SAS/STAT® 14.1 User's Guide

The SURVEYIMPUTE Procedure

This document is an individual chapter from *SAS/STAT® 14.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2015. *SAS/STAT® 14.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 14.1 User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

July 2015

Chapter 110
The SURVEYIMPUTE Procedure

Contents

Overview: SURVEYIMPUTE Procedure

The SURVEYIMPUTE procedure imputes missing values of an item in a data set by replacing them with observed values from the same item. The principles by which the imputation is performed are particularly useful for survey data. PROC SURVEYIMPUTE also computes replicate weights (such as jackknife weights) that account for the imputation and that can be used for replication-based variance estimation for complex surveys. The procedure implements a fractional hot-deck imputation technique (Kim and Fuller 2004; Fuller 2009; Kim and Shao 2014) in addition to some traditional hot-deck imputation techniques (Andridge and Little 2010).

Nonresponse is a common problem in almost all surveys of human populations. Estimators that are based on survey data that include nonresponse can suffer from nonresponse bias if the nonrespondents are different from the respondents. Estimators that use complete cases (only the observed units) might also be less precise. Imputation techniques are important tools for reducing nonresponse bias and producing efficient estimators.

The main objectives of any imputation technique are to eliminate the nonresponse bias and to provide an imputed data set that results in consistent analyses conducted with the imputed data. In addition, a variance estimator must be available that accounts for both the sampling variance and the imputation variance. Imputation techniques use implicit or explicit models. Some model-based imputation techniques include multiple imputation, mean imputation, and regression imputation. For more information about multiple imputation in SAS/STAT, see Chapter 75, "The MI Procedure," and Chapter 76, "The MIANALYZE Procedure."

Imputation techniques that do not use explicit models include hot-deck imputation, cold-deck imputation, and fractional imputation. PROC SURVEYIMPUTE implements imputation techniques that do not use explicit models. It also produces replicate weights that can be used with any survey analysis procedure in SAS/STAT to estimate both the sampling variability and the imputation variability.

Hot-deck imputation is the most commonly used imputation technique for survey data. A donor is selected for a recipient unit, and the observed values of the donor are imputed for the missing items of the recipient. Although the imputation method is straightforward, the variance estimator that accounts for imputation variance might not be simple and is often ignored in practice. PROC SURVEYIMPUTE does not create imputation-adjusted replicate weights for hot-deck imputation.

Fractional hot-deck imputation (Kalton and Kish 1984; Fay 1996; Kim and Fuller 2004; Fuller and Kim 2005), also known as fractional imputation (FI), is a variation of hot-deck imputation in which one missing item for a recipient is imputed from multiple donors. Each donor donates a fraction of the original weight of the recipient such that the sum of the fractional weights from all the donors is equal to the original weight of the recipient. For fully efficient fractional imputation (FEFI), all observed values in an imputation cell are used as donors for a recipient unit in that cell (Kim and Fuller 2004).

The SURVEYIMPUTE procedure implements single and multiple hot-deck imputation and FEFI. Available donor selection techniques include simple random selection with or without replacement, probability proportional to weights selection (Rao and Shao 1992), and approximate Bayesian bootstrap selection (Rubin and Schenker 1986).

The remaining sections of this chapter are organized as follows:

- "Getting Started: SURVEYIMPUTE Procedure" on page 8925 introduces PROC SURVEYIMPUTE with an example.

- "Syntax: SURVEYIMPUTE Procedure" on page 8930 describes the syntax of the procedure.
- "Details: SURVEYIMPUTE Procedure" on page 8941 summarizes the imputation techniques that PROC SURVEYIMPUTE uses.
- "Examples: SURVEYIMPUTE Procedure" on page 8958 includes some additional examples of useful applications.

Getting Started: SURVEYIMPUTE Procedure

This example shows how you can use PROC SURVEYIMPUTE to impute missing values and compute imputation-adjusted statistics for sample survey data. The example uses simulated data from a customer satisfaction survey for a student information system (SIS), which is a software product that provides modules for student registration, class scheduling, attendance, grade reporting, and other functions.

The software company conducted a survey of school personnel who use the SIS. A probability sample of SIS users was selected from the study population, which included SIS users at middle schools and high schools in three states, Georgia, South Carolina, and North Carolina. The sample design for this survey was a two-stage stratified design. A first-stage sample of schools was selected from the list of schools in the three states that use the SIS. The list of schools, which are the primary sampling units (PSU), was stratified by state and by customer status (whether the school was a new user or a renewal user of the system). Within the strata, schools were selected with probability proportional to size and with replacement, where the size measure was school enrollment. From each sample school, five staff members were randomly selected with replacement as the second-stage units to complete the SIS satisfaction questionnaire. These staff members include both teachers and administrators.

The SAS data set SIS_Survey_Sub contains the survey results and the sample design information that is needed to analyze the data. The data set contains the following items:

- State: state where the school is located
- NewUser: 1 if the school is a new user of SIS or 0 if not
- School: school identification (PSU)
- SamplingWeight: sampling weight
- Department: 0 for teachers and 1 for administrators
- Response: coded from 1 to 5, where 1 represents "Very Unsatisfied" and 5 represents "Very Satisfied"

The following statements request the imputation of missing values for Department and Response by using the fully efficient fractional imputation (FEFI) method:

```
proc surveyimpute data=SIS_Survey_Sub method=fefi varmethod=jackknife;
   class Department Response;
   var Department Response;
   strata State NewUser;
   cluster School;
```

```
   weight SamplingWeight;
   output out=SIS_Survey_Imputed outjkcoefs=SIS_JKCoefs;
run;
```

The PROC SURVEYIMPUTE statement invokes the procedure. The DATA= option in the PROC SURVEYIMPUTE statement specifies the input data set containing the missing values, the METHOD=FEFI option requests the fully efficient fractional imputation method, and the VARMETHOD= option requests the imputation-adjusted jackknife replicate weights. The CLASS statement specifies the classification variables. The STRATA, CLUSTER, and WEIGHT statements specify the strata, clusters (PSUs), and weight variables. The VAR statement specifies the variables to be imputed (Department and Response). By default, both the variables Department and Response are imputed jointly. Therefore, the missing values for Department will be imputed conditionally on the observed levels of Response, and the missing values for Response will be imputed conditionally on the observed levels of Department. Observations that contain missing values for both Department and Response will be imputed by using the joint observed levels of Department and Response. The OUT= option in the OUTPUT statement names a SAS data set to save the imputed data. The OUTJKCOEFS= option in the OUTPUT statement names a SAS data set to save the jackknife coefficients.

Summary information about the data, CLASS levels, and survey design is shown in Figure 110.1. The "Imputation Information" table summarizes the imputation information. The "Number of Observations" table displays the number of observations that PROC SURVEYIMPUTE reads and uses. This table also displays the sum of weights that are read and used. The sum of weights read (6,468) can be used as an estimator of the population size. For example, the 235 observation units in the SIS_Survey_Sub data set represent 6,468 teachers and administrative staff in the population. The "Class Level Information" table shows that Department has two levels and Response has five levels. The "Design Summary" table shows that 47 schools are selected in the sample from six strata.

Figure 110.1 Summary Information

The SURVEYIMPUTE Procedure

Imputation Information	
Data Set	WORK.SIS_SURVEY_SUB
Weight Variable	SamplingWeight
Stratum Variables	State
	NewUser
Cluster Variable	School
Imputation Method	FEFI

Number of Observations Read	235
Number of Observations Used	235
Sum of Weights Read	6468
Sum of Weights Used	6468

Class Level Information		
Class	**Levels**	**Values**
Department	2	0 1
Response	5	1 2 3 4 5

Design Summary	
Number of Strata	6
Number of Clusters	47

The "Missing Data Patterns" table in Figure 110.2 lists distinct missing data patterns along with their corresponding frequencies and weighted percentages. An "X" means that the variable is observed in the corresponding group, and a "." means that the variable is missing. The table also displays group-specific variable means. In this hypothetical example, five respondents have unit nonresponse (both variables in the VAR statement contain missing values), 73 respondents have item nonresponse (only one variable in the VAR statement contains a missing value), and 157 respondents have complete response (no variables in the VAR statement contain missing values). Among the 73 item nonrespondents, for 52 respondents, Department is observed but Response is not observed; for 21 respondents, Response is observed but Department is not observed. The estimated percentages in the sample for unit nonresponse, item nonresponse, and complete response are 2.1%, 31.1%, and 66.8%, respectively.

Figure 110.2 Missing Data Patterns

Missing Data Patterns

Group	Department	Response	Freq	Sum of Weights	Unweighted Percent	Weighted Percent
1	X	X	157	4272	66.81	66.05
2	X	.	52	1480	22.13	22.88
3	.	X	21	586	8.94	9.06
4	.	.	5	130	2.13	2.01

Missing Data Patterns

Group Means

Group	Department 0	Department 1	Response 1	Response 2	Response 3	Response 4	Response 5
1	0.440309	0.559691	0.184457	0.206695	0.265684	0.209738	0.133427
2	0.641892	0.358108	.	.	.	.	.
3	.	.	0.261092	0.235495	0.230375	0.085324	0.187713
4	.	.	.	.	.	.	.

The "Imputation Summary" table in Figure 110.3 lists the number of nonmissing observations, missing observations, and imputed observations. There are 78 observations that have missing values for at least one variable, and all 78 missing observations are imputed.

Figure 110.3 Imputation Summary

Imputation Summary

Observation Status	Number of Observations	Sum of Weights
Nonmissing	157	4272
Missing	78	2196
Missing, Imputed	78	2196
Missing, Not Imputed	0	0
Missing, Partially Imputed	0	0

The output data set SIS_Survey_Imputed contains the observed data and the imputed values for Department and Response. In addition, this data set contains the imputation-adjusted full-sample weight (ImpWt), observation unit identification (UnitId), recipient index (Recipient), and imputation-adjusted jackknife replicate weights (ImpRepWt_1, . . . , ImpRepWt_47).

Suppose you want to compute frequency tables by using the imputed data set. The following statements request one-way tables for Department and Response and a two-way table for Department by Response. The analyses include the imputed values and account for both the design variance and the imputation variance.

```
proc surveyfreq data=SIS_Survey_Imputed varmethod=jackknife;
   table department response  department*response;
   weight ImpWt;
   repweights ImpRepWt: / jkcoefs=SIS_JKCoefs;
run;
```

The DATA= option in the PROC SURVEYFREQ statement specifies the input data set for analysis, SIS_Survey_Imputed, which contains the observed values and the imputed values for Department and Response. The FEFI technique uses multiple donor cells for a missing item. Therefore, the number of rows in the SIS_Survey_Imputed data set is greater than the number of rows in the observed data set, SIS_Survey_Sub. Each row in the SIS_Survey_Sub data set represents an observation unit, but this is not true for the SIS_Survey_Imputed data set. Therefore, it is very important to use only the weighted statistics from SIS_Survey_Imputed. The WEIGHT statement specifies the weight variable ImpWt, which is adjusted for the FEFI method. The imputation-adjusted jackknife replicate weights are saved in the variables ImpRepWt_1, ..., ImpRepWt_47 in the SIS_Survey_Imputed data set. The REPWEIGHTS statement names the replicate weight variables and the jackknife coefficients data set, SIS_JKCOEFS. You should not use the unadjusted full-sample weights (SamplingWeight) or unadjusted replicate weights along with the imputed data.

Figure 110.4 displays some summary information. Note that the sum of weights in Figure 110.4 matches the sum of weights read from Figure 110.1, but the number of observations in Figure 110.4 (509) does not match the number of observations from Figure 110.1 (235). The sum of weights from both PROC SURVEYIMPUTE and PROC SURVEYFREQ represents the population size. The number of observations in Figure 110.1 represents the number of observation units, but the number of observations in Figure 110.4 represents the number of rows in the data set that include the observed units and the imputed rows. The number of replicates is 47, which is the same as the number of schools (PSUs).

Figure 110.4 One-Way Table

The SURVEYFREQ Procedure

Data Summary	
Number of Observations	509
Sum of Weights	6468

Variance Estimation	
Method	Jackknife
Replicate Weights	SIS_SURVEY_IMPUTED
Number of Replicates	47

Figure 110.5 displays one-way tables for Department and Response. The Frequency column does not represent frequencies for observation units from the SIS_Survey_Sub data set. These frequencies represent the frequency of data lines in the SIS_Survey_Sub data set, which also include the imputed rows. The Weighted Frequency, Std Err of Wgt Freq, Percent, and Std Err of Percent columns use the imputation-adjusted full-sample weight and replicate weights. You should use the weighted statistics from these columns. For example, an estimated 49.47% of SIS users are teachers, with a standard error of 6.64%. An estimate of "Very Satisfied" users is 14.19%, with a standard error of 3.77%.

Figure 110.5 One-Way Table

Table of Department					
Department	**Frequency**	**Weighted Frequency**	**Std Err of Wgt Freq**	**Percent**	**Std Err of Percent**
0	278	3200	429.52229	49.4729	6.6407
1	231	3268	429.52229	50.5271	6.6407
Total	509	6468	5.3369E-11	100.000	

Table of Response					
Response	**Frequency**	**Weighted Frequency**	**Std Err of Wgt Freq**	**Percent**	**Std Err of Percent**
1	100	1256	291.92305	19.4153	4.5133
2	103	1371	361.02585	21.1976	5.5817
3	112	1710	305.26968	26.4371	4.7197
4	100	1213	283.69298	18.7598	4.3861
5	94	917.82544	243.87967	14.1903	3.7706
Total	509	6468	3.2868E-11	100.000	

Figure 110.6 displays the two-way table for Department by Response. The Weighted Frequency, Std Err of Wgt Freq, Percent, and Std Err of Percent columns use the imputation-adjusted full-sample weight and replicate weights. You should use the weighted statistics from these columns. Among the teachers, 8.10% are estimated to be "Very Satisfied," with a standard error of 3.11%. Among the administrators, 6.09% are "Very Satisfied," with a standard error of 2.43%.

Figure 110.6 Crosstabulation

Table of Department by Response						
Department	**Response**	**Frequency**	**Weighted Frequency**	**Std Err of Wgt Freq**	**Percent**	**Std Err of Percent**
0	**1**	57	637.83724	246.18741	9.8614	3.8062
	2	55	743.50947	334.28335	11.4952	5.1683
	3	64	951.95811	258.23015	14.7180	3.9924
	4	49	342.84458	150.44168	5.3006	2.3259
	5	53	523.75680	200.99126	8.0977	3.1075
	Total	278	3200	429.52229	49.4729	6.6407
1	**1**	43	617.94159	209.04386	9.5538	3.2320
	2	48	627.55128	185.53346	9.7024	2.8685
	3	48	757.99401	237.82609	11.7191	3.6770
	4	51	870.53830	262.37514	13.4592	4.0565
	5	41	394.06863	156.95381	6.0926	2.4266
	Total	231	3268	429.52229	50.5271	6.6407
Total	**1**	100	1256	291.92305	19.4153	4.5133
	2	103	1371	361.02585	21.1976	5.5817
	3	112	1710	305.26968	26.4371	4.7197
	4	100	1213	283.69298	18.7598	4.3861
	5	94	917.82544	243.87967	14.1903	3.7706
	Total	509	6468	4.7965E-12	100.000	

Syntax: SURVEYIMPUTE Procedure

The following statements are available in the SURVEYIMPUTE procedure. Items within < > are optional.

PROC SURVEYIMPUTE < *options* > **;**
    BY *variables* **;**
    CELLS *variables* **;**
    CLASS *variable* < **(***options***)** > < . . . *variable* < **(***options***)** > > < / *options* > **;**
    CLUSTER *variables* **;**
    ID *variable* **;**
    IMPJOINT < *variables* > **;**
    OUTPUT < **OUT=***SAS-data-set* >< **OUTJKCOEFS=***SAS-data-set* >< *keyword***=***name*
        . . . *keyword***=***name* > **;**
    REPWEIGHTS *variables* **;**
    STRATA *variables* **;**
    VAR *variables* **;**
    WEIGHT *variable* **;**

The PROC SURVEYIMPUTE and VAR statements are required.

The following sections describe the PROC SURVEYIMPUTE statement and then describe the other statements in alphabetical order.

PROC SURVEYIMPUTE Statement

PROC SURVEYIMPUTE < *options* > **;**

The PROC SURVEYIMPUTE statement invokes the SURVEYIMPUTE procedure. The DATA= option identifies the data set to be analyzed. Table 110.1 summarizes the options available in the PROC SURVEYIMPUTE statement.

Table 110.1 Options Available in the PROC SURVEYIMPUTE Statement

Option	Description
DATA=	Names the input data set
METHOD=	Specifies the imputation method
NDONORS=	Specifies the number of donors for a recipient
NOPRINT	Suppresses all displayed output
ORDER=	Specifies the sort order of CLASS variables
SEED=	Specifies the random number seed
VARMETHOD=	Specifies the variance estimation method

You can specify the following *options*.

DATA=*SAS-data-set*

names the SAS data set that contains the data to be analyzed. If you omit the DATA= option, PROC SURVEYIMPUTE uses the most recently created SAS data set.

METHOD=FEFI | HOTDECK | HD < (*method-option***) >**

specifies the imputation method to impute missing values for all variables in the VAR statement. By default, METHOD=FEFI.

Table 110.2 summarizes the available *method-options*.

Table 110.2 Imputation Methods

METHOD=	Imputation Method	*Method-Options*
FEFI	Fully efficient fractional imputation method	ABSEMWTCONV=
		MAXDONORCELLS=
		MAXEMITER=
		RELEMWTCONV=
HOTDECK | HD	Approximate Bayesian bootstrap	SELECTION=ABB
	Simple random sampling without replacement	SELECTION=SRSWOR
	Simple random sampling with replacement	SELECTION=SRSWR
	Weighted selection	SELECTION=WEIGHTED

By default, if all variables that you specify in the VAR statement are also specified in the CLASS statement, then METHOD=FEFI. Otherwise, the default imputation method is METHOD=HOTDECK. You can specify the following values:

FEFI < (*method-options***) >**

requests the fully efficient fractional imputation (FEFI) method. For more information, see the section "Fully Efficient Fractional Imputation" on page 8944.

You can specify the following *method-options*:

ABSEMWTCONV=*r*

specifies the absolute weighted convergence criterion. The expectation maximization (EM) algorithm stops when the maximum absolute difference between the fractional weights from the previous iteration and the fractional weights from the current iteration is less than *r*. The default value of *r* is 0.00001. For more information, see the section "FEFI Algorithm" on page 8945.

RELEMWTCONV=*r*

specifies a relative weighted convergence criterion. The expectation maximization (EM) algorithm stops when the maximum relative absolute difference between the weights from the previous iteration and the weights from the current iteration is less than *r*. The default value of *r* is 0.001. For more information, see the section "FEFI Algorithm" on page 8945.

MAXDONORCELLS=*i*
specifies the maximum number (*i*) of donor cells allowed for a recipient unit. By default, MAXDONORCELLS=5000.

MAXEMITER=*i*
specifies the maximum number (*i*) of iterations for the expectation maximization (EM) algorithm. By default, MAXEMITER=100.

HOTDECK < (SELECTION=*selection-option*) >

HD < (SELECTION=*selection-option*) >
requests the hot-deck imputation method. For more information, see the section "Hot-Deck Imputation" on page 8949.

By default, SELECTION=SRSWR for METHOD=HOTDECK if you do not use the WEIGHT statement, and SELECTION=WEIGHTED for METHOD=HOTDECK if you use the WEIGHT statement. You can specify one of the following donor selection *selection-options*:

ABB
requests donor selection by using the approximate Bayesian bootstrap method. For more information, see the section "Approximate Bayesian Bootstrap" on page 8951.

SRSWOR
requests donor selection by using simple random samples without replacement. For more information, see the section "Simple Random Samples without Replacement" on page 8951.

SRSWR
requests donor selection by using simple random samples with replacement. For more information, see the section "Simple Random Samples with Replacement" on page 8951.

WEIGHTED
requests donor selection by using probability proportional to respondent weights with replacement. For more information, see the section "Weighted Selection" on page 8951.

NDONORS=*r*
specifies the number of donor units to impute for a recipient unit when METHOD=HOTDECK. If you specify NDONORS=0, then no imputation is performed. When METHOD=FEFI, the SURVEYIMPUTE procedure performs fully efficient fractional imputation, for which the NDONORS= option does not apply. When METHOD=HOTDECK, NDONORS=1 by default.

NOPRINT
suppresses all displayed output. This option temporarily disables the Output Delivery System (ODS); for more information about ODS, see Chapter 20, "Using the Output Delivery System."

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sort order for the levels of the classification variables (which are specified in the CLASS statement).

This option applies to the levels for all classification variables, except when you use the (default) ORDER=FORMATTED option with numeric classification variables that have no explicit format. In that case, the levels of such variables are ordered by their internal value.

The ORDER= option can take the following values:

Value of ORDER=	Levels Sorted By
DATA	Order of appearance in the input data set
FORMATTED	External formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	Descending frequency count; levels with the most observations come first in the order
INTERNAL	Unformatted value

By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine-dependent.

For more information about sort order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

SEED=*number*

specifies the initial seed for random number generation that is used to select donor units for METHOD=HOTDECK. The *number* should be a positive integer. If you do not specify this option or if *number* is 0, PROC SURVEYIMPUTE uses the time of day from the computer's clock to obtain the initial seed.

VARMETHOD=BRR < (*method-options***) > | JACKKNIFE | NONE**

REPWEIGHTSTYPE=BRR < (*method-options***) > | JACKKNIFE | NONE**

computes imputation-adjusted replicate weights. You can specify the following values:

BRR < (*method-options***) >**

computes the imputation-adjusted balanced repeated replication (BRR) weights. The BRR method requires a stratified sample design with two primary sampling units (PSUs) in each stratum. If you specify the VARMETHOD=BRR option, you must also use a STRATA statement unless you provide replicate weights by using a REPWEIGHTS statement. For more information, see the section "Balanced Repeated Replication (BRR) Method" on page 8952.

You can specify the following *method-options* in parentheses after the VARMETHOD=BRR option:

FAY <=*value***>**

requests Fay's method, which is a modification of the BRR method. For more information, see the section "Unadjusted Fay's BRR Replicate Weights" on page 8953.

You can specify the *value* of the Fay coefficient, which is used in converting the original sampling weights to replicate weights. The Fay coefficient must be a nonnegative number less than 1. By default, the value of the Fay coefficient is 0.5.

HADAMARD=*SAS-data-set*

H=*SAS-data-set*

names a SAS data set that contains the Hadamard matrix for BRR replicate construction. If you do not provide a Hadamard matrix by using this *method-option*, PROC SURVEYIMPUTE generates an appropriate Hadamard matrix for replicate construction. For more information, see the sections "Balanced Repeated Replication (BRR) Method" on page 8952 and "Hadamard Matrix" on page 8953.

If a Hadamard matrix of a particular dimension exists, it is not necessarily unique. Therefore, if you want to use a specific Hadamard matrix, you must provide the matrix as a SAS data set in the HADAMARD= *method-option*.

In the HADAMARD= input data set, each variable corresponds to a column of the Hadamard matrix, and each observation corresponds to a row of the matrix. You can use any variable names in the HADAMARD= data set. All values in the data set must equal either 1 or –1. You must ensure that the matrix you provide is indeed a Hadamard matrix—that is, $\mathbf{A}'\mathbf{A} = R\mathbf{I}$, where $\mathbf{A}$ is the Hadamard matrix of dimension R and $\mathbf{I}$ is an identity matrix. PROC SURVEYIMPUTE does not check the validity of the Hadamard matrix that you provide.

The HADAMARD= input data set must contain at least H variables, where H denotes the number of first-stage strata in your design. If the data set contains more than H variables, PROC SURVEYIMPUTE uses only the first H variables. Similarly, the HADAMARD= input data set must contain at least H observations.

If you do not specify the REPS= *method-option*, then the number of replicates is equal to the number of observations in the HADAMARD= input data set. If you specify the number of replicates—for example, REPS=*nreps*—then the procedure uses the first *nreps* observations in the HADAMARD= data set to construct the replicates.

You can specify the PRINTH *method-option* to display the Hadamard matrix that the procedure uses to construct replicates for BRR.

PRINTH

displays the Hadamard matrix that is used to construct replicates for BRR. When you provide the Hadamard matrix in the HADAMARD= *method-option*, PROC SURVEYIMPUTE displays only the rows and columns that are actually used to construct replicates. For more information, see the sections "Balanced Repeated Replication (BRR) Method" on page 8952 and "Hadamard Matrix" on page 8953.

The PRINTH *method-option* is not available when you use a REPWEIGHTS statement to provide replicate weights, because the procedure does not use a Hadamard matrix in this case.

REPS=*number*

specifies the number of replicates for BRR variance estimation. The value of *number* must be an integer greater than 1.

If you do not provide a Hadamard matrix by using the HADAMARD= *method-option*, the number of replicates should be greater than the number of strata and should be a multiple of 4. For more information, see the section "Balanced Repeated Replication (BRR) Method" on page 8952. If a Hadamard matrix cannot be constructed for the REPS= value that you specify, the value is increased until a Hadamard matrix of that dimension can be constructed. Therefore, it is possible for the actual number of replicates used to be larger than the REPS= value that you specify.

If you provide a Hadamard matrix by using the HADAMARD= *method-option*, the value of REPS= must not be greater than the number of rows in the Hadamard matrix. If you provide a Hadamard matrix and do not specify the REPS= *method-option*, the number of replicates equals the number of rows in the Hadamard matrix.

If you do not specify the REPS= or HADAMARD= *method-option* and do not include a REPWEIGHTS statement, the number of replicates equals the smallest multiple of 4 that is greater than the number of strata.

If you provide replicate weights by using a REPWEIGHTS statement, PROC SURVEYIMPUTE does not use the REPS= *method-option*. When you use a REPWEIGHTS statement, the number of replicates equals the number of REPWEIGHTS variables.

JACKKNIFE

JK

computes the imputation-adjusted jackknife replicate weights. For more information, see the section "Jackknife Method" on page 8954.

NONE

does not compute replicate weights.

By default, VARMETHOD=JACKKNIFE when METHOD=FEFI, and VARMETHOD=NONE when METHOD=HOTDECK.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SURVEYIMPUTE to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the SURVEYIMPUTE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

Note that using a BY statement provides completely separate imputation within the BY groups. For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

CELLS Statement

CELLS *variables* **;**

The CELLS statement names the variables that identify the imputation cells. The imputation cells divide the data into groups of similar units. The combination of levels of CELLS variables defines the imputation cells. If you do not use this statement, then all observation units are assumed to be in one imputation cell.

CLASS Statement

CLASS *variable* < **(***options***)** > *. . .* < *variable* < **(***options***)** > > < */ options* > **;**

The CLASS statement names the classification variables for the analysis. Most *options* can be specified either as individual variable *options* or as global *options*. You can specify *options* for each variable by enclosing the options in parentheses after the variable name. You can also specify global *options* for the CLASS statement by placing the *options* after a slash (/). Global *options* are applied to every variable that is specified without an *option* in the CLASS statement. However, individual CLASS variable *options* override the global *options*. You can specify the following *options* either as global options or as individual options:

DESCENDING

DESC

reverses the sort order of the classification variable. If you specify both the DESCENDING and ORDER= options, PROC SURVEYIMPUTE orders the categories according to the ORDER= option and then reverses that order.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sort order for the levels of classification variables.

The following table shows how PROC SURVEYIMPUTE interprets values of the ORDER= option:

Value of ORDER=	Levels Sorted By
DATA	Order of appearance in the input data set
FORMATTED	External formatted values, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) values
FREQ	Descending frequency count; levels with more observations come earlier in the order
INTERNAL	Unformatted value

By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine-dependent. When ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. For ORDER=FREQ, the frequency counts are unweighted. For more information about sort order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

TRUNCATE < =*n*>
specifies the length *n* of CLASS variable values to use in determining CLASS variable levels. The default is to use the full formatted length of the CLASS variable. If you specify the TRUNCATE option without the length *n*, the first 16 characters of the formatted values are used. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases before SAS 9. The TRUNCATE option is available only as a global option.

CLUSTER Statement

CLUSTER *variables* **;**

The CLUSTER statement names variables that identify the first-stage clusters in a clustered sample design. First-stage clusters are also known as primary sampling units (PSUs). The combinations of categories of CLUSTER variables define the clusters in the sample. If you also use the STRATA statement, clusters are nested within strata.

If your sample design has clustering at multiple stages, you should specify only the first-stage clusters (PSUs) in the CLUSTER statement. For more information, see the section "Specifying the Sample Design" on page 8941.

If you use a REPWEIGHTS statement to provide replicate weights for the BRR or jackknife variance estimation method, you do not need to use a CLUSTER statement.

The CLUSTER *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric, but PROC SURVEYIMPUTE treats them as categorical variables. The formatted values of the CLUSTER variables determine the cluster variable levels. Thus, you can use formats to group values into levels. For more information, see the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Formats and Informats: Reference*.

You can use multiple CLUSTER statements to specify CLUSTER variables. PROC SURVEYIMPUTE uses variables from all CLUSTER statements to create clusters.

ID Statement

ID *variable* **;**

The ID statement names a *variable* in the DATA= input data set to identify observation units. When you use an OUTPUT statement, the OUT= data set includes the ID variable. PROC SURVEYIMPUTE uses the ID variable values to identify the donor observations. If you do not use an ID statement, the procedure creates a new variable named UnitID that uses the observation numbers to identify the donor observations.

IMPJOINT Statement

IMPJOINT < *variables*> **;**

The IMPJOINT statement specifies the names of variables that are to be imputed jointly for the fully efficient fractional imputation (FEFI) method. If you do not use the IMPJOINT statement, then all the variables that you specify in the VAR statement are imputed jointly. You can use multiple IMPJOINT statements. The levels of the variables in the IMPJOINT statement describe a nonparametric imputation model for the expectation maximization (EM) step for the fractional imputation.

If you use the following IMPJOINT statements, then the variables y1, y2, and y3 are imputed jointly, and the variables y4 and y5 are imputed jointly:

```
IMPJOINT y1 y2 y3;
IMPJOINT y4 y5;
```

Analysis variables, which you specify in the VAR statement, can appear in only one IMPJOINT statement. However, auxiliary variables that you do not specify in the VAR statement can be specified in multiple IMPJOINT statements.

If you do not specify any variable names in the IMPJOINT statement, then every variable in the VAR statement is imputed marginally.

The IMPJOINT statement is ignored when you specify the METHOD=HOTDECK option in the PROC SURVEYIMPUTE statement.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > < **OUTJKCOEFS=***SAS-data-set* > < *keyword***=***name* . . . *keyword***=***name* > **;**

The OUTPUT statement creates a SAS data set that contains the imputed data. You must use the OUTPUT statement to store the imputed data in a SAS data set. If you use multiple OUTPUT statements, then PROC SURVEYIMPUTE uses only the first OUTPUT statement and ignores the rest. The OUTPUT OUT= data set contains all the variables from the DATA= input data set, imputed values for missing values for the variables in the VAR statement, and some observation-level quantities. These quantities can include the fractionally adjusted weights, replicate weights, recipient numbers, and donor identifications.

You can specify the following in the OUTPUT statement:

OUT=*SAS-data-set*

names the output data set. If you use the OUTPUT statement but omit the OUT= option, then the output data set is named by using the DATA*n* convention. For more information, see the section "OUT= Output Data Set" on page 8956.

OUTJKCOEFS=*SAS-data-set*

names a SAS data set that contains the jackknife coefficients for VARMETHOD=JACKKNIFE.

keyword < *=name* >

specifies the quantities to include in the output data set and optionally names the new variables that contain the quantities. Specify a *keyword* for each desired quantity (see the following list of *keywords*), optionally followed by an equal sign and a variable name to contain the quantity. If you specify a *keyword* without a variable *name*, then PROC SURVEYIMPUTE uses default names. You can specify the following *keywords*:

DONORID< =*name* **>**
: requests a name for the identification variable for the donor units. If you do not specify this keyword, the donor IDs are not saved in the output data set. If you specify this keyword but do not specify a *name*, then the donor IDs are stored in a new variable named DonorID. This keyword is available only when METHOD=HOTDECK.

FRACTIONALWEIGHTS=*name*
: includes the fractional weights of donor cells in the output data set and specifies the corresponding variable *name*. If you do not specify this keyword, the fractional weights are not saved in the output data set. This keyword is available only when METHOD=FEFI.

IMPADJWEIGHTS=*name*
: includes the imputation-adjusted weights in the output data set and specifies the corresponding variable *name*. The imputation-adjusted weights are computed by multiplying the base weights by the fractional weights. If you do not specify this keyword, then the imputation-adjusted weights are stored in a new variable named ImpWt. This keyword is available only when METHOD=FEFI.

IMPSTATUS=*name*
: includes an imputation status index with the values shown in Table 110.3.

Table 110.3 Imputation Status Index

Index	**Imputation Status**
0	All items are observed
1	All missing items are imputed
2	No missing items are imputed
3	Some missing items are imputed but some missing items are not imputed
4	Invalid observation; observation is not used in the imputation

OBSID=*name*
: includes an index variable to contain the unique numeric identification of every unit from the input data set in the output data and specifies the corresponding variable *name*. If you do not specify this keyword, then the default unit ID is stored in a new variable named UnitID. The OBSID= option is not applicable when the ID statement is specified.

RECIPIENT=*name*
: includes the recipient index in the output data set and specifies the corresponding variable *name*. The recipient index can be 0 (which indicates a nonmissing unit) or any positive integer (which represents multiple donor units for a recipient unit). If you do not specify this keyword, then the recipient index is stored in a new variable named Recipient.

REPWEIGHTS Statement

REPWEIGHTS *variables* **;**

The REPWEIGHTS statement names variables that provide replicate weights.

If you use a REPWEIGHTS statement and you specify METHOD=HOTDECK in the PROC SURVEYIMPUTE statement, the procedure does not adjust the replicate weights.

If you use a REPWEIGHTS statement and you specify METHOD=FEFI in the PROC SURVEYIMPUTE statement, the procedure adjusts the replicate weights.

Each REPWEIGHTS variable should contain the weights for a single replicate, and the number of replicates should equal the number of REPWEIGHTS variables. The REPWEIGHTS variables must be numeric, and the variable values must be nonnegative numbers.

If you provide replicate weights in a REPWEIGHTS statement, you do not need to use a CLUSTER or STRATA statement.

If you use a REPWEIGHTS statement but do not use a WEIGHT statement, PROC SURVEYIMPUTE uses the average of each observation's replicate weights as the observation's weight.

STRATA Statement

STRATA *variables* ;

The STRATA statement names one or more *variables* that form the strata in a stratified sample design. The combinations of levels of STRATA variables define the strata in the sample, where strata are nonoverlapping subgroups that were sampled independently.

If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. For more information, see the section "Specifying the Sample Design" on page 8941.

If you provide replicate weights in a REPWEIGHTS statement, you do not need to use a STRATA statement.

The STRATA *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric, but PROC SURVEYIMPUTE treats them as categorical variables. The formatted values of the STRATA variables determine the STRATA variable levels. Thus, you can use formats to group values into levels. For more information, see the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in the *SAS Formats and Informats: Reference*.

VAR Statement

VAR *variables* ;

The VAR statement names the analysis variables to be imputed. The analysis variables can be either character or numeric. The categorical variables in the VAR statement, which can be either character or numeric, must also be specified in the CLASS statement. Only variables that you specify in the VAR statement will be imputed. If you specify METHOD=FEFI, then all numeric variables that you specify in the VAR statement must also be specified in the CLASS statement. METHOD=FEFI is not available for continuous variables in SAS/STAT 14.1.

A *variable* in the VAR statement should not appear in any of the BY, CLUSTER, REPWEIGHTS, STRATA, and WEIGHT statements.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement names the variable that contains the sampling weights. This variable must be numeric, and the sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then PROC SURVEYIMPUTE omits that observation from the analysis. For more information, see the section "Missing Values" on page 8942.

If you do not use a WEIGHT statement but you provide replicate weights in a REPWEIGHTS statement, PROC SURVEYIMPUTE uses the average of each observation's replicate weights as the observation's weight.

If you use neither a WEIGHT statement nor a REPWEIGHTS statement, PROC SURVEYIMPUTE assigns all observations a weight of 1.

Details: SURVEYIMPUTE Procedure

Specifying the Sample Design

PROC SURVEYIMPUTE produces replicate weights that are based on the sample design that is used to collect the survey data. You can use PROC SURVEYIMPUTE for single-stage or multistage designs, with or without stratification, and with or without unequal weighting. To create imputation-adjusted replicate weights for your survey data, you need to provide sample design information to PROC SURVEYIMPUTE. This information can include design (or variance) strata, clusters, and sampling weights. You provide sample design information by using the STRATA, CLUSTER, WEIGHT, and REPWEIGHTS statements.

If you use the REPWEIGHTS statement to provide replicate weights, you do not need to use a STRATA or CLUSTER statement. Otherwise, you should use STRATA and CLUSTER statements whenever your design includes stratification and clustering. If your design includes unequal sampling weights, you should use the WEIGHT statement.

For a multistage sample design, PROC SURVEYIMPUTE uses only the first stage of the sample design to create replicate weights. Therefore, the required input includes only the first-stage cluster (PSU) identification and first-stage stratum identification. You do not need to input design information about any additional stages of sampling.

Stratification

If your sample design is stratified at the first stage of sampling, use the STRATA statement to name the variables that form the strata. The combinations of categories of STRATA variables define the strata in the sample, where strata are nonoverlapping subgroups that were sampled independently. If your sample design has stratification at multiple stages, then identify only the first-stage strata in the STRATA statement.

If you use a REPWEIGHTS statement to provide replicate weights, you do not need to use a STRATA statement. Otherwise, you should use a STRATA statement whenever your design includes stratification. If you do not use a STRATA statement or a REPWEIGHTS statement, then PROC SURVEYIMPUTE assumes there is no stratification at the first stage; that is, the procedure assumes that all observation units are in the same stratum.

Clustering

If your sample design selects clusters at the first stage of sampling, use the CLUSTER statement to name the variables that identify the first-stage clusters, which are also called primary sampling units (PSUs). The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata. If your sample design has clustering at multiple stages, you should specify only the first-stage clusters (PSUs) in the CLUSTER statement. PROC SURVEYIMPUTE assumes that each cluster that is defined by the variables in the CLUSTER statement represents a PSU in the sample.

If you use a REPWEIGHTS statement to provide replicate weights, you do not need to use a CLUSTER statement. Otherwise, you should use a CLUSTER statement whenever your design includes clustering at the first stage of sampling. If you do not use a CLUSTER statement, then PROC SURVEYIMPUTE treats each observation as a PSU.

Weighting

If your sample design includes unequal weighting, use the WEIGHT statement to name the variable that contains the sampling weights. Sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then PROC SURVEYIMPUTE omits that observation from the analysis. For more information, see the section "Missing Values" on page 8942.

If you do not use a WEIGHT statement but you include a REPWEIGHTS statement, PROC SURVEYIMPUTE uses the average of each observation's replicate weights as the observation's weight. If you use neither a WEIGHT statement nor a REPWEIGHTS statement, PROC SURVEYIMPUTE assumes that all observations have a weight of 1.

Replicate Weights

If you have replicate weights available for your survey data, use the REPWEIGHTS statement to name the variables that contain the replicate weights. Replicate weights must be positive numbers. If an observation has a replicate weight that is nonpositive or missing, then PROC SURVEYIMPUTE does not perform any imputation. For more information, see the section "Missing Values" on page 8942.

Missing Values

You might have missing values in your data set for several reasons. Some common reasons are data entry error, ineligible items or units, and nonresponse. You should complete your data preparation (identify data entry error or ineligibility) and adjustment (fill in deterministic values or edits) before using the SURVEYIMPUTE procedure. Use PROC SURVEYIMPUTE to impute missing values that arise only from nonresponse. If you have observations that have missing values in variables that are not specified in the VAR statement, then the procedure does not impute those observations. The following subsections describe how PROC SURVEYIMPUTE treats missing values in the variables that are specified in some statements.

WEIGHT Statement Variable

If an observation has a missing value or a nonpositive value for the variable in the WEIGHT statement, then PROC SURVEYIMPUTE excludes that observation from the analysis. However, if you use the OUTPUT statement, the observation is included in the output data set.

REPWEIGHTS Statement Variables

If you provide replicate weights by using a REPWEIGHTS statement, the values for all variables in that statement must be nonmissing and nonnegative. PROC SURVEYIMPUTE does not perform the analysis when any replicate weight value is missing or nonpositive.

Variables in the CLUSTER and STRATA Statements

An observation is excluded from the analysis if it has a missing value for any variable in a CLUSTER or STRATA statement. However, if you use the OUTPUT statement, the observation is included in the output data set.

Variables in the CELLS Statement

An observation is excluded from the imputation if it has a missing value for any variable in the CELLS statement. However, if you specify the VARMETHOD=JK option in the PROC SURVEYIMPUTE statement, then the observation unit is used to create replicate weights, unless the observation unit has missing values in any of the variables in the STRATA, CLUSTER, or WEIGHT statement. If you use the OUTPUT statement, the observation is also included in the output data set.

Auxiliary Variables in the IMPJOINT Statement

Variables that you specify in the IMPJOINT statement but do not specify in the VAR statement are used as auxiliary variables in the imputation. If you have missing values in the auxiliary variables, then that observation unit is not used in the imputation. However, if you specify the VARMETHOD=JK option in the PROC SURVEYIMPUTE statement, then the observation unit is used to create replicate weights, unless the observation unit has missing values in any of the variables in the STRATA, CLUSTER, or WEIGHT statement. If you use the OUTPUT statement, the observation unit is also included in the output data set.

Variable in the ID Statement

If an observation unit has a missing value for the variable in the ID statement, then that observation is used in the imputation unless it also has missing values for variables in the STRATA, CLUSTER, or WEIGHT or CELLS statement. If the observation is selected as a donor unit for a recipient unit, then the donor identification for that recipient unit will also be missing.

Missing Data Patterns

The SURVEYIMPUTE procedure displays the missing data patterns in groups in a “Missing Data Patterns” table; the groups are based on whether the analysis variables are observed or missing. The input data set does not need to be sorted in any order.

For example, when a data set contains variables Z_1, Z_2, and Z_3 (in that order), up to eight groups of observations can be formed from the data set. Figure 110.7 displays the eight groups of observations and an unique missing pattern for each group.

Figure 110.7 Missing Data Patterns

Missing Data Patterns

Group	Z1	Z2	Z3
1	X	X	X
2	X	X	.
3	X	.	X
4	X	.	.
5	.	X	X
6	.	X	.
7	.	.	X
8	.	.	.

An "X" in Figure 110.7 means that the variable is observed, and a "." means that the variable is missing.

The order of the groups is determined by the order in which you list the variables in the VAR statement. If you specify a different order of variables in the VAR statement, then the results are different even if the other specifications remain the same.

Random Number Generation

The donor selection methods available in PROC SURVEYIMPUTE use random numbers in their selection algorithms. PROC SURVEYIMPUTE uses a uniform random number function to generate streams of pseudorandom numbers from an initial starting point, or seed. You can use the SEED= option to specify the initial seed. You can specify the same SEED= value (along with the same options and the same input data) to reproduce the imputation. If you do not specify the SEED= option, PROC SURVEYIMPUTE uses the time of day from the computer's clock to obtain the initial seed. For information about specifying the initial seed, see the SEED= option.

PROC SURVEYIMPUTE uses the Mersenne twister random number generator. The Mersenne twister generator (Matsumoto and Nishimura 1998) has a very long period ($2^{19937} - 1$) and good statistical properties. The algorithm is a twisted generalized feedback shift register. This is the same random number generator that PROC SURVEYSELECT uses by default and that the RAND function provides for the uniform distribution. For more information, see *SAS Functions and CALL Routines: Reference*.

Fully Efficient Fractional Imputation

The fully efficient fractional imputation (FEFI) method uses multiple donor units for a recipient unit. The observation unit that contains the missing values is known as the recipient unit, and the observation unit that provides the value for imputation is known as the donor unit. The number of donor units for a recipient unit is equal to the number of observed levels for the missing items, given the observed levels for the nonmissing items of the recipient unit. Each donor donates a fraction of the original weight of the recipient unit such that the sum of the fractional weights from all the donors is equal to the original weight of the recipient. The fraction of the recipient weight that a donor unit contributes to the recipient unit is known as the *fractional weight*. The method is called fully efficient because it does not introduce additional variability that is caused by the selection of donor units (Kim and Fuller 2004). One disadvantage of the FEFI method is that it can greatly increase the size of the imputed data set. For more information, see Kalton and Kish (1984), Fuller (2009, Section 5.2.2), and Kim and Shao (2014, Section 4.6).

FEFI Algorithm

Suppose you want to impute P items jointly (by using the IMPJOINT statement in PROC SURVEYIMPUTE). Let $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iP})$ be the true response for the P items for unit i. $\mathbf{Z}_i$ is completely known if all P items are observed for unit i. However, the true response might not be known for some units because of item nonresponse. Let Z_{ij} be categorical and have J levels for item j. Denote $\mathbf{Z}_{i,\texttt{obs}}$ as the observed part and $\mathbf{Z}_{i,\texttt{miss}}$ as the missing part of $\mathbf{Z}_i$. Let the population proportion that falls in category $(Z_{i1} = \kappa_1, Z_{i2} = \kappa_2, \cdots, Z_{iP} = \kappa_P)$ be $\pi(\kappa_1\kappa_2\cdots\kappa_P)$. Assume that it is possible to estimate the population proportion from the observed sample. That is, for example, the conditional probability, $P(Z_{i1} = \kappa_1, Z_{i2} = \kappa_2 | Z_{i3} = \kappa_3, \ldots, Z_{iP} = \kappa_p)$, in the observed data is the same as the conditional probability in the data where $(Z_{i1} = \kappa_1, Z_{i2} = \kappa_2)$ are missing. The conditional probabilities are estimated by

$$\hat{P}(Z_{i1} = \kappa_1, Z_{i2} = \kappa_2 | Z_{i3} = \kappa_3, \ldots, Z_{iP} = \kappa_p) = \left\{ \sum_{\kappa_1\kappa_2} \hat{\pi}(\kappa_1\kappa_2\cdots\kappa_P) \right\}^{-1} \hat{\pi}(\kappa_1\kappa_2\cdots\kappa_P)$$

where

$$\hat{\pi}(\kappa_1\kappa_2\cdots\kappa_P) = \left\{ \sum_i w_i \right\}^{-1} \sum_i w_i I(Z_{i1} = \kappa_1, \ldots, Z_{iP} = \kappa_P)$$

is the estimated joint probability, $I(.)$ is an indicator function, and w_i is the observation weight for unit i.

The FEFI method uses an EM-by-weighting algorithm similar to that of Ibrahim (1990). The detailed algorithm is described in Kim and Fuller (2013). The following steps describe the imputation technique. If you do not specify imputation cells by using the CELLS statement, PROC SURVEYIMPUTE uses the entire data set as one imputation cell. If you specify imputation cells, then all the probabilities in these steps are computed by using observations from the same imputation cell as the recipient unit. To simplify notation, subscripts are not used for imputation cells in the following description.

For given i, let $\mathbf{Z}_{i,\texttt{obs}}$ and $\mathbf{Z}_{i,\texttt{miss}}$ be the observed part and the missing part, respectively, of unit i. Let $\mathcal{A}_c$ be the index set for the complete respondents. Suppose you want to impute the missing part of $\mathbf{Z}_i$, $\mathbf{Z}_{i,\texttt{miss}}$. The index set $d_i = \{k : k \in \mathcal{A}_c \wedge \mathbf{Z}_{k,\texttt{obs}} = \mathbf{Z}_{i,\texttt{obs}}\}$ contains the indexes for the all possible donor units for $\mathbf{Z}_i$. Let $l = 1, 2, \ldots, M_l$ be all the observed combinations of $\{\mathbf{Z}_{k,\texttt{miss}} : k \in d_i\}$. The set of all observed combinations for unit i defines the donor cells (all possible realizations) for unit i. Let $\mathbf{Z}_{i,\texttt{miss}[l]}$ be the lth imputed value of $\mathbf{Z}_{i,\texttt{miss}}$. You must assume that at least one imputed value is available; otherwise the observation is not imputed.

1. *Initialization:* For each observation that has missing items, determine the number of donor cells by using the number of unique combinations of observed levels for the missing items for the responding units in the imputation cell. Compute the initial fractional weight from donor cell l to unit i, $w_{il(0)}$, by

$$w_{il(0)} = \left\{ \sum_{k=1}^{M_l} \tilde{\pi}_{(0)}(\mathbf{Z}_{i,\texttt{obs}}, \mathbf{Z}_{i,\texttt{miss}[k]}) \right\}^{-1} \tilde{\pi}_{(0)}(\mathbf{Z}_{i,\texttt{obs}}, \mathbf{Z}_{i,\texttt{miss}[l]})$$

where $l = 1, 2, \ldots, M_l$ is the number of donor cells, and

$$\tilde{\pi}_{(0)}(\kappa_1, \ldots, \kappa_P) = \left\{ \sum_{i \in \mathcal{A}_c} w_i \right\}^{-1} \sum_{i \in \mathcal{A}_c} w_i I(Z_{i1} = \kappa_1, \ldots, Z_{iP} = \kappa_P)$$

The sum of the fractional weights over all the donor cells is 1 for every observation unit; that is, $\sum_l w_{il(0)} = 1$, for all i. The lth imputed row for unit i is created by keeping the observed items unchanged, replacing the missing items with the observed levels from the lth donor cell, and computing the fractional weight by $w_i w_{il(0)}$. Only the complete respondents are used to compute the fractional weights in this step. If unit i has no missing items, then $w_{i1(0)} = 1$. The initial FEFI data set contains all the observed units, the imputed rows for observation that had missing items, and the corresponding fractional weights.

2. *M-step:* The tth M-step computes the joint probabilities by using the fractional weights from the (t–1)th E-step,

$$\tilde{\pi}_{(t)}(\kappa_1, \ldots, \kappa_P) = \left\{ \sum_i \sum_l w_i w_{il(t-1)} \right\}^{-1} \sum_i \sum_l w_i w_{il(t-1)} I(Z_{i1} = \kappa_1, \ldots, Z_{iP} = \kappa_P)$$

for all i, all l, and $t > 0$. Note that for $t > 0$, $\tilde{\pi}_{(t)}$ uses all observation units including observations that have missing items that are imputed in the initialization step.

3. *E-step:* The tth E-step computes the fractional weights by using the joint probabilities $\tilde{\pi}_{(t)}(\kappa_1, \ldots, \kappa_P)$ from the tth M-step. The tth fractional weight for unit i and donor cell l is given by

$$w_{il(t)} = \left\{ \sum_{k=1}^{M_l} \tilde{\pi}_{(t)}(\mathbf{Z}_{i,\mathtt{obs}}, \mathbf{Z}_{i,\mathtt{miss}[k]}) \right\}^{-1} \tilde{\pi}_{(t)}(\mathbf{Z}_{i,\mathtt{obs}}, \mathbf{Z}_{i,\mathtt{miss}[l]})$$

4. *Repetition:* The EM steps are repeated for $t = 1, 2, \ldots,$ until the changes in fractional weights over all observation units between two successive EM steps are negligible or the maximum number of EM repetitions is reached.

   The maximum absolute difference convergence criterion, ϵ_{AD}, at step t is defined as

$$\max_{i,l} |w_{il(t)} - w_{il(t-1)}| / w_{il} \le \epsilon_{\mathrm{AD}}$$

   The maximum absolute relative difference convergence criterion, ϵ_{RD}, at step t is defined as

$$\max_{i,l} |w_{il(t)} - w_{il(t-1)}| / w_{il(t-1)} \le \epsilon_{\mathrm{RD}}$$

   where $w_{il(t-1)} > 0$.

The replicate weights are created by computing a replicated version of $\tilde{\pi}_{(t)}(\kappa_1\kappa_2 \cdots \kappa_p)$, $\tilde{\pi}^{(k)}_{(t)}(\kappa_1\kappa_2 \cdots \kappa_p)$, and by repeating the EM-by-weighting algorithm as described earlier. For the kth replicate sample, $\tilde{\pi}^{(k)}_{(t)}(\kappa_1\kappa_2 \cdots \kappa_p)$ is computed by

$$\tilde{\pi}^{(k)}_{(t)}(\kappa_1, \ldots, \kappa_P) = \left\{ \sum_i \sum_l w_i^{(k)} w_{il(t-1)}^{(k)} \right\}^{-1} \sum_i \sum_l w_i^{(k)} w_{il(t-1)}^{(k)} I(Z_{i1} = \kappa_1, \ldots, Z_{iP} = \kappa_P)$$

Example of FEFI

The small data set shown in Figure 110.8 is used to illustrate the imputation technique. The data set contains nine observation units, and each unit has two items (X and Y). The variable Unit contains the observation identification. In this example, X is missing for units 5 and 9, and Y is missing for units 2 and 9.

Figure 110.8 Sample Data with Missing Items

Unit	X	Y
1	0	0
2	0	.
3	0	1
4	0	0
5	.	1
6	1	0
7	1	1
8	1	1
9	.	.

The following SAS statements request joint imputation of X and Y by using the FEFI method. These statements also request imputation-adjusted replicate weights for the jackknife replication method. The CLASS statement specifies that both X and Y are CLASS variables. The OUTPUT statement stores the imputed values in the data set Imputed and stores the jackknife coefficients in the data set Ojkc. The FRACTIONALWEIGHTS= option in the OUTPUT statement saves the fractional weights in the Imputed data set.

```
proc surveyimpute data=test varmethod=jackknife;
   class x y;
   var x y;
   id Unit;
   output out=Imputed fractionalweights=FracWgt outjkcoefs=Ojkc;
run;
```

The initial fractional weights, FracWgt, after the initialization step are displayed in Figure 110.9.

- Observation unit 1 has no missing value. Therefore, the Recipient value is 0, the FracWgt value is 1, and the values of X and Y are the same as the observed values for observation unit 1 in Figure 110.9. Because all observation units have a weight of 1, the fractional weights, FracWgt, and the imputation-adjusted weights, ImpWt, are the same for all rows.

- Observation unit 2 has a missing Y. The observed level for X for unit 2 is 0. For X = 0, two levels for Y are observed: Y = 0, which has a proportion (FracWgt) of 0.67, and Y = 1, which has a proportion of 0.33. Therefore, observation unit 2 receives two donor cells (Recipient = 1 and Recipient = 2), whose initial fractional weights are 0.67 and 0.33, respectively. Because X is observed, the X values in both rows for unit 2 are the same as the observed value. However, the first recipient row for unit 2 has an imputed Y value of 0, the second recipient row for unit 2 has an imputed Y value of 1, and each has a corresponding initial fractional weight.

- Observation unit 5 has a missing X. The observed level for Y for unit 5 is 1. To impute X, note that two levels of X are observed when Y = 1: X = 0 with a proportion of 0.33 and X = 1 with a proportion of

0.67. The two recipient rows for observation unit 5 contain the initial fractional weights in the FracWgt column and the imputed X values.

- Observation unit 9 has missing values for both X and Y. From the observed data, X and Y can take the following values: (X = 0, Y = 0) with probability 0.33, (X = 0, Y = 1) with probability 0.17, (X = 1, Y = 0) with probability 0.17, and (X = 1, Y = 1) with probability 0.33. The four imputed rows (Recipient 1, Recipient 2, Recipient 3, and Recipient 4) for observation unit 9 represent the four observed combinations for X and Y along with their initial fractional weights.

The resulting data set contains 14 rows. There are six rows for fully observed units (Recipient = 0), two rows for unit 2, two rows for unit 5, and four rows for unit 9. The sum of initial fractional weights is 1 for all units.

Figure 110.9 Fractional Imputation after Initialization

Unit	Recipient	ImpWt	FracWgt	X	Y
1	0	1.00000	1.00000	0	0
2	1	0.66667	0.66667	0	0
2	2	0.33333	0.33333	0	1
3	0	1.00000	1.00000	0	1
4	0	1.00000	1.00000	0	0
5	1	0.33333	0.33333	0	1
5	2	0.66667	0.66667	1	1
6	0	1.00000	1.00000	1	0
7	0	1.00000	1.00000	1	1
8	0	1.00000	1.00000	1	1
9	1	0.33333	0.33333	0	0
9	2	0.16667	0.16667	0	1
9	3	0.16667	0.16667	1	0
9	4	0.33333	0.33333	1	1

The EM algorithm repeats the computation of the joint probabilities and the fractional weights until convergence. The fractional weights, FracWgt, after the EM step and the imputation-adjusted replicate weights (ImpRepWt_1, . . . , ImpRepWt_9) are displayed in Figure 110.10.

Figure 110.10 Fractional Imputation after the EM

Unit	Recipient	ImpWt	FracWgt	X	Y	ImpRepWt_1	ImpRepWt_2	ImpRepWt_3	ImpRepWt_4
1	0	1.00000	1.00000	0	0	0.00000	1.12500	1.12500	1.12500
2	1	0.58601	0.58601	0	0	0.46072	0.00000	1.12498	0.46072
2	2	0.41399	0.41399	0	1	0.66428	0.00000	0.00002	0.66428
3	0	1.00000	1.00000	0	1	1.12500	1.12500	0.00000	1.12500
4	0	1.00000	1.00000	0	0	1.12500	1.12500	1.12500	0.00000
5	1	0.41399	0.41399	0	1	0.49821	0.37510	0.00002	0.49821
5	2	0.58601	0.58601	1	1	0.62679	0.74990	1.12498	0.62679
6	0	1.00000	1.00000	1	0	1.12500	1.12500	1.12500	1.12500
7	0	1.00000	1.00000	1	1	1.12500	1.12500	1.12500	1.12500
8	0	1.00000	1.00000	1	1	1.12500	1.12500	1.12500	1.12500
9	1	0.32330	0.32330	0	0	0.22659	0.32143	0.48214	0.22659
9	2	0.22840	0.22840	0	1	0.32669	0.21434	0.00001	0.32669
9	3	0.12500	0.12500	1	0	0.16071	0.16071	0.16071	0.16071
9	4	0.32330	0.32330	1	1	0.41101	0.42851	0.48214	0.41101

ImpRepWt_5	ImpRepWt_6	ImpRepWt_7	ImpRepWt_8	ImpRepWt_9
1.12500	1.12500	1.12500	1.12500	1.12500
0.75009	0.65906	0.62682	0.62682	0.65877
0.37491	0.46594	0.49818	0.49818	0.46623
1.12500	1.12500	1.12500	1.12500	1.12500
1.12500	1.12500	1.12500	1.12500	1.12500
0.00000	0.46601	0.66443	0.66443	0.46623
0.00000	0.65899	0.46057	0.46057	0.65877
1.12500	0.00000	1.12500	1.12500	1.12500
1.12500	1.12500	0.00000	1.12500	1.12500
1.12500	1.12500	1.12500	0.00000	1.12500
0.42862	0.41563	0.41109	0.41109	0.00000
0.21424	0.29384	0.32672	0.32672	0.00000
0.16071	0.00000	0.16071	0.16071	0.00000
0.32143	0.41553	0.22648	0.22648	0.00000

Hot-Deck Imputation

Imputation techniques that use observed values from the sample to impute (fill in) missing values are known as hot-deck imputation. For more information, see Fellegi and Holt (1976), Lohr (2010, Section 8.6.3), Andridge and Little (2010), Fuller (2009, Section 5.2.1), Särndal and Lundström (2005), and Bethlehem (2009, Section 8.3). The observation unit that contains the missing values is known as the recipient unit, and the observation unit that provides the value for imputation is known as the donor unit. It is common to group similar observation units in one imputation cell and then select the donor units from the same imputation cell as the recipient unit. This imputation technique is also known as hot-deck imputation within classes (Särndal, Swensson, and Wretman 1992, p. 593). If the donor unit is selected randomly for a recipient unit, then the imputation technique is called random hot-deck imputation.

PROC SURVEYIMPUTE implements cell-based random hot-deck imputation methods. You identify imputation cells by using the CELLS statement and specify a random selection method by using the

SELECTION= suboption for METHOD=HOTDECK in the PROC SURVEYIMPUTE statement. If an observation unit does not contain any missing values in the analysis variables, then the observation unit is considered as a donor unit. If an observation unit contains at least one missing value in the analysis variables, then the observation unit is treated as a recipient unit. You specify the analysis variables in the VAR statement. If no donors are found for an observation unit in the imputation cell, then the missing items are not imputed for that observation unit.

To illustrate the technique, consider the data set in Figure 110.11. Assume these six units are in the same imputation cell and you want to use the hot-deck imputation to impute the missing values in units 3, 4, and 6.

Figure 110.11 Units in an Imputation Cell

Unit	Age	Gender	Pregnancy
1	48	Male	0
2	22	Female	1
3	31		2
4	.	Male	0
5	22	Female	0
6	35		.

The following SAS statements use units 1, 2, and 5, which contain no missing values as the donor units for all recipient units 3, 4, and 6. For every recipient unit, PROC SURVEYIMPUTE selects a donor unit at random from all available donor units and replaces the missing values in the recipient unit with the observed values from the selected donor unit.

Both Age and Pregnancy are missing for unit 6, and PROC SURVEYIMPUTE uses the same donor to impute both items. Using the same donor unit to impute multiple items helps preserve the observed multivariate relationship.

However, it is possible to generate impossible responses. For example, if observation unit 1 is randomly selected as the donor unit for observation unit 3, then observation unit 3 will have Gender=Male but Pregnancy=2—a biological impossibility! To deal with such situations, consider filling the deterministic values before using imputation. For example, unit 3 is reported to be pregnant twice, and thus must be a female respondent. So you assign Gender=Female for unit 3 before using PROC SURVEYIMPUTE.

```
proc surveyimpute method=hotdeck(selection=srswr);
   var Age Gender Pregnancy;
   output out=JointHotDeck;
run;
```

If you do not want to preserve the multivariate relationship among the items, then you can impute the items marginally. The following SAS statements impute Age marginally. The recipient unit is 4, and the possible donor units are 1, 2, 3, 5, and 6.

```
proc surveyimpute method=hotdeck(selection=srswr);
   var Age;
   output out=MarginalHotDeck;
run;
```

The random selection of donors preserves the expectations within the imputation cells, but the random selection process increases the variance (Fuller 2009, p. 289). The variance estimator must include both the sampling variability and the imputation variability (Särndal and Lundström 2005).

PROC SURVEYIMPUTE implements the random selection methods that are described in the following subsections.

Approximate Bayesian Bootstrap

Suppose there are m recipient units and r donor units in an imputation cell. The approximate Bayesian bootstrap technique uses the following two steps for donor selection:

1. Select a sample of size r from the r donor units by using a simple random sample with replacement. The selected set is called the donor set for this imputation cell.
2. Select m donor units from the donor set by using a simple random sample with replacement.

To account for the imputation variance, you must select multiple donor units for every recipient unit. You can use the NDONORS= option in the PROC SURVEYIMPUTE statement to select multiple donor units. The procedure repeats the preceding two steps independently to select multiple donor units for every recipient unit. If you have a stratified design, Little and Rubin (2002, p. 89) suggest defining the imputation cells that are nested within strata. By default, the procedure does not assume that the cells are nested within the strata. You must specify the STRATA variables in the CELLS statement to define the imputation cells that are nested within the strata. For more information about the approximate Bayesian bootstrap method, see Rubin and Schenker (1986), Little and Rubin (2002, p. 89), and Kim (2002).

Simple Random Samples without Replacement

Suppose there are m recipient units and r donor units in an imputation cell. PROC SURVEYIMPUTE selects a simple random sample without replacement of size m from the r donors. One requirement for this selection method is that the number of donor units must be greater than or equal to the number of the recipient units. PROC SURVEYIMPUTE uses the selection-rejection method described in Tillé (2006, p. 48). If you select multiple (d) donor units for each recipient unit (by using the NDONORS= option in the PROC SURVEYIMPUTE statement), then the procedure selects d simple random samples independently.

Simple Random Samples with Replacement

Suppose there are m recipient units and r donor units in an imputation cell. PROC SURVEYIMPUTE selects a simple random sample with replacement of size m from the r donors. If you select multiple (d) donor units for each recipient unit (by using the NDONORS= option in the PROC SURVEYIMPUTE statement), then the procedure selects d simple random samples independently.

Weighted Selection

Suppose there are m recipient units and r donor units in an imputation cell. Let w_i be the weight of the donor unit i. PROC SURVEYIMPUTE selects a probability proportional to donor weight, w_i, with replacement sample of size m from the r donors. The procedure uses the probability proportional to size sampling algorithm described in Särndal, Swensson, and Wretman (1992, p. 97). For more information about the weighted hot-deck method, see Shao and Tu (1995, p. 271), and Rao and Shao (1992).

Replication Variance Estimation

Replication methods are useful for estimating variances that account for both the sampling variability and the imputation variability. If you specify the METHOD=FEFI option in the PROC SURVEYIMPUTE statement, then, by default, the procedure creates imputation-adjusted jackknife replicate weights unless you also specify the VARMETHOD=NONE option in the same statement. If you specify your own replicate weights by using the REPWEIGHTS statement and if you specify the METHOD=FEFI option in the PROC SURVEYIMPUTE statement, then the procedure creates new replicate weights by adjusting the replicate weights that you provide for imputation. It does not create imputation-adjusted replicate weights when you specify the METHOD=HOTDECK option in the PROC SURVEYIMPUTE statement.

The SURVEYIMPUTE procedure does not compute any variances. The replicate weights that are created can be used in any SAS/STAT survey procedure for variance computation. For an example, see the section “Getting Started: SURVEYIMPUTE Procedure” on page 8925.

Replication methods draw multiple replicates (also called subsamples) from a full sample according to a specific resampling scheme. The most commonly used resampling schemes are the balanced repeated replication (BRR) method and the jackknife method. For each replicate, the original weights are modified for the primary sampling units (PSUs) in the replicates to create replicate weights. The parameters of interest are estimated by using the replicate weights for each replicate. These estimates are also known as replicate estimates. Then the variances of parameters of interest are estimated by estimating variability among the replicate estimates. The SURVEYIMPUTE procedure automatically creates replicate weights based on the replication method that you specify; alternatively you can use the REPWEIGHTS statement to provide your own replicate weights.

The following subsections provide details about how the replication weights are created for each variance estimation method.

Balanced Repeated Replication (BRR) Method

The balanced repeated replication (BRR) method requires that the full sample be drawn by using a stratified sample design with two primary sampling units (PSUs) per stratum. The BRR method constructs half-sample replicates by deleting one PSU per stratum according to a Hadamard matrix and doubling the original weight of the other PSU in that stratum. If you use the FEFI method, then the unadjusted BRR weights are adjusted for the imputation to create the *imputation-adjusted replicate weights*. The sections “Unadjusted BRR Replicate Weights” on page 8952 and “Unadjusted Fay’s BRR Replicate Weights” on page 8953 describe how the unadjusted replicate weights are created, and the section “Imputation-Adjusted Replicate Weights” on page 8955 describes how the imputation-adjusted replicate weights are created.

Unadjusted BRR Replicate Weights

Let H be the total number of strata. The total number of replicates, R, is the smallest multiple of 4 that is greater than H. However, if you prefer a larger number of replicates, you can specify the REPS=n *method-option*. If an $n \times n$ Hadamard matrix cannot be constructed, the number of replicates is increased until a Hadamard matrix becomes available.

Each replicate is obtained by deleting one PSU per stratum according to a corresponding Hadamard matrix and adjusting the original weights for the remaining PSUs. The new weights are called replicate weights.

Replicates are constructed by using the first H columns of the $R \times R$ Hadamard matrix. The rth ($r = 1, 2, \ldots, R$) replicate is drawn from the full sample according to the rth row of the Hadamard matrix as follows:

- If the (r, h) element of the Hadamard matrix is 1, then the first PSU of stratum h is included in the rth replicate and the second PSU of stratum h is excluded.
- If the (r, h) element of the Hadamard matrix is –1, then the second PSU of stratum h is included in the rth replicate and the first PSU of stratum h is excluded.

The replicate weights of the remaining PSUs in each half sample are then doubled to their original weights. For more information about the BRR method, see Wolter (2007) and Lohr (2010).

By default, PROC SURVEYIMPUTE generates an appropriate Hadamard matrix automatically to create the replicates. You can display the Hadamard matrix by specifying the VARMETHOD=BRR(PRINTH) *method-option*. If you provide a Hadamard matrix by specifying the VARMETHOD=BRR(HADAMARD=) *method-option*, then the replicates are generated according to the provided Hadamard matrix.

For more information about how the BRR variance estimators are computed for related statistics, see the section "Balanced Repeated Replication (BRR) Method" in each of the following chapters: Chapter 109, "The SURVEYFREQ Procedure," Chapter 111, "The SURVEYLOGISTIC Procedure," Chapter 112, "The SURVEYMEANS Procedure," Chapter 113, "The SURVEYPHREG Procedure," and Chapter 114, "The SURVEYREG Procedure."

Unadjusted Fay's BRR Replicate Weights

The traditional BRR method constructs half-sample replicates by deleting one PSU per stratum according to a Hadamard matrix and doubling the original weight of the other PSU. Fay's BRR method uses the Fay coefficient, ϵ ($0 \leq \epsilon < 1$), and instead of deleting one PSU per stratum, it multiplies the original weight by the coefficient ϵ. The original weight of the remaining PSU in that stratum is multiplied by $2 - \epsilon$. PROC SURVEYIMPUTE uses $\epsilon = 0.5$ as the default value; alternatively, you can specify a value for ϵ by using the FAY= *method-option*. When $\epsilon = 0$, Fay's method becomes the traditional BRR method. For more information, see Dippo, Fay, and Morganstein (1984); Fay (1984, 1989); Judkins (1990). Because the traditional BRR method uses only half of the total sample in every replicate, some observed levels of the analysis variables might not be available in the replicate samples. Fay's BRR method is especially useful in this situation because it uses all the sampled units in every replicate.

For more information about how Fay's BRR variance estimators are computed for related statistics, see the section "Balanced Repeated Replication (BRR) Method" in each of the following chapters: Chapter 109, "The SURVEYFREQ Procedure," Chapter 111, "The SURVEYLOGISTIC Procedure," Chapter 112, "The SURVEYMEANS Procedure," Chapter 113, "The SURVEYPHREG Procedure," and Chapter 114, "The SURVEYREG Procedure."

Hadamard Matrix

PROC SURVEYIMPUTE uses a Hadamard matrix to construct replicates for BRR variance estimation. You can provide a Hadamard matrix for replicate construction by using the HADAMARD= *method-option* for VARMETHOD=BRR. Otherwise, PROC SURVEYIMPUTE generates an appropriate Hadamard matrix. You can display the Hadamard matrix by specifying the PRINTH *method-option*.

A Hadamard matrix $\mathbf{A}$ of dimension R is a square matrix that has all elements equal to 1 or –1 such that $\mathbf{A}'\mathbf{A} = R\mathbf{I}$, where $\mathbf{I}$ is an identity matrix of appropriate order. The dimension of a Hadamard matrix must equal 1, 2, or a multiple of 4.

For example, the following matrix is a Hadamard matrix of dimension $k = 8$:

$$\begin{array}{rrrrrrrr}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{array}$$

For BRR replicate construction, the dimension of the Hadamard matrix must be at least H, where H denotes the number of first-stage strata in your design. If a Hadamard matrix of a particular dimension exists, it is not necessarily unique. Therefore, if you want to use a specific Hadamard matrix, you must provide the matrix as a SAS data set in the HADAMARD= *method-option*. You must ensure that the matrix that you provide is actually a Hadamard matrix; PROC SURVEYIMPUTE does not check the validity of your Hadamard matrix.

For more information about how the Hadamard matrix is used to construct replicates for BRR variance estimation, see the section "Unadjusted BRR Replicate Weights" on page 8952.

Jackknife Method

The jackknife method of variance estimation deletes one PSU at a time from the full sample to create replicates. This method is also known as the delete-1 jackknife method because it deletes exactly one PSU in every replicate. The total number of replicates R is the same as the total number of PSUs. In each replicate, the sampling weights of the remaining PSUs are modified by the jackknife coefficient α_r. The modified weights are called *replicate weights*. If you use the FEFI method, then the unadjusted replicate weights are adjusted for the imputation to create the *imputation-adjusted replicate weights*. The section "Unadjusted Jackknife Replicate Weights" on page 8954 describes how the unadjusted replicate weights are created, and the section "Imputation-Adjusted Replicate Weights" on page 8955 describes how the imputation-adjusted replicate weights are created.

Unadjusted Jackknife Replicate Weights

Let PSU i in stratum h_r be omitted for the rth replicate. Then the jackknife coefficient, α_r, and replicate weights, $w_{hij}^{(r)}$, are computed as

$$\alpha_r = \begin{cases} \frac{n_{h_r}-1}{n_{h_r}} & \text{for a stratified design} \\ \frac{R-1}{R} & \text{for designs without stratification} \end{cases}$$

$$w_{hij}^{(r)} = \begin{cases} w_{hij} & \text{if observation unit } j \text{ is not in donor stratum } h_r \\ 0 & \text{if observation unit } j \text{ is in PSU } i \text{ of donor stratum } h_r \\ w_{hij}/\alpha_r & \text{if observation unit } j \text{ is not in PSU } i \text{ but is in donor stratum } h_r \end{cases}$$

If you use the hot-deck imputation method, then you can use the OUTPUT statement in PROC SURVEY-IMPUTE to store the unadjusted replicate weights. The unadjusted replicate weights are not saved for the FEFI method. You should use the imputation-adjusted replicate weights for variance estimation from a fractionally imputed data set. Use the OUTJKCOEFS= option in the OUTPUT statement to store the jackknife coefficients in a SAS data set.

For more information about how the jackknife variance estimators are computed for related statistics, see the section "Jackknife Method" in each of the following chapters: Chapter 109, "The SURVEYFREQ Procedure," Chapter 111, "The SURVEYLOGISTIC Procedure," Chapter 112, "The SURVEYMEANS Procedure," Chapter 113, "The SURVEYPHREG Procedure," and Chapter 114, "The SURVEYREG Procedure."

Imputation-Adjusted Replicate Weights

If you use the hot-deck imputation technique by specifying the METHOD=HOTDECKoption in the PROC SURVEYIMPUTE statement, the procedure does not create imputation-adjusted replicate weights. Naive variance estimators that do not use imputation-adjusted replicate weights and assume the imputed data as the observed data might underestimate the true variance. For more information, see Haziza (2009); Särndal and Lundström (2005); Rao and Shao (1992).

If you use the FEFI method by specifying the METHOD=FEFI option in the PROC SURVEYIMPUTE statement, the procedure adjusts the replicate weights for imputation. The imputation-adjusted replicate weights should be used with other SAS/STAT survey procedures to estimate the variance of an estimator that uses the imputed data. For more information, see Fuller (2009, Section 5.2.2) and Kim and Shao (2014, Section 4.6).

Let $w_i^{(r)}$ be the unadjusted replicate weight for observation unit i. To facilitate discussion, separate subscripts for strata, clusters, and imputation cells are omitted. The unadjusted replicate weights can come from a jackknife method as described in the section "Unadjusted Jackknife Replicate Weights" on page 8954 or from a BRR method as described in the section "Unadjusted BRR Replicate Weights" on page 8952, or they can be specified by using the REPWEIGHTS statement. The adjustment follows the similar EM-by-weighting algorithm that is described in the section "Fully Efficient Fractional Imputation" on page 8944 but uses the replicate weights, $w_i^{(r)}$, instead of the full sample weight, w_i.

In particular, the joint probabilities for the tth M-step and the rth replicate weight are computed by

$$\tilde{\pi}_{(t)}^{(r)}(\kappa_1, \ldots, \kappa_P) = \left\{ \sum_i \sum_l w_i^{(r)} w_{il(t-1)}^{(r)} \right\}^{-1} \sum_i \sum_l w_i^{(r)} w_{il(t-1)}^{(r)} I(Z_{i1} = \kappa_1, \ldots, Z_{iP} = \kappa_P)$$

for all i, l, and $t > 0$.

The rth replicate fractional weights for the tth E-step is computed by

$$w_{il(t)}^{(r)} = \left\{ \sum_{k=1}^{M_l} \tilde{\pi}_{(t)}^{(r)}(\mathbf{Z}_{i,\mathtt{obs}}, \mathbf{Z}_{i,\mathtt{miss}[k]}) \right\}^{-1} \tilde{\pi}_{(t)}^{(r)}(\mathbf{Z}_{i,\mathtt{obs}}, \mathbf{Z}_{i,\mathtt{miss}[l]})$$

where M_l is the number of donor cells.

Output Data Sets

PROC SURVEYIMPUTE creates an output data set to store the imputed data and the replicate weights, and an output data set to store the jackknife coefficients for jackknife variance estimation. You can use the Output Delivery System (ODS) to create a SAS data set from any piece of PROC SURVEYIMPUTE output. For more information, see the section "Displayed Output" on page 8956.

OUT= Output Data Set

You can use the OUTPUT statement to create a data set to store the imputed data. The OUTPUT OUT= data set contains all the variables from the input data set, imputed values for missing values for the variables in the VAR statement, and some observation-level quantities. These quantities can include the fractionally adjusted weights, replicate weights, recipient numbers, and donor identifications.

Jackknife Coefficients Output Data Set

If you specify the OUTJKCOEFS= option in the OUTPUT statement, PROC SURVEYIMPUTE stores the jackknife coefficients in an output data set. The OUTJKCOEFS= output data set contains one observation for each replicate. The OUTJKCOEFS= data set contains the following variables:

- Replicate: the replicate number for the jackknife coefficient
- JKCoefficient: the jackknife coefficient for the replicate
- DonorStratum: the stratum of the PSU that was deleted to construct the replicate, if you use a STRATA statement

You can use the JKCOEFS= option in the REPWEIGHTS statement in any SAS/STAT survey procedure to provide jackknife coefficients for that procedure. If the jackknife coefficients are different from $(R–1)/R$, where R is the total number of replicates, then you must provide the jackknife coefficients to correctly estimate the variance.

Displayed Output

If you use the NOPRINT option in the PROC SURVEYIMPUTE statement, the procedure does not display any output. Otherwise, PROC SURVEYIMPUTE displays results of the imputation in a collection of tables, which are described in the following subsections.

Class Level Information

If you use a CLASS statement, PROC SURVEYIMPUTE displays a “Class Level Information” table, which lists the categories of every CLASS variable that is used in the imputation. The ODS name of the “Class Level Information” table is ClassLevelInfo.

Convergence Status

If you specify METHOD=FEFI, PROC SURVEYIMPUTE displays a “Convergence Status” table, which shows the convergence status of the EM optimization routine. If the optimization routine converges, then the Status is set to 0; otherwise, the Status is set to 1. The ODS name of the “Convergence Status” table is ConvergenceStatus.

Design Summary

If you use a STRATA or CLUSTER statement, PROC SURVEYIMPUTE displays a “Design Summary” table, which provides information about the sample design. The table displays the total number of strata that are read and used, and the total number of clusters that are read and used. The ODS name of the "Design Summary" table is DesignSummary.

Hadamard Matrix

If you specify the PRINTH *method-option* for VARMETHOD=BRR, PROC SURVEYIMPUTE displays the Hadamard matrix that is used to construct replicates for BRR variance estimation. If you provide a Hadamard matrix by using the HADAMARD= *method-option* for VARMETHOD=BRR but the procedure does not use the entire matrix, the procedure displays only the rows and columns that are actually used to construct replicates. The ODS name of the "Hadamard Matrix" table is Hadamard.

Imputation Information

By default, PROC SURVEYIMPUTE displays an "Imputation Information" table, which provides information about the imputation method. The table displays the two-level name of the input data set, the name and label of the WEIGHT variable, the name and label of each STRATA variable, the name and label of each CLUSTER variable, the name of the imputation method used, and the random number seed. The ODS name of the "Imputation Information" table is ImputationInfo.

Imputation Summary

By default, PROC SURVEYIMPUTE displays an "Imputation Summary" table, which provides summary information about the imputation. The table displays the number of observations and the sum of weights for the following:

- Nonmissing observations – all variables specified in the VAR statement have nonmissing values
- Missing – at least one variable specified in the VAR statement has a missing value
- Missing, Imputed – all missing values have been imputed
- Missing, Not Imputed – no missing values are imputed
- Missing, Partially Imputed – missing values in some variables are imputed, but missing values in some other variables are not imputed

The ODS name of the "Imputation Summary" table is ImputationSummary.

Iteration History

If you specify METHOD=FEFI, PROC SURVEYIMPUTE displays an "Iteration History" table, which provides information about the iteration history for the EM algorithm. The table displays iteration numbers, maximum absolute differences, and maximum relative absolute differences for the fractional weights over all the observations. The ODS name of the "Iteration History" table is IterationHistory.

Missing Data Patterns

By default, PROC SURVEYIMPUTE displays a "Missing Data Patterns" table, which provides information about the missing data patterns. The table displays the missing data pattern groups, "X" if the variable is observed in the group, and "." if the variable is missing in that group. In addition, it displays observation frequencies, the sum of weights, unweighted percentages, and weighted percentages for each group. The ODS name of the “Missing Data Patterns” table is MissPattern.

Number of Observations

By default, PROC SURVEYIMPUTE displays a “Number of Observations” table, which shows the number of observations that are read and used, and the sum of weights that are read and used in the imputation. The ODS name of the “Number of Observations” table is NObs.

ODS Table Names

PROC SURVEYIMPUTE assigns a name to each table that it creates. You can use this name to refer to the table when using the Output Delivery System (ODS) to select tables and create output data sets. For more information about ODS, see Chapter 20, “Using the Output Delivery System.” Table 110.4 lists the table names.

Table 110.4 ODS Tables Produced by PROC SURVEYIMPUTE

ODS Table Name	Description	Statement	Option
ClassLevelInfo	CLASS variable levels	CLASS	
ConvergenceStatus	Convergence status	PROC	METHOD=FEFI
DesignSummary	Design summary	STRATA or CLUSTER	
HadamardMatrix	Hadamard matrix	PROC	VARMETHOD=BRR (PRINTH)
ImputationInfo	Imputation information	PROC	
ImputationSummary	Imputation summary	PROC	
IterationHistory	Iteration history	PROC	METHOD=FEFI
MissPattern	Missing data patterns	PROC	
NObs	Number of observations	PROC	

Examples: SURVEYIMPUTE Procedure

Example 110.1: Approximate Bayesian Bootstrap Imputation

This example illustrates the approximate Bayesian bootstrap hot-deck imputation method by using a simulated data set from a fictitious survey of drug abusers. A stratified clustered sample of drug abuse treatment centers is taken from a list of available treatment centers. The list is first stratified based on geographic locations. From each strata, two or three treatment centers are sampled as the primary sampling units (PSU). Data are collected from individual patients within the selected treatment centers. The survey collects information about the substances that the patients used (such as drugs, alcohol, and marijuana) along with insurance information and treatment information.

The data set contains 736 observation units in 35 PSUs and 10 strata. The sum of the weights is 19,600. Therefore, the survey data represent a population of 19,600 patients from the study area. Some participants did not respond to all questions. The data set contains missing values in many variables.

To impute the missing items, you first need to decide whether to impute within imputation cells. Imputation cells divide the data into groups of similar units such that the recipient units share similar characteristics with the donor units in the same group. For example, it is reasonable to believe that different age groups, races, and income categories might have different responses to the drug abuse survey. You can use these characteristics to create imputation cells. Characteristics for imputation cells might come from the same survey or might come from other sources such as census data or previous surveys. In this example, assume that the imputation cells are available as a variable called ImputationCell in the data set.

The data set DrugAbuse contains the following items:

- Strata: stratum identification
- PSU: PSU identification (treatment centers)
- ObsWeight: observation weight for patients
- ImputationCell: imputation cell identification
- Age: age, in years
- Sex: 1 for female and 2 for male
- Race: 1 for white, 2 for black, and 3 for others
- Insurance: 1 if the patient has any insurance, and 2 otherwise
- Drug: 1 if the patient used any drugs in the past three months, and 2 otherwise
- Alcohol: 1 if the patient consumed any alcohol in the past month, and 2 otherwise
- Treatment: 1 if the patient is being treated for the first time, and 2 otherwise

```
data DrugAbuse;
   input Strata PSU ObsWeight ImputationCell Age Sex Race Insurance
         Drug Alcohol Treatment;
   datalines;
 1    1     5        1      74  1   1      3      2     2     1
 1    1     5        1      20  0   3      1      2     2     1
 1    1     5        3      42  1   2      1      1     2     1
 1    1     5        3      65  1   3      2      1     2     1
 1    1     5        2      53  1   1      1      1     1     1
 1    1     5        3      49  1   1      1      1     2     1
 1    1     5        3      51  0   2      1      2     1     1
 1    1     5        2      77  0   3      1      1     1     1
 1    1     5        2      26  1   1      1      1     2     1
 1    1     5        3      28  0   3      1      1     1     1
 1    1     5        1      71  1   1      1      2     2     1
 1    1     5        2      72  1   1      3      2     2     1
 1    1     5        3      24  1   1      1      1     1     1
 1    1     5        2      65  1   1      2      1     1     2
```

```
 1    1    5         3       47  1    1      1       1      1        1
 1    1    5         2       37  1    1      2       1      2        1
 1    1    5         2       46  1    1      3       1      1        1
 1    1    5         2       52  1    1      1       1      2        2
 1    1    5         3       60  0    3      1       1      2        1
 1    1    5         1       31  0    1      1       1      2        1
 1    2    5         1       23  0    3      3       1      1        1
 1    2    5         2       78  1    1      .       .      .        .
 1    2    5         2       29  1    1      1       1      1        1
 1    2    5         2       21  .    .      .       .      .        .

   ... more lines ...

10    4  55.5556      1       40  0    3      2       1      1        2
10    4  55.5556      1       32  1    3      1       2      2        1
10    4  55.5556      3       68  0    1      2       2      1        2
10    4  55.5556      3       35  1    1      2       1      2        2
;
```

The following statements request that the missing items be imputed by using the approximate Bayesian bootstrap hot-deck imputation method:

```
proc surveyimpute data=DrugAbuse method=hotdeck(selection=abb)
                  ndonors=5 seed=773269;
   var Sex Race Insurance Drug Alcohol Treatment;
   cells ImputationCell;
   output out=DrugAbuseABB;
run;
```

The PROC SURVEYIMPUTE statement invokes the procedure, the DATA= option specifies the input data set DrugAbuse, the METHOD= option requests the hot-deck imputation method, the METHOD=HOTDECK(SELECTION=ABB) option requests the approximate Bayesian bootstrap method, the NDONORS= option requests five donor units for every recipient unit, and the SEED= option specifies the random number generator seed. The VAR statement specifies the variables that are to be imputed, the CELLS statement identifies the imputation cell variable ImputationCell, and the OUT= option in the OUTPUT statement names the output data set DrugAbuseABB.

You do not need to use WEIGHTS, STRATA, and CLUSTER statements for the approximate Bayesian bootstrap method unless you want to create the jackknife replication weights by including the VARMETHOD=JACKKNIFE option in the PROC SURVEYIMPUTE statement. The selection of donors does not use the design information. However, if you want to select donors from the same strata or the same group of clusters, then you must include that information in the imputation cell.

Summary information about the imputation method, number of observations, and missing data patterns is shown in Output 110.1.1. The “Imputation Information” table summarizes the imputation method. The “Number of Observations” table shows that PROC SURVEYIMPUTE read and used all 736 observations. The “Missing Data Pattern” table displays the missing patterns in the data set. There are four different missing data pattern groups: all items observed, one item missing, four items missing, and all items missing. Of the observation units, 92.53% have all items observed; 4.64% have missing values in Treatment; 1.77% have missing values in Insurance, Drug, Alcohol, and Treatment; and 1.09% have missing values in all variables. Because the WEIGHT statement is not specified, these percentages represent the percentages of missing units in the input data.

Output 110.1.1 Imputation Summary

The SURVEYIMPUTE Procedure

Imputation Information	
Data Set	WORK.DRUGABUSE
Imputation Method	HOTDECK
Selection Method	ABB
Random Number Seed	773269

Number of Observations Read	736
Number of Observations Used	736

Missing Data Patterns

Group	Sex	Race	Insurance	Drug	Alcohol	Treatment	Freq	Sum of Weights	Unweighted Percent	Weighted Percent
1	X	X	X	X	X	X	681	681	92.53	92.53
2	X	X	X	X	X	.	34	34	4.62	4.62
3	X	X	.	.	.	.	13	13	1.77	1.77
4	.	.	.	.	.	.	8	8	1.09	1.09

Missing Data Patterns

	Group Means					
Group	Sex	Race	Insurance	Drug	Alcohol	Treatment
1	0.566814	1.491924	1.718062	1.292217	1.716593	1.201175
2	0.500000	1.441176	1.588235	1.294118	1.588235	.
3	0.692308	1.230769	.	.	.	.
4	.	.	.	.	.	.

Imputation Summary

Observation Status	Number of Observations	Sum of Weights
Nonmissing	681	681
Missing	55	55
Missing, Imputed	55	55
Missing, Not Imputed	0	0

Some selected observations from the output data set are displayed in Output 110.1.2. The output data set DrugAbuseABB contains the unit identification, the recipient index, and all the variables from the input data set DrugAbuse. Units that are complete respondents have one row, but units that are incomplete respondents have five rows in the output data set. For example, unit 21 is a complete respondent, so it has only one row in the output data set and its Recipient value is 0. Unit 22 is an incomplete respondent, so it has five rows in the output data set and its Recipient values range from 1 to 5.

Output 110.1.2 Observations for Some Selected Units

UnitID	Recipient	Strata	PSU	ObsWeight	ImputationCell	Age	Sex	Race	Insurance	Drug	Alcohol	Treatment
20	0	1	1	5	1	31	0	1	1	1	2	1
21	0	1	2	5	1	23	0	3	3	1	1	1
22	1	1	2	5	2	78	1	1	1	1	1	1
22	2	1	2	5	2	78	1	1	1	1	1	1
22	3	1	2	5	2	78	1	1	2	1	2	1
22	4	1	2	5	2	78	1	1	3	1	2	1
22	5	1	2	5	2	78	1	1	1	1	2	2
23	0	1	2	5	2	29	1	1	1	1	1	1
24	1	1	2	5	2	21	1	2	2	1	2	2
24	2	1	2	5	2	21	1	3	1	1	1	1
24	3	1	2	5	2	21	1	2	2	1	2	1
24	4	1	2	5	2	21	1	1	2	1	2	1
24	5	1	2	5	2	21	1	2	2	2	1	1
25	0	1	2	5	3	85	0	1	3	1	2	1

Suppose you want to perform a logistic regression analysis by using the imputed data set. If you want to use the multiple imputation variance estimator that is available in the MIANALYZE procedure with the imputed data set, then you need to create one complete data set for every imputation. The following SAS statements create five complete data sets and then merge the five data sets into one. Each complete data set contains the complete respondents and only one donor unit for the incomplete respondents. Each data set also contains the imputation number (_Imputation_).

```
data DAIMP;
   set DrugAbuseABB;
   if (Recipient = 0) then do;  /* Include complete respondents */
      do _Imputation_=1 to 5;   /* in all imputations.           */
         output;
      end;
   end;
   else do;                     /* Put incomplete respondents   */
      _Imputation_ = Recipient; /* in separate imputations.     */
      output;
   end;
proc sort data=DAIMP;
   by _Imputation_ UnitID;
run;
```

The following SAS statements first use the SURVEYLOGISTIC procedure (see Chapter 111, "The SURVEYLOGISTIC Procedure") to perform separate logistic regression analyses within the imputed data sets and use the MIANALYZE procedure (Chapter 76, "The MIANALYZE Procedure") to combine the logistic regression results from five imputed data sets:

```
ods select none;
proc surveylogistic data=DAIMP;
   by _imputation_;
   class Treatment Insurance Sex Race;
   strata Strata;
   cluster PSU;
   weight ObsWeight;
   model Drug=Treatment Insurance Age Sex Race / covb;
   ods output parameterestimates=Estimates covb=Covariances;
run;
ods select all;

proc mianalyze parms(classvar=classval)=Estimates
               covb(effectvar=stacking)=Covariances
               edf=25;
   class Treatment Insurance Sex Race;
   modeleffects Intercept Treatment Insurance Age Sex Race;
   ods output parameterestimates=ABBLogisticAnalysis;
run;
```

Although the survey design information was not directly used in the imputation, you must use the complete design information, including strata, clusters, and weights, to estimate the design variance within each imputed data set. The STRATA, CLUSTER, and WEIGHT statements in PROC SURVEYLOGISTIC specify the design information. However, separate logistic regression results from any single imputed data set should not be used for inference.

Degrees of freedom values for survey data are often much less than the number of observation units. In this example, there are 736 observation units, but there are 35 PSUs in 10 strata. The degrees of freedom for the Taylor series linearized variance estimator is 25 (35 – 10). You should specify the reduced degrees of freedom by using the EDF= option in PROC MIANALYZE. For more information, see the section "EDF=*number*" on page 5997 in Chapter 76, "The MIANALYZE Procedure"; also see Barnard and Rubin (1999).

The estimated regression parameters and their standard errors from a multiply imputed data set are shown in Output 110.1.3.

Output 110.1.3 Logistic Regression Analysis Using a Multiply Imputed Data Set

The MIANALYZE Procedure

Parameter Estimates (5 Imputations)

Parameter	Treatment	Insurance	Sex	Race	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum
Intercept					0.527080	0.232094	0.04656	1.007603	22.658	0.492600	0.574832
Treatment	1				0.094937	0.153297	-0.22225	0.412121	22.916	0.078459	0.114233
Insurance		1			-0.128475	0.144781	-0.42822	0.171268	22.671	-0.157661	-0.108998
Insurance		2			-0.038353	0.119353	-0.28536	0.208658	22.816	-0.059264	-0.024690
Age					0.001926	0.004635	-0.00767	0.011524	22.577	0.000858	0.002539
Sex			0		-0.088823	0.088702	-0.27265	0.095000	22.279	-0.101860	-0.063631
Race				1	0.436564	0.156414	0.11307	0.760060	23.092	0.418673	0.443484
Race				2	-0.111091	0.195368	-0.51509	0.292904	23.16	-0.118899	-0.096429

Parameter Estimates (5 Imputations)

Parameter	Treatment	Insurance	Sex	Race	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept					0	2.27	0.0330
Treatment	1				0	0.62	0.5418
Insurance		1			0	-0.89	0.3842
Insurance		2			0	-0.32	0.7509
Age					0	0.42	0.6817
Sex			0		0	-1.00	0.3274
Race				1	0	2.79	0.0104
Race				2	0	-0.57	0.5751

Example 110.2: Fully Efficient Fractional Imputation

This example illustrates the fully efficient fractional imputation (FEFI) method by using the data set DrugAbuse from a fictitious survey of drug abusers from Example 110.1. The survey collects information about substance that are used (such as drugs, alcohol, and marijuana) along with insurance information and treatment information. Some participants did not respond to all questions. The data set contains 736 observation units in 35 PSUs and 10 strata. The sum of the weights is 19,600. The data set contains missing values for many variables.

As in Example 110.1, to impute the missing items, you first need to decide whether to impute within imputation cells. Imputation cells divide the data into groups of similar units such that the recipient units share similar characteristics with the donor units in the same group. For example, it is reasonable to believe that different age groups, races, and income categories might have different responses to the drug abuse survey. You can use these characteristics to create imputation cells. Characteristics for imputation cells might come from the same survey or from other sources such as census data or previous surveys. In this example, assume that the imputation cells are available as a variable called ImputationCell in the data set.

The following statements request that the missing items be imputed by using the FEFI method:

```
proc surveyimpute data=DrugAbuse method=FEFI varmethod=Jackknife;
   class Sex Race Insurance Drug Alcohol Treatment;
   var   Sex Race Insurance Drug Alcohol Treatment;
   cells ImputationCell;
   strata Strata;
```

```
      cluster PSU;
      weight ObsWeight;
      output out=DrugAbuseFEFI outjkcoefs=DrugAbuseJKCOEFS;
   run;
```

The PROC SURVEYIMPUTE statement invokes the procedure, the DATA= option specifies the input data set DrugAbuse, the METHOD= option requests the FEFI method, and the VARMETHOD= option requests that imputation-adjusted jackknife replicate weights be created. The VAR statement specifies the variables that are to be imputed, and the CELLS statement identifies the imputation cell variable ImputationCell. Because no IMPJOINT statements are specified, all the variables in the VAR statement are to be imputed by using their joint categories. For more information, see the section "IMPJOINT Statement" on page 8937. The STRATA, CLUSTER, and WEIGHT statements specify the strata, cluster, and weight variables. The OUT= option in the OUTPUT statement names the output data set DrugAbuseFEFI to store the imputed values, and the OUTJKCOEFS= option in the OUTPUT statement names the output data set DrugAbuseJKCOEFS to store the jackknife coefficients.

Summary information about the imputation method, number of observations, and survey design is shown in Output 110.2.1. The "Imputation Information" table summarizes the imputation method. The "Number of Observations" table displays the number of observations that are read and used (736) and the weighted number of observation that are read and used (19,600) by PROC SURVEYIMPUTE. The "Design Information" table shows that there are 35 PSUs and 10 strata.

Output 110.2.1 Imputation Information

The SURVEYIMPUTE Procedure

Imputation Information	
Data Set	WORK.DRUGABUSE
Weight Variable	ObsWeight
Stratum Variable	Strata
Cluster Variable	PSU
Imputation Method	FEFI

Number of Observations Read	736
Number of Observations Used	736
Sum of Weights Read	19599.99
Sum of Weights Used	19599.99

Design Summary	
Number of Strata	10
Number of Clusters	35

Selected observations for some variables from the output data set are displayed in Output 110.2.2. The output data set, DrugAbuseFEFI, contains unit identification, the recipient index, imputation-adjusted full-sample weights, imputation-adjusted jackknife weights, and all the variables from the input data set DrugAbuse. The output data set contains 35 sets of replicate weights, but only the first three sets of replicate weights are shown in Output 110.2.2. Units that are complete respondents have one row, but units that are incomplete respondents have multiple rows in the output data set. For example, unit 21 is a complete respondent, so it has only one row in the output data set and its Recipient value is 0. Unit 22 is an incomplete respondent; it has 20 rows in the output data set, its Recipient values range from 1 to 20, and its imputation-adjusted full-sample weights (ImpWt) range from 0.02 to 1.02. The sum of ImpWt for all rows (donor cells) for observation unit 22 is 5, which is the full sample weight for unit 22.

Output 110.2.2 Observations for Selected Units

UnitID	Recipient	ImpWt	Strata	PSU	ObsWeight	ImputationCell	Sex	Race	Insurance	Drug	Alcohol	Treatment	ImpRepWt_1	ImpRepWt_2	ImpRepWt_3
20	0	5.00000	1	1	5	1	0	1	1	1	2	1	0.00000	6.66667	6.66667
21	0	5.00000	1	2	5	1	0	3	3	1	1	1	6.66667	0.00000	6.66667
22	1	0.49238	1	2	5	2	1	1	1	1	1	1	0.65757	0.00000	0.66513
22	2	1.01597	1	2	5	2	1	1	1	1	2	1	1.36836	0.00000	1.33813
22	3	0.14661	1	2	5	2	1	1	1	1	2	2	0.18561	0.00000	0.19845
22	4	0.11496	1	2	5	2	1	1	1	2	1	1	0.15441	0.00000	0.15268
22	5	0.03894	1	2	5	2	1	1	1	2	1	2	0.05231	0.00000	0.05172
22	6	0.32730	1	2	5	2	1	1	1	2	2	1	0.44358	0.00000	0.43861
22	7	0.21661	1	2	5	2	1	1	1	2	2	2	0.29074	0.00000	0.28749
22	8	0.24024	1	2	5	2	1	1	2	1	1	1	0.32268	0.00000	0.31907
22	9	0.10453	1	2	5	2	1	1	2	1	1	2	0.12908	0.00000	0.14255
22	10	0.46726	1	2	5	2	1	1	2	1	2	1	0.61595	0.00000	0.62442
22	11	0.19382	1	2	5	2	1	1	2	1	2	2	0.26067	0.00000	0.25730
22	12	0.37531	1	2	5	2	1	1	2	2	2	1	0.50409	0.00000	0.49845
22	13	0.03697	1	2	5	2	1	1	2	2	2	2	0.04966	0.00000	0.04911
22	14	0.15533	1	2	5	2	1	1	3	1	1	1	0.19731	0.00000	0.21002
22	15	0.08672	1	2	5	2	1	1	3	1	1	2	0.11647	0.00000	0.11517
22	16	0.69416	1	2	5	2	1	1	3	1	2	1	0.93613	0.00000	0.92566
22	17	0.03688	1	2	5	2	1	1	3	1	2	2	0.04953	0.00000	0.04897
22	18	0.02106	1	2	5	2	1	1	3	2	1	1	0.02828	0.00000	0.02796
22	19	0.18442	1	2	5	2	1	1	3	2	2	1	0.23638	0.00000	0.24865
22	20	0.05053	1	2	5	2	1	1	3	2	2	2	0.06788	0.00000	0.06712
23	0	5.00000	1	2	5	2	1	1	1	1	1	1	6.66667	0.00000	6.66667

You can use the imputed data set and the imputation-adjusted replicate weights to compute any estimators from your imputed data. You can use the REPWEIGHTS statement in any SAS/STAT survey analysis procedures to specify the imputation-adjusted replicate weights. For example, the following statements use PROC SURVEYLOGISTIC to perform logistic regression analysis of the imputed data:

```
proc surveylogistic data=DrugAbuseFEFI varmethod=Jackknife;
   class Treatment Insurance Sex Race;
   model Drug=Treatment Insurance Age Sex Race;
   weight ImpWt;
   repweights ImpRepWt_: / jkcoefs=DrugAbuseJKCOEFS;
   ods output parameterestimates=FEFILogisticAnalysis;
run;
```

The WEIGHT statement specifies the imputation-adjusted full-sample weight (ImpWt), and the REPWEIGHTS statement specifies the imputation-adjusted replicate weights (ImpRepWt_1, . . . , ImpRepWt_35). The JKCOEFS= option in the REPWEIGHTS statement specifies the jackknife coefficients.

The parameter estimates and their standard errors are displayed in Output 110.2.3. The variance estimators correctly account for both the design variability and the imputation variability.

Output 110.2.3 Logistic Regression Analysis of the Fractionally Imputed Data Set

The SURVEYLOGISTIC Procedure

Analysis of Maximum Likelihood Estimates

Parameter		Estimate	Standard Error	t Value	Pr > |t|
Intercept		0.5105	0.2281	2.24	0.0317
Treatment	1	0.1048	0.1600	0.65	0.5168
Insurance	1	-0.1152	0.1454	-0.79	0.4337
Insurance	2	-0.0461	0.1183	-0.39	0.6988
Age		0.00195	0.00459	0.42	0.6736
Sex	0	-0.0719	0.0931	-0.77	0.4452
Race	1	0.4463	0.1574	2.84	0.0075
Race	2	-0.1212	0.2110	-0.57	0.5695

NOTE:
The degrees of freedom for the t tests is 35.

Example 110.3: Fully Efficient Fractional Imputation, Fay's Balanced Repeated Replication, and Domain Analysis

This example demonstrates the FEFI method by using data from the third National Health and Nutrition Examination Survey (NHANES III). The data set contains a set of BRR replicate weights. The REPWEIGHTS statement in PROC SURVEYIMPUTE is used to create imputation-adjusted replicate weights. The imputed data set and the imputation-adjusted replicate weights are then used in PROC SURVEYFREQ to create crosstabulation tables and to perform domain analysis.

The objective of NHANES is to study the health and nutritional status of the US population. NHANES uses a multistage stratified area sample with typically two PSUs per stratum. Strata are created based on geographic location, Metropolitan Statistical Areas (MSAs), and other demographics. An MSA or a group of counties are selected as PSUs from each stratum. Sampling weights are unequal because of different selection probabilities among different subgroups and for reasons such as nonresponse and undercoverage. For more information about NHANES, see http://www.cdc.gov/nchs/nhanes/about_nhanes.htm.

NHANES III data contain missing values in many items. Multiple imputation was used to impute some of the missing items. Five multiply imputed data sets are available for public use. Because FEFI will be used in this example to impute the missing values, you need the observed data, the missing (or imputation) flag for every item, and only one imputed data set. The data sets Core and IMP1 have been downloaded from http://www.cdc.gov/nchs/nhanes/nh3data.htm#7a. The Core data set contains the demographic variables, full sample weights, replicate weights, and imputation flags. The replicate weights are created by using Fay's BRR method with a Fay coefficient of 0.3. The IMP1 data set contains the first version of the five multiply imputed data sets.

For this example, a new data set, Smoke, is created by merging the Core and IMP1 data sets by the observation sequence number, SEQN. The Smoke data set contains the following items:

- SEQN: observation sequence number
- WTPFQX6: observation weight
- WTPQRP1 to WTPQRP52: 52 replicate weights from the BRR method
- DMARETHN: race-ethnicity; 1 for white, 2 for black, 3 for Mexican American, and 4 for other
- HSSEX: gender; 1 for male and 2 for female
- HFF1IF: imputation flag for HFF1MI; 1 for observed and 2 for imputed
- HAN6SRIF: imputation flag for HAN6SRMI; 0 for not applicable, 1 for observed, and 2 for imputed
- HAR3RIF: imputation flag for HAR3RMI; 0 for not applicable, 1 for observed, and 2 for imputed
- HAT28IF: imputation flag for HAT28MI; 0 for not applicable, 1 for observed, and 2 for imputed
- HFF1MI: anyone smokes cigarettes in the home; 1 for yes, and 2 for no
- HAN6SRMI: beer, wine, or liquor per month; –9 for not applicable, 1 for 0 time in the past month, 2 for 1 to 10 times in the past month, and 3 for more than 10 times in the past month
- HAR3RMI: smoke cigarettes now; –9 for not applicable, 1 for yes, and 2 for no
- HAT28MI: activity level compared to others; –9 for not applicable, 1 for more active, 2 for less active, and 3 for about the same
- Education: highest education attained; levels are elementary, high school, college, and unknown

For donor-based imputation methods, auxiliary information is used to create imputation cells. Imputation cells divide the data into groups of similar units such that the recipient units share similar characteristics with the donor units in the same group. Characteristics for imputation cells might come from the same survey or from other auxiliary sources such as census data or previous surveys. The cell identification is known for every unit in the sample. Categorical levels of auxiliary variables are often used to create imputation cells. For a helpful review, see Brick and Kalton (1996). For the purpose of this example, seven imputation cells were created by using only two demographic variables: race-ethnicity status (DMARETHN) and gender

(HSSEX). Both variables are available in the Core data set, and both have no missing values. The imputation cells are identified by the variable ImputationCells in the Smoke data set.

The following DATA step creates the imputation cells and the variable Education, and replaces the multiply imputed values with missing values:

```
/*--Create education levels, imputation cells and
    assign . to missing items --*/
data Smoke; set Smoke;
   if HFA7 <=8                    then Education='Elementary ';
   if HFA7 > 8  and HFA7 <= 12 then Education='High School';
   if HFA7 > 12 and HFA7 <= 17 then Education='College    ';
   if HFA7 > 17                   then Education='Unknown    ';
   if DMARETHN = 1 & HSSEX = 1 then ImputationCells=1;
   if DMARETHN = 1 & HSSEX = 2 then ImputationCells=2;
   if DMARETHN = 2 & HSSEX = 1 then ImputationCells=3;
   if DMARETHN = 2 & HSSEX = 2 then ImputationCells=4;
   if DMARETHN = 3 & HSSEX = 1 then ImputationCells=5;
   if DMARETHN = 3 & HSSEX = 2 then ImputationCells=6;
   if DMARETHN = 4             then ImputationCells=7;
   if HFF1IF   = 2 then HFF1MI   = .;
   if HAN6SRIF = 2 then HAN6SRMI = .;
   if HAR3RIF  = 2 then HAR3RMI  = .;
   if HAT28IF  = 2 then HAT28MI  = .;
run;
```

The following statements request that the missing values be imputed by using the FEFI method:

```
proc surveyimpute data=Smoke method=FEFI varmethod=BRR;
   weight wtpfqx6;
   repweights wtpqrp:;
   id seqn;
   class hff1mi han6srmi har3rmi hat28mi;
   var   hff1mi han6srmi har3rmi hat28mi;
   cells ImputationCells;
   output out=SmokeImputed;
run;
```

The PROC SURVEYIMPUTE statement invokes the procedure, the DATA= option specifies the input data set Smoke, the METHOD= option requests the FEFI method, and the VARMETHOD= option requests the imputation-adjusted BRR replication weights. The WEIGHT statement specifies the weight variable, and the REPWEIGHTS statement specifies the unadjusted BRR replicate weights. Because you specify replicate weights by using the REPWEIGHTS statement, you do not need to specify the Fay coefficient in PROC SURVEYIMPUTE. The variable SEQN in the ID statement identifies the observation units. The VAR statement specifies the variables to be imputed, the CELLS statement identifies the imputation cell variable ImputationCells, and the OUT= option in the OUTPUT statement names the output data set SmokeImputed. You request that all four variables be imputed jointly and that the imputed data be saved in the SmokeImputed data set.

Note that this example creates imputation-adjusted BRR replicate weights from the unadjusted BRR replicate weights that are available for these data. If the unadjusted BRR replicate weights are not available to you, then PROC SURVEYIMPUTE first creates the unadjusted BRR replicate weights and then updates the unadjusted weights for imputation to create the imputation-adjusted BRR replicate weights. For more information, see the section “Balanced Repeated Replication (BRR) Method” on page 8952.

Summary information about the number of observations and class level information are shown in Output 110.3.1. The “Number of Observations” table displays the number of observations (33,994) that are read and used. The weighted number of observations that are read shows that the 33,994 observation units in the sample represent over 251,000,000 observation units in the population. The “Class Level Information” table displays the observed levels for the analysis variables. The “Missing Data Patterns” table shows an arbitrary missing pattern. There are 12 different missing pattern groups. An “X” denotes that the variable is observed in that group, and a “.” denotes that the variable is missing. Almost 94% of the observation units have no missing values (Group 1), 4.5% of the observation units have missing values for the variable HAN6SRMI (Group 4), and 1% of the observation units have missing values for the variable HAT28MI (Group 2).

Output 110.3.1 Imputation Information

The SURVEYIMPUTE Procedure

Number of Observations Read	33994
Number of Observations Used	33994
Sum of Weights Read	2.511E8
Sum of Weights Used	2.511E8

Class Level Information

Class	Levels	Values
HFF1MI	2	1 2
HAN6SRMI	4	-9 1 2 3
HAR3RMI	3	-9 1 2
HAT28MI	4	-9 1 2 3

Output 110.3.2 Missing Data Patterns

Missing Data Patterns

									Group Means		
Group	**HFF1MI**	**HAN6SRMI**	**HAR3RMI**	**HAT28MI**	**Freq**	**Sum of Weights**	**Unweighted Percent**	**Weighted Percent**	**HFF1MI 1**	**HFF1MI 2**	**HAN6SRMI -9**
1	X	X	X	X	31916	2.2918E8	93.89	91.27	0.370883	0.629117	0.275976
2	X	X	X	.	383	3584033	1.13	1.43	0.312278	0.687722	0
3	X	X	.	X	2	6696.18	0.01	0.00	0	1.000000	0
4	X	.	X	X	1536	17201676	4.52	6.85	0.417227	0.582773	.
5	X	.	X	.	25	255366.8	0.07	0.10	0.512251	0.487749	.
6	X	.	.	.	1	1137.27	0.00	0.00	0	1.000000	.
7	.	X	X	X	106	723162	0.31	0.29	.	.	0.277402
8	.	X	X	.	5	21175.81	0.01	0.01	.	.	0
9	.	X	.	.	2	43867.66	0.01	0.02	.	.	0
10	.	.	X	X	4	11751.6	0.01	0.00	.	.	.
11	.	.	X	.	1	4483.45	0.00	0.00	.	.	.
12	.	.	.	.	13	59745.61	0.04	0.02	.	.	.

Missing Data Patterns

	Group Means									
Group	**HAN6SRMI 1**	**HAN6SRMI 2**	**HAN6SRMI 3**	**HAR3RMI -9**	**HAR3RMI 1**	**HAR3RMI 2**	**HAT28MI -9**	**HAT28MI 1**	**HAT28MI 2**	**HAT28MI 3**
1	0.365368	0.219609	0.139048	0.275976	0.199395	0.524629	0.275976	0.239253	0.160019	0.324752
2	0.533112	0.322946	0.143942	0	0.231400	0.768600	.	.	.	.
3	0	0.695886	0.304114	.	.	.	0	0.695886	0	0.304114
4	.	.	.	0	0.362741	0.637259	0	0.339890	0.222725	0.437384
5	.	.	.	0	0.371613	0.628387	.	.	.	.
6	.	.	.	.	.	.	.	.	.	.
7	0.570801	0.099198	0.052599	0.277402	0.156784	0.565813	0.277402	0.185712	0.192402	0.344483
8	0.414365	0.585635	0	0	0.078044	0.921956	.	.	.	.
9	0.262239	0.737761	0	.	.	.	.	.	.	.
10	.	.	.	0	0	1.000000	0	0.664584	0	0.335416
11	.	.	.	0	0	1.000000	.	.	.	.
12	.	.	.	.	.	.	.	.	.	.

The "Iteration History" table shown in Output 110.3.3 displays the maximum absolute and relative differences of the fractional weights for the EM algorithm for the full sample. The algorithm converged after four iterations. The "Imputation Summary" table shown in Output 110.3.4 displays the number of observed units (31,961), the number of missing units (2,078), and the number of imputed units. All units that have missing values have been imputed.

Output 110.3.3 Iteration History for the EM

Iteration History		
Iteration Number	Maximum Absolute Difference	Maximum Relative Difference
1	830.4733	0.18278
2	93.33655	0.00904
3	16.14668	0.00237
4	4.138731	0.00061

Output 110.3.4 Imputation Summary

Imputation Summary		
Observation Status	Number of Observations	Sum of Weights
Nonmissing	31916	2.2918E8
Missing	2078	21913095
Missing, Imputed	2078	21913095
Missing, Not Imputed	0	0
Missing, Partially Imputed	0	0

The imputed data set SmokeImputed contains the imputation-adjusted weight (ImpWt) and 52 imputation-adjusted replicate weights (ImpRepWt_1 to ImpRepWt_52). The SmokeImputed data set has 38,701 data lines. The number of imputed values for an observation unit ranges from two to six, but around 80% of the units are imputed by using two or three imputed values.

You can use the imputed data set, the imputation-adjusted replicate weights, and the appropriate Fay coefficient to compute any estimators from your imputed data. You should use the REPWEIGHTS statement in SAS/STAT survey analysis procedures to specify the imputation-adjusted replicate weights. This example uses PROC SURVEYFREQ to perform the following analyses:

- estimate the percentage of smokers and nonsmokers in the population
- describe the smoking habits of an individual and of anyone who smokes in the home
- perform a domain analysis of activity levels for different levels of education

The PROC SURVEYFREQ statement invokes the procedure, the DATA= option names the imputed data set SmokeImputed, and the VARMETHOD=option requests the BRR variance estimation. The FAY= option for VARMETHOD=BRR specifies the Fay coefficient 0.3. Because your replicate weights come from Fay's BRR method, you must specify the FAY= option in the SAS/STAT survey analysis procedures to appropriately estimate the variance. The VARHEADER=LABEL option in the PROC SURVEYFREQ statement requests that the labels of the variables be displayed in the output. The WEIGHT statement specifies the imputation-adjusted full sample weights, and the REPWEIGHTS statement specifies the imputation-adjusted replicate weights. Note that the imputation-adjusted full sample and replicate weights are created by

PROC SURVEYIMPUTE, and they are different from the unadjusted weights available in the Smoke data set. The first TABLE statement requests a two-way frequency analysis for HFF1MI and HAR3RMI. The second TABLE statement requests a domain analysis for HAT28MI, where the variable Education is used as the domain variable. The ROW option in the TABLE statement is required in order to compute the distribution of HAT28MI for different levels of Education. The NOTOTAL, NOFREQ, and NOWT options suppress some output columns.

```
proc surveyfreq data=SmokeImputed varmethod=brr(fay=0.3) varheader=label;
   weight ImpWt;
   repweights ImpRepWt_:;
   table HFF1MI*HAR3RMI;
   table Education*HAT28MI / row nototal nofreq nowt;
run;
```

The data summary and the variance estimation information are displayed in Output 110.3.5. There are 38,701 data lines in the SmokeImputed data set. These 38,701 data lines represent the 33,994 observation units in the Smoke data set. The observation units are identified by the variable SEQN. The sum of weights is over 251,000,000, which is the same as the sum of weights in the Smoke data set. The sum of weights is an estimate of the population size. The "Variance Estimation" table shows that 52 replicate weights from Fay's BRR method are used for variance estimation with the Fay coefficient 0.3.

Output 110.3.5 Summary Information

The SURVEYFREQ Procedure

Data Summary	
Number of Observations	38701
Sum of Weights	251097002

Variance Estimation	
Method	BRR
Replicate Weights	SMOKEIMPUTED
Number of Replicates	52
Fay Coefficient	0.300

A two-way table for the smoking habit of the observation unit and smoking in the home is shown in Output 110.3.6. There are 21% smokers and 54% nonsmokers in the population. Nearly 19% of the individuals are smokers and live in a home where at least one person smokes in the home, but only 2% of the individuals are smokers and live in a home where no other household member smokes in the home. However, almost 9% of the individuals are nonsmokers but live in a home where at least one household member smokes in the home. The standard errors that are reported in the table properly account for the imputation.

Output 110.3.6 Two-Way Table for Smoking Status

Table of Anyone living here smoke cigs in home by Smoke cigarettes now (recode)						
Anyone living here smoke cigs in home	**Smoke cigarettes now (recode)**	**Frequency**	**Weighted Frequency**	**Std Err of Wgt Freq**	**Percent**	**Std Err of Percent**
1	**-9**	5431	24762830	717089	9.8619	0.2856
	1	5788	47166758	1361381	18.7843	0.5422
	2	3341	21790932	614088	8.6783	0.2446
	Total	14560	93720520	2180565	37.3244	0.8684
2	**-9**	8582	38702422	701855	15.4133	0.2795
	1	881	5837874	397358	2.3249	0.1582
	2	14678	112836186	1602093	44.9373	0.6380
	Total	24141	157376482	2180565	62.6756	0.8684
Total	**-9**	14013	63465252	260068	25.2752	0.1036
	1	6669	53004633	1276926	21.1092	0.5085
	2	18019	134627118	1328833	53.6156	0.5292
	Total	38701	251097002	2.25270	100.000	

Suppose you want to perform a domain analysis by using the imputed data. If a list of domain variables is available before the imputation, then sometimes it is desirable to use the domain variables to create the imputation cells. However, requests for domain analyses often come after the imputation. In addition, data users might use domain variables that are different from what are used to create the imputation cells. In this example, the domain variable Education was not used to create the imputation cells. Although education level is not used in the imputation, it is reasonable to use the imputed data to perform domain analysis for every level of education. Domain analysis for activity levels for different education levels is shown in Output 110.3.7. If the highest education level is college, then 38% are reported as more active and 21% are reported as less active than their peers. If the highest education level is high school, then 28% are reported as more active and 20% are reported as less active than their peers. The standard errors that are reported in the table properly account for the imputation.

Output 110.3.7 Domain Analysis for Activity Levels by Education

Table of Education by Compare own activity level to others

Education	Compare own activity level to others	Percent	Std Err of Percent	Row Percent	Std Err of Row Percent
College	**-9**	0.0017	0.0016	0.0055	0.0052
	1	11.9908	0.4117	38.3446	0.9792
	2	6.5292	0.3253	20.8795	0.8461
	3	12.7494	0.4641	40.7704	0.9978
Elementary	**-9**	21.8961	0.1269	74.2986	0.9154
	1	1.8571	0.1193	6.3015	0.3596
	2	2.0051	0.1500	6.8039	0.4513
	3	3.7121	0.1970	12.5961	0.5420
High School	**-9**	3.2191	0.1238	8.3653	0.3251
	1	10.6977	0.2990	27.7997	0.6634
	2	7.8486	0.2782	20.3959	0.6333
	3	16.7160	0.4958	43.4391	0.7062
Unknown	**-9**	0.1583	0.0289	20.3674	3.2674
	1	0.1981	0.0463	25.4966	3.7217
	2	0.1502	0.0346	19.3311	3.3973
	3	0.2704	0.0426	34.8049	3.2211

References

Andridge, R. R., and Little, R. J. A. (2010). “A Review of Hot Deck Imputation for Survey Non-response.” *International Statistical Review* 78:40–64.

Barnard, J., and Rubin, D. B. (1999). “Small-Sample Degrees of Freedom with Multiple Imputation.” *Biometrika* 86:948–955.

Bethlehem, J. (2009). *Applied Survey Methods: A Statistical Perspective*. Hoboken, NJ: John Wiley & Sons.

Brick, J. M., and Kalton, G. (1996). “Handling Missing Data in Survey Research.” *Statistical Methods in Medical Research* 5:215–238.

Dippo, C. S., Fay, R. E., and Morganstein, D. H. (1984). “Computing Variances from Complex Samples with Replicate Weights.” In *Proceedings of the Survey Research Methods Section*, 489–494. Alexandria, VA: American Statistical Association.

Fay, R. E. (1984). “Some Properties of Estimates of Variance Based on Replication Methods.” In *Proceedings of the Survey Research Methods Section*, 495–500. Alexandria, VA: American Statistical Association.

Fay, R. E. (1989). “Theory and Application of Replicate Weighting for Variance Calculations.” In *Proceedings of the Survey Research Methods Section*, 212–217. Alexandria, VA: American Statistical Association.

Fay, R. E. (1993). “Valid Inferences from Imputed Survey Data.” In *Proceedings of the Survey Research Methods Section*, 41–48. Alexandria, VA: American Statistical Association.

Fay, R. E. (1996). “Alternative Paradigms for the Analysis of Imputed Survey Data.” *Journal of the American Statistical Association* 91:490–498.

Fellegi, I. P., and Holt, D. (1976). "A Systematic Approach to Automatic Edit and Imputation." *Journal of the American Statistical Association* 71:17–35.

Fuller, W. A. (2009). *Sampling Statistics*. Hoboken, NJ: John Wiley & Sons.

Fuller, W. A., and Kim, J. K. (2005). "Hot Deck Imputation for the Response Model." *Survey Methodology* 31:139–149.

Haziza, D. (2009). "Imputation and Inference in the Presence of Missing Data." In *Sample Surveys: Design, Methods, and Applications*, edited by D. Pfeffermann, and C. R. Rao, 215–246. Vol. 29A of Handbook of Statistics. Amsterdam: North-Holland.

Ibrahim, J. G. (1990). "Incomplete Data in Generalized Linear Models." *Journal of the American Statistical Association* 85:765–769.

Judkins, D. R. (1990). "Fay's Method for Variance Estimation." *Journal of Official Statistics* 6:223–239.

Kalton, G., and Kasprzyk, D. (1986). "The Treatment of Missing Survey Data." *Survey Methodology* 12:1–16.

Kalton, G., and Kish, L. (1984). "Some Efficient Random Imputation Methods." *Communications in Statistics—Theory and Methods* 13:1919–1939.

Kim, J. K. (2002). "A Note on Approximate Bayesian Bootstrap Imputation." *Biometrika* 89:470–477.

Kim, J. K., and Fuller, W. A. (2004). "Fractional Hot Deck Imputation." *Biometrika* 91:559–578.

Kim, J. K., and Fuller, W. A. (2013). "Hot Deck Imputation for Multivariate Missing Data." In *Proceedings of the Fifty-Ninth ISI World Statistics Congress*, 924–929. The Hague: International Statistics Institute.

Kim, J. K., and Shao, J. (2014). *Statistical Methods for Handling Incomplete Data*. Boca Raton, FL: CRC Press.

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

Lohr, S. L. (2010). *Sampling: Design and Analysis*. 2nd ed. Boston: Brooks/Cole.

Matsumoto, M., and Nishimura, T. (1998). "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-random Number Generator." *ACM Transactions on Modeling and Computer Simulation* 8:3–30.

Myers, T. A. (2011). "Goodbye, Listwise Deletion: Presenting Hot Deck Imputation as an Easy and Effective Tool for Handling Missing Data." *Communication Methods and Measures* 5:297–310.

Rao, J. N. K., and Shao, J. (1992). "Jackknife Variance Estimation with Survey Data under Hot Deck Imputation." *Biometrika* 79:811–822.

Rubin, D. B., and Schenker, N. (1986). "Multiple Imputation for Interval Estimation from Simple Random Samples with Ignorable Nonresponse." *Journal of the American Statistical Association* 81:366–374.

Särndal, C. E., and Lundström, S. (2005). *Estimation in Surveys with Nonresponse*. Chichester, UK: John Wiley & Sons.

Särndal, C. E., Swensson, B., and Wretman, J. (1992). *Model Assisted Survey Sampling*. New York: Springer-Verlag.

Shao, J., and Tu, D. (1995). *The Jackknife and Bootstrap*. New York: Springer-Verlag.

Tillé, Y. (2006). *Sampling Algorithms*. New York: Springer.

Wolter, K. M. (2007). *Introduction to Variance Estimation*. 2nd ed. New York: Springer.

Subject Index

Syntax Index