PharmaSUG China 2025 - Paper SA-148

Handling Missing Data in Clinical Trials with Multiple Imputation Methods

Zhihua Zhang and
Guangzhao Li, Everest Clinical Research

ABSTRACT

While analyzing data sets, encountering missing data is often unavoidable. Missing data poses a common challenge to the validity of research outcomes in clinical trials. To reduce the potential bias of study results caused by missing values, multiple imputation (MI) methods are widely employed. This paper presents a systematic approach to handling missing data, beginning with the imputation of missing values from the original data sets using the PROC MI procedure to generate an analysis-ready database, followed by the integration of these processed data sets to derive reliable study conclusions.

Furthermore, this paper explains how to analyze results using the integrated data with the PROC MIANALYZE procedure, including a discussion of potential warning messages and how to resolve them. The proposed methodology offers a versatile tool applicable to a wide range of analyses involving incomplete data sets. This MI process can be encapsulated in a macro and reused across different analytical tasks, such as generating descriptive statistics or fitting statistical models.

While this paper focuses on the general workflow using the predictive mean matching (PMM) method as an example, it also briefly discusses key considerations for selecting an appropriate MI method and incorporating it into the proposed macro.

By combining rigorous imputation techniques with transparent reporting practices, this framework enhances the validity and reproducibility of clinical research outcomes.

INTRODUCTION

In clinical trials, as sample size increases, encountering missing values is more often unavoidable. It is impossible to have an analysis data set filled with all non-missing values. Collected observations may have missing value due to patient dropout, measurement errors or absence at some time point for some subjects. For example, lab technician accidentally drops random blood samples or older patients are more likely to miss later visits. The missing data will create analytical complexities by bringing potential bias for estimation, which represents a critical threat to the validity and integrity of clinical research. Thus, how to deal with the missing values to reduce the bias of study results will be a challenge. It will be valuable to present this paper to examine the theoretical foundations, practical implementation, and regulatory considerations of modern missing data methodologies, proving a roadmap for robust inference in the presence of incomplete data sets. In order to address both technical nuances and real-world applicability across therapeutic domains, it is significant to evaluate performance through simulation studies and clinical trial case examples.

There are many basic methods to deal with missing data with the assumption of **MAR (Missing at Random),** which missing values only depend on the observed data. **LOCF (Last Observation Carried Forward)** is the most widely used method for simple imputation. This method imputed missing values by using the last observed values. **BOCF (Baseline Observation Carried Forward)** is used to impute values by carrying forward the baseline value for the subsequent timepoints, which possibly underestimates the post-baseline results with the assumption that the treatment has no efficacy. There are also other methods that imputing missing values with worst value, best value, mean value, median value and so on. Here, in this paper, the focus will be **the Multiple Imputation Method**.

Multiple Imputation Method is a systematic approach which is always used for handling missing data by imputing data based on observed values multiple times. Usually, there are two parts of this procedure. The first part is generating multiple data sets filling in the missing values from an imputation model by using **PROC MI**. And the result imputed database will be analyzed to derive a wide range of descriptive results, such as mean, standard error, etc. or fit statistic models for analyses. The second part is to

combine the integrated result data and pooling reliable study conclusion using Rubin's Rule with **PROC MIANALYZE**.

STEP-BY-STEP ILLUSTRATION OF MI USING AN EXAMPLE

DATA PREPARATION: TRANSFORMING ORIGINAL DATA TO THE CORRECT FORMAT

First of all, there is an assumption that each row of input database of PROC MI represents one independent subject. Therefore, before using PROC MI, it is necessary to check if the data set is in "long format", which is one subject has multiple rows for repeated measurement or in "wide format", which is one row per subject with different timepoints in multiple variables. If the data is in "long format", you need to use PROC TRANSPOSE to convert to "wide format" before imputation. Otherwise, PROC MI will treat each row as a separate subject, corrupting the imputation model.

There is an example with 50 patients for two treatment groups used for exploring the baseline and change from baseline among different visits. In this database, for example, the data set has a baseline visit and four post-baseline visits for each subject. And suppose there is one covariate variable, Sex Group. In this experiment, there are missing values for some random post-baseline records. (Notice that baseline record (AVISITN = 1) cannot be missing)

SUBJID	AVISITN	TRT01PN	SEX	AVAL
10001	1	1	F	50
10001	2	1	F	54
10001	3	1	F	?
10001	4	1	F	55
10001	5	1	F	56
10002	1	2	M	75
...	2	2	M	?

Table 1. Data Preparation: Example Data in the Original Format

As you can see, the example data is in "long format", so there is a necessary step to transpose the original data to generate the following data in "wide format" in order to use PROC MI.

Obs	SUBJID	TRT01PN	SEX	_NAME_	_LABEL_	v1	v2	v3	v4	v5
1	10001	1	F	AVAL	AVAL	50	54	.	55	56
2	10002	2	M	AVAL	AVAL	75	.	78	77	79

Display 1. Example Data in the Desired Format

THE MULTIPLE IMPUTATION PROCESS: USING PROC MI IN SAS

This process derives a random sample of the missing values from the original distribution. The main goal of PROC MI is to derive an estimation of the model with unbiased by missing data. Therefore, PROC MI will replace missing values with multiple imputations. If imputing 30 times, there will be 30 complete data sets with non-missing data. In default, you need 10-20 observations to impute one variable value. In other words, if one subject has one missing value at visit 2 and second subject has one missing value at visit 3. You will need at least 20 subjects to impute. Otherwise, you will get an error "Not enough observations to fit regression models for variable with an FCS regression method."

As for the code, **MIPUTE** and **SEED** are the key options you need to set when using PROC MI.

NIMPUTE Specifies the times of imputations. According to the SAS help website, the default is NIMPUTE=5, however, using more than 20 imputations will be good for the analysis. And FDA recommends 50-100. If you need to check the errors or other information, setting MIMPUTE=0 is a good way to skip the imputation to check.

SEED uses a positive integer for reproducibility. The default is generated from the time of day from your computer's clock. If you want identical results with multiple execution, you must use the same integer for this option.

Besides, there are some main methods in PROC MI procedure.

MCMC (Markov Chain Monte Carlo) uses Bayesian modeling to impute missing continuous data with the assumption of joint multivariate normality with large sample size.

MNORMAL (Multivariate Normal Imputation) is also based on the assumption of joint multivariate normality with continuous variables. Different from MCMC, it uses **EM (expectation-maximization)** algorithm to estimate the mean vector and covariance matrix of the data and find MLE for the parameters.

MOMOTONE will be used if your data is with monotone missing data patterns. For example, if you have missing data at visit x for the subject, then the later visit for the same subject will also be missing.

However, if the missing values are intermittent, then you will need to use **FCS (Fully Conditional Specification)**. This method is widely used in PROC MI, and can be used for different types of variables, such as continuous, binary, ordinal, nominal, etc. Initially, it will fill missing values with simple imputations with basic statistics, such as mean. Then it will have an iterative imputation cycle to predict missing values with the condition on the current imputed variables. It repeats the iteration until the model is stable.

In the example, there are continuous values in small sample sizes, it is good to use REGPMM statement for FCS method. REGPMM can avoid impossible value under imputation with unbiased results. Covariate variables Sex Group will be in CLASS statement and VAR statement.

```
proc mi data=weightt seed=21965 nimpute=30 out=outmi;
    by trt01pn;
    class sex ;
    fcs regpmm (v2)
    regpmm (v3)
    regpmm (v4)
    regpmm (v5);
    var sex  v1 v2 v3 v4 v5;
run;
```

Obs	_Imputation_	SUBJID	TRT01PN	SEX	_NAME_	_LABEL_	v1	v2	v3	v4	v5
1	1	10001	1	F	AVAL	AVAL	50	54	52	55	56
2	2	10001	1	F	AVAL	AVAL	50	54	52	55	56
3	3	10001	1	F	AVAL	AVAL	50	54	48	55	56
4	4	10001	1	F	AVAL	AVAL	50	54	48	55	56
5	5	10001	1	F	AVAL	AVAL	50	54	48	55	56
6	6	10001	1	F	AVAL	AVAL	50	54	48	55	56
7	7	10001	1	F	AVAL	AVAL	50	54	52	55	56
8	8	10001	1	F	AVAL	AVAL	50	54	48	55	56
9	9	10001	1	F	AVAL	AVAL	50	54	54	55	56
10	10	10001	1	F	AVAL	AVAL	50	54	53	55	56
11	11	10001	1	F	AVAL	AVAL	50	54	52	55	56
12	12	10001	1	F	AVAL	AVAL	50	54	52	55	56
13	13	10001	1	F	AVAL	AVAL	50	54	51	55	56
14	14	10001	1	F	AVAL	AVAL	50	54	52	55	56
15	15	10001	1	F	AVAL	AVAL	50	54	54	55	56
16	16	10001	1	F	AVAL	AVAL	50	54	52	55	56
17	17	10001	1	F	AVAL	AVAL	50	54	52	55	56
18	18	10001	1	F	AVAL	AVAL	50	54	52	55	56
19	19	10001	1	F	AVAL	AVAL	50	54	52	55	56
20	20	10001	1	F	AVAL	AVAL	50	54	53	55	56
21	21	10001	1	F	AVAL	AVAL	50	54	52	55	56
22	22	10001	1	F	AVAL	AVAL	50	54	53	55	56
23	23	10001	1	F	AVAL	AVAL	50	54	52	55	56
24	24	10001	1	F	AVAL	AVAL	50	54	53	55	56
25	25	10001	1	F	AVAL	AVAL	50	54	52	55	56
26	26	10001	1	F	AVAL	AVAL	50	54	54	55	56
27	27	10001	1	F	AVAL	AVAL	50	54	46	55	56
28	28	10001	1	F	AVAL	AVAL	50	54	46	55	56
29	29	10001	1	F	AVAL	AVAL	50	54	51	55	56
30	30	10001	1	F	AVAL	AVAL	50	54	52	55	56
31	1	10002	2	M	AVAL	AVAL	75	75	78	77	79
32	2	10002	2	M	AVAL	AVAL	75	74	78	77	79

Display 2. The Completed Data After Imputation

As a result, there will be 50 subjects multiple 30 imputations, which is a database with 1500 non-missing records in “wide format” and need to transpose back to “long format” for analysis. The data after transposing will have 50 subjects * 30 imputation * 5 visit = 7500 non-missing records.

Obs	SUBJID	TRT01PN	_Imputation_	_NAME_	AVAL
1	10001	1	1	v1	50
2	10001	1	1	v2	54
3	10001	1	1	v3	52
4	10001	1	1	v4	55
5	10001	1	1	v5	56
6	10001	1	2	v1	50
7	10001	1	2	v2	54
8	10001	1	2	v3	52
9	10001	1	2	v4	55
10	10001	1	2	v5	56

Display 3. Final Data set in Transformed Format

POOLING IMPUTATION RESULTS: ANALYZING DATA WITH PROC MIANALYZE IN SAS

After deriving the output data with all non-missing values, it is easy to generate analyzing results either for modelized or non-modelized.

As for non-modelized results, such as mean value. You can easily derive mean values with PROC MEANS or PROC SUMMARY for each treatment group for each imputation number. Therefore, there will

be 30 data sets with mean values for each group.

Obs	_NAME_	TRT01PN	_Imputation_	_TYPE_	_FREQ_	n	mean	se	sd
91	v2	2	1	0	26	26	64.884615385	2.4352077296	12.417171733
92	v2	2	2	0	26	26	64.846153846	2.4291133601	12.386096424
93	v2	2	3	0	26	26	64.961538462	2.4491636056	12.488333017
94	v2	2	4	0	26	26	64.884615385	2.4352077296	12.417171733
95	v2	2	5	0	26	26	64.884615385	2.4352077296	12.417171733
96	v2	2	6	0	26	26	65	2.4570150746	12.528367811
97	v2	2	7	0	26	26	64.961538462	2.4491636056	12.488333017
98	v2	2	8	0	26	26	64.884615385	2.4352077296	12.417171733
99	v2	2	9	0	26	26	64.961538462	2.4491636056	12.488333017
100	v2	2	10	0	26	26	64.846153846	2.4291133601	12.386096424
101	v2	2	11	0	26	26	64.961538462	2.4491636056	12.488333017
102	v2	2	12	0	26	26	64.961538462	2.4491636056	12.488333017
103	v2	2	13	0	26	26	64.846153846	2.4291133601	12.386096424
104	v2	2	14	0	26	26	64.961538462	2.4491636056	12.488333017
105	v2	2	15	0	26	26	64.846153846	2.4291133601	12.386096424
106	v2	2	16	0	26	26	64.884615385	2.4352077296	12.417171733
107	v2	2	17	0	26	26	64.961538462	2.4491636056	12.488333017
108	v2	2	18	0	26	26	64.961538462	2.4491636056	12.488333017
109	v2	2	19	0	26	26	64.961538462	2.4491636056	12.488333017
110	v2	2	20	0	26	26	64.846153846	2.4291133601	12.386096424
111	v2	2	21	0	26	26	64.846153846	2.4291133601	12.386096424
112	v2	2	22	0	26	26	64.884615385	2.4352077296	12.417171733
113	v2	2	23	0	26	26	64.846153846	2.4291133601	12.386096424
114	v2	2	24	0	26	26	64.884615385	2.4352077296	12.417171733
115	v2	2	25	0	26	26	64.884615385	2.4352077296	12.417171733
116	v2	2	26	0	26	26	64.846153846	2.4291133601	12.386096424
117	v2	2	27	0	26	26	64.884615385	2.4352077296	12.417171733
118	v2	2	28	0	26	26	65	2.4570150746	12.528367811
119	v2	2	29	0	26	26	64.884615385	2.4352077296	12.417171733
120	v2	2	30	0	26	26	64.846153846	2.4291133601	12.386096424

Display 4. Non-Modeled Results Using the Mean as an Example

Before using PROC MIANALYZE procedure to combine 30 imputation result data sets into one data set, the important step is to check if all 30 values are identical or not. Otherwise, there will be a warning message “Between-imputation variance is zero for variable mean” if there are no missing values in the original data for some timepoints and all 30 imputation data sets are identical due to non-imputation.

Therefore, it is necessary to separate identical records and non-identical records first. Then take **NODUPKEY** step for identical records and use PROC MIANANLYZE for non-identical records. This step can be encapsulated in a macro since it is always reused before taking PROC MIANALYZE step.

Here is the example code for checking if records are identical.

```
data datain;
  set datain;
  by trt01pn _name_ _Imputation_;
  if dif(value)=0 then idtncl+1;
  if first._name_ then idtncl=1;
run;

proc sql;
  create table prop_trt_flt as
  select *, count(distinct _Imputation_) as cnt,
    case when calculated cnt = max(idtncl) then 'Y' else 'N' end as idfl
      from &datin
      group by trt01pn, _name_
      order by trt01pn, _name_, _Imputation_;
quit;

data prop_trt_flt1 prop_trt_flt2;
    set prop_trt_flt;
    if idfl = 'N' then output prop_trt_flt1;
```

```
    if idfl = 'Y' then output prop_trt_flt2;
run;
```

For the database with identical records, it can be directly removed the duplicate records. And for the database with non-identical records, **PROC MIANALYZE** needs to be used.

PROC MIANALYZE combines the result using Rubin's rules (Rubin, 1987). This accounts for both within-imputation variability and between-imputation variability, providing statistics such as estimates, standard errors, confidence intervals, and p-values for each treatment group and each timepoint. It will be required **MODELEFFECTS** statement to list the fixed effects which is the variables need to analyze. And **STDERR** statement lists the standard errors with the effects in the **MODELEFFECTS** statement.

Here is the sample code of **PROC MIANALYZE** for this example. The analysis variable is MEAN.

```
proc mianalyze data=prop_trt_flt1;
    by trt01pn _name_;
    modeleffects mean;
    stderr sd;
    ods output VarianceInfo = Variance1;
    ods output ParameterEstimates = result;
run;
```

And here is the output with the above code. Estimate will be the combined result for the mean among 30 imputation data sets. In this situation, since there are only missing values at visit 2, visit 3 for TRT01PN = 2 and visit 3, visit 5 for TRT01PN = 1 in the original data, the result of **PROC MIANALYZE** only includes the result of these visits. Therefore, the final output data needs to set with identical records after taking **NODUPKEY**.

Obs	TRT01PN	_NAME_	NImpute	Parm	Estimate	StdErr	LCLMean	UCLMean	Probt
1	2	v2	30	mean	64.903846	2.439637	60.12224	69.68545	<.0001
2	1	v3	30	mean	61.675000	2.640636	56.49945	66.85055	<.0001
3	2	v3	30	mean	63.338462	2.697473	58.05151	68.62541	<.0001
4	1	v5	30	mean	62.647222	2.454902	57.83570	67.45874	<.0001

Display 5. Results from PROC MIANALYZE Using the Mean as an Example

If there is necessary to generate LS Means, P values or other statistics with models such as MMRM, ANCOVA, etcs. **PROC MIANALYZE** is also can be used by similar steps.

For example, you can derive MMRM model with PROC MIXED procedure to get the results with 30 imputation results data sets.

```
proc mixed data = midata METHOD=REML;
class trt01pn _name_   subjid;
    model chg = trt01pn _name_   trt01pn*_name_ base / s ddfm= kr;
    Repeated _name_ / type = cs subject = subjid;
    lsmeans trt01pn*_name_/pdiff cl alpha=0.05;
    lsmestimate trt01pn*_name_
    'Control vs placebo'  0 0 0 1 1  0 0 0 -1 -1  divisor = 2/ CL;
    by _imputation_;
run;
```

Like doing for non-modelized results, it is significant to check the identity before using **PROC MIANALYZE**. And then combining the 30 imputation result data sets and generating the result by **PROC MIANLAYZE** for non-identical records and **NODUPKEY** for identical records.

The whole process of MI method is systematic and can be used for every statistic for analysis.

KEY CONSIDERATIONS FOR SELECTING AN APPROPRIATE MULTIPLE IMPUTATION (MI) METHOD

The **predictive mean matching (PMM)** method presented above is a widely used imputation technique for continuous variables. However, it may not be suitable for all studies. This section briefly discusses key considerations for selecting an appropriate MI method and integrating it into the proposed process.

STRUCTURED APPROACH FOR SELECTING AN MI METHOD

Several factors should be considered when choosing the most appropriate MI method. Below is a structured approach to guide the selection:

1. **Understand the Missingness Mechanism**
   Determine whether the missing data are:
   - Missing Completely at Random (MCAR)
   - Missing at Random (MAR)
   - Missing Not at Random (MNAR)
2. **Assess the Missingness Pattern**
   Identify whether the missing data follow a monotone or arbitrary pattern.
3. **Consider Variable Types**
   Different MI methods are appropriate for **continuous**, **binary/categorical**, or **mixed** data types.
4. **Role of the Variables with Missing Values**
   Consider whether the missing values are in predictor variables, response variables, or both. This distinction may influence the choice of model used for imputation.
5. **Conduct Sensitivity Analysis**
   Perform sensitivity analyses to evaluate the robustness of results under varying missing data assumptions.

The details of each step are discussed in the following sections.

MECHANISM OF MISSING VALUES AND MISSINGNESS PATTERN

When missing values are Missing Completely at Random (MCAR)— where the probability of missingness is independent of observed or unobserved data—single imputation methods may suffice. Examples include last observation carried forward (LOCF), mean/median/mode imputation, regression imputation, and hot-deck imputation. These methods are relatively simple to implement in SAS using a DATA step.

For **Missing at Random (MAR)** or **Missing Not at Random (MNAR)**, multiple imputation methods are generally preferred.

There are two important missingness patterns to be aware of: **monotone missingness** and **arbitrary (non-monotone) missingness**.

A database exhibits a monotone missingness pattern if the variables can be ordered in such a way that, for any given observation, if a value is missing for a particular variable, then all variables that follow in that order are also missing. In SAS, the MONOTONE statement can be used in conjunction with appropriate imputation methods.

Below is an example SAS code demonstrating this approach. It requests the regression method for imputing variable V2 and the predictive mean matching (PMM) method for imputing variable V3. The resulting imputed data set is named outmi. The MU0= option requests t-tests for the hypotheses that the population means corresponding to the variables in the VAR statement are V2=65 and V3=75.

```
proc mi data=in mu0= 0 65 75
        seed=21965 nimpute=30 out=outmi;
   monotone reg(V2/ details)
            regpmm(V3= V1 V2 V1*V2/ details);
   var V1 V2 V3;
run;
```

While MCMC is well-suited for both monotone and arbitrary missing data patterns, it is important to take steps to avoid bias when generating minimum and maximum values (e.g., 1 and 100). In PROC MI, the minimum and maximum scores are set to the observed minimum minus 0.49 and the observed maximum plus 0.49, respectively. The imputed values are then rounded to the nearest integer.

```
proc mi data=in out=outmi seed=21965 nimpute=30;
    mcmc
        minimum=(0.51 0.51 0.51 0.51 0.51)
        maximum=(100.49 100.49 100.49 100.49 100.49);
    by trt01pn;
    var v1 v2 v3 v4 v5;
run;
```

In the case of MNAR, the probability of missingness depends on the unobserved values themselves. This scenario typically arises when there is moderate to high missingness (greater than 5%).

The implementation of the pattern-mixture model approach often involves control-based pattern imputation. In this approach, the imputation model for missing observations in the treatment group is constructed using the observed data from the control group, rather than from the treatment group itself. The same model is also used to impute missing observations in the control group.

In SAS, the MNAR statement can be invoked to perform imputation using the pattern-mixture model approach. The MODELoption specifies that only observations with trt01pn = 2 (i.e., the control group) are used to derive the imputation model for variables v1–v5.

```
proc mi data= in out=outmi seed=21965 nimpute=30
  class sex;
  fcs reg (v1-v5);
  mnar model(v1-v5/ modelobs =(trt01pn =2));
  var sex v1-v5;
run;
```

CONSIDER THE VARIABLE TYPES

As with all statistical analyses, choosing an appropriate multiple imputation (MI) method requires careful consideration of the data types involved.

The FCS (Fully Conditional Specification) statement in SAS allows for multivariate imputation using different methods tailored to each variable type. With FCS, variables are imputed sequentially in the order specified in the VAR statement.

- For continuous variables, you can use a regression or predictive mean matching method.
- For nominal categorical variables, you can use either a discriminant function method or logistic regression(generalized logit model) without considering the ordering of class levels.
- For ordinal categorical variables, you can use logistic regression (cumulative logit model), which accounts for the ordering of the categories.
- For binary variables, either a discriminant function method or a logistic regression method can be applied.

```
proc mi data= in out=outmi seed=21965 nimpute=30;
   class sex;
   monotone reg( v1/ details)
            logistic( v3= v1 v2 v1*v2/ details);
   var v1-v3;
run;
```

By default, **regression** is used for continuous variables and **discriminant function** is used for classification variables.

The table below summarizes the available imputation methods in PROC MI, accounting for both the missingness pattern (as discussed earlier) and variable type.

Missingness Pattern	Variable Type	Available Methods
Monotone	Continuous	Monotone regression Monotone predicted mean matching Monotone propensity score
Monotone	Classification (ordinal)	Monotone logistic regression
Monotone	Classification (nominal)	Monotone discriminant function
Arbitrary	Continuous	MCMC full-data imputation MCMC monotone-data imputation FCS regression FCS predicted mean matching
Arbitrary	Classification (ordinal)	FCS logistic regression
Arbitrary	Classification (nominal)	FCS discriminant function

Table 2. Imputation Methods in PROC MI

ROLE OF MISSING VALUES

The propensity score method was originally designed for randomized experiments with repeated measures on the response variables. Its goal is to impute missing values in the response variables. The method uses only covariate information associated with the missingness of the imputed variable; it does not rely on correlations among variables.

To impute values for a variable Y with missing data, the propensity score method involves the following steps:

1. Fit a logistic regression model where the outcome is a missingness indicator (1 if missing, 0 if observed).
2. Generate a propensity score for each observation, estimating the probability that the value is missing.
3. Divide observations into a fixed number of strata (typically five) based on these propensity scores.
4. Impute missing values within each stratum using observed data (e.g., mean imputation or regression).

The propensity score method can be used to impute continuous variables in datasets with monotone missing patterns. Below is an example SAS code:

```
proc mi data= in out=outmi seed=21965 nimpute=30;
monotone propensity;
var v1-v3;
run;
```

During the imputation process, within each stratum (typically five strata), the missing values of v2 are first imputed using the observed values of v1. Then, the missing values of v3 are imputed using the observed values of both v1 and v2 within that same stratum.

SENSITIVITY ANALYSIS

According to the estimand framework outlined in the ICH E9(R1) Guideline (ICH E9(R1) Addendum, Section 5.2): *"Sensitivity analyses should explore the impact of departures from the primary missing data assumptions,"* it is essential to conduct sensitivity analyses to evaluate the robustness of study conclusions. These analyses serve as sensitivity estimators and should include an assessment of the assumptions underlying the chosen imputation methods.

For example, it is recommended to compare inferential results obtained under the Missing at Random (MAR) assumption to those obtained under the Missing Not at Random (MNAR) assumption. This comparison allows for an evaluation of the sensitivity of the study conclusions to the MAR assumption. If the MAR assumption leads to conclusions that differ substantially from those derived under MNAR scenarios, the validity of the MAR assumption may be called into question.

Tipping Point Analysis (TPA) in clinical trials is a sensitivity analysis used to assess the robustness of trial results, particularly when dealing with missing data. It determines how much the results would need to change (i.e., the "tipping point") to alter the trial's primary conclusion (e.g., statistical significance or clinical relevance).

CONCLUSION

This paper provides a step-by-step guide for applying multiple imputation in SAS and a framework for selecting appropriate methods. The recommended approaches aim to ensure valid, reliable results during data preprocessing, supporting robust clinical decision-making.

By systematically addressing missing data and transparently reporting its impact, this guide enhances confidence and reproducibility in research findings.

REFERENCES

Afkanpour, M., Tehrany Dehkordy, D., Momeni, M. *et al.* Conceptual framework as a guide to choose appropriate imputation method for missing values in a clinical structured dataset. *BMC Med Res Methodol* **25**, 43 (2025). https://doi.org/10.1186/s12874-025-02496-3

SAS Institute Inc. 2015. SAS/STAT® 14.1 User's Guide. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

Zhihua Zhang
Everest Clinical Research Corporation
jenny.zhang@ecrscorp.com

Guangzhao Li
Everest Clinical Research Corporation
rachel.li@ecrscorp.com