Paper CT06

From Missing to Meaningful: Validating Multiple Imputation Techniques for Reliable Analysis

Karthick Jayaraman, SESAN Biosciences, Chennai, India

ABSTRACT

Handling missing data effectively is a persistent challenge in applied research, particularly in large-scale observational studies and clinical datasets. Multiple imputation (MI) has become a robust method for handling missing values by creating several plausible datasets and combining results to reflect the uncertainty due to missingness. This presentation focuses on validating MI techniques to ensure their reliability and applicability across varied data scenarios. It systematically compares key MI methods - including multivariate imputation by chained equations (MICE), predictive mean matching (PMM), and Bayesian approaches - under different missing data scenarios (MCAR, MAR, and MNAR).

It presents a thorough validation framework that helps programmers better assess and justify their use of MI, ensuring greater reliability and transparency in empirical findings.

Practical implementations using SAS/STAT® (PROC MI and PROC MIANALYZE) and R (MICE package) are demonstrated for imputing incomplete multivariate datasets, analyzing imputed outputs, and comparing results across methods.

INTRODUCTION

Missing data poses a significant concern in statistical analysis because it can affect both the accuracy and validity of study results. When data are missing, the effective sample size is reduced, which in turn lowers statistical power and misleads conclusions. This loss of information can lead to wider confidence intervals and less precise estimates. More critically, missing data can introduce bias. Commonly used methods such as complete-case analysis (listwise deletion) exclude observations with missing values, which may systematically differ from those with complete data, thereby compromising representativeness. Similarly, simple imputation methods like mean substitution, LOCF or BOCF fail to account for the uncertainty introduced by missing data and can distort the underlying data distribution. Therefore, understanding the consequences of missing data and using appropriate techniques to manage it is essential.

Multiple Imputation (MI) method is a statistically robust and widely accepted method tailored to address these limitations. Instead of substituting each missing value with a single fixed estimate, MI generates multiple plausible values that capture the uncertainty associated with the missing observations. Each resulting complete dataset is analyzed independently using conventional statistical methods, and the findings are then combined to obtain a reliable study conclusion using Rubin's Rule.

MISSING DATA PATTERNS

Missing data patterns describe the structure and arrangement of missing values within a dataset and provide important insight into how missingness occurs. There are two missing data patterns:

1. Monotone missing pattern:
   If a variable is missing for an individual, all subsequent variables/measurement in that order are also missing.
2. Arbitrary (non-monotone) missing pattern:
   Missing values occur in no specific pattern or sequence.

Figure 1 Monotone missing pattern

SUBJID	BASE	V1	V2	V3
1001	X	X	X	Missing
1002	X	X	Missing	Missing
1003	X	Missing	Missing	Missing

Monotone Pattern

Figure 2 Arbitrary (non-monotone) missing pattern

SUBJID	BASE	V1	V2	V3
1001	X	Missing	X	X
1002	X	X	Missing	X
1003	X	Missing	X	Missing

Arbitrary Pattern

MISSING DATA MECHANISMS

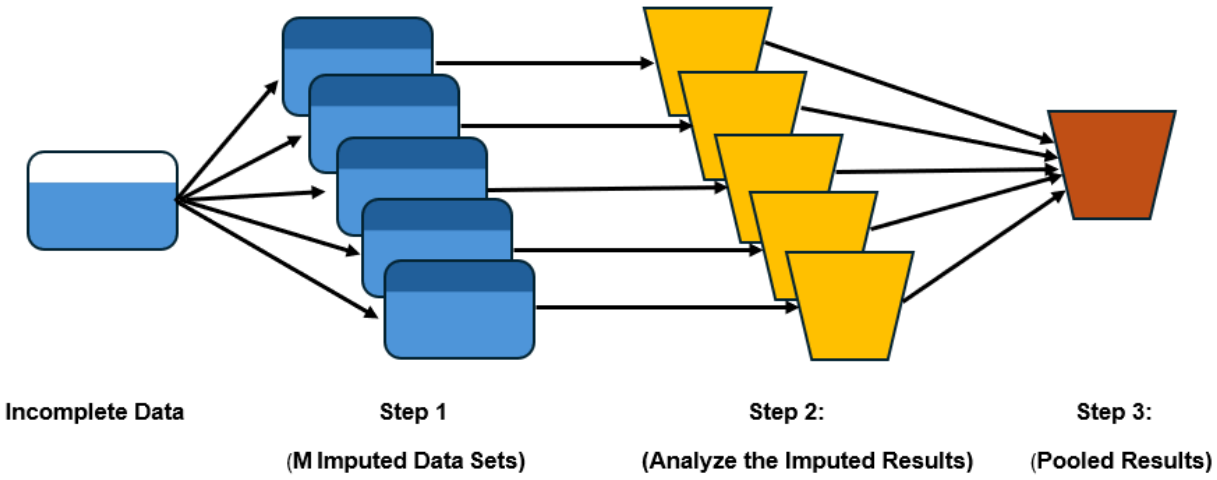
Understanding these mechanisms is crucial for choosing appropriate imputation methods to avoid biased results. Missing data mechanisms describe the relationship between the probability that a value is missing and the observed or unobserved data. Missing data can be classified into three categories: Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR).

Missing Completely at Random (MCAR), occurs when the missingness is completely unrelated to both observed and unobserved values in the dataset. In simpler terms, the absence of data occurs randomly.
Missing At Random (MAR), occurs when missingness is related to the observed data, but not to the unobserved values themselves.
Missing Not at Random (MNAR) occurs when the missingness itself is related to the unobserved values. In this case, the missing data is not random and is associated with specific reasons or patterns.

MULTIPLE IMPUTATION STEPS

Multiple Imputation (MI) is not simply a technique for imputing missing data. It is also a method for obtaining estimates and correct inferences for statistics ranging from simple descriptive statistics to the parameters of complex multivariate models. As illustrated in Figure 3, a complete multiple imputation analysis can be organized into three sequential steps.

Figure 3 Multiple Imputation (MI) Process



IMPUTATION STEP

The imputation step is a core component of the multiple imputation process and involves replacing missing values with plausible estimates based on observed data. During this step, an appropriate imputation model is specified for each variable with missing data, considering its distribution, variable type, and relationship with other relevant variables. This process is repeated several times to create m times of completed datasets. These imputed datasets are identical for the observed data entries but differ in the imputed values.

ANALYSIS STEP

Step 2 uses the imputed datasets created in step 1 and analyzes each of the M imputed datasets separately using standard statistical procedures specified for the analysis. During this step, it is important to output the parameter estimates and their standard errors from each analysis, which will be required in next step.

POOLING STEP

In Step 3, the analysis results obtained from the preceding analysis step are combined into a single estimation with standard error using Rubin's rules.

MULTIPLE IMPUTATION IN SAS

CONSIDERATIONS

a) Number of Imputations:
   There are no strict rules for the number of imputations. However recent studies suggest the use of a much larger number of imputation than the classic recommendation of 3 to 4 imputations. Thus, the default number of imputation in PROC MI has been changed from NIMPUTE=5 to NIMPUTE=25.
b) Arbitrary missing data handling: Generally handled by two methods:
   i. Two-step imputation: First, impute only the intermittent missing data, so each dataset will only have missing ending data or a monotone missing pattern, Then, impute the remaining missing ending data using a regression imputation model.
   ii. One-step imputation: Directly impute all the missing data using a regression imputation model.
c) Imputed value out of range
   i. Manually assign boundary: After imputation, if there are results outside of the regular range, manually assign the values to the upper bound and lower bound.
   ii. Constraint in Proc MI: In Proc MI, option MIN= and MAX= can be specified.

Several methods are available in SAS PROC MI to generate multiple imputation. The choice of these methods depends on pattern of missing data and type of variables to be imputed. Below table shows some of the available methods in PROC MI.

Table 1: SAS PROC MI IMPUTATION METHODS

Missing Data Pattern	Imputed Variable Type	Method	PROC MI Statement
Monotone	Continuous	Linear regression	MONOTONE REG
		Predictive mean matching	MONOTONE REGPMM
		Propensity score	MONOTONE PROPENSITY
	Binary/ordinal	Logistic regression	MONOTONE LOGISTIC
	Nominal	Discriminant function	MONOTONE DISCRIM
Arbitrary	Continuous	With continuous covariates: MCMC monotone method MCMC full-data imputation	 MCMC IMPUTE=MONOTONE MCMC IMPUTE=FULL
		With mixed covariates: FCS regression FCS predictive mean matching	 FCS REG FCS REGPMM
	Binary/ordinal	FCS logistic regression	FCS LOGISTIC
	Nominal	FCS discriminant function	FCS DISCRIM

This section presents a step-by-step example demonstrating the implementation of multiple imputation (MI) in SAS.

STEP 0: EXAMINE MISSING DATA PATTERN

PROC MI requires input data in a horizontal format with one record per subject. However, data to be imputed are often stored in a vertical ADaM BDS structure, so the dataset must be transposed to a horizontal format before applying PROC MI.

```
/* step 0a: Transpose input data to be in a horizontal format */
PROC TRANSPOSE DATA= adeff OUT= indset;
      BY usubjid trt;
      ID avisit;
      VAR aval;
RUN;
```

Obs	SUBJID	TRT	BASELINE	WK_12	WK_24	WK_36	WK_48	WK_60	WK_72
1	1001	Active	30	.	30	.	.	.	.
2	1002	Active	26	26	25	27	.	.	.
3	1003	Placebo	27	28	.	.	.	.	.
4	1004	Placebo	33	33	33	33	.	34	33

Output 1: Input data in horizontal format after transpose

Before proceeding to imputation step, it is important to examine the missing data pattern which can be done by PROC MI using zero imputation (NIMPUTE=0). The code below demonstrates this step.

```
/* step 0b: Examine the missing data patterns */
PROC MI DATA = indset NIMPUTE = 0;
   VAR BASELINE WK_12 WK_24 WK_36 WK_48 WK_60 WK_72;
 RUN;
```

Missing Data Patterns																	
										Group Means							
Group	BASELINE	WK_12	WK_24	WK_36	WK_48	WK_60	WK_72	Freq	Percent	BASELINE	WK_12	WK_24	WK_36	WK_48	WK_60	WK_72
1	X	X	X	X	X	X	X	54	88.52	22.151235	22.009259	22.691358	22.469136	23.098765	23.364198	23.654321
2	X	X	X	X	X	O	O	2	3.28	34.000000	33.500000	34.000000	32.500000	34.000000	.	.
3	X	X	X	X	.	X	X	2	3.28	19.333333	18.833333	17.500000	17.666667	.	21.500000	21.500000
4	X	X	X	X	O	O	O	1	1.64	26.000000	26.000000	25.333333	27.000000	.	.	.
5	X	X	O	O	O	O	O	1	1.64	21.666667	16.000000	.	.	.	.	.
6	X	.	X	O	O	O	O	1	1.64	30.000000	.	30.000000	.	.	.	.

Output 2: Missing Data Patterns

STEP 1: IMPUTATION
Since the missing data pattern of the input dataset is arbitrary (non-monotone), either the two-step PROC MI approach or the FCS REG/REGPMM method can be used.

Method 1: Imputation by two-step PROC MI approach assuming missing values are MNAR.
First, using MCMC method to impute just enough values to transform the non-monotone missing data pattern into a monotone missing pattern. Then, impute the remaining missing ending data using a monotone regression imputation method.

The code below shows how to obtain a monotone missing data pattern from arbitrary (non-monotone) missing patterns using MCMC algorithm. It is recommended that all variables - both those being imputed and the associated factors must be continuous.

```
/*Step 1a: MCMC method to impute intermediate missing to produce 5 datasets
with monotone missing pattern */
PROC MI DATA= indset OUT= indset_mono SEED=12345 NIMPUTE=5
      MINIMUM= . 0 0 0 0 0 MAXIMUM= . 36 36 36 36 36;
      BY trt;
      MCMC CHAIN = multiple IMPUTE=monotone;
      VAR BASELINE WK_12 WK_24 WK_36 WK_48 WK_60 WK_72;
RUN;
```

IMPUTATION	SUBJID	TRT	BASELINE	WK_12	WK_24	WK_36	WK_48	WK_60	WK_72
1	1001	Active	30	29.7737	30	.	.	.	.
2	1001	Active	30	31.3339	30	.	.	.	.
3	1001	Active	30	31.5557	30	.	.	.	.
4	1001	Active	30	29.6653	30	.	.	.	.
5	1001	Active	30	28.2933	30	.	.	.	.

Output 3: Partially imputed data by MCMC method to obtain monotone missing pattern.

The resulting output dataset indset_mono will have 5 copies of the original dataset where each copy will have a monotone missing pattern. Then directly move on to next step to performing multiple imputation using monotone regression assuming missing values are MNAR.

The code below demonstrates how multiple imputations are created using monotone regression method in PROC MI.

```
PROC SORT DATA= indset_mono;
      BY _Imputation_ trtn;
RUN;

/* Step 1b: Impute missing ending data using monotone regression using CCMV
assuming that missing values are MNAR */
PROC MI DATA= indset_mono OUT=imp_indset_monoreg NIMPUTE=1 SEED=12345
      MAXIMUM=. 36 36 36 36 36 MINIMUM=. 0 0 0 0 0;
      BY _Imputation_;
      VAR BASELINE WK_12 WK_24 WK_36 WK_48 WK_60 WK_72;
      MONOTONE REG (BASELINE WK_12 WK_24 WK_36 WK_48 WK_60 WK_72/details);
      MNAR MODEL (WK_12 WK_24 WK_36 WK_48 WK_60 WK_72/ modelobs= CCMV);
RUN;
```

IMPUTATION	SUBJID	TRT	BASELINE	WK_12	WK_24	WK_36	WK_48	WK_60	WK_72
1	1001	Active	30	29.7737	30	31.8103	33.4266	36.0000	33.7092
2	1001	Active	30	31.3339	30	29.7793	30.7383	33.8649	34.0435
3	1001	Active	30	31.5557	30	32.6640	30.5264	32.8720	28.0543
4	1001	Active	30	29.6653	30	29.3554	29.7644	30.5748	30.0389
5	1001	Active	30	28.2933	30	29.2694	31.5780	29.2217	31.9757

Output 4: Completed data after imputation by monotone regression method.

Method 2: Imputation by Fully Conditional specification (FCS) Regression.
The code below shows PROC MI using FCS regression method.

```
/*perform multiple imputation using FCS regression*/
PROC MI DATA= indset OUT=imp_indset_fcsreg NIMPUTE=5 SEED=12345
      MAXIMUM=. . . . . . 36 36 36 36 36
      MINIMUM=. . . . . . 0 0 0 0 0;
      CLASS trt region sex;
      VAR region sex BASELINE WK_12 WK_24 WK_36 WK_48 WK_60 WK_72;
      FCS REG (BASELINE WK_12 WK_24 WK_36 WK_48 WK_60 WK_72);
RUN;
```

STEP 2: ANALYZE THE M IMPUTED DATASETS

Step 2 uses the imputed datasets created by PROC MI (Step 1) and analyzes each of the M imputed datasets separately using standard SAS procedures with a BY _IMPUTATION_ statement. During this step, it is important to output the parameter estimates and their standard errors from each analysis, which will be required in next step.

Before analysing, it is needed to transpose data to get it vertically oriented and specify the new variable to be used in proc mixed.

```
PROC SORT DATA= imp_indset_monoreg;
      BY _Imputation_ usubjid trtn age base;
RUN;

/*Step 2a: transpose data to vertically oriented data */
PROC TRANSPOSE DATA= imp_indset_monoreg OUT= indset1(rename=(col1=aval_imp));
      BY _imputation_ usubjid trtn age base;
      VAR BASELINE WK_12 WK_24 WK_36 WK_48 WK_60 WK_72;
RUN;
/*Step 2b: perform MMRM analysis for each imputation*/
PROC MIXED DATA = indset1;
      BY _Imputation_;
      CLASS usubjid trtn avisitn;
      MODEL chg_imp = trtn base avisitn age trt*avisitn / solution;
      REPEATED avisitn/TYPE=UN SUBJECT=usubjid;
      LSMEANS trtn*avisit / DIFF = control('0') cl;
      ODS OUTPUT diffs = lsdiffs lsmeans = lsm solutionf = parms;
RUN;
```

IMPUTATION	EFFECT	TRTN	AVISITN	_TRTN	_AVISITN	ESTIMATE	STDERR	DF	TVALUE	PROBT	ALPHA	LOWER	UPPER
1	ESTIMATE	1	9	0	9	-1.1616	0.5302	115	-2.19	0.0305	0.05	-2.2118	-0.1113
2	ESTIMATE	1	9	0	9	-1.1644	0.5364	115	-2.17	0.0320	0.05	-2.2269	-0.1019
3	ESTIMATE	1	9	0	9	-1.3932	0.5384	115	-2.59	0.0109	0.05	-2.4596	-0.3268
4	ESTIMATE	1	9	0	9	-1.1873	0.5134	115	-2.31	0.0225	0.05	-2.2042	-0.1704
5	ESTIMATE	1	9	0	9	-1.2862	0.5206	115	-2.47	0.0150	0.05	-2.3174	-0.2549

Output 5: PROC PRINT Output Data Set from PROC MIXED

STEP 3: POOLING RESULTS

In Step 3, the analysis results obtained from the preceding analysis step are combined into a single estimation with standard error using PROC MIANALYZE.

```
PROC SORT DATA= lsdiffs; BY trtn avisitn _Imputation_; RUN;
/*Step 3: Combine estimates of each imputation*/
PROC MIANALYZE parms(classvar=full)= lsdiffs;
      CLASS trtn;
      MODELEFFECTS estimate;
      ODS OUTPUT parameterestimates=lsdiffs_out;
RUN;
```

Parameter Estimates (5 Imputations)										
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > |t|
estimate	-1.238527	0.539190	-2.29587	-0.18118	2321.1	-1.393190	-1.161558	0	-2.30	0.0217

Output 6: PROC PRINT Output Data Set from PROC MIANALYZE.

MULTIPLE IMPUTATION IN R

Multiple Imputation in R is commonly implemented using the **mice** (Multivariate Imputation by Chained Equations) package, though other packages like Amelia, missForest and Hmisc are also popular. It provides a flexible and powerful framework for handling missing data under the MAR assumption.

STEP 1A: LOAD AND EXPLORE THE DATA

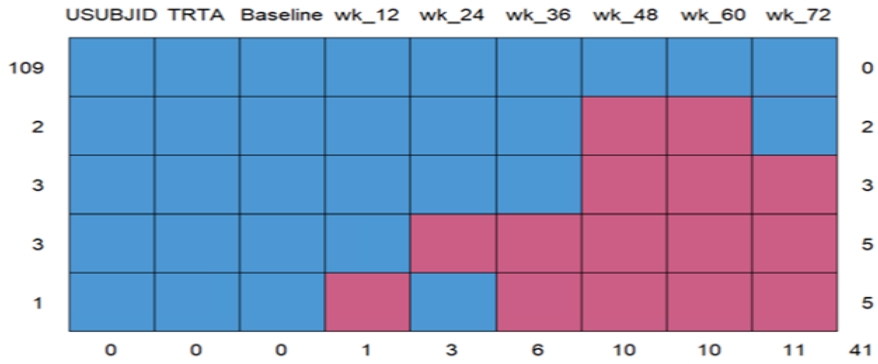
```
#Install and Load the Required Package
> install.packages(c("mice", "mmrm", "emmeans"))
> library(mice) # for multiple imputation pooling
> library(mmrm) # for MMRM analysis
> library(emmeans) # for LSmeans and treatment difference

#Import the dataset
# Use forward slashes in the file path, especially on Windows, to avoid errors
> indset <- read_sas("../ADaM/adeff.sas7bdat")
```

STEP 1B: EXAMINE MISSING DATA PATTERN

```
# Transpose input data to be in a horizontal format
> wide_data - indset %>%
      pivot_wider(id_cols = c("USUBJID", "TRTA"),
                  names_from = AVISIT
                     values_from = AVAL)

# Visualize missingness pattern
> md.pattern(wide_data)
```



Output 7: Missing data pattern in R mice package

STEP 2: IMPUTE MISSING DATA

```
#Perform Multiple Imputation
> imp <- mice(wide_data,
                  m=5, #no of imputed datasets
                  maxit=100, #no of iterations
                  method = "pmm", #Imputation method
                  seed = 12345)
```

The **mice** function creates an object containing 5 imputed datasets. The method = 'pmm' argument specifies the predictive mean matching method for continuous variables. Other commonly used methods are

"logreg" - binary variables
"polyreg" - unordered categorical variables (e.g., Race, Sex)
"polr" - ordered categorical variables (e.g., "None", "Mild pain", "Moderate pain", "Severe pain")

```
#Check/update imputation methods used in each variable
> imp$method
```

```
> imp$method
 USUBJID      TRTA Baseline    wk_12    wk_24    wk_36    wk_48    wk_60    wk_72
      ""        ""       ""    "pmm"    "pmm"    "pmm"    "pmm"    "pmm"    "pmm"
```

Output 8: Imputation methods

```
** imputation methods of each variable can be changed/removed using
method=c("", "", "", "pmm", "pmm", "pmm", "pmm", " pmm ", "pmm")

# Get all imputed data frames as a list
> imp_data <- complete(imp, action="long", include = TRUE)

# Transpose data back to vertically oriented and re-derive CHG with imp AVAL
> imp_data_long <- imp_data %>%
      pivot_longer(cols=c(Baseline,wk_12,wk_24,wk_36,wk_48,wk_60,wk_72),
                   names_to= "AVISIT",
                   values_to= "AVAL") %>%
      arrange(USUBJID, TRTA, BASE, .imp, .id) %>%
      mutate(CHG_IMP=AVAL-BASE)

# Convert long imputed data frame to mids.object
> imp_data_long <- imp_data_long %>%
      arrange(.imp, .id, USUBJID, AVISITN) %>%
      group_by(.imp) %>%
      mutate(idn= row_number()) %>%
      ungroup() %>% mutate(.id=idn) #re-assign .id values to avoid duplicate
                                     USUBJID AVISITN issue in mmrm() function

> imp_data_derived <= as.mids(imp_data_long)
```

STEP 3: ANALYZE IMPUTED DATA
You can analyze each imputed dataset using standard R functions. The with function from the mice package facilitates this.

```
# MMRM Analysis for each imputed dataset
  > model_fit <- with(imp_data_derived,
            mmrm(formula= CHG_IMP ~ TRTA + BASE + AVISITN + TRTA * AVISITN
                                + us(AVISITN | USUBJID)))
```

us(AVISITN | USUBJID) - specifies an unstructured (us) covariance matrix based on the time points (AVISIT) nested within each subject (USUBJID)

STEP 4: POOL RESULTS
After analyzing each imputed dataset, use the pool function to combine the results.

```
# Pool Results
> pooled_results <- pool(fit)
> summary(pooled_results, conf.int = TRUE)
```

The pool function combines the estimates from each imputed dataset and calculates appropriate standard errors.

Filter

term	AVISITN	estimate	std.error	conf.low	conf.high	df	p.value
Active - Placebo	72	-1.308748	0.430991	-2.153593	-0.4639028	8653.782	0.002399741

Output 9: Summary of pooled results.

CONCLUSION

In conclusion, effective handling of missing data is critical to ensuring the reliability, validity, and interpretability of statistical results. This paper presents a comprehensive, step-by-step approach to implementing multiple imputation using SAS and R, guiding users through the key stages of the process. In addition, it provides a structured framework for selecting suitable imputation methods based on data characteristics and missingness patterns, thereby supporting more robust and scientifically sound analyses.

REFERENCES

"Multiple Imputation of Missing Data Using SAS". Patricia Berglund, Steven G. Heeringa.
SAS Institute Inc. 2015. SAS/STAT® 14.1 User's Guide. Cary, NC: SAS Institute Inc.
"Multiple imputation for nonresponse in surveys". Rubin, D.B. (1987), New York: John Wiley & Sons, Inc.
"Multiple imputation as a valid way of dealing with missing data", Vadym Kalinichenko, Kharkiv, 2019.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Karthick Jayaraman
Company: SENSAN Biosciences Pvt. Ltd.
Address: Temple Steps, No.184-187, 3rd Floor, A-Block, Little Mount, Anna Salai, Chennai - 600 015.
Email: Karthickj@sensanbio.com
Website: https://sensanbio.com/