)=male ed1 ed2 ed3;
 ods output parameterestimates=surveyphregest;
 by _imputation_;
run;
```

The PROC MIANALYZE syntax below specifies the PARMS= statement to read in the parameter data set produced by PROC SURVEYPHREG and the EDF= option to set the degrees of freedom used to 42 (approximate complete data degrees of freedom for the NCS-R complex sample design). Note that since the indicator variables for the reference category parameterization of SEX and EDUCAT were created in a DATA step and used directly in the PROC SURVEYPHREG model statement, the form of the parameter estimates output and the subsequent input to PROC MIANALYZE is a bit simpler that the alternative of declaring these variables as categorical predictors in a CLASS statement. However, either method will produce the same estimates of the Cox

PH model parameters.

```
proc mianalyze parms=surveyphregest edf=42;
 modeleffects male ed1 ed2 ed3;
run;
```

Output 8.9: Model Information and Parameter Estimates from PROC MIANALYZE

The MIANALYZE Procedure

Model Information	
PARMS Data Set	WORK.SURVEYPHREGEST
Number of Imputations	5

Variance Information

Parameter	Variance			DF	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
	Between	Within	Total				
male	0.000001072	0.009744	0.009746	40.128	0.000132	0.000132	0.999974
ed1	0.000626	0.028053	0.028804	38.828	0.026793	0.026426	0.994743
ed2	0.000129	0.011629	0.011783	39.541	0.013265	0.013176	0.997372
ed3	0.000533	0.016033	0.016672	38.055	0.039869	0.039046	0.992251

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
male	-0.561261	0.098720	-0.76076	-0.36176	40.128	-0.562457	-0.560095	0	-5.69	<.0001
ed1	-0.266795	0.169718	-0.61013	0.07654	38.828	-0.297359	-0.235700	0	-1.57	0.1241
ed2	-0.199316	0.108552	-0.41879	0.02015	39.541	-0.208947	-0.179954	0	-1.84	0.0739
ed3	0.148490	0.129122	-0.11289	0.40987	38.055	0.116908	0.178211	0	1.15	0.2573

Output 8.9 includes Model Information which echoes the names of the data set read into the procedure and number of imputations performed: SURVEYPHREGEST and five repetitions. The Parameter Estimates table includes standard errors that are both design-adjusted in PROC SURVEYPHREG and further adjusted for the variability due to imputation in PROC MIANALYZE. Based on Output 8.9, all else being equal, being male has a significant negative impact on the hazard for onset of GAD, as compared to women.

8.6 Summary

This chapter has provided several examples of preparation of output data sets from step 2 of the MI process for input into PROC MIANALYZE. There are numerous types of output data sets for use as input to PROC MIANALYZE, and in this chapter we have provided examples for procedures not already

demonstrated in earlier applications.

References

Allison, P. D. 2001. *Missing Data (Quantitative Applications in the Social Sciences)*. Thousand Oaks, CA: Sage Publications.

———. 2005. Imputation of categorical variables with PROC MI. Proceedings of the SAS Users Group International (SUGI). Volume 30, paper 113-30, 1–14.

Amemiya, T. 1985. *Advanced Econometrics*. Cambridge, MA: Harvard University Press.

Andridge, R. R., and R. J. A. Little. 2011. Proxy pattern-mixture analysis for survey nonresponse. *Journal of Official Statistics*, 27, 2, 153–180.

Barnard, J., and D. B. Rubin. 1999. Small-sample degrees of freedom with multiple imputation. *Biometrika*, 86, 948–955.

Bobb, J. F., D. O. Scharfstein, M. J. Daniels, F. S. Collins, and S. N. Kelada. 2011. Multiple imputation of missing phenotype data for QTL mapping. *Statistical Applications in Genetics and Molecular Biology*, 10, 1, 1–27.

Bodner, T. E. 2008. What improves with increased missing data imputations? *Structural Equation Modeling: A Multidisciplinary Journal* 15, 651–675.

Buck, S. F. 1960. A method of estimation of missing values in multivariate data suitable for use with an electronic computer. *Journal of the Royal Statistical Society*, Series B (Statistical Methodology), 22, 2, 302–306.

Carlin, J. B., J. C. Galati, and P. Royston, P. 2008. A new framework for managing and analyzing multiply imputed data in Stata. *The Stata Journal*, 8, 1, 49–67.

Carpenter, J. R., and M. G. Kenward. 2013. *Multiple Imputation and its Application*. Chichester: Wiley.

Cox, D. R. 1972. Regression models and life tables. *Journal of the Royal Statistical Society*, Series B, 20, 187–220, with discussion.

Enders, C. K. 2001. A primer on maximum likelihood algorithms for use with missing data. *Structural Equation Modeling: A Multidisciplinary Journal*, 8, 1, 128–141.

Fay, R. E., 1996. Alternative paradigms for the analysis of imputed survey data. *Journal of the American Statistical Association*, 91, 434, 490–498.

Gelman, A., and D. B. Rubin. 1992. Inference from iterative simulation using

multiple sequences. *Statistical Science*, 7, 4, 457–472.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin. 1995. *Bayesian Data Analysis*. London: Chapman and Hall.

Goodnight, J. H. 1979. A tutorial on the SWEEP operator. *American Statistician*, 33, 3, 149–158.

Guex, N., E. Migliavacca, and I. Xenarios. 2010. Multiple imputations applied to the DREAM3 phosphoproteomics challenge: A winning strategy. *PLoS ONE*, 5, 1, e8012.

Heckman, J. J. 1976. The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models. *Annals of Economic Social Measurement*, 5, 4, 475–492.

Heeringa, S. G., R. J. A. Little, and T. E. Raghunathan. 2002. Multivariate imputation of coarsened survey data on household wealth. In *Survey Nonresponse*, ed. R. M. Groves et al. New York: Wiley.

Heeringa, S. G., B. T. West, and P. A. Berglund. 2010. *Applied Survey Data Analysis*. London: Chapman and Hall.

Herzog, T. N., and D. B. Rubin. 1983. Using multiple imputations to handle nonresponse in sample surveys. In *Incomplete Data in Sample Surveys*, vol. 2: *Theory and Bibliography*, ed. W. G. Madow, I. Olkin, and D. B. Rubin. New York: Academic Press.

Hosmer, D. W., and S. Lemeshow. 2000. *Applied Logistic Regression*, 2nd ed. New York: Wiley.

Kalton, G. 1983. *Introduction to Survey Sampling*. Beverly Hills, CA: Sage.

Kalton, G., and D. Kasprzyk. 1986. The treatment of missing survey data. *Survey Methodology*, 12, 1–16.

Keel Data Set Repository, J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. 2011. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17, 2–3, 255–287.

Kessler, R. C., P. Berglund, W. T. Chiu, O. Demler, S. Heeringa, E. Hiripi, et al.. 2004. The US National Comorbidity Survey Replication (NCS-R): Design and field procedures. *International Journal of Methods in Psychiatric Research*, 13, 2, 69–92.

Kim, J. K. 2007. Regression fractional hot deck imputation. *Journal of the Korean Statistical Society*, 36, 423–434.

Kim, J. K. 2011. Parametric fractional imputation for missing data analysis.

Biometrika, 98, 119–132.

Kim, J. K., J. M. Brick, W. A. Fuller, and G. Kalton. 2006. On the bias of the multiple-imputation variance estimator in survey sampling. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68, 3, 509–521.

Kim, J. K., and W. A. Fuller. 2004. Fractional hot deck imputation. *Biometrika*, 91, 559–578.

Korn, E. L., and B. I. Graubard. 1999. *Analysis of Health Surveys*. New York: Wiley.

Li, K. H., T. E. Raghunathan, and D. B. Rubin. 1991. Large-sample significance levels from multiply imputed data using moment-based statistics and an F reference distribution. *Journal of the American Statistical Association*, 86, 1065–1073.

Liang, K. Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika*, 73, 1, 13–22.

Little, R. J. A. 1985. A note about models for selectivity bias. *Econometrica*, 53, 6, 1469–1474

———. 1988. A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, 83, 404, 1198–1202.

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

Madow, W. G., and I. Olkin, eds. 1983. *Incomplete Data in Sample Surveys*, vol. 3: *Proceedings of the Symposium*. New York: Academic Press.

Marchini, J., B. Howie, S. Myers, G. McVean, and P. Donnelly. 2007. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature Genetics*, 39, 906–913.

Rao, J. N. K., and J. Shao. 1992. Jackknife variance estimation with survey data under hot deck imputation. *Biometrika*, 79, 4, 811–822.

Raghunathan, T. E., and J. E. Grizzle. 1995. A split questionnaire survey design. *Journal of the American Statistical Association*, 90, 429, 54–63.

Raghunathan, T. E., J. M. Lepkowski, J. Van Hoewyk, and P. Solenberger. 2001. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, 27, 1, 85–95.

Raghunathan, T. E., P. W. Solenberger, and J. Van Hoewyk. 2002. *IVEware: Imputation and Variance Estimation Software User's Guide*. Ann Arbor, MI: Institute for Social Research, University of Michigan.

Reiter, J. P., and T. E. Raghunathan. 2007. The multiple adaptations of multiple imputation. *Journal of the American Statistical Association*, 102, 1462–1471.

Reiter, J. P., T. E. Raghunathan, and S. K. Kinney. 2006. The importance of modeling the sampling design in multiple imputation for missing data. *Survey Methodology*, 32, 2, 143–149.

Royston, P. 2005. Multiple imputation of missing values: Update of ice. *The Stata Journal*, 5, 4, 527–536.

Rubin, D. B. 1980. *Handling Nonresponse in Sample Surveys by Multiple Imputations*. U.S. Department of Commerce, Bureau of the Census Monograph.

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

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

Rubin, D. B., and N. Schenker. 1986. Multiple imputation for interval estimation from simple random samples with ignorable nonresponse. *Journal of the American Statistical Association*, 81, 394, 366–374.

Sande, I. 1983. Hot deck imputation procedures. In *Incomplete Data in Sample Surveys*, vol. 2: *Theory and Bibliographies*, ed. W. G. Madow, H. Nisselson, and I. Olkin. New York: Academic Press, 339–349.

Schafer, J. L. 1996. *MIX: Multiple Imputation for Mixed Continuous and Categorical Data*. Software library for S-PLUS. Written in S-PLUS and Fortran-77. Available at http://www.stat.psu.edu/~jls/misoftwa.html.

———. 1997. *Analysis of Incomplete Multivariate Data*. London: Chapman & Hall.

———. 1999. *NORM: Multiple Imputation of Incomplete Multivariate Data under a Normal Model*, Version 2. Software for Windows 95/98/NT, available from http://www.stat.psu.edu/~jls/misoftwa.html.

Schafer, J. L., T. M. Ezatti-Rice, W. Johnson, M. Khare, R. J. A. Little, and D. B. Rubin, D.B. 1996. The NHANES III multiple imputation project. *Proceedings of the Survey Research Methods Section, American Statistical Association*, 696–701.

Schenker, N, T. E. Raghunathan, P. L. Chiu, D. M. Makuc, G. Y. Zhang, and A. J. Cohen. 2006. Multiple imputation for missing income data in the National Health Interview Survey. *Journal of the American Statistical Association*, 101, 924–933.

Schenker, N., T. E. Raghunathan, and I. Bondarenko, I. 2010. Improving on analyses of self-reported data in a large-scale health survey by using information from an examination-based survey. *Statistics in Medicine*, 29, 533–545.

Taylor, L., and X. H. Zhou. 2009. Multiple imputation methods for treatment noncompliance and nonresponse in randomized clinical trials. *Biometrics*, 65, 1, 88–95.

Thall, P. F., and S. C. Vail. 1990. Some covariance models for longitudinal count data with overdispersion. *Biometrics*, 46, 3, 657–671

Thomas, N., T. E. Raghunathan, N. Schenker, M. J. Katzoff, and C. L. Johnson. 2006. An evaluation of matrix sampling methods using data from the National Health and Nutrition Examination Survey. *Survey Methodology*, 32, 2, 217–231.

van Buuren, S. 2011. Multiple imputation of multilevel data. In *The Handbook of Advanced Multilevel Analysis*, ed. J. J. Hox and J. K. Roberts. New York: Routledge. Pp. 173–196.

van Buuren S. 2012. *Flexible Imputation of Missing Data*. Boca Raton, FL: CRC Press.

van Buuren, S., H. C. Boshuizen, and D. L. Knook. 1999. Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine*, 18, 681–694.

West, B. T., P. Berglund, P., and S. G. Heeringa. 2008. A closer examination of subpopulation analysis of complex sample survey data. *Stata Journal*, 8, 4, 520–531.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Table of Contents